Symbolic Logic

An Accessible Introduction to Serious Mathematical Logic

Tony Roy

From first print edition, 2023
(Third revision, July 2025)

Additional resources for this work are available at
https://tonyroyphilosophy.net/symbolic-logic/

For my girls,
Rose, Hannah, Christina

Preface

There is, I think, a gap between what many students learn in their first course in formal logic, and what they are expected to know for their second. While courses in mathematical logic with metalogical components often cast only the barest glance at mathematical induction or even the very idea of reasoning from definitions, a first course may also leave these untreated, and fail explicitly to lay down the definitions upon which the second course is based. The aim of this text is to integrate material from these courses and, in particular, to make serious mathematical logic accessible to students I teach. The first parts introduce classical symbolic logic as appropriate for beginning students; the last parts build to Gödel's completeness and incompleteness results. A distinctive feature of the last section is a complete development of Gödel's second incompleteness theorem.

Accessibility, in this case, includes components which serve to locate this text among others: First, assumptions about background knowledge are minimal. I do not assume particular content about computer science, or about mathematics much beyond high school algebra. Officially, everything is introduced from the ground up. No doubt, the material requires a certain sophistication—which one might acquire from other courses in critical reasoning, mathematics, or computer science. But the requirement does not extend to particular contents from any of these areas.

Second, I aim to build skills, and to keep conceptual distance for different applications of 'so' relatively short. Authors of books that are completely correct and precise may assume skills and require readers to recognize connections not fully explicit. It may be that this accounts for some of the reputed difficulty of the material. The results are often elegant. But this can exclude a class of students capable of grasping and benefiting from the material, if only it is adequately explained. Thus I attempt explanations and examples to put the student at every stage in a position to understand the next. In some cases, I attempt this by introducing relatively concrete methods for reasoning. The methods are, no doubt, tedious or unnecessary for the experienced logician. However, I have found that they are valued by students, insofar as students are presented with an occasion for success. These methods are not meant to wash over or substitute for understanding details, but rather to expose and clarify them. Clarity, beauty, and power come, I think, by getting at details, rather than burying or ignoring them.

Third, the discussion is ruthlessly directed at core results. Results may be rendered inaccessible to students, who have many constraints on their time and schedules, simply because the results would come up in, say, a second course rather than a first. My idea is to exclude side topics and problems, and to go directly after (what I see as) the core. One manifestation is the way definitions and results from earlier sections feed into ones that follow. Thus simple integration is a benefit. Another is the way predicate logic with identity is introduced as a whole in Part I. Though it is possible to isolate sentential logic from the first parts of Chapter 2 through Chapter 6, and so to use the text for separate treatments of sentential and predicate logic, the guiding idea is to avoid repetition that would be associated with independent treatments for sentential logic, or perhaps monadic predicate logic, the full predicate logic, and predicate logic with identity.

Also (though it may suggest I am not so ruthless about extraneous material as I would like to think), I try to offer some perspective about what is accomplished along the way. Some of this is by organization; some by asides to the main text; and some built into the main content. So for example the text may be of particular interest to those who have, or desire, an exposure to natural deduction in formal logic. In this case, insight arises from the nature of the system. In the first part, I introduce both axiomatic and natural derivation systems; and in Part III, show how they are related.

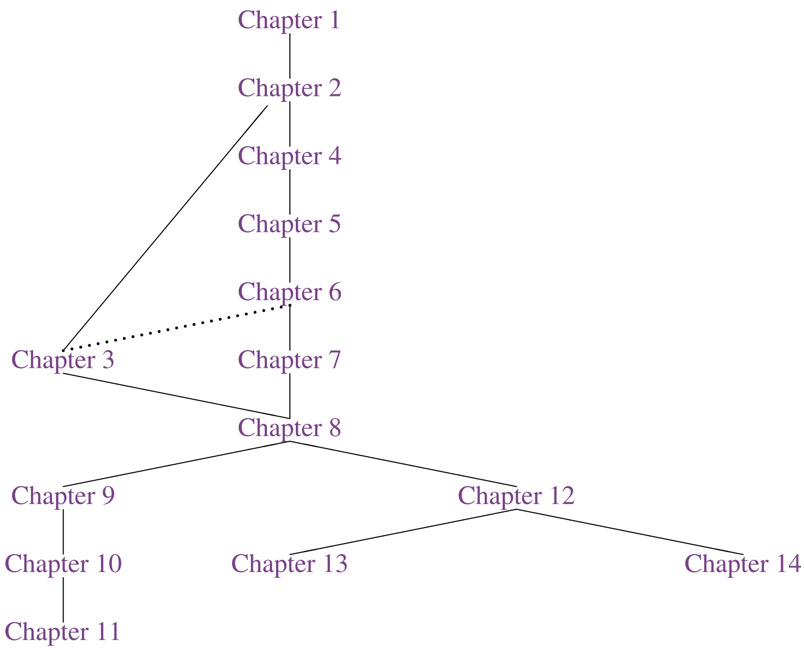
There are different ways to organize a course around this text. Chapters locate and order material *conceptually*. But in many contexts the conceptual order will be other than the best pedagogical order, and content may be taken in different ways. For students who are likely to complete the whole, a straightforward option is to proceed sequentially through the text from beginning to end (but postponing Chapter 3 until after Chapter 6). Taken as wholes, Part II depends on Part I; parts III and IV on parts I and II. At the level of whole chapters, dependencies are as in the box on the next page. At a more fine-grained level, one might construct a sequence, like one I have regularly offered, as follows:

informal notions:	Chapter 1
sentential logic:	first parts of chapters 2, 4, 5, 6
predicate logic:	latter parts of chapters 2, 4, 5, 6
transitional:	chapters 3, 7, first parts of 8
advanced topics:	*metalogic:* 8.3, Part III; and/or *incompleteness:* 8.4, Part IV

For predicate logic I have preferred to cover material in the order 2, 6, 4, 5 to convey a sense of the formal language "by immersion" prior to chapters 4 and 5. Thus the text is compatible with different course organizations—and may (should) be customized to your own needs!

A remark about Chapter 7 especially for the instructor: By a formally restricted system for reasoning with semantic definitions, Chapter 7 aims to leverage derivation skills from earlier chapters to informal reasoning with definitions. I have had a difficult time convincing instructors to try this material—and even been told flatly that these

Chapter dependencies. Though there are cross references throughout, the following represent reasonable sequences for study:



The relation between Chapter 6 and Chapter 3 is pedagogical rather than logical, and might be ignored for students with sufficient technical background (but see the caution in the box on page 69 of Chapter 3).

skills "cannot be taught." In my experience, this is false (and when I have been able to convince others to try the chapter, they have quickly seen its value). Perhaps the difficulty is just that the strategy is unfamiliar. Of course, if one is presented with students whose mathematical sophistication is sufficient for advanced work, it is not necessary. But if (as is often the case, especially for students in philosophy) one obtains one's mathematical sophistication *from* courses in logic, this chapter is an important part of the bridge from earlier material to later. Additionally, the chapter is an important "takeaway" even for students who will not continue to later material. The chapter closes an open question from Chapter 4—how it is possible to demonstrate quantificational validity. But further, the ability to reason closely with definitions is a skill from which students in (sentential or) predicate logic, even though they never go on to formalize another sentence or do another derivation, will benefit both in philosophy and more generally.

Another remark about the (long) sections 13.3, 13.4, and 13.5. These develop in PA the "derivability conditions" for Gödel's second incompleteness theorem. They are perhaps for enthusiasts. Still, in my experience many students are enthusiasts and, especially from an introduction, benefit by seeing the conditions derived—else the very idea of proving in a formal theory results about provability may remain mysterious. There are different ways to treat the sections. One might work through them in some detail. However, even if you skim demonstrations lightly, there is an advantage having a panorama at which to gesture and say "thus it is accomplished!"

Naturally, results in this book are not innovative. If there is anything original, it is in presentation. Even here, I am greatly indebted to others, especially perhaps Bergmann, Moor, and Nelson, *The Logic Book*, Mendelson, *Introduction to Mathematical Logic*, and Smith, *An Introduction to Gödel's Theorems*. I thank my first logic teacher, G.J. Mattey, who communicated to me his love for the material. And I thank especially my colleagues John Mumma and Darcy Otto for many helpful comments. Hannah Baehr and Catlin Andrade provided comments and some of the answers to exercises. In addition I have received helpful feedback from Ramachandran Venkataraman and Steve Johnson, along with students in different logic classes at California State University San Bernardino. Hannah and Steve Baehr produced the cover.

This text evolved over a number of years starting modestly from notes originally provided as a supplement to other texts. The current version divides naturally into two volumes, the first (including Parts I and II) for reasoning *in* logic, and the second (including Parts III and IV) for reasoning *about* it. These volumes are available in hardcopy from Amazon.com. In addition, both the text and answers to selected exercises are available as PDF downloads at `https://tonyroyphilosophy.net/symbolic-logic/`. The website includes also a forum for comment and discussion. I recommend working from the hardcopy: for a text that you do not merely read but rather *work through*, it makes a difference to see more than a "screen's worth" at a time, mark on pages, and such. (The hardcopy is available at the lowest allowable price, without royalties for me, so this is no self-interested recommendation.) Of course, the electronic version is useful too—an advantage over the hardcopy is that its many internal links are live. Further, the assiciated Symbolic Logic APPlication (SLAPP) is freely available from the website. In its current version (3.0) it is sufficient for the production of exercises from (at least) Volume 1 of the text. In addition SLAPP checks derivations from chapters 3 and 6, and has a help function for the basic I/E-rule systems of chapter 6. It is well-worth using this tool.

I think this is fascinating material, and consider it great reward when students respond "cool!" as they sometimes do. I hope you will have that response more than once along the way.

T.R. July 2025

Contents

Preface iv

Contents viii

Quick Reference Guides xi

I The Elements: Four Notions of Validity 1

1 Logical Validity and Soundness 5
- 1.1 Consistent Stories . . . 6
- 1.2 The Definitions . . . 12
- 1.3 Some Consequences . . . 23

2 Formal Languages 31
- 2.1 Introductory . . . 31
- 2.2 Sentential Languages . . . 33
- 2.3 Quantificational Languages . . . 46

3 Axiomatic Deduction 65
- 3.1 General . . . 66
- 3.2 Sentential . . . 69
- 3.3 Quantificational . . . 77
- 3.4 Application: PA . . . 85

4 Semantics 93
- 4.1 Sentential . . . 93
- 4.2 Quantificational . . . 110

5 Translation 134
- 5.1 General . . . 134
- 5.2 Sentential . . . 136
- 5.3 Quantificational . . . 164

6 Natural Deduction **198**
6.1 General . . . 198
6.2 Sentential . . . 207
6.3 Quantificational . . . 263
6.4 Applications: Q and PA . . . 299

II Transition: Reasoning About Logic **315**

7 Direct Semantic Reasoning **317**
7.1 Introductory . . . 318
7.2 Sentential . . . 320
7.3 Quantificational . . . 335

8 Mathematical Induction **361**
8.1 General Characterization . . . 361
8.2 Preliminary Examples . . . 367
8.3 Further Examples (for Part III) . . . 380
8.4 Additional Examples (for Part IV) . . . 390

III Classical Metalogic: S̊oundness and C̊ompleteness **406**

9 Preliminary Results **409**
9.1 Semantic Validity Implies Logical Validity . . . 409
9.2 Validity in *AD* Implies Validity in *ND* . . . 414
9.3 Validity in *ND* Implies Validity in *AD* . . . 421
9.4 Extending to *ND+* . . . 440

10 Main Results **446**
10.1 S̊oundness . . . 447
10.2 Sentential C̊ompleteness . . . 453
10.3 Quantificational C̊ompleteness: Basic Version . . . 464
10.4 Quantificational C̊ompleteness: Full Version . . . 476

11 More Main Results **491**
11.1 Expressive Completeness . . . 491
11.2 Unique Readability . . . 495
11.3 Independence . . . 499
11.4 Beginning Model Theory . . . 503

IV Logic and Arithmetic: Ĩncompleteness and Computability **560**

12 Recursive Functions and Q **564**
12.1 Recursive Functions 566
12.2 Expressing Recursive Functions 573
12.3 Capturing Recursive Functions 587
12.4 More Recursive Functions 602
12.5 Essential Results 623

13 Gödel's Theorems **637**
13.1 Gödel's First Theorem 637
13.2 Gödel's Second Theorem: Overview 643
13.3 The Derivability Conditions: Definition 650
13.4 The Second Condition: $\Box(\mathscr{P} \to \mathscr{Q}) \to (\Box\mathscr{P} \to \Box\mathscr{Q})$ 695
13.5 The Third Condition: $\Box\mathscr{P} \to \Box\Box\mathscr{P}$ 716
13.6 Reflections on the Second Theorem 735

14 Logic and Computability **751**
14.1 Turing Computable Functions 751
14.2 Essential Results 764
14.3 Church's Thesis 771

Concluding Remarks **791**

Bibliography **794**

Index **799**

Quick Reference Guides

Negation and Quantity . . . 21
Classical Validity . . . 28
Countability . . . 35
Parts of a Formula . . . 39
More on Countability . . . 48
Grammar Quick Reference . . . 61
ADs Quick Reference . . . 78
AD Quick Reference . . . 84
Peano Arithmetic (*AD*) . . . 92
Greek Characters . . . 102
Semantics Quick Reference (sentential) . . . 109
Basic Notions of Set Theory . . . 112
Semantics Quick Reference (quantificational) . . . 132
Definitions for Translation . . . 142
'Neither nor' and 'Not both' . . . 155
Cause and Conditional . . . 159
Definitions for Auxiliary Assumptions . . . 206
NDs Quick Reference . . . 225
NDs+ Quick Reference . . . 259
ND Quick Reference . . . 291
ND+ Quick Reference . . . 298
$\mathscr{L}_{NT}$ Quick Reference . . . 301
Robinson and Peano Arithmetic (*ND*+) . . . 313
Metalinguistic Quick Reference (sentential) . . . 333
Metalinguistic Quick Reference (quantificational) . . . 352
On the Semantics of Variables . . . 353
Theorems of Chapter 7 . . . 359
Induction Schemes . . . 368
First Theorems of Chapter 8 . . . 390
Final Theorems of Chapter 8 . . . 405
Numeral and Number . . . 419
Theorems of Chapter 9 . . . 445

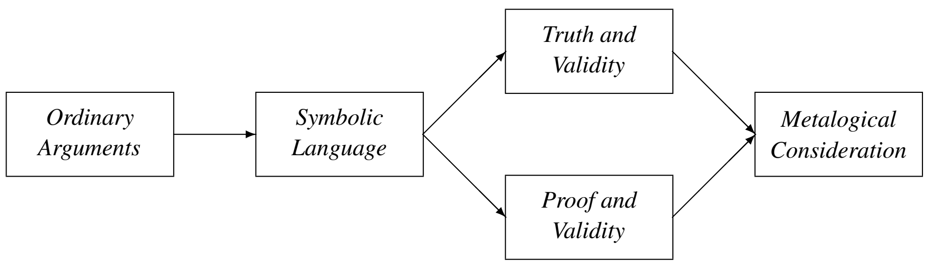
Some Arithmetic Relevant to Gödel Numbering 458
More Arithmetic Relevant to Gödel Numbering 468
Theorems of Chapter 10 490
First Theorems of Chapter 11 514
Basic Definitions for Model Theory 522
Extensions of Classical Logic 527
Cantor's Theorem 537
Final Theorems of Chapter 11 559
The Recursion Theorem 570
Arithmetic for the Beta Function 583
Sigma-1 and Pi-1 Formulas 591
First Results of Chapter 12 606
Final Results of Chapter 12 636
Additional Theorems of PA 649
First Theorems of Chapter 13 660
Font Conventions 694
Second Theorems of Chapter 13 715
Substitution Vectors 729
Final Theorems of Chapter 13 750
Binary Numbers 753
Simple Time Dilation 775
Enumerating Primitive Recusive Functions 778
Theorems of Chapter 14 790

Part I

The Elements: Four Notions of Validity

Introductory

Symbolic logic is a tool for argument evaluation. In this part of the text we introduce the basic elements of that tool. Those parts are represented in the following diagram:



The starting point is ordinary arguments. Such arguments come in various forms and contexts—from politics and ordinary living, to mathematics and philosophy. Here is a classic, simple case:

(A)
All humans are mortal.
Socrates is a human.
―――――
Socrates is mortal.

This argument has *premises* listed above a line, with a *conclusion* listed below. The premises are supposed to demonstrate the conclusion. Here is another case which may seem less simple:

(B) If the maid did it, then it was done with a revolver only if it was done in the parlor. But if the butler is innocent, then the maid did it unless it was done in the parlor. The maid did it only if it was done with a revolver, while the butler is guilty if it did happen in the parlor. So the butler is guilty.

It is fun to think about this; from the given evidence, it follows that the butler did it! Here is an argument that is both controversial and significant:

(C) There is evil. If god is good, then there is no evil unless god has morally sufficient reasons for allowing it. If god is both omnipotent and omniscient, then god does not have morally sufficient reasons for allowing evil. So god is not good, omnipotent, and omniscient.

A being is *omnipotent* if it is all-powerful, and *omniscient* if all-knowing. Thus this is a version of the famous "problem of evil" for traditional theism. It matters whether the conclusion is true! Roughly, an argument is good if it does what it is supposed to do, if its premises demonstrate the conclusion; and an argument is bad if it does not do what it is supposed to do, if its premises fail to demonstrate the conclusion. So a theist (someone who accepts that there is a god) may want to hold that (C) is a bad argument, but an atheist (someone who denies that there is a god) that it is good.

We begin in Chapter 1 with an account of success for ordinary arguments (the leftmost box). So we say what it is for an argument to be good or bad. This introduces us to the fundamental notions of *logical validity* and *logical soundness*. These will be our core concepts for argument evaluation. But just as it is one thing to know what honesty is, and another to know whether someone is honest, so it is one thing to know what logical validity and logical soundness are, and another to know whether an argument is valid or sound. In some cases, it may be obvious. But others are not so clear—as, for example, cases (B) or (C) above, along with complex arguments in mathematics and philosophy. Thus symbolic logic is introduced as a sort of machine or tool to identify validity and soundness.

This machine begins with certain symbolic representations of ordinary arguments (the box second from the left). That is why it is *symbolic* logic. We introduce these representations in Chapter 2, and translate from ordinary arguments to the symbolic representations in Chapter 5. Once arguments have this symbolic representation, there are different methods of operating upon them. We develop three such methods, each with its own distinctive advantages and disadvantages.

An account of truth and validity is developed for the symbolic representations in Chapter 4 and Chapter 7 (the upper box). On this account, truth and validity are associated with clearly defined criteria for their evaluation. And validity from this upper box implies logical validity for the ordinary arguments that are symbolically represented. Thus we obtain clearly defined criteria to identify the logical validity of arguments we care about. Evaluation of validity for the butler and evil cases is entirely routine given the methods from Chapter 2, Chapter 4, and Chapter 5—though the soundness of (C) will remain controversial!

One account of proof and validity is developed for the symbolic representations in Chapter 3, and another in Chapter 6. So there are separate applications of the proof method (the lower box). Again, on these accounts, proof and validity are associated with clearly defined criteria for their evaluation. And validity by the proof methods implies logical validity for the ordinary arguments that are symbolically represented. In each case the result is another well-defined approach to the identification of logical validity. Evaluation of validity for the butler and evil cases is entirely routine given the methods from, say, Chapter 2, Chapter 3, and Chapter 5, or alternatively, Chapter 2, Chapter 5, and Chapter 6—though, again, the soundness of (C) will remain controversial.

These, then, are the elements of our logical "machine"—we start with the fundamental notion of logical validity, then there are symbolic representations of ordinary reasonings, along with approaches to evaluation from truth and validity, and from proof and validity. These elements are developed in this part. In later parts we turn to thinking about how these parts work together (the right-hand box). In particular, we begin thinking *how* to reason about logic (Part II), *demonstrate* that the same arguments come out valid by the truth method as by the proof methods (Part III), and establish limits on application of logic and computing to arithmetic (Part IV). But first we have to say what the elements are. And that is the task we set ourselves in this part.

Chapter 1

Logical Validity and Soundness

We have said that symbolic logic is a tool or machine for the identification of argument goodness. In this chapter we begin, not with the machine, but with an account of this "argument goodness" that the machinery is supposed to identify. In particular, we introduce the notions of *logical validity* and *logical soundness*.

An argument is made up of sentences one of which is taken to be supported by the others.

AR An *argument* is some sentences, one of which (the *conclusion*) is taken to be supported by the remaining sentences (the *premises*).

(Important definitions are often offset and given a short name as above; then there may be appeal to the definition by its name, in this case, 'AR'.) So an argument has premises which are taken to support a conclusion. Such support is often indicated by words or phrases of the sort, 'so', 'it follows', 'therefore', or the like. We will typically represent arguments in *standard form* with premises listed as complete sentences above a line, and the conclusion under. Roughly, an argument is good if the premises do what they are taken to do, if they actually support the conclusion. An argument is bad if they do not accomplish what they are taken to do, if they do not actually support the conclusion.

Logical validity and soundness correspond to different ways an argument can go wrong. Consider the following two arguments:

(A)	Only citizens can vote Hannah is a citizen ——— Hannah can vote	(B)	All citizens can vote Hannah is a citizen ——— Hannah can vote

The line divides premises from conclusion, indicating that the premises are supposed to support the conclusion. Thus these are arguments. But these arguments go wrong in different ways. The premises of argument (A) are true; as a matter of fact, only citizens can vote, and Hannah (my daughter) is a citizen. But she cannot vote; she is not old enough. So the conclusion is false. Thus, in argument (A), the relation between the premises and the conclusion is defective. Even though the premises are true, there is no guarantee that the conclusion is true as well. We will say that this argument is *logically invalid*. In contrast, argument (B) is logically valid. If its premises were true, the conclusion would be true as well. So the *relation* between the premises and conclusion is not defective. The problem with this argument is that the premises are not true—not all citizens can vote. So argument (B) is defective, but in a different way. We will say that it is *logically unsound*.

The task of this chapter is to define and explain these notions of logical validity and soundness. I begin with some preliminary notions in section 1.1, then turn to official definitions of logical validity and soundness (section 1.2), and finally to some consequences of the definitions (section 1.3).

1.1 Consistent Stories

Given a certain notion of a *possible* or *consistent* story, it is easy to state definitions for logical validity and soundness. So I begin by identifying the kind of stories that matter. Then we will be in a position to state the definitions, and apply them in some simple cases.

Let us begin with the observation that there are different sorts of possibility. Consider, say, 'Hannah could make it in the WNBA (that is, in the Women's National Basketball Association)'. This seems true. She is reasonably athletic, and if she were to devote herself to basketball over the next few years, she might very well make it in the WNBA. But wait! Hannah is only a kid—she rarely gets the ball even to the rim from the top of the key—so there is no way she could make it in the WNBA. So we have said both that she could and that she could not make it. But this cannot be right. What is going on? Here is a plausible explanation: Different sorts of possibility are involved. When we hold fixed current abilities, we are inclined to say there is no way she could make it. When we hold fixed only general physical characteristics, and allow for development, it is natural to say that she might. Similarly, I sometimes ask students if it is possible to drive the 60 miles from our campus in San Bernardino to Los Angeles in 30 minutes. From natural assumptions about Los Angeles traffic, law enforcement, and the like, most say it is not. But some, under different assumptions, allow that it can be done! In each example, the scope of what is possible varies with whatever constraints are in play: the weaker the constraints, the broader the range of what is possible. In ordinary contexts, constraints are understood—so when you ask a friend if she can make it to your party in thirty minutes, rocket ships and jet cars are not an option. That is how we manage to communicate.

The sort of possibility we are interested in is *very* broad, and constraints are correspondingly weak. We will allow that a story is *possible* or *consistent* so long as it involves no *internal* contradiction. A story is impossible when it collapses from within. For this it may help to think about the way you respond to ordinary fiction. Consider, say, J.K. Rowling's *Harry Potter and the Prisoner of Azkaban*. Harry and his friend Hermione are at wizarding school. Hermione acquires a "time turner" which allows time travel, and uses it in order to take classes that are offered at the same time. Such devices are no part of the actual world, but they fit into the wizarding world of Harry Potter. So far, then, the story does not contradict itself. So you go along.

At one stage, though, Harry is at a lakeshore under attack by a bunch of fearsome "dementors." His attempts to save himself appear to have failed when a figure across the lake drives the dementors away. But the figure who saves Harry is Harry himself who has come back from the future. Somehow then, as often happens in these stories, the past depends on the future, at the same time as the future depends on the past: Harry is saved only insofar as he comes back from the future, but he comes back from the future only insofar as he is saved. This, rather than the time travel itself, generates an *internal* conflict. The story makes it the case that you cannot have Harry's rescue apart from his return, and cannot have Harry's return apart from his rescue. This might make sense if time were always repeating in an eternal loop. But, according to the story, there were times before the rescue and after the return. So the story faces *internal* collapse. Notice: the objection does not have *anything* to do with the way things actually are—with existence of time turners or the like; it has rather to do with the way the story hangs together internally.[1] Similarly, we want to ask whether stories hold together *internally*. If a story holds together internally, it counts for our purposes as consistent and possible. If a story does not hold together, it is not consistent or possible.

In some cases, stories may be consistent with things we know are true in the real world. Thus Hannah could grow up to play in the WNBA. There is nothing about our world that rules this out. But stories may remain consistent though they do not fit with what we know to be true in the real world. Here are cases of time travel and the like. Stories become inconsistent when they collapse internally—as when a story says that some time both can and cannot happen apart from another.

As with a movie or novel, we can say that different things are true or false *in* our stories. In *Harry Potter* it is true that Harry and Hermione travel through time with a timer turner, but false that they go through time in a DeLorean (as in the *Back to the Future* films). In the real world, of course, it is false that there are time turners, and false that DeLoreans go through time. Officially, a complete story is always *maximal*

---

[1]In more consistent cases of time travel (in fiction) time seems to move on different paths so that after today and tomorrow, there is *another* today and *another* tomorrow. So time does not return to the very point at which it first turns back. In the trouble cases, time seems to move in a sort of "loop" so that a point on the path to today (this very day) goes through tomorrow. With this in mind, it is interesting to think about say, the *Terminator* (1984, 1991) and *Back to the Future* (1985, 1989, 1990) films along with, maybe more consistent, *Groundhog Day* (1993) and, very much like it, *Happy Death Day* (2017). Even if I am wrong, and the Potter story is internally consistent, the overall point should be clear. And it should be clear that I am not saying anything serious about time travel.

in the sense that *any* sentence is either true or false in it. A story is *inconsistent* when it makes some sentence both true and false. Since, ordinarily, we do not describe every detail of what is true and what is false when we tell a story, what we tell is only part of a maximal story. In practice, however, it will be sufficient for us merely to give or fill in whatever details are relevant in a particular context.

But there are a couple of cases where we cannot say when sentences are true or false in a story. The first is when stories we tell do not fill in relevant details. In *The Wizard of Oz* (film, 1939) it is true that Dorothy wears red shoes. But the film has nothing to say about whether she likes Twinkies. By itself, then, the film does not give us enough information to say that 'Dorothy likes Twinkies' is either true or false in the story. Similarly, there is a problem when stories are inconsistent. Suppose according to some story,

(a) All dogs can fly

(b) Fido is a dog

(c) Fido cannot fly

Given (a), all dogs fly; but from (b) and (c), it seems that not all dogs fly. Given (b), Fido is a dog; but from (a) and (c) it seems that Fido is not a dog. Given (c), Fido cannot fly; but from (a) and (b) it seems that Fido can fly. The problem is not that inconsistent stories say too little, but rather that they say too much. When a story is inconsistent, we will refuse to say that it makes any sentence (simply) true or false.[2]

It will be helpful to consider some examples of consistent and inconsistent stories:

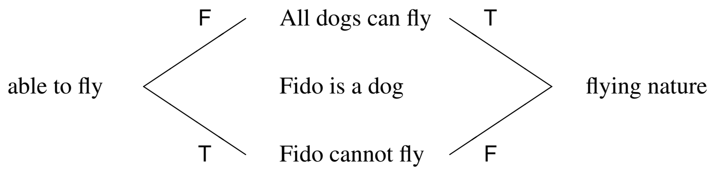
(a) The real story, "Everything is as it actually is." Since no contradiction is actually true, this story involves no contradiction; so it is internally consistent and possible.

(b) "All dogs can fly: over the years, dogs have developed extraordinarily large and muscular ears; with these ears, dogs can fly." It is bizarre, but not obviously inconsistent. If we allow the consistency of stories according to which monkeys fly, as in *The Wizard of Oz*, or elephants fly, as in *Dumbo* (films 1941, 2019), then we should allow that this story is consistent as well.

(c) "All dogs can fly, but my dog Fido cannot; Fido's ear was injured while he was chasing a helicopter, and he cannot fly." This is *not* internally consistent. If all dogs can fly and Fido is a dog, then Fido can fly. You might think that Fido retains a sort of flying nature—just because Fido remains a dog. In evaluating internal consistency, however, we require that *meanings remain the same*.

[2]The intuitive picture developed above should be sufficient for our purposes. However, we are on the verge of vexed issues. For further discussion, you may want to check out the vast literature on "possible worlds." Contributions of my own include the introductory article, "Modality," in *The Continuum Companion to Metaphysics*.

(C)



If 'can fly' means 'is able to fly' then in the story it is true that Fido cannot fly, but not true that all dogs can fly (since Fido cannot). If 'can fly' means 'has a flying nature' then in the story it is true that all dogs can fly, but not true that Fido cannot (because he remains a dog). The only way to keep both 'all dogs fly' and 'Fido cannot fly' true is to *switch* the sense of 'can fly' from one use to another. So long as 'can fly' means the same in each use, the story is sure to fall apart insofar as it says both that Fido is and is not that sort of thing.

(d) "Germany won WWII; the United States never entered the war; after a long and gallant struggle, England and the rest of Europe surrendered." It did not happen; but the story does not contradict itself. For our purposes, then, it counts as possible.

(e) "$1 + 1 = 3$; the numerals '2' and '3' are switched (the numerals are '1', '3', '2', '4', '5', '6', ...); so that one and one are three." This story does not hang together. Of course numerals can be switched—so that people would correctly say, '$1 + 1 = 3$'. But this does not make it the case that one and one are three! We tell stories in our own language (imagine that you are describing a foreign-language film in English). Take a language like English except that 'fly' means 'bark'; and consider a movie where dogs are ordinary, so that people in the movie correctly assert, in their language, 'dogs fly'. But changing the words people use to describe a situation does not change the situation. It would be a mistake to tell a friend, in English, that you saw a movie in which there were flying dogs. Similarly, according to our story, people correctly assert, in their language, '$1 + 1 = 3$'. But it is a mistake to say in English (as our story does), that this makes one and one equal to three.

Last notes:

- Some authors prefer talk of "possible worlds," "possible situations," or the like to that of consistent stories. It is conceptually simpler to stick with stories, as I have, than to have situations and distinct descriptions of them. However, it is worth recognizing that our consistent stories are or describe possible situations, so that the one notion matches up directly with the others.

- It is essential to success that you work a significant body of exercises successfully and independently: In learning logic, you acquire a *skill*. Just as a coach might help you to understand how to hit a baseball—but you learn to hit only by practice—so an instructor (or this book) may help you to understand concepts of logic, but you gain the skill only by practice. So do not neglect exercises!

- As you approach exercises, note that answers to problems indicated by star are available at `https://tonyroyphilosophy.net/symbolic-logic/`. In addition, the website makes available the Symbolic Logic APPlication (SLAPP). In version 3.0, SLAPP is an exercise editor sufficient for production of exercises from (at least) the first volume of this text (it also adds check and help for derivation exercises). Even without check and help SLAPP adds structure for responses of the sort required in this chapter, and becomes increasingly valuable as symbolic elements are introduced. So it is worth getting started with it now.

E1.1. Say whether each of the following stories is internally consistent or inconsistent. In either case, explain why.

*a. Smoking cigarettes greatly increases the risk of lung cancer, although most people who smoke cigarettes do not get lung cancer.

b. Joe is taller than Mary, but Mary is taller than Joe.

*c. Abortion is always morally wrong, though abortion is morally right in order to save a woman's life.

d. Mildred is Dr. Saunders's daughter, although Dr. Saunders is not Mildred's father.

*e. No rabbits are nearsighted, though some rabbits wear glasses.

f. Ray got an 'A' on the final exam in both Phil 200 and Phil 192. But he got a 'C' on the final exam in Phil 192.

*g. Barack Obama was never president of the United States, although Michelle is president right now.

h. Egypt, with about 100 million people is the most populous country in Africa, and Africa contains the most populous country in the world. But the United States has over 200 million people.

*i. The death star is a weapon more powerful than that in any galaxy, though there is, in a galaxy far, far away, a weapon more powerful than it.

j. Luke and the Rebellion valiantly battled the evil Empire, only to be defeated. The story ends there.

E1.2. For each of the following, (i) say whether the sentence is true or false in the real world and then (ii) say, if you can, whether the sentence is true or false according to the accompanying story. In each case, explain your answers. Do not forget about contexts where we refuse to say whether sentences are simply true or false. The first problem is worked as an example.

a. *Sentence:* Aaron Burr was never a president of the United States.

   Story: Aaron Burr was the first president of the United States, however he turned traitor and was impeached and then executed.

   (i) It is *true* in the real world that Aaron Burr was never a president of the United States. (ii) But the story makes the sentence *false*, since the story says Burr was the first president.

*b. *Sentence:* In 2006, there were still buffalo.

   Story: A thundering herd of buffalo overran Phoenix, Arizona in early 2006. The city no longer exists.

*c. *Sentence:* After overrunning Phoenix in early 2006, a herd of buffalo overran Newark, New Jersey.

   Story: A thundering herd of buffalo overran Phoenix, Arizona in early 2006. The city no longer exists.

*d. *Sentence:* There has been an all-out nuclear war.

   Story: After the all-out nuclear war, John Connor organized the resistance against the machines—who had taken over the world for themselves.

*e. *Sentence:* Barack Obama has swum the Atlantic.

   Story: No human being has swum the Atlantic. Barack Obama and Leonardo DiCaprio and you are all human beings, and at least one of you swam all the way across.

f. *Sentence:* Some people have died as a result of nuclear explosions.

   Story: As a result of a nuclear blast that wiped out most of this continent, you have been dead for over a year.

g. *Sentence:* Your instructor is not a human being.

   Story: Your instructor is a human being. However he has traveled widely and received his degree from a logic academy located on one of Saturn's moons.

h. *Sentence:* Lassie is a dog who cannot fly.

   Story: Dogs have super-big ears and have learned to fly. Indeed, all dogs can fly. However Lassie and Rin Tin Tin are dogs who cannot fly.

i. *Sentence:* The Yugo is the most expensive car in the world.

   Story: Jaguar and Rolls Royce are expensive cars. But the Yugo is more expensive than either of them.

j. *Sentence:* Lassie is a bird who has learned to fly.

   Story: Dogs have super-big ears and have learned to fly. Indeed, all dogs can fly. Among the many dogs are Lassie and Rin Tin Tin.

1.2 The Definitions

The definition of logical validity depends on what is true and false in consistent stories. The definition of soundness builds directly on the definition of validity. Note: in offering these definitions, I *stipulate* the way the terms are to be used; there is no attempt to say how they are used in ordinary conversation; rather, we say what they will mean for us in this context.

LV An argument is *logically valid* if and only if (iff) there is no consistent story in which all the premises are true and the conclusion is false.

LS An argument is *logically sound* iff it is logically valid and all of its premises are true in the real world.

Observe that logical validity has entirely to do with what is true and false in consistent stories. Only with logical soundness is validity combined with premises true in the real world.

Logical (deductive) validity and soundness are to be distinguished from *inductive* validity and soundness. For the inductive case, it is natural to focus on the *plausibility* or the *probability* of stories—where an argument is relatively strong when stories that make the premises true and conclusion false are relatively implausible. Logical (deductive) validity and soundness are thus a sort of limiting case, where stories that make premises true and conclusion false are not merely implausible, but impossible. In a deductive argument, conclusions are supposed to be *guaranteed;* in an inductive argument, conclusions are merely supposed to be made probable or plausible. For mathematical logic, we set the inductive case to the side, and focus on the deductive.

Also, do not confuse *truth* with validity and soundness. A sentence is true in the real world when it correctly represents how things are in the real world, and true in a story when it correctly represents how things are in the story. An argument is valid when there is no consistent story that makes the premises true and conclusion false, and sound when it is valid and all its premises are true in the real world. The definitions for validity and soundness depend on truth and falsity for the *premises* and *conclusion* in stories and then in the real world. But truth and falsity do not even apply to arguments: just as it is a "category" mistake to say that the number three is tall or short, so it is a mistake to say that an argument is true or false.[3]

1.2.1 Invalidity

It will be easiest to begin thinking about *invalidity*. From the definition, if an argument is logically valid, there is no consistent story that makes the premises true and

[3] From an introduction to philosophy of language, one might wonder (with good reason) whether the proper bearers of truth are sentences rather than, say, *propositions*. This question is not relevant to the simple point made above.

conclusion false. So to show that an argument is invalid, it is enough to *produce* even one consistent story that makes premises true and conclusion false. Perhaps there are stories that result in other combinations of true and false for the premises and conclusion; this does not matter for the definition. However, if there is even one story that makes premises true and conclusion false then, by definition, the argument is not logically valid—and if it is not valid, by definition, it is not logically sound.

We can work through this reasoning by means of a simple *invalidity test*. Given an argument, this test has the following four stages:

IT

a. List the premises and negation of the conclusion.
b. Produce a consistent story in which the statements from (a) are all true.
c. Apply the definition of validity.
d. Apply the definition of soundness.

We begin by considering what needs to be done to show invalidity. Then we do it. Finally we apply the definitions to get the results. For a simple example, consider the following argument:

(D)

Eating brussels sprouts results in good health
Ophelia has good health
———
Ophelia has been eating brussels sprouts

We apply the invalidity test (IT), to show that this argument is both invalid and unsound.

The definition of validity has to do with whether there are consistent stories in which the premises are true and the conclusion false. Thus, in the first stage, we simply write down what would be the case in a story of this sort.

a. List premises and negation of conclusion.	In any story with the premises true and conclusion false, 1. Eating brussels sprouts results in good health 2. Ophelia has good health 3. Ophelia has not been eating brussels sprouts

Observe that the conclusion is reversed! At this stage we are not giving an argument. Rather we merely list what is the case when the premises are true and conclusion false. Thus there is no line between premises and the last sentence, insofar as there is no suggestion of support. It is easy enough to repeat the premises for (1) and (2). Then for (3) we say what is required for the conclusion to be *false*. Thus, 'Ophelia has been eating brussels sprouts' is false if Ophelia has not been eating brussels sprouts. I return to this point below, but that is enough for now.

An argument is invalid if there is even one consistent story that makes the premises true and the conclusion false—so, since the conclusion is reversed, an argument is invalid if there is even one consistent story in which the statements from (a) are all true. Thus, to show invalidity, it is enough to *produce* a consistent story that "hits the target" from (a).

b. Produce a consistent story in which the statements from (a) are all true.	Story: Eating brussels sprouts results in good health, but eating spinach does so as well; Ophelia is in good health but has been eating spinach, not brussels sprouts.

For the statements listed in (a): the story satisfies (1) insofar as eating brussels sprouts results in good health; (2) is satisfied since Ophelia is in good health; and (3) is satisfied since Ophelia has not been eating brussels sprouts. The story *explains* how she manages to maintain her health without eating brussels sprouts, and so the consistency of (1)–(3) together. The story does not have to be true—and, of course, many different stories will do. All that matters is that there is a *consistent* story in which the premises of the original argument are true, and the conclusion is false.

Producing a story that makes the premises true and conclusion false is the creative part. What remains is to apply the definitions of validity and soundness. By LV, an argument is logically valid only if there is no consistent story in which the premises are true and the conclusion is false. So if, as we have demonstrated, there is such a story, the argument cannot be logically valid.

c. Apply the definition of validity.	This is a consistent story that makes the premises true and the conclusion false; thus, by definition, the argument is not logically valid.

By LS, for an argument to be sound, it must have its premises true in the real world *and* be logically valid. Thus if an argument fails to be logically valid, it automatically fails to be logically sound.

d. Apply the definition of soundness.	Since the argument is not logically valid, by definition, it is not logically sound.

Given an argument, the definition of validity depends on stories that make the premises true and the conclusion false. Thus, in step (a) we simply list claims required of any such story. To show invalidity, in step (b), we produce a consistent story that satisfies each of those claims. Then in steps (c) and (d) we apply the definitions to get the final results.

It may be helpful to think of stories as a sort of "wedge" to pry the premises of an argument off its conclusion. We pry the premises off the conclusion if there is a consistent way to make the premises true and the conclusion not. If it is possible to insert such a wedge between the premises and conclusion, then a defect is exposed in the way premises are connected to the conclusion. Observe that the flexibility we allow in consistent stories (with flying dogs and the like) corresponds directly to the strength of the required connection between premises and conclusion. If the connection is sufficient to resist all such attempts to wedge the premises off the conclusion, then it is significant indeed. Observe also that our method reflects what we did with argument (A) at the beginning of the chapter: Faced with the premises that only citizens can vote and Hannah is a citizen, it was natural to worry that she might be underage and so cannot vote. But this is precisely to produce a story that makes the premises true and conclusion false. Thus our method is not "strange" or "foreign"! Rather, it makes explicit what has seemed natural from the start.

Here is another example of our method. Though the argument may seem on its face not to be a very good one, we can expose its failure by our methods—in fact, again, our method may formalize or make rigorous a way you very naturally think about cases of this sort. Here is the argument:

(E)
I shall run for president
———
I shall be one of the most powerful men on earth

To show that the argument is invalid, we turn to our standard procedure:

a. In any story with the premise true and conclusion false,

   1. I shall run for president
   2. I shall not be one of the most powerful men on earth

b. Story: I do run for president, but get no financing and gain no votes; I lose the election. In the process, I lose my job as a professor and end up begging for scraps outside a Domino's Pizza restaurant. I fail to become one of the most powerful men on earth.

c. This is a consistent story that makes the premise true and the conclusion false; thus, by definition, the argument is not logically valid.

d. Since the argument is not logically valid, by definition, it is not logically sound.

This story forces a wedge between the premise and the conclusion. Thus we use the definition of validity to explain why the conclusion does not properly follow from the premises. It is, perhaps, obvious that *running* for president is not enough to make me one of the most powerful men on earth. Our method forces us to be very explicit about why: running for president leaves open the option of losing, so that the premise does not force the conclusion. Once you get used to it, then, our method may appear as a natural approach to argument evaluation.

If you follow this method for showing invalidity, the place where you are most likely to go wrong is stage (b), telling stories where the premises are true and the conclusion false. Be sure that your story is consistent, and that it verifies *each* of the claims from stage (a). If you do this, you will be fine.

E1.3. Use our invalidity test to show that each of the following arguments is not logically valid, and so not logically sound. Understand terms in their most natural sense.

*a. If Joe works hard, then he will get an 'A'
Joe will get an 'A'
———
Joe works hard

b. Harry had his heart ripped out by a government agent
———
Harry is dead

c. Everyone who loves logic is happy
Jane does not love logic
———
Jane is not happy

d. Our car will not run unless it has gasoline
Our car has gasoline
———
Our car will run

e. Only citizens can vote
Hannah is a citizen
———
Hannah can vote

1.2.2 Validity

Suppose I assert that no student at California State University San Bernardino is from Beverly Hills, and attempt to prove it by standing in front of the library and buttonholing students to ask if they are from Beverly Hills—I do this for a week and never find anyone from Beverly Hills. Is the claim that no CSUSB student is from Beverly Hills thereby proved? Of course not, for there may be students I never meet. Similarly, failure to find a story to make the premises true and conclusion false does not show that there is not one—for all we know, there might be some story we have not thought of yet. So, to show validity, we need another approach. If we could show that every story which makes the premises true and conclusion false is *inconsistent*, then we could be sure that no *consistent* story makes the premises true and conclusion false—and so, from the definition of validity, we could conclude that the argument is valid. Again, we can work through this by means of a procedure, this time a *validity test*.

VT
a. List the premises and negation of the conclusion.
b. Expose the inconsistency of such a story.
c. Apply the definition of validity.
d. Apply the definition of soundness.

In this case, we begin in just the same way. The key difference arises at stage (b). For an example, consider this argument:

(F)

No car is a person
My mother is a person
———
My mother is not a car

We apply the validity test (VT), to show that this argument is valid and then to evaluate soundness.

Since LV has to do with stories where the premises are true and the conclusion false, as before, we begin by listing the premises together with the negation of the conclusion.

a. List premises and negation of conclusion.	In any story with the premises true and conclusion false, 1. No car is a person 2. My mother is a person 3. My mother is a car

Any story where 'My mother is not a car' is false, is one where my mother is a car (perhaps along the lines of the 1965 TV series, *My Mother the Car*).

For invalidity, we would produce a consistent story in which (1)–(3) are all true. In this case, to show that the argument is valid, we show that this *cannot* be done. That is, we show that no story that makes each of (1)–(3) true is a consistent story.

b. Expose the inconsistency of such a story.	In any such story, Given (1) and (3), 4. My mother is not a person Given (2) and (4), 5. My mother is and is not a person

The reasoning should be clear if you focus *just on the specified lines*. Given (1) and (3), if no car is a person and my mother is a car, then my mother is not a person. But then my mother is a person from (2) and not a person from (4). So we have our goal: any story with (1)–(3) as members contradicts itself and therefore is not consistent. Observe that we could have reached this result in other ways. For example, we might have reasoned from (1) and (2) that (4′), my mother is not a car; and then from (3) and (4′) to the result that (5′) my mother is and is not a car. Either way, an inconsistency is exposed. Thus, as before, there are different options for this creative part.

Now we are ready to apply the definitions of logical validity and soundness. First,

c. Apply the definition of validity.	So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

For the invalidity test, we produce a consistent story that "hits the target" from stage (a) to show that the argument is invalid. For the validity test, we show that any attempt to hit the target from stage (a) must collapse into inconsistency: No consistent story includes each of the elements from stage (a) so that *there is no consistent story in which the premises are true and the conclusion false*. So by application of LV the argument is logically valid.

Given that the argument is logically valid, LS makes logical soundness depend on whether the premises are true in the real world. Suppose we think the premises of our argument are in fact true. Then,

d. Apply the definition of soundness.	In the real world no car is a person and my mother is a person, so all the premises are true; so since the argument is also logically valid, by definition, it is logically sound.

Observe that LS requires for logical soundness that an argument is logically valid and that its *premises* are true in the *real world*. Validity depends just on truth and falsity in consistent stories; setting stories to the side, soundness requires in addition that premises really are true. And we do not say anything at this stage about claims other than the premises of the original argument. Thus we do not make any claim about the truth or falsity of the conclusion, 'My mother is not a car'. Rather, the observations have entirely to do with the two premises, 'No car is a person' and 'My mother is a person'. When an argument is valid and the premises are true in the real world, by LS, it is logically sound.

But it will not always be the case that a valid argument has true premises. Say *My Mother the Car* is (surprisingly) a documentary about a person rein*car*nated as a car (the premise of the show) and therefore a true account of some car that is a person. Then some cars are persons and the first premise is false; so you would have to respond as follows:

d′. Since in the real world some cars are persons, the first premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

Another option is that you are in doubt about reincarnation into cars, and in particular about whether some cars are persons. In this case you might respond as follows:

d″. Although in the real world my mother is a person, I cannot say whether no car is a person; so I cannot say whether the first premise is true. So though the argument is logically valid, I cannot say whether it is logically sound.

So once we decide that an argument is valid, for soundness there are three options:

(i) You are in a position to identify all of the premises as true in the real world. In this case, you should do so, and apply the definition for the result that the argument is logically sound.

(ii) You are in a position to say that one or more of the premises is false in the real world. In this case, you should do so, and apply the definition for the result that the argument is not logically sound.

(iii) You cannot identify any premise as false, but neither can you identify them all as true. In this case, you should explain the situation and apply the definition for the result that you are not in a position to say whether the argument is logically sound.

So given a valid argument, there remains a substantive question about soundness. In some cases, as for example (C) on page 2, this can be the most controversial part.

Again, given an argument, we say in step (a) what would be the case in any story that makes the premises true and the conclusion false. Then, at step (b), instead of finding a consistent story in which the premises are true and conclusion false, we show that there is no such thing. Steps (c) and (d) apply the definitions for the final results.

Notice that there is an "inverse relation" between stories and validity: Stories with premises true and conclusion false *attack* an argument. If some attack succeeds, the argument fails; and if all attacks fail, the argument succeeds. So IT shows that an argument fails by finding a successful attack; VT shows that an argument succeeds by showing that attacks fail. Observe also that only one method can be correctly applied in a given case. If we can produce a consistent story according to which the premises are true and the conclusion is false, then it is not the case that no consistent story makes the premises true and the conclusion false. Similarly, if no consistent story makes the premises true and the conclusion false, then we will not be able to produce a consistent story that makes the premises true and the conclusion false.

For showing validity, the most difficult steps are (a) and (b), where we say what happens in every story where the premises true and the conclusion false. For an example, consider the following argument:

All collies can fly
(G) All collies are dogs
———
All dogs can fly

It is invalid. We can easily tell a story that makes the premises true and the conclusion false—say one where collies fly but dachshunds do not. Suppose, however, that we proceed with the validity test as follows:

a. In any story with the premises true and conclusion false,

 1. All collies can fly
 2. All collies are dogs
 3. No dogs can fly

b. In any such story,

 Given (1) and (2),
 4. Some dogs can fly
 Given (3) and (4),
 5. Some dogs can and cannot fly

c. So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

d. Since in the real world collies cannot fly, the first premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

The reasoning at (b), (c), and (d) is correct. Any story with (1)–(3) is inconsistent. But something is wrong. (Can you see what?) There is a mistake at (a): It is not the case that every story that makes the premises true and conclusion false includes (3). The negation of 'All dogs can fly' is not 'No dogs can fly', but rather, 'Not all dogs can fly' (or 'Some dogs cannot fly'). All it takes to falsify the claim that *all dogs fly* is some dog that does not (on this, see the extended discussion on the following page). So for argument (G) we have indeed shown that every story of a certain sort is inconsistent, but have not shown that every story which makes the premises true and conclusion false is inconsistent. In fact, as we have seen, there are consistent stories that make the premises true and conclusion false.

Similarly, in step (b) it is easy to get confused if you consider too much information at once. Ordinarily, if you focus on sentences singly or in pairs, it will be clear what must be the case in every story including those sentences. It does not matter which sentences you consider in what order, so long as in the end, you reach a contradiction according to which something is and is not so.

So far, we have seen our procedures applied in contexts where it is given ahead of time whether an argument is valid or invalid. But not all situations are so simple. In the ordinary case, it is not given whether an argument is valid or invalid. In this case, there is no magic way to say ahead of time which of our two tests, IT or VT applies. The only thing to do is to try one way—if it works, fine. If it does not, try the other. It is perhaps most natural to begin by looking for stories to pry the premises off the conclusion. If you can find a consistent story to make the premises true and conclusion false, the argument is invalid. If you cannot find any such story, you may begin to suspect that the argument is valid. This suspicion does not itself amount to a demonstration of validity. But you might try to turn your suspicion into such a demonstration by attempting the validity method. Again, if one procedure works, the other better not!

E1.4. Use our validity procedure to show that each of the following is logically valid, and decide (if you can) whether it is logically sound.

*a. If Barack is president, then Michelle is first lady
Michelle is not first lady
———
Barack is not president

Negation and Quantity

In general you want to be careful about negations. To negate any claim $\mathscr{P}$ it is always correct to write simply, *it is not the case that* $\mathscr{P}$. So 'It is not the case that all dogs can fly' negates 'All dogs can fly'. You may choose this approach for conclusions in the first step of our procedures. At some stage, however, you will need to understand what the negation comes to. It is easy enough to see that,

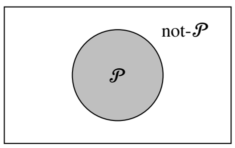
My mother is a car and My mother is not a car

negate one another. However, there are cases where caution is required. This is particularly the case with terms involving quantities.

Say the conclusion of your argument is, 'There are at least ten apples in the basket'. Clearly a story according to which there are, say, three apples in the basket makes this conclusion false. However, there are other ways to make the conclusion false—as if there are two apples or seven. Any of these are fine for showing invalidity.

But when you show that an argument is valid, you must show that *any* story that makes the premises true and conclusion false is inconsistent. So it is not sufficient to show that stories with (the premises true and) three apples in the basket contradict. Rather, you need to show that any story that includes the premises and *fewer than ten* apples fails. Thus in step (a) of our procedure we always say what is so in *every* story that makes the premises true and conclusion false. So in (a) you would have the premises and, 'There are fewer than ten apples in the basket'.

If a statement is included in some range of consistent stories, then its negation says what is so in all the others—all the ones where it is not so.



all consistent stories

That is why the negation of 'there are at least ten' is 'there are fewer than ten'.

The same point applies with other quantities. Consider some grade examples: First, if a professor says that everyone will not get an 'A', she says something disastrous—nobody in your class will get an 'A'. In order to deny it, to show that she is wrong, all you need is at least one person that gets an 'A'. In contrast, if she says that someone will not get an 'A', she says only what you expect from the start—that not everyone will get an 'A'. To deny this, you would need that everyone gets an 'A'. Thus the following pairs negate one another:

Everyone will not get an 'A'	and	Someone will get an 'A'
Someone will not get an 'A'	and	Everyone will get an 'A'

It is difficult to give rules to cover all the cases. The best is just to think about what you are saying, perhaps with reference to examples like these.

b. Only fools find love
Elvis was no fool
———
Elvis did not find love

c. If there is a good and omnipotent god, then there is no evil
There is evil
———
There is no good and omnipotent god

d. All sparrows are birds
All birds fly
———
All sparrows fly

e. All citizens can vote
Hannah is a citizen
———
Hannah can vote

E1.5. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound. Hint: You may have to do some experimenting to decide whether the arguments are logically valid or invalid—and so decide which procedure applies.

a. If Barack is president, then Michelle is first lady
Barack is president
———
Michelle is first lady

b. Most professors are insane
TR is a professor
———
TR is insane

*c. Some dogs have red hair
Some dogs have long hair
———
Some dogs have long, red hair

d. If you do not strike the match, then it does not light
The match lights
———
You strike the match

e. Brittney is taller than Steph
Steph is at least as tall as TR
———
Steph is taller than TR

1.3 Some Consequences

We now know what logical validity and soundness are, and should be able to identify them in simple cases. Still, it is one thing to know what validity and soundness are, and another to know why they matter. So in this section I turn to some consequences of the definitions.

1.3.1 Soundness and Truth

First, a consequence we want: The conclusion of every sound argument is true in the real world. Observe that this is *not* part of what we require to show that an argument is sound. LS requires just that an argument is valid and that its *premises* are true. However it is a consequence of validity plus true premises that the conclusion is true as well.

$$\text{sound} \quad \Longrightarrow \quad \frac{\text{valid} + \text{true premises}}{\text{true conclusion}}$$

By themselves, neither validity nor true premises guarantee a true conclusion. However, taken together they do. To see this, consider a two-premise argument. Say the *real* story describes the real world; so the sentences of the real story are all true in the real world. Then in the real story, the premises and conclusion of our argument must fall into one of the following combinations of true and false:

1	2	3	4	5	6	7	8	
T	T	T	F	T	F	F	F	combinations for
T	T	F	T	F	T	F	F	the real story
T	F	T	T	F	F	T	F	

These are all the combinations of T and F. Say the premises are true in the real story; this leaves open that the real story has the conclusion true as in (1) or false as in (2); so the conclusion of an argument with true premises may or may not be true in the real world. Say the argument is logically valid; then no consistent story makes the premises true and the conclusion false; but the real story is a consistent story; so we can be sure that the real story does not result in combination (2); again, though, this leaves open any of the other combinations and so that the conclusion of a valid argument may or may not be true in the real world. Now say the argument is *sound;* then it is valid and all its premises are true in the real world; again, since it is valid, the real story does not result in combination (2); and since the premises of a sound argument are true in the real world, the premises do not fall into any of the combinations (3)–(8); (1) is the only combination left: in the real story, and so in the real world, the conclusion of a sound argument is true. And not only in this case but in general, if an argument is sound then its conclusion is true in the real world: Since a sound argument is valid, there is no consistent story where its premises are true and conclusion false; so the real story does not have the premises true and conclusion false; and since the premises really are true, the conclusion is not false—and so (given the maximality of our stories) true. Put another way, if an argument is sound, its premises are true in the real story; but then if the conclusion is not true, and so false, the real story has the premises true and conclusion false—and since there is such a story, the argument is not valid. So if an argument is sound, if it is valid and its premises are true, it has a true conclusion.

Note again: We do not need that the conclusion is true in the real world in order to *decide* that an argument is sound; saying that the conclusion is true is no part of our procedure for validity or soundness. Rather, by discovering that an argument is logically valid and that its *premises* are true, we *establish* that it is sound; this gives us the result that its conclusion therefore is true. And that is just what we want.

1.3.2 Validity and Form

It is worth observing a connection between what we have done and argument form. Some of the arguments we have seen so far are of the same general *form*. Thus both arguments at (H) have the form on the right.

(H)	If Joe works hard, then he will get an ‘A’ Joe works hard ——— Joe will get an ‘A’	If Hannah is a citizen, then she can vote Hannah is a citizen ——— Hannah can vote	If $\mathcal{P}$ then $\mathcal{Q}$ $\mathcal{P}$ ——— $\mathcal{Q}$

As it turns out, all arguments of this form are valid. In contrast, the following arguments with the indicated form are not.

(I)	If Joe works hard, then he will get an ‘A’ Joe will get an ‘A’ ——— Joe works hard	If Hannah can vote, then she is a citizen Hannah is a citizen ——— Hannah can vote	If $\mathcal{P}$ then $\mathcal{Q}$ $\mathcal{Q}$ ——— $\mathcal{P}$

There are stories where, say, Joe cheats for the ‘A’, or Hannah is a citizen but not old enough to vote. In these cases, it may be that $\mathcal{P}$ results in $\mathcal{Q}$, although there are ways to have $\mathcal{Q}$ without $\mathcal{P}$—this is what the stories bring out. And, generally, it is often possible to characterize arguments by their forms, where a form is *valid* iff it has no instance that makes the premises true and the conclusion false. On this basis, form (H) above is valid, and (I) is not.

In chapters to come, we take advantage of certain very general formal or structural features of arguments to identify ones that are valid and ones that are invalid. For now, though, it is worth noting that some presentations of critical reasoning (which you may or may not have encountered), take advantage of patterns like those above, listing typical ones that are valid, and typical ones that are not (for example, Cederblom and Paulsen, *Critical Reasoning*). A student may then identify valid and invalid arguments insofar as they match the listed forms. This approach has the advantage of simplicity—and one may go quickly to applications of the logical notions for concrete cases. But the approach is limited to application of listed forms, and so to a very narrow range of arguments. LV has application to any argument whatsoever. And for our logical machine, within a certain range, we shall develop an account of validity for quite arbitrary forms. So we are pursuing a general account or theory of validity that goes well beyond the mere lists of these other more traditional approaches.

1.3.3 Relevance

Another consequence seems less welcome. Consider the following argument:

(J)
Snow is white
Snow is not white
____
All dogs can fly

It is natural to think that the premises are not connected to the conclusion in the right way—for the premises have nothing to do with the conclusion—and that this argument therefore should not be logically valid. But if it is not valid, by definition, there is a consistent story that makes the premises true and the conclusion false. And in this case there is no such story, for *no consistent story makes the premises true;* so no consistent story makes the premises true *and* the conclusion false; so, by definition, this argument is logically valid. The procedure applies in a straightforward way. Thus,

a. In any story with the premises true and conclusion false,

   1. Snow is white
   2. Snow is not white
   3. Some dogs cannot fly

b. In any such story,

   Given (1) and (2),
   4. Snow is and is not white

c. So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

d. Since in the real world snow is white, the second premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

This seems bad! Intuitively, there is something wrong with the argument. But, on our official definition, it is logically valid. One might rest content with the observation that, even though the argument is logically valid, it is not logically sound. But this does not remove the general worry. For this argument,

(K) There are fish in the sea
———
Nothing is round and not round

has all the problems of the other and is logically *sound* as well. (Why?) One might, on the basis of examples of this sort, decide to reject the (classical) account of validity with which we have been working. Some do just this.[4] But, for now, let us see what can be said in defense of the classical approach. (And the classical approach is, no doubt, the approach you have seen or will see in any standard course on critical thinking or logic.)

As a first line of defense, one might observe that the conclusion of every sound argument is true and ask, "What more do you want?" We use arguments to demonstrate the truth of conclusions. And nothing we have said suggests that sound arguments do not have true conclusions: An argument whose premises are inconsistent is sure to be unsound. And an argument whose conclusion cannot be false is sure to have a true conclusion. So soundness may seem sufficient for our purposes. Even though we accept that there remains something about argument goodness that soundness leaves behind, we can insist that soundness is useful as an intellectual tool. Whenever it is the truth or falsity of a conclusion that matters, we can profitably employ the classical notions.

But one might go further, and dispute even the suggestion that there is something about argument goodness that soundness leaves behind. Consider the following two argument forms:

(ds) $\dfrac{\mathcal{P} \text{ or } \mathcal{Q},\ \text{not-}\mathcal{P}}{\mathcal{Q}}$ (add) $\dfrac{\mathcal{P}}{\mathcal{P} \text{ or } \mathcal{Q}}$

According to ds (*disjunctive syllogism*), if you are given that $\mathcal{P}$ or $\mathcal{Q}$ and that not-$\mathcal{P}$, you can conclude that $\mathcal{Q}$. If you have cake or ice cream, and you do not have cake, you have ice cream; if you are in California or New York, and you are not in California, you are in New York; and so forth. Thus ds seems hard to deny. And similarly for add (*addition*). Where 'or' means 'one or the other or both', when you are given that $\mathcal{P}$, you can be sure that $\mathcal{P}$ or anything. Say you have cake, then you have cake or ice cream, cake or brussels sprouts, and so forth; if grass is green, then grass is green or pigs have wings, grass is green or dogs fly, and so forth.

Return now to our problematic argument. As we have seen, it is valid according to the classical definition LV. We get a similar result when we apply the ds and add principles.

[4]Especially the so-called "relevance" logicians. For an introduction, see Graham Priest, *Non-Classical Logics*. But his text presumes mastery of material corresponding to Part I and Part II of this one. So the non-classical approaches develop or build on the classical one developed here.

1.	Snow is white	premise
2.	Snow is not white	premise
3.	Snow is white or all dogs can fly	from 1 and add
4.	All dogs can fly	from 2 and 3 and ds

If snow is white, then snow is white or anything. So snow is white or dogs fly. So we use line 1 with add to get line 3. But if snow is white or dogs fly, and snow is not white, then dogs fly. So we use lines 2 and 3 with ds to reach the final result. So our principles ds and add go hand in hand with the classical definition of validity. The argument is valid on the classical account; and with these principles, we can move from the premises to the conclusion. If we want to reject the validity of this argument, we will have to reject not only the classical notion of validity, but also one of our principles ds or add. And it is not obvious that one of the principles should go. If we decide to retain both ds and add then, seemingly, the classical definition of validity should stay as well. If we have intuitions according to which ds and add should stay, and also that the definition of validity should go, we have conflicting intuitions. Thus our intuitions might, at least, sensibly be resolved in the classical direction.

These issues are complex, and a subject for further discussion. For now, it is enough for us to treat the classical approach as a useful tool: It is useful in contexts where what we care about is whether conclusions are true. And alternate approaches to validity typically develop or modify the classical approach. So it is natural to begin where we are, with the classical account. At any rate, this discussion constitutes a sort of acid test: If you understand the validity of the "snow is white" and "fish in the sea" arguments (J) and (K), you are doing well—you understand *how* the definition of validity works, with its results that may or may not now seem controversial. If you do not see what is going on in those cases, then you have not yet understood how the definitions work and should return to section 1.2 with these cases in mind.

E1.6. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound. Hint: You may have to do some experimenting to decide whether the arguments are logically valid or invalid—and so decide which procedure applies.

a. Bob is over six feet tall
   Bob is under six feet tall
   ———
   Bob is disfigured

b. Marilyn is not over six feet tall
   Marilyn is not under six feet tall
   ———
   Marilyn is not in the WNBA

c. There are fish in the sea
   ———
   Nothing is round and not round

Classical Validity

As we have mentioned, there are approaches to validity other than classical. But the classical account remains the one developed in any standard course on critical reasoning or logic. Not every course "exposes" cases like (J) but, insofar as the classical definition is employed, all have the same result. Still, there are different *formulations* of the classical account which may obscure underlying equivalence. Here are some different formulations, the first three bad, the last three good:

(1) Sometimes it is said that an argument is valid iff the premises *logically entail* the conclusion. On its face, this defines validity by a notion equally in need of definition. It might be made adequate by an account of logical entailment, perhaps along the lines of one of the accounts below.

(2) It will not do to characterize valid arguments saying, "if the premises are true then the conclusion is true." For consider a true conclusion, as 'Dogs bark'; then any premises are such that if they are true then the conclusion is true. But, say, the argument "There are fish in the sea, so Dogs bark" has stories with the premise true and conclusion false and so is not logically valid.

(3) Similarly it is a mistake to characterize valid arguments saying "*if* the premises are true *then* the conclusion must be true. For consider a valid argument as, "I am less than 100 miles from Los Angeles, so I am less than 200 miles from Los Angeles." The premise is true (of me now); so on this account, the conclusion must be true; but the conclusion 'I am less than 200 miles from Los Angeles' is not such that *it* must be true—there are consistent stories where I am, say, in London right now.

(4*) Perhaps, though, (3) is a sloppy way of saying, "it *must* be that if the premises are true then the conclusion is true." So the conditional, not the conclusion, is true in all consistent stories. This is equivalent to LV. The conditional is necessarily true iff every consistent story with the premises true has the conclusion true; and this is so just in case none has the premises true and conclusion false.

(5*) Given the match between stories and possibility, LV is straightforwardly equivalent to an account on which an argument is logically valid iff it is *not possible* for the premises to be true and the conclusion false—although, by the appeal to stories, we have attempted to give some substance to the relevant notion of possibility.

(6*) Another option is to say an argument is valid iff it has some valid *form* (see section 1.3.2). This is not equivalent to LV, but remains a version of the classical account. Formally valid arguments are logically valid. But an argument can be logically valid without being formally valid. Return to the example from (3). It is valid by LV. But it has form "$\mathscr{P}$ so $\mathscr{Q}$" of which there are (many) instances with the premise true and conclusion false. Still, (J) has form "$\mathscr{P}$, not-$\mathscr{P}$, so $\mathscr{Q}$" of which there are no instances that make the premises true—thus the form comes out valid, and (J) as well.

*d. Cheerios are square
Chex are round
―――――
There is no round square

e. All dogs can fly
Fido is a dog
Fido cannot fly
―――――
I am blessed

E1.7. Respond to each of the following.

*a. Create another argument of the same form as the first set of examples (H) from section 1.3.2, and then use our regular procedures to decide whether it is logically valid and sound. Is the result what you expect? Explain.

b. Create another argument of the same form as the second set of examples (I) from section 1.3.2, and then use our regular procedures to decide whether it is logically valid and sound. Is the result what you expect? Explain.

E1.8. Which of the following are true, and which are false? In each case, explain your answers, with reference to the relevant definitions. The first is worked as an example.

a. A logically valid argument is always logically sound.

False. An argument is sound iff it is logically valid and all of its premises are true in the real world. Thus an argument might be valid but fail to be sound if one or more of its premises is false in the real world.

b. A logically sound argument is always logically valid.

*c. If the conclusion of an argument is true in the real world, then the argument must be logically valid.

d. If the premises and conclusion of an argument are true in the real world, then the argument must be logically sound.

*e. If a premise of an argument is false in the real world, then the argument cannot be logically valid.

f. If an argument is logically valid, then its conclusion is true in the real world.

*g. If an argument is logically sound, then its conclusion is true in the real world.

h. If an argument has contradictory premises (its premises are true in no consistent story), then it cannot be logically valid.

*i. If the conclusion of an argument cannot be false (is false in no consistent story), then the argument is logically valid.

j. The premises of every logically valid argument are relevant to its conclusion.

E1.9. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Logical validity.

b. Logical soundness.

E1.10. Do you think we should accept the classical account of validity? In an essay of about two pages, explain your position, with special reference to difficulties raised in section 1.3.3.

Chapter 2

Formal Languages

Having said in Chapter 1 what validity and soundness are, we now turn to our logical machine. As depicted in the picture of elements for symbolic logic on page 2, this machine begins with symbolic representations of ordinary reasoning. In this chapter we introduce the formal languages by introducing their *grammar* or *syntax*. After some brief introductory remarks in section 2.1, the chapter divides into sections that introduce grammar for a *sentential* language $\mathscr{L}_s$ (section 2.2), and then the grammar for an extended *quantificational* language $\mathscr{L}_q$ (section 2.3).

2.1 Introductory

There are different ways to introduce a formal language. It is natural to introduce expressions of a new language in relation to expressions of one that is already familiar. Thus a traditional course in a foreign language is likely to present vocabulary lists of the sort,

chou:	cabbage
petit:	small
⋮	

But the terms of a foreign language are not *originally defined* by such lists. Rather French, in this case, has conventions of its own such that sometimes *'chou'* corresponds to 'cabbage' and sometimes it does not. It is not a legitimate criticism of a Frenchman who refers to his sweetheart as *mon petit chou* to observe that she is no cabbage! (Indeed, in this context, *chou* is *chou à la crème*—a "cabbage-shaped" cream puff—and works like 'sweet' or 'honey' in English.) Although it is possible to use such lists to introduce the conventions of a new language, it is also possible to introduce a language "as itself"—the way a native speaker learns it. In this case, one avoids the danger of importing conventions and patterns from one language onto the other. Similarly, the expressions of a formal language might be introduced in correlation with expressions of, say, English. But this runs the risk of obscuring just what the official definitions accomplish. Since we will be concerned extensively with what follows from the definitions, it is best to introduce our languages in their "pure" forms.

In this chapter, we develop the *grammar* of our formal languages. Consider the following algebraic expressions:

$$a + b = c \qquad\qquad a + \ = c$$

Until we know what numbers are assigned to the terms (as $a = 1, b = 2, c = 3$), we cannot evaluate the first for truth or falsity. Still, we can say that it is grammatical and so *capable* of truth and falsity in a way that the other is not. Similarly, we shall be able to evaluate the grammar of formal expressions apart from truth and falsity—we do not have to know what the language represents in order to decide if its expressions are grammatically correct. Or, again, just as a computer can check the spelling and grammar of English without reference to meaning, so we can introduce the vocabulary and grammar of our formal languages without reference to what their expressions mean or what makes them true. The grammar, taken alone, is completely straightforward. Taken this way, we work directly from the definitions, without "pollution" from associations with English or whatever.

So we want the definitions. Even so, it may be helpful to offer some hints that foreshadow how things will go. Do not take these as defining anything! Still, it is nice to have a sense of how it fits together. Consider some simple sentences of an ordinary language, say, 'The butler is guilty' and 'The maid is guilty'. It will be convenient to introduce capital letters corresponding to these, say, B and M. Such sentences may combine to form ones that are more complex as, '*It is not the case that* the butler is guilty' or '*If* the butler is guilty, *then* the maid is guilty'. We shall find it convenient to express these, '$\sim$the butler is guilty' and 'the butler is guilty $\rightarrow$ the maid is guilty', with operators $\sim$ and $\rightarrow$. Putting these together we get, $\sim B$ and $B \rightarrow M$. Operators may be combined in obvious ways so that $B \rightarrow \sim M$ says that if the butler is guilty then the maid is not. And so forth. We shall see that incredibly complex expressions of this sort are possible!

In this case, simple sentences, 'The butler is guilty' and 'The maid is guilty' are "atoms" and complex sentences are built out of them. This is characteristic of the *sentential* languages to be considered in section 2.2. For the *quantificational* languages of section 2.3, certain sentence *parts* are taken as atoms. So quantificational languages expose structure beyond that for the sentential case. Perhaps, though, this will be enough to give you a glimpse of the overall strategy and aims for the formal languages of which we are about to introduce the grammar.

2.2 Sentential Languages

Just as algebra or English have their own vocabulary or symbols and then grammatical rules for the way the vocabulary is combined, so our formal language has its own vocabulary and then grammatical rules for the way the vocabulary is combined. In this section we introduce the vocabulary for a sentential language, introduce the grammatical rules, and conclude with some discussion of abbreviations for official expressions.

2.2.1 Vocabulary

We begin, then, with the vocabulary. In this section, we say which symbols are included in the language, and introduce some conventions for talking about the symbols.

For any sentential language $\mathcal{L}$, vocabulary includes,

VC (p) Punctuation symbols: ( )

(o) Operator symbols: $\sim$ $\rightarrow$

(s) A non-empty countable collection of sentence letters

And that is all. $\sim$ is *tilde* and $\rightarrow$ is *arrow*.[1] In order to fully specify the vocabulary of any particular sentential language, we need to identify its sentence letters—so far as definition VC goes, different languages may differ in their collections of sentence letters. The only constraint on such specifications is that the collections of sentence letters be non-empty and countable. A collection is *non-empty* iff it has at least one member. So any sentential language has at least one sentence letter. A collection is *countable* iff its members can be matched one-to-one with all (or some) of the non-negative integers. Thus we might let the sentence letters be $A, B, \ldots, Z$, where these correlate with the integers $1 \ldots 26$. Or we might let there be infinitely many sentence letters, $S_0, S_1, S_2, \ldots$ where the letters are correlated with the integers by their subscripts.

So there is room for different sentential languages. Having made this point, though, we immediately focus on a standard sentential language $\mathcal{L}_s$ whose sentence letters are Roman italics $A \ldots Z$ with or without positive integer subscripts. Thus,

$$A \qquad B \qquad K \qquad Z$$

are sentence letters of $\mathcal{L}_s$. Similarly,

$$A_1 \qquad B_3 \qquad K_7 \qquad Z_{23}$$

[1] Sometimes sentential languages are introduced with different symbols, for example, $\neg$ for $\sim$, or $\supset$ for $\rightarrow$. It should be easy to convert between presentations of the different sorts. And sometimes sentential languages include operators in addition to $\sim$ and $\rightarrow$ (for example, $\vee$, $\wedge$, $\leftrightarrow$). Such symbols will be introduced in due time—but as abbreviations for complex official expressions.

are sentence letters of $\mathcal{L}_s$. We will not use the subscripts very often, but they do guarantee that we never run out of sentence letters. Perhaps surprisingly, as described in the box on the next page (and E2.2), these letters too can be correlated with the non-negative integers. Official sentences of $\mathcal{L}_s$ are built out of this vocabulary.

To proceed, we need some conventions for talking *about* expressions of a language like $\mathcal{L}_s$. Here, $\mathcal{L}_s$ is an *object* language—the thing we want to talk about; and we require conventions for the *metalanguage*—for talking about the object language. In general, for any formal object language $\mathcal{L}$, an *expression* is a sequence of one or more elements of its vocabulary. Thus $(A \to B)$ is an expression of $\mathcal{L}_s$, but $(A \star B)$ is not. (What is the difference?) We shall use script characters $\mathcal{A} \dots \mathcal{Z}$ as variables that range over expressions. Then '$\sim$', '$\to$', '(', and ')' represent themselves. And concatenated or joined symbols in the metalanguage represent the concatenation of the symbols they represent.

To see how this works, think of metalinguistic expressions as "mapping" to object-language ones. Thus, for example, where $\mathcal{S}$ represents an arbitrary sentence letter, $\sim\mathcal{S}$ may represent any of, $\sim A$, $\sim B$, or $\sim Z$. But $\sim\mathcal{S}$ does not represent $\sim(A \to B)$, for it does not consist of a tilde followed by a sentence letter. With $\mathcal{S}$ restricted to sentence letters, there is a straightforward map from $\sim\mathcal{S}$ onto $\sim A$, $\sim B$, or $\sim Z$, but not from $\sim\mathcal{S}$ onto $\sim(A \to B)$.

$$\text{(A)} \qquad \begin{array}{cccc} \sim\mathcal{S} & \sim\mathcal{S} & \sim\mathcal{S} & \sim\mathcal{S} \\ \downarrow\downarrow & \downarrow\downarrow & \downarrow\downarrow & \downarrow \\ \sim A & \sim B & \sim Z & \sim\overbrace{(A \to B)}^{?} \end{array}$$

In the first three cases, $\sim$ maps to itself, and $\mathcal{S}$ to a sentence letter. In the last case there is no map. We might try mapping $\mathcal{S}$ to A or B; but this would leave the rest of the expression unmatched. While there is no map from $\sim\mathcal{S}$ to $\sim(A \to B)$, there is a map from $\sim\mathcal{P}$ to $\sim(A \to B)$ if we let $\mathcal{P}$ represent any arbitrary expression, for $\sim(A \to B)$ consists of a tilde followed by an expression of some sort. Metalinguistic expressions give the *form* of ones in the object language. An object-language expression has some form just when there is a complete map from the metalinguistic expression to it.

Say $\mathcal{P}$ represents any arbitrary expression. Then by similar reasoning, $(A \to B) \to (A \to B)$ is of the form $\mathcal{P} \to \mathcal{P}$.

$$\text{(B)} \qquad \begin{array}{ccc} & \mathcal{P} \to \mathcal{P} & \\ \swarrow & \downarrow & \searrow \\ \overbrace{(A \to B)} & \to & \overbrace{(A \to B)} \end{array}$$

In this case, $\mathcal{P}$ maps to all of $(A \to B)$ and $\to$ to itself. A constraint on our maps is that the use of the metavariables $\mathcal{A} \dots \mathcal{Z}$ must be consistent within a given map. Thus $(A \to B) \to (B \to B)$ is not of the form $\mathcal{P} \to \mathcal{P}$.

Countability

To see the full range of languages which are allowed under VC, observe how multiple infinite series of sentence letters may satisfy the countability constraint. Thus, for example, suppose we have two series of sentence letters, $A_0, A_1, \dots$ and $B_0, B_1, \dots$. These can be correlated with the non-negative integers as follows:

$$\begin{array}{ccccccc} A_0 & B_0 & A_1 & B_1 & A_2 & B_2 & \\ | & | & | & | & | & | & \dots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

For any non-negative integer n, A_n is matched with $2n$, and B_n with $2n + 1$. So each sentence letter is matched with some non-negative integer; so the sentence letters are countable. If there are three series, they may be correlated,

$$\begin{array}{ccccccc} A_0 & B_0 & C_0 & A_1 & B_1 & C_1 & \\ | & | & | & | & | & | & \dots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

so that every sentence letter is matched to some non-negative integer. And similarly for any finite number of series. And there might be 26 such series, as for our language $\mathcal{L}_s$.

In fact even this is not the most general case. If there are *infinitely* many series of sentence letters, we can still line them up and correlate them with the non-negative integers. Here is one way to proceed. Order the letters as follows:

$$\begin{array}{cccccccc} A_0 & \to & A_1 & & A_2 & \to & A_3 & \cdots \\ & \swarrow & & \nearrow & & \swarrow & & \\ B_0 & & B_1 & & B_2 & & B_3 & \\ \downarrow & \nearrow & & \swarrow & & & & \\ C_0 & & C_1 & & C_2 & & C_3 & \\ & \swarrow & & & & & & \\ D_0 & & D_1 & & D_2 & & D_3 & \\ \vdots & & & & & & & \end{array}$$

so that any letter appears somewhere along the arrows. Then following the arrows, match them accordingly with the non-negative integers,

$$\begin{array}{ccccccc} A_0 & A_1 & B_0 & C_0 & B_1 & A_2 & \\ | & | & | & | & | & | & \dots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

so that, again, any sentence letter is matched with some non-negative integer. It may seem odd that we can line symbols up like this, but it is hard to dispute that we have done so. Thus we may say that VC is compatible with a wide variety of specifications, but also that all legitimate specifications have something in common: If a collection is countable, it is possible to sort its members into a series with a first member, a second member, and so forth.

(C)
$$\begin{array}{c} \mathcal{P} \to \mathcal{P} \\ \swarrow \quad \downarrow \qquad ? \\ \overbrace{(A \to B)} \to \overbrace{(B \to B)} \end{array} \qquad \text{or} \qquad \begin{array}{c} \mathcal{P} \to \mathcal{P} \\ ? \qquad \downarrow \quad \searrow \\ \overbrace{(A \to B)} \to \overbrace{(B \to B)} \end{array}$$

We are free to associate $\mathcal{P}$ with whatever we want. However, within a given map, once $\mathcal{P}$ is associated with some expression, we have to use it consistently within that map.

Observe again that $\sim\mathcal{S}$ and $\mathcal{P} \to \mathcal{P}$ are not expressions of $\mathcal{L}_s$. Rather, we use them to talk about expressions of $\mathcal{L}_s$. And it is important to see how we can use the metalanguage to make claims about a range of expressions all at once. Given that $\sim A$, $\sim B$, and $\sim Z$ are all of the form $\sim\mathcal{S}$, when we make some claim about expressions of the form $\sim\mathcal{S}$, we say something about each of them—but not about $\sim(A \to B)$. Similarly, if we make some claim about expressions of the form $\mathcal{P} \to \mathcal{P}$, we say something with application to a range of expressions. In the next section, for the specification of *formulas*, we use the metalanguage in just this way.

E2.1. Assuming that $\mathcal{S}$ may represent any sentence letter, and $\mathcal{P}$ any arbitrary expression of $\mathcal{L}_s$, use maps to determine whether each of the following expressions is (i) of the form $(\mathcal{S} \to \sim\mathcal{P})$ and then (ii) whether it is of the form $(\mathcal{P} \to \sim\mathcal{P})$. In each case, explain your answers.

a. $(A \to \sim A)$

b. $(A \to \sim(R \to \sim Z))$

c. $(\sim A \to \sim(R \to \sim Z))$

d. $((R \to \sim Z) \to \sim(R \to \sim Z))$

*e. $((\to \sim) \to \sim(\to \sim))$

E2.2. On the pattern of examples from the countability guide on page 35, show that the sentence letters of $\mathcal{L}_s$ are countable—that is, that they can be correlated with the non-negative integers. On the scheme you produce, what numbers correlate with A, B_1, and C_{10}? Hint: Supposing that A without subscript is like A_0, for any subscript n, you should be able to produce a formula for the position of A_n, and similarly for B_n, C_n, and the like. Then it will be easy to find the position of any letter, even if the question is about, say, L_{125}.

2.2.2 Formulas

We are now in a position to say which expressions of a sentential language are its grammatical *formulas* and *sentences*. The specification itself is easy. We will spend a bit more time explaining how it works. For a given sentential language $\mathcal{L}$, FR (s) If $\mathcal{S}$ is a sentence letter, then $\mathcal{S}$ is a *formula*.

(∼) If $\mathcal{P}$ is a formula, then $\sim\mathcal{P}$ is a *formula*.

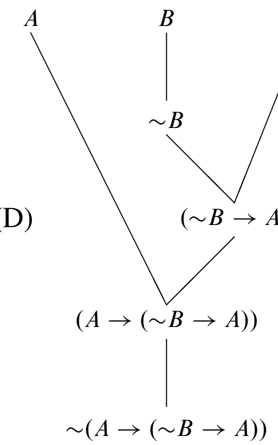
(→) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *formula*.

(CL) Any formula may be formed by repeated application of these rules.

And we simply identify the formulas with the sentences. For any sentential language $\mathcal{L}$, an expression is a *sentence* iff it is a formula.

FR is a first example of a *recursive* definition. Such definitions always build from the parts to the whole. Frequently we can use "tree" diagrams to see how they work. Thus, for example, by repeated applications of the definition, $\sim(A \rightarrow (\sim B \rightarrow A))$ is a formula and sentence of $\mathcal{L}_s$.

(D)



These are formulas by FR(s)

Since B is a formula, this is a formula by FR(∼)

Since $\sim B$ and A are formulas, this is a formula by FR(→)

Since A and $(\sim B \rightarrow A)$ are formulas, this is a formula by FR(→)

Since $(A \rightarrow (\sim B \rightarrow A))$ is a formula, this is a formula by FR(∼)

By FR(s), the sentence letters, A, B, and A are formulas; given this, clauses FR(∼) and FR(→) let us conclude that other, more complex, expressions are formulas as well. Notice that, in the definition, $\mathcal{P}$ and $\mathcal{Q}$ may be any expressions that are formulas: By FR(∼), if B is a formula, then tilde followed by B is a formula; but similarly, if $\sim B$ and A are formulas, then an opening parenthesis followed by $\sim B$, followed by → followed by A and then a closing parenthesis is a formula; and so forth as on the tree above. You should follow through each step very carefully.

A recursive definition always involves some "basic" starting elements, in this case, sentence letters. These occur across the top row of our tree. Other elements are constructed, by the definition, out of ones that come before. The last, *closure*, clause tells us that any formula is built this way. To demonstrate that an expression is a formula and a sentence, it is sufficient to construct it, according to the definition, on a tree. If an expression is not a formula, there will be no way to construct it according to the rules. Thus $(A{\sim}B)$ for example, is not a formula. A is a formula and $\sim B$ is a formula; but there is no way to put them together, by the definition, without → in between.

Here are a couple of last examples which emphasize the point that *you must maintain and respect parentheses* in the way you construct a formula. Thus consider,

(E)

$A \qquad\qquad B$	These are formulas by FR(s)
$(A \rightarrow B)$	Since A and B are formulas, this is a formula by FR($\rightarrow$)
$\sim(A \rightarrow B)$	Since $(A \rightarrow B)$ is a formula, this is a formula by FR($\sim$)

And compare it with,

(F)

$A \qquad\qquad B$	These are formulas by FR(s)
$\sim A$	Since A is a formula, this is a formula by FR($\sim$)
$(\sim A \rightarrow B)$	Since $\sim A$ and B are formulas, this is a formula by FR($\rightarrow$)

Once you have $(A \rightarrow B)$ as in the first case, the only way to apply FR($\sim$) puts the tilde on the outside. To get the tilde inside the parentheses it has to go on first, as in the second case. The significance of this point emerges immediately below.

It will be helpful to have some additional definitions, each of which may be introduced in relation to the trees. Restrict attention to trees that branch in the usual way: without extraneous nodes not required for the result, and without nodes used more than once (so for every node, there is a unique upward path from the root to it). Then for any formula $\mathcal{P}$, each formula which appears in the tree for $\mathcal{P}$ including $\mathcal{P}$ itself is a *subformula* of $\mathcal{P}$. Thus $\sim(A \rightarrow B)$ has subformulas:

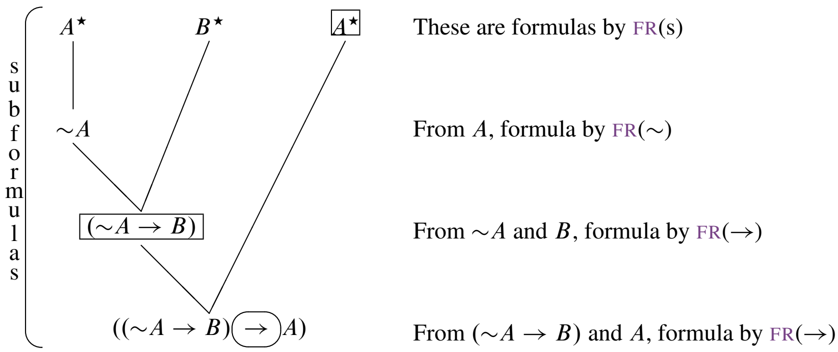
$$A \qquad\qquad B \qquad\qquad (A \rightarrow B) \qquad\qquad \sim(A \rightarrow B)$$

In contrast, $(\sim A \rightarrow B)$ has subformulas:

$$A \qquad\qquad B \qquad\qquad \sim A \qquad\qquad (\sim A \rightarrow B)$$

So it matters for the subformulas how the tree is built. The *immediate* subformulas of a formula $\mathcal{P}$ are the subformulas to which $\mathcal{P}$ is directly connected by lines. Thus $\sim(A \rightarrow B)$ has one immediate subformula, $(A \rightarrow B)$; $(\sim A \rightarrow B)$ has two, $\sim A$ and B. The *atomic* subformulas of a formula $\mathcal{P}$ are the sentence letters that appear across the top row of its tree. Thus both $\sim(A \rightarrow B)$ and $(\sim A \rightarrow B)$ have A and B as their atomic subformulas. Finally, the *main operator* of a formula $\mathcal{P}$ is the last operator added in its tree. Thus $\sim$ is the main operator of $\sim(A \rightarrow B)$, and $\rightarrow$ is the main operator of $(\sim A \rightarrow B)$. So, again, it matters how the tree is built. We sometimes speak of a formula by means of its main operator: A formula of the form $\sim\mathcal{P}$ is a *negation;* a formula of the form $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *(material) conditional*, where $\mathcal{P}$ is the *antecedent* of the conditional and $\mathcal{Q}$ is the *consequent*. Because it operates on the two immediate subformulas, $\rightarrow$ is a *binary* operator; because it has just one $\sim$ is *unary*.

E2.3. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathcal{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator. A first case for $((\sim A \to B) \to A)$ is worked as an example.



*a. A

b. $\sim\sim\sim A$

c. $\sim(\sim A \to B)$

d. $(\sim C \to \sim(A \to \sim B))$

e. $(\sim(A \to B) \to (C \to \sim A))$

E2.4. Explain why the following expressions are not formulas or sentences of $\mathcal{L}_s$. Hint: You may find that an attempted tree will help you see or explain what is wrong.

a. $(A \supset B)$

Parts of a Formula

The parts of a formula are here defined in relation to its tree.

SB Each formula which appears in the tree for formula $\mathcal{P}$ including $\mathcal{P}$ itself is a *subformula* of $\mathcal{P}$.

IS The *immediate* subformulas of a formula $\mathcal{P}$ are the subformulas to which $\mathcal{P}$ is directly connected by lines.

AS The *atomic* subformulas of a formula $\mathcal{P}$ are the sentence letters that appear across the top row of its tree.

MO The *main operator* of a formula $\mathcal{P}$ is the last operator added in its tree.

*b. $(\mathcal{P} \rightarrow \mathcal{Q})$

c. $(\sim B)$

d. $(A \rightarrow \sim B \rightarrow C)$

e. $((A \rightarrow B) \rightarrow \sim(A \rightarrow C) \rightarrow D)$

E2.5. For each of the following expressions, determine whether it is a formula and sentence of $\mathcal{L}_s$. If it is, show it on a tree, and exhibit its parts as in E2.3. If it is not, explain why as in E2.4.

*a. $\sim((A \rightarrow B) \rightarrow (\sim(A \rightarrow B) \rightarrow A))$

b. $\sim(A \rightarrow B \rightarrow (\sim(A \rightarrow B) \rightarrow A))$

*c. $\sim(A \rightarrow B) \rightarrow (\sim(A \rightarrow B) \rightarrow A)$

d. $\sim\sim\sim(\sim\sim\sim A \rightarrow \sim\sim\sim A)$

e. $((\sim(A \rightarrow B) \rightarrow (\sim C \rightarrow D)) \rightarrow \sim(\sim(E \rightarrow F) \rightarrow G))$

2.2.3 Abbreviations

We have completed the official grammar for our sentential languages. So far, the languages are relatively simple. When we turn to reasoning about logic (in later parts), it will be good to have our languages as simple as we can. However, for applications of logic it will be advantageous to have additional expressions which, though redundant with expressions of the language already introduced, simplify the work. I begin by introducing these additional expressions, and then turn to the question about how to understand the redundancy.

Abbreviating. As may already be obvious, formulas of a sentential language like $\mathcal{L}_s$ can get complicated quickly. Abbreviated forms give us ways to manipulate official expressions without undue pain. First, for any formulas $\mathcal{P}$ and $\mathcal{Q}$,

AB ($\vee$) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \rightarrow \mathcal{Q})$

($\wedge$) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})$

($\leftrightarrow$) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \rightarrow \mathcal{P}))$

The last of these is easier than it looks; I say something about this below. $\vee$ is *wedge*, $\wedge$ is *caret*, and $\leftrightarrow$ is *double arrow*. An expression of the form $(\mathcal{P} \vee \mathcal{Q})$ is a *disjunction* with $\mathcal{P}$ and $\mathcal{Q}$ as *disjuncts;* it has the standard reading, ($\mathcal{P}$ *or* $\mathcal{Q}$). An expression of the form $(\mathcal{P} \wedge \mathcal{Q})$ is a *conjunction* with $\mathcal{P}$ and $\mathcal{Q}$ as *conjuncts;*

it has the standard reading, ($\mathcal{P}$ *and* $\mathcal{Q}$). An expression of the form $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *(material) biconditional;* it has the standard reading, ($\mathcal{P}$ *iff* $\mathcal{Q}$).[2] Again, we do not use ordinary English to define our symbols. All the same, this should suggest how the extra operators extend the range of what we are able to say in a natural way.

With the abbreviations, we are in a position to introduce derived clauses for FR. Suppose $\mathcal{P}$ and $\mathcal{Q}$ are formulas; then by FR($\sim$), $\sim\mathcal{P}$ is a formula; so by FR($\rightarrow$), $(\sim\mathcal{P} \rightarrow \mathcal{Q})$ is a formula; but this is just to say that $(\mathcal{P} \vee \mathcal{Q})$ is a formula. And similarly in the other cases. (If you are confused by such reasoning, work it out on a tree.) Thus we arrive at the following conditions:

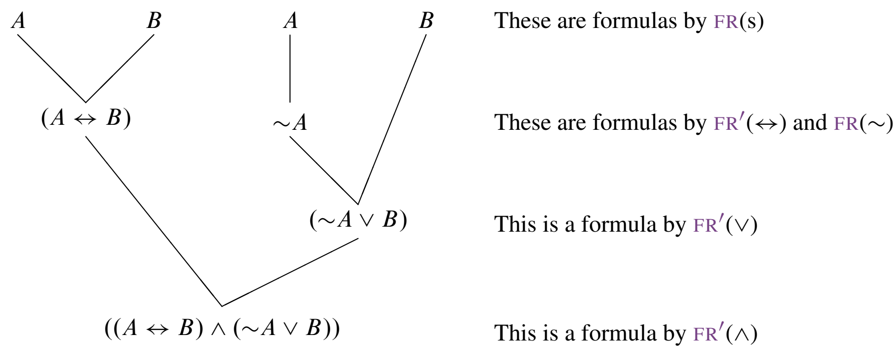
FR$'$ ($\vee$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.

($\wedge$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.

($\leftrightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.

Once FR is extended in this way, the additional conditions may be applied directly in trees. Thus, for example, if $\mathcal{P}$ is a formula and $\mathcal{Q}$ is a formula, we can safely move in a tree to the conclusion that $(\mathcal{P} \vee \mathcal{Q})$ is a formula by FR$'$($\vee$). Similarly, for a more complex case, $((A \leftrightarrow B) \wedge (\sim A \vee B))$ is a formula.

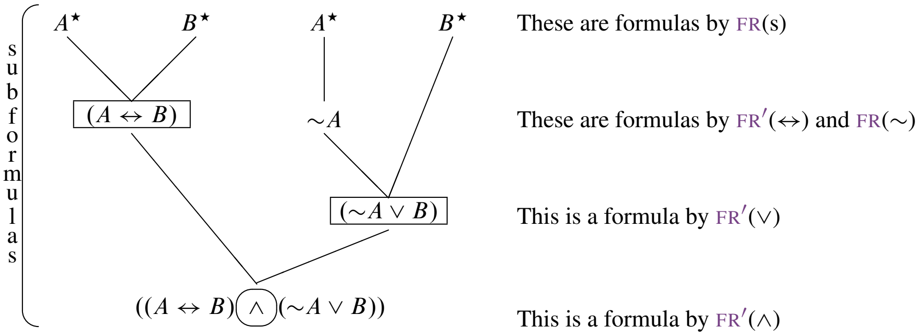
(G)



In a derived sense, expressions with the new symbols have *subformulas*, *atomic* subformulas, *immediate* subformulas, and *main operator* all as before. Thus on the diagram immediately above, with notation from exercises—bracket for subformulas, star for atomic subformulas, box for immediate subformulas, and circle for main operator:

[2] Common alternatives are & for $\wedge$, and $\equiv$ for $\leftrightarrow$. Less common nowadays is a dot (period) for $\wedge$.

(H)



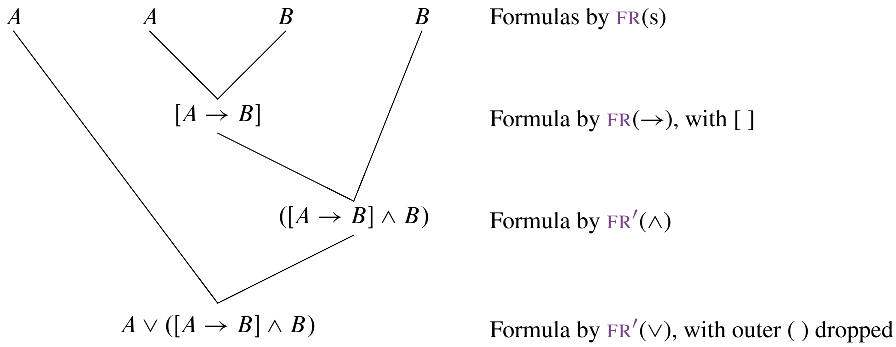
In the derived sense, $((A \leftrightarrow B) \wedge (\sim A \vee B))$ has immediate subformulas $(A \leftrightarrow B)$ and $(\sim A \vee B)$, and main operator $\wedge$.

Return to the case of $(\mathcal{P} \leftrightarrow \mathcal{Q})$ and observe that it can be thought of as based on a simple abbreviation of the sort we expect. That is, $((\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P}))$ is of the sort $(\mathcal{A} \wedge \mathcal{B})$; so by AB($\wedge$), it abbreviates $\sim(\mathcal{A} \to \sim\mathcal{B})$; but with $(\mathcal{P} \to \mathcal{Q})$ for $\mathcal{A}$ and $(\mathcal{Q} \to \mathcal{P})$ for $\mathcal{B}$, this is just, $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$ as in AB($\leftrightarrow$). So you may think of $(\mathcal{P} \leftrightarrow \mathcal{Q})$ as an abbreviation of $((\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P}))$, which in turn abbreviates the more complex $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$. This is what we expect: a double arrow is like an arrow going from $\mathcal{P}$ to $\mathcal{Q}$ *and* an arrow going from $\mathcal{Q}$ to $\mathcal{P}$.

A couple of additional abbreviations concern parentheses. First, it is sometimes convenient to use a pair of square brackets [ ] in place of parentheses ( ). This is purely for visual convenience; for example ((()())) may be more difficult to absorb than ([()()]). Second, if the very last step of a tree for some formula $\mathcal{P}$ is justified by FR($\to$), FR$'$($\wedge$), FR$'$($\vee$), or FR$'$($\leftrightarrow$), we feel free to abbreviate $\mathcal{P}$ with the *outermost* set of parentheses or brackets dropped. Again, this is purely for visual convenience. Thus, for example, we might write, $A \to (B \to C)$ in place of $(A \to (B \to C))$. As it turns out, where $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are formulas, there is a difference between $((\mathcal{A} \to \mathcal{B}) \to \mathcal{C})$ and $(\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$, insofar as the main operator shifts from one case to the other. In $(\mathcal{A} \to \mathcal{B} \to \mathcal{C})$, however, it is not clear which arrow should be the main operator. That is why we do not count the latter as a grammatical formula or sentence. Similarly there is a difference between $\sim(\mathcal{A} \to \mathcal{B})$ and $(\sim\mathcal{A} \to \mathcal{B})$; again, the main operator shifts. However, there is no room for ambiguity when we drop just an outermost pair of parentheses and write $(\mathcal{A} \to \mathcal{B}) \to \mathcal{C}$ for $((\mathcal{A} \to \mathcal{B}) \to \mathcal{C})$; and similarly when we write $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$ for $(\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$. The same reasoning applies for abbreviations with $\wedge$, $\vee$, or $\leftrightarrow$. So dropping outermost parentheses counts as a legitimate abbreviation.

An expression which uses the extra operators, square brackets, or drops outermost parentheses is a formula just insofar as it is a sort of shorthand for an official formula which does not. But we will not usually distinguish between the shorthand expressions and official formulas. Thus, again, the new conditions may be applied directly in trees and, for example, the following is a legitimate tree to demonstrate that $A \vee ([A \to B] \wedge B)$ is a formula:

(I)

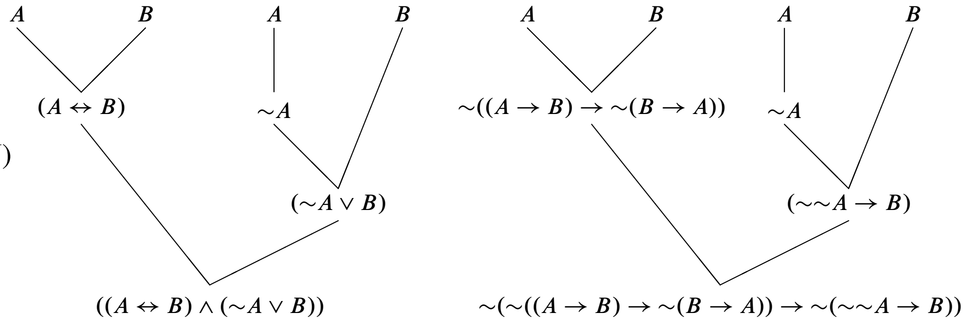


So we use our extra conditions for FR$'$, introduce square brackets instead of parentheses, and drop parentheses in the very last step. The *only* case where you can omit parentheses is if they would have been added in the very last step of the tree. So long as we do not distinguish between shorthand expressions and official formulas, we regard a tree of this sort as sufficient to demonstrate that an expression is a formula and a sentence.

Unabbreviating. As we have suggested, there is a certain tension between the advantages of a simple language, and one that is more complex. When a language is simple, it is easier to reason about; when it has additional resources, it is easier to use. Expressions with $\wedge$, $\vee$, and $\leftrightarrow$ are redundant with expressions that do not have them—though it is easier to work with a language that has $\wedge$, $\vee$, and $\leftrightarrow$ than with one that does not (something like reciting the Pledge of Allegiance in English, and then in Morse code; you can do it in either, but it is easier in the former). If all we wanted was a simple language to reason about, we would forget about the extra operators. If all we wanted was a language easy to use, we would forget about keeping the language simple. To have the advantages of both, we have adopted the position that expressions with the extra operators *abbreviate*, or are a shorthand for, expressions of the original language. It will be convenient to work with abbreviations in many contexts. But when it comes to reasoning about the language, we set the abbreviations to the side and focus on the official language itself.

For this to work, we have to be able to undo abbreviations when required. It is, of course, easy enough to substitute parentheses back for square brackets, or to replace outermost dropped parentheses. For formulas with the extra operators, it is always possible to work through trees, using AB to replace formulas with unabbreviated forms, one operator at a time. Consider an example:

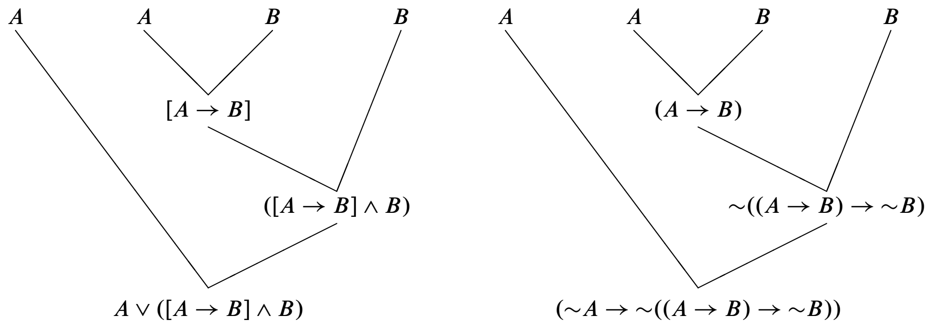
(J)



The tree on the left is (G) from above. The tree on the right uses AB to "unpack" each of the expressions on the left. Atomics remain as before. Then, at each stage, given an unabbreviated version of the parts, we give an unabbreviated version of the whole. First, $(A \leftrightarrow B)$ abbreviates $\sim((A \rightarrow B) \rightarrow \sim(B \rightarrow A))$; this is a simple application of AB($\leftrightarrow$). $\sim A$ is not an abbreviation and so remains as before. From AB($\vee$), $(\mathscr{P} \vee \mathscr{Q})$ abbreviates $(\sim\mathscr{P} \rightarrow \mathscr{Q})$; in this case, $\mathscr{P}$ is $\sim A$ and $\mathscr{Q}$ is B; so we take tilde the $\mathscr{P}$ arrow the $\mathscr{Q}$ (so that we get two tildes). For the final result, we combine the input formulas according to the unabbreviated form for $\wedge$. It is more a bookkeeping problem than anything: There is one formula $\mathscr{P}$ that is the unabbreviated version of $(A \leftrightarrow B)$, another $\mathscr{Q}$ that is the unabbreviated version of $(\sim A \vee B)$; these are combined into $(\mathscr{P} \wedge \mathscr{Q})$ and so by AB($\wedge$) into $\sim(\mathscr{P} \rightarrow \sim\mathscr{Q})$. You should be able to see that this is just what we have done. There is a tilde and a parenthesis; then the $\mathscr{P}$; then an arrow and a tilde; then the $\mathscr{Q}$; and a closing parenthesis. Not only is the abbreviation more compact but, as we shall see, there is a corresponding advantage when it comes to grasping what an expression says.

Here is another example, this time from (I). In this case, we replace also square brackets and restore dropped outer parentheses.

(K)



In the right-hand tree, we reintroduce parentheses for the square brackets. Similarly, we apply AB($\wedge$) and AB($\vee$) to unpack shorthand symbols. And outer parentheses are reintroduced at the very last step. Thus $A \vee ([A \rightarrow B] \wedge B)$ is a shorthand for the unabbreviated expression, $(\sim A \rightarrow \sim((A \rightarrow B) \rightarrow \sim B))$.

Observe that these right-hand trees are *not* ones of the sort you would use directly to show that an expression is a formula by FR! FR does not let you move directly from that $(A \rightarrow B)$ is a formula and B is a formula, to the result that $\sim((A \rightarrow B) \rightarrow \sim B)$

is a formula as just above. Of course, if $(A \to B)$ and B are formulas, then $\sim((A \to B) \to \sim B)$ is a formula, and nothing stops a tree to show it. This is the point of our derived clauses for FR′. In fact, this is a good check on your unabbreviations: If the result is not a formula, you have made a mistake. But you should not think of trees as on the right as involving application of FR. Rather they are *unabbreviating* trees, having exactly one node corresponding to each node on the left; by AB the unabbreviating tree unpacks each expression from the left into its unabbreviated form. The combination of a formula constructed with FR′ and then unabbreviated by AB always results in an expression that meets all the requirements from FR.

E2.6. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathcal{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator.

*a. $(A \wedge B) \to C$

b. $\sim([A \to \sim K_{14}] \vee C_3)$

c. $B \to (\sim A \leftrightarrow B)$

d. $(B \to A) \wedge (C \vee A)$

e. $(A \vee \sim B) \leftrightarrow (C \wedge A)$

*E2.7. For each of the formulas in E2.6a–e, produce an unabbreviating tree to find the unabbreviated expression it represents.

*E2.8. For each of the unabbreviated expressions from E2.7a–e, produce a complete tree to show by direct application of FR that it is an official formula.

E2.9. In the text, we introduced derived clauses to FR by reasoning as follows: "Suppose $\mathcal{P}$ and $\mathcal{Q}$ are formulas; then by FR($\sim$), $\sim\mathcal{P}$ is a formula; so by FR($\to$), $(\sim\mathcal{P} \to \mathcal{Q})$ is a formula; but this is just to say that $(\mathcal{P} \vee \mathcal{Q})$ is a formula. And similarly in the other cases" (page 41). Supposing that $\mathcal{P}$ and $\mathcal{Q}$ are formulas, produce the similar reasoning to show that $(\mathcal{P} \wedge \mathcal{Q})$ and $(\mathcal{P} \leftrightarrow \mathcal{Q})$ are formulas. Hint: Again, it may help to think about trees.

E2.10. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The vocabulary for a sentential language, and use of the metalanguage.

b. A formula of a sentential language.

c. The parts of a formula.

d. The abbreviation and unabbreviation for an official formula of a sentential language.

2.3 Quantificational Languages

The *methods* by which we define the grammar of a quantificational language are very much the same as for a sentential language. Of course, in the quantificational case, additional expressive power is associated with additional complications. We will introduce a class of *terms* before we get to the formulas, and there will be a distinction between formulas and sentences. As before, however, there is the *vocabulary* and then the grammatical elements. After introducing the vocabulary, we build to *terms*, *formulas*, and *sentences*. The chapter concludes with some discussion of abbreviations, and of a particular language with which we shall be concerned later in the text.

Here is a brief intuitive picture: At the start of section 2.2 we introduced 'The butler is guilty' and 'The maid is guilty' as atoms for sentential languages, and the rest of the section went on to fill out that picture. For the quantificational languages of this section, our atoms are certain sentence parts. Thus we introduce a class of *individual terms* which work to pick out objects. In the simplest case, we might introduce b and m to pick out the butler and the maid. Similarly, we introduce a class of *predicate* expressions as (x is guilty) and (x killed y) indicating them by capitals as G^1 or K^2 (with the superscript to indicate the number of object *places*). Then G^1b says that the butler is guilty, and K^2bm that the butler killed the maid. We shall read $\forall x G^1 x$ to say for any thing x, *it* is guilty—that *everything* is guilty. (The upside-down 'A' for *all* is the *universal* quantifier.) As indicated by this reading, the variable x works very much like a pronoun in ordinary language. And, of course, our notions may be combined. Thus, $\forall x G^1 x \wedge K^2bm$ says that everything is guilty and the butler killed the maid. Thus we expose structure buried in sentence letters from before. Insofar as the language includes quantifiers (upside-down 'A') and predicates (as G^1 or K^2) it is said to be a language for *quantificational* (or *predicate*) logic. Of course we have so-far done nothing to define such a language. But this should give you a picture of the direction in which we aim to go.

2.3.1 Vocabulary

We begin by specifying the *vocabulary* or symbols of our quantificational languages. For now, do not worry about what the symbols mean or how they are used. Our task is to identify the symbols and give some conventions for talking about them. For any quantificational language $\mathscr{L}$ the vocabulary consists of,

VC (p) Punctuation symbols: ( )

(o) Operator symbols: $\sim$ $\rightarrow$ $\forall$

(v) A countably infinite collection of variable symbols

(s) A countable collection of sentence letters

(c) A countable collection of constant symbols

(f) For any integer $n \geq 1$, a countable collection of n-place function symbols

(r) For any integer $n \geq 1$, a countable collection of n-place relation symbols

Each of the countable collections may be empty except that there are always the variable symbols, and there must be at least one relation symbol. Unless otherwise noted, '$=$' is always included among the 2-place relation symbols, and the variable symbols are $i \ldots z$ with or without positive integer subscripts. Notice that all the punctuation symbols, operator symbols and sentence letters remain from before (except that the collection of sentence letters may be empty). There is one new operator symbol, with the new variable symbols, constant symbols, function symbols, and relation symbols.

> This definition VC is parallel to definition vc from section 2.2. For definitions with both sentential and quantificational versions, I adopt the convention of naming the initial sentential version in small caps, and the quantificational version in large.

In order to fully specify the vocabulary of any particular language, we need to specify its variable symbols, sentence letters, constant symbols, function symbols, and relation symbols. Our general definition VC leaves room for languages with different collections of these symbols. As before, the requirement that the collections be countable is compatible with multiple series; for example, there may be sentence letters $A, A_1, A_2, \ldots, B, B_1, B_2, \ldots$. So, again VC is compatible with a wide variety of specifications, but legitimate specifications always require that variable symbols, sentence letters, constant symbols, function symbols, and relation symbols can be sorted into series with a first member, a second member, and so forth.

As a sample for these specifications, we shall adopt a generic quantificational language $\mathscr{L}_q$ which includes the standard variables, the equality symbol '$=$' and,

More on Countability

Given what was said on page 35, one might think that every collection is countable. However, this is not so. This amazing and simple result was proved by G. Cantor in 1873. Consider the collection which includes every countably infinite series of digits 0 through 9 (or, if you like, decimal representations of real numbers between 0 and 1). Suppose that the members of this collection can be correlated one-to-one with the non-negative integers. Then there is some list,

$$\begin{array}{lllllllll}
0 & - & \boldsymbol{a_0} & a_1 & a_2 & a_3 & a_4 & \cdots \\
1 & - & b_0 & \boldsymbol{b_1} & b_2 & b_3 & b_4 \\
2 & - & c_0 & c_1 & \boldsymbol{c_2} & c_3 & c_4 \\
3 & - & d_0 & d_1 & d_2 & \boldsymbol{d_3} & d_4 \\
4 & - & e_0 & e_1 & e_2 & e_3 & \boldsymbol{e_4} \\
\vdots
\end{array}$$

which matches each series of digits with a non-negative integer. For any digit x, say x' is the digit after it in the standard ordering (where 0 follows 9). Now consider the digits along the diagonal, $a_0, b_1, c_2, \ldots$ and ask: does the series $a'_0, b'_1, c'_2, \ldots$ appear anywhere in the list? It cannot be the first member, because $a_0 \neq a'_0$; it cannot be the second, because $b_1 \neq b'_1$; it cannot be the third because $c_2 \neq c'_2$; and similarly for every member. So $a'_0, b'_1, c'_2, \ldots$ does not appear in the list. So we have *failed* to match *all* the infinite series of digits with non-negative integers.

One might suggest simply *adding* $a'_0, b'_1, c'_2, \ldots$, say at position 0 and pushing the other members down—but then, from the diagonal of this *new* list, $a''_0, a'_1, b'_2, c'_3, \ldots$ is missing. And similarly for any attempt! Insofar as its members cannot be matched to the non-negative integers, the set of all infinite series of digits is *uncountable*.

As an example, consider the following attempt to line up the non-negative integers with the series of digits: For each non-negative integer, repeat its digits, except that for "duplicate" cases—1 and 11, 2 and 22, 12 and 1212—prefix enough 0s so that no later series duplicates an earlier one.

0	–	**0**	0	0	0	0	0	0	0	0	0	0	0	0	⋯
1	–	1	**1**	1	1	1	1	1	1	1	1	1	1	1	
2	–	2	2	**2**	2	2	2	2	2	2	2	2	2	2	
3	–	3	3	3	**3**	3	3	3	3	3	3	3	3	3	
4	–	4	4	4	4	**4**	4	4	4	4	4	4	4	4	
5	–	5	5	5	5	5	**5**	5	5	5	5	5	5	5	
6	–	6	6	6	6	6	6	**6**	6	6	6	6	6	6	
7	–	7	7	7	7	7	7	7	**7**	7	7	7	7	7	
8	–	8	8	8	8	8	8	8	8	**8**	8	8	8	8	
9	–	9	9	9	9	9	9	9	9	9	**9**	9	9	9	
10	–	1	0	1	0	1	0	1	0	1	0	**1**	0	1	
11	–	0	1	1	1	1	1	1	1	1	1	1	**1**	1	
12	–	1	2	1	2	1	2	1	2	1	2	1	2	**1**	
⋮															

Then, by the above method, from the diagonal,

1 2 3 4 5 6 7 8 9 0 2 2 2 ⋯

cannot appear anywhere on the list. And similarly, any list has some missing series.

> Sentence letters: uppercase Roman italics $A \dots Z$ with or without positive integer subscripts
>
> Constant symbols: lowercase Roman italics $a \dots h$ with or without positive integer subscripts
>
> Function symbols: for any integer $n \geq 1$, superscripted lowercase Roman italics $a^n \dots z^n$ with or without positive integer subscripts
>
> Relation symbols: for any integer $n \geq 1$, superscripted uppercase Roman italics $A^n \dots Z^n$ with or without positive integer subscripts.

Observe that constant symbols and variable symbols partition the lowercase alphabet: $a \dots h$ for constants, and $i \dots z$ for variables. Function symbols are distinguished from constant and variable symbols by their superscripts; similarly relation symbols are distinguished from sentence letters by their superscripts. Function symbols with a superscript 1 ($a^1 \dots z^1$) are *one-place* function symbols; function symbols with a superscript 2 ($a^2 \dots z^2$) are *two-place* function symbols; and so forth. Similarly, relation symbols with a superscript 1 ($A^1 \dots Z^1$) are *one-place* relation symbols; relation symbols with a superscript 2 ($A^2 \dots Z^2$) are *two-place* relation symbols; and so forth. Subscripts merely guarantee that we never run out of symbols of the different types. Notice that superscripts and subscripts suffice to distinguish all the different symbols from one another. Thus for example A and A^1 are different symbols—one a sentence letter, and the other a one-place relation symbol; A^1, A^1_1, and A^2 are distinct as well—the first two are one-place relation symbols, distinguished by the subscript, the latter is a completely distinct two-place relation symbol. In practice, again, we will not see subscripts very often. (And we shall even find ways to abbreviate away some superscripts.)

The *metalanguage* works very much as before. We use script letters $\mathcal{A} \dots \mathcal{Z}$ and $\mathscr{a} \dots \mathscr{z}$ to represent expressions of an object language like $\mathcal{L}_q$. Again, '$\sim$', '$\rightarrow$', '$\forall$', '$=$', '(', and ')' represent themselves. And concatenated or joined symbols of the metalanguage represent the concatenation of the symbols they represent. As before, the metalanguage lets us make general claims about ranges of expressions all at once. Thus, where $\mathscr{x}$ is a variable, $\forall \mathscr{x}$ is a *universal $\mathscr{x}$-quantifier*. Here, '$\forall \mathscr{x}$' is not an expression of an object language like $\mathcal{L}_q$ (Why?) Rather, we have said of object language expressions that $\forall x$ is a universal x-quantifier, $\forall y_2$ is a universal y_2-quantifier, and so forth. In the metalinguistic expression, '$\forall$' stands for itself, and '$\mathscr{x}$' for the arbitrary variable. Again, as in section 2.2.1, it may help to use maps to see whether an expression is of a given form. Thus given that $\mathscr{x}$ maps to any variable, $\forall x$ and $\forall y$ are of the form $\forall \mathscr{x}$, but $\forall c$ and $\forall f^1 z$ are not.

$$\text{(L)} \quad \begin{array}{cccc} \forall \mathscr{x} & \forall \mathscr{x} & \forall \mathscr{x} & \forall \mathscr{x} \\ \downarrow\downarrow & \downarrow\downarrow & \downarrow & \downarrow \\ \forall x & \forall y & \forall c? & \forall\ \overbrace{f^1 z}^{?} \end{array}$$

In the leftmost two cases, $\forall$ maps to itself, and $\mathscr{x}$ to a variable. In the next, 'c' is a *constant* so there is no variable to which $\mathscr{x}$ can map. In the rightmost case, there is a variable z in the object expression, but if $\mathscr{x}$ is mapped to it, the function symbol f^1 is left unmatched. So the rightmost two expressions are not of the form $\forall\mathscr{x}$.

E2.11. Assuming that $\mathscr{R}^1$ may represent any one-place relation symbol, $\mathscr{h}^2$ any two-place function symbol, $\mathscr{x}$ any variable, and $\mathscr{c}$ any constant of $\mathscr{L}_q$, use maps to determine whether each of the following expressions is (i) of the form, $\forall\mathscr{x}(\mathscr{R}^1\mathscr{x} \to \mathscr{R}^1\mathscr{c})$ and then (ii) of the form, $\forall\mathscr{x}(\mathscr{R}^1\mathscr{x} \to \mathscr{R}^1\mathscr{h}^2\mathscr{x}\mathscr{c})$.

*a. $\forall k(A^1k \to A^1d)$

b. $\forall h(J^1h \to J^1b)$

c. $\forall w(S^1w \to S^1g^2wb)$

d. $\forall w(S^1w \to S^1c^2xc)$

e. $\forall vL^1v \to L^1yh^2$

2.3.2 Terms

With the vocabulary of a language in place, we can turn to specification of its grammatical expressions. For this, in the quantificational case, we begin with *terms*.

TR (v) If t is a variable $\mathscr{x}$, then t is a *term*.

(c) If t is a constant $\mathscr{c}$, then t is a *term*.

(f) If $\mathscr{h}^n$ is an n-place function symbol and $t_1 \dots t_n$ are n terms, then $\mathscr{h}^n t_1 \dots t_n$ is a *term*.

(CL) Any term may be formed by repeated application of these rules.

TR is another example of a recursive definition. As before, we can use tree diagrams to see how it works. This time, basic elements are constants and variables. Complex elements are put together by clause (f). Thus, for example, $f^1g^2h^1xc$ is a term of $\mathscr{L}_q$.

(M)

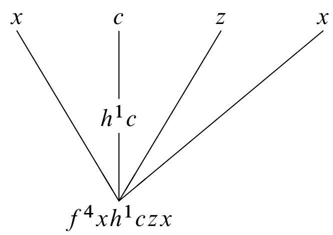
x $\quad$ c	x is a term by TR(v), and c is a term by TR(c)
h^1x	since x is a term, this is a term by TR(f)
g^2h^1xc	since h^1x and c are terms, this is a term by TR(f)
$f^1g^2h^1xc$	since g^2h^1xc is a term, this is a term by TR(f)

Superscripts of a function symbol indicate the number of places that take terms. Thus x is a term, and h^1 followed by x to form h^1x is another term. But then, given that h^1x and c are terms, g^2 followed by h^1x and then c is another term. And so forth.

Just as a formula is made up of operator symbols and other formulas, so a complex term is made of function symbols and other terms. While terms may have other terms as parts, each stage in the tree counts as a single unit for the next. Thus in the third row of (M), g^2 is followed by the two terms h^1x and c. And in the last stage, f^1 is followed by the one term g^2h^1xc. In contrast, neither h^1xc nor f^1h^1xc are terms—in each case, the problem is that the one-place function symbol is followed by two terms: x and c are terms, and h^1x and c are terms, but a one-place function symbol followed by two terms does not form a term. And similarly, g^2h^1x and g^2c are not terms—the function symbol g^2 must be followed by a pair of terms to form a new term. You will find that there is always only one way to build a term on a tree.

Here is another example:

(N)



these are terms by TR(v), TR(c), TR(v), and TR(v)

since c is a term, this is a term by TR(f)

given the four input terms, this is a term by TR(f)

Again, there is always just one way to build a term by the definition. If you are confused about the makeup of a term, build it on a tree, and all will be revealed. To demonstrate that an expression is a term, it is sufficient to construct it, according to the definition, on such a tree. If an expression is not a term, there will be no way to construct it according to the rules.

E2.12. For each of the following expressions, demonstrate that it is a term of $\mathcal{L}_q$ with a tree.

a. f^1c

b. g^2yf^1c

*c. h^3cf^1yx

d. $g^2h^3xyf^1cx$

e. $h^3f^1f^1xcg^2f^1za$

E2.13. Explain why the following expressions are not terms of $\mathcal{L}_q$. Hint: You may find that an attempted tree will help you see what is wrong.

a. X

b. g^2

c. zc

*d. $g^2 y f^1 x c$

e. $h^3 f^1 f^1 c g^2 f^1 z a$

E2.14. For each of the following expressions, determine whether it is a term of $\mathscr{L}_q$; if it is, demonstrate with a tree; if not, explain why.

*a. $g^2 g^2 x y f^1 x$

*b. $h^3 c f^2 y x$

c. $f^1 g^2 x h^3 y f^2 y c$

d. $f^1 g^2 x h^3 y f^1 y c$

e. $h^3 g^2 f^1 x c g^2 f^1 z a f^1 b$

2.3.3 Formulas

With the terms in place, we are ready for the central notion of a formula. Again, the definition is recursive.

FR (s) If $\mathscr{S}$ is a sentence letter, then $\mathscr{S}$ is a *formula*.

(r) If $\mathscr{R}^n$ is an n-place relation symbol and $t_1 \ldots t_n$ are n terms, then $\mathscr{R}^n t_1 \ldots t_n$ is a *formula*.

($\sim$) If $\mathscr{P}$ is a formula, then $\sim\mathscr{P}$ is a *formula*.

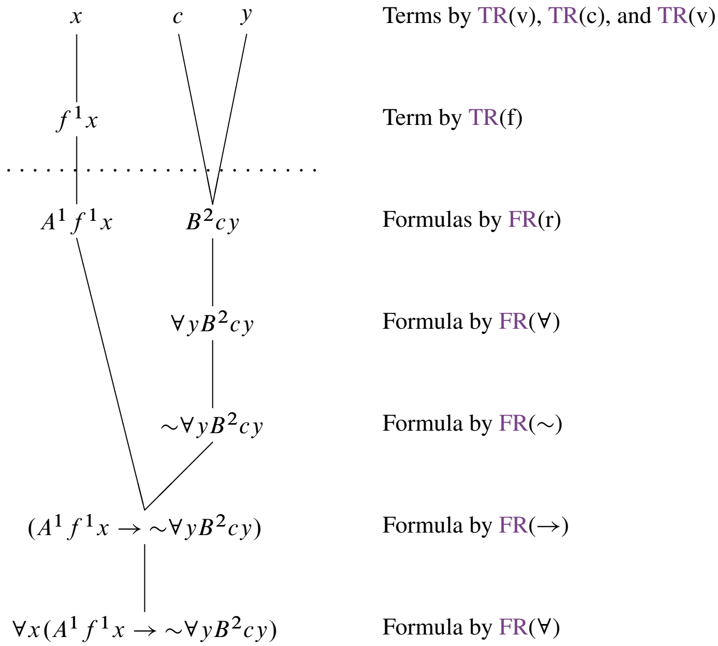
($\rightarrow$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $(\mathscr{P} \rightarrow \mathscr{Q})$ is a *formula*.

($\forall$) If $\mathscr{P}$ is a formula and x is a variable, then $\forall x \mathscr{P}$ is a *formula*.

(CL) Any formula can be formed by repeated application of these rules.

Again, we can use trees to see how it works. In this case, FR(r) depends on which expressions are terms. So it is natural to split the diagram into two, with applications of TR above a division, and FR below. Then, for example, $\forall x(A^1 f^1 x \rightarrow \sim\forall y B^2 c y)$ is a formula.

(O)



By now, the basic strategy should be clear. We construct terms by TR just as before. Given that f^1x is a term, FR(r) gives us that A^1f^1x is a formula, for it consists of a one-place relation symbol followed by a single term; and given that c and y are terms, FR(r) gives us that B^2cy is a formula, for it consists of a two-place relation symbol followed by two terms. From the latter, by FR(∀), $\forall yB^2cy$ is a formula. Then FR(∼) and FR(→) work just as before. The final step is another application of FR(∀).

For another example consider tree (P) in the upper box on page 55. By the tree, $\forall x{\sim}(L \rightarrow \forall yB^3f^1ycx)$ is a formula of $\mathcal{L}_q$. L is a sentence letter; so it does not require any terms to be a formula. B^3 is a three-place relation symbol, so by FR(r) it takes three terms to make a formula. After that, other formulas are constructed out of ones that come before.

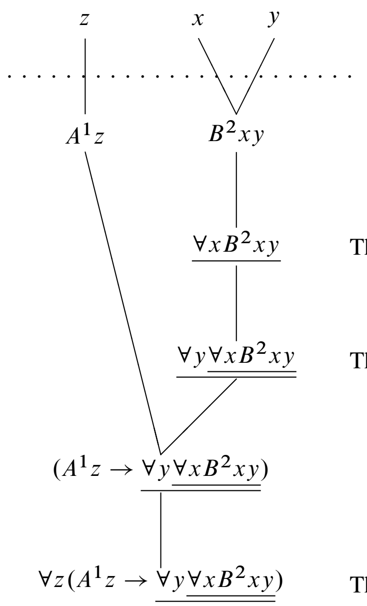
If an expression is not a formula, then there is no way to construct it by the rules. Thus, for example, (A^1x) is not a formula of $\mathcal{L}_q$. A^1x is a formula; but the only way parentheses are introduced is in association with →; the parentheses in (A^1x) are not introduced that way; so there is no way to construct it by the rules, and it is not a formula. Similarly, A^2x and A^2f^2xy are not formulas; in each case, the problem is that the two-place relation symbol is followed by just *one* term. You should be clear about these in your own mind, particularly for the second case.

Before turning to the official notion of a *sentence*, we introduce some additional definitions, each directly related to the trees—and to notions you have seen before. Again, require that trees branch in the usual way: without extraneous nodes, and without nodes used more than once. Then where '→', '∼', and any $\forall x$ is an *operator*, a formula's *main* operator is the last operator added in its tree. Every formula in the formula portion of a diagram for $\mathcal{P}$, including $\mathcal{P}$ itself, is a *subformula* of $\mathcal{P}$. Notice that terms are not formulas, and so are not subformulas. An *immediate* subformula of $\mathcal{P}$ is a subformula to which $\mathcal{P}$ is directly connected by lines. A subformula is *atomic* iff it contains no operators and so appears in the top line of the formula part of a tree.

Thus with notation from exercises before—bracket for subformulas, star for atomic subformulas, box for immediate subformulas, and circle for main operator—tree (Q) in the lower box on the next page identifies the parts from tree (P). The main operator is $\forall x$, and the immediate subformula is $\sim(L \to \forall y B^3 f^1 ycx)$. The atomic subformulas are L and $B^3 f^1 ycx$. The atomic subformulas are the most basic *formulas*. Given this, everything is as one would expect from before. In general, if $\mathcal{P}$ and $\mathcal{Q}$ are some formulas and x is a variable, then the main operator of $\forall x \mathcal{P}$ is the quantifier, and the immediate subformula is $\mathcal{P}$; the main operator of $\sim\mathcal{P}$ is the tilde, and the immediate subformula is $\mathcal{P}$; the main operator of $(\mathcal{P} \to \mathcal{Q})$ is the arrow, and the immediate subformulas are $\mathcal{P}$ and $\mathcal{Q}$—for you would build these formulas by getting $\mathcal{P}$, or $\mathcal{P}$ and $\mathcal{Q}$, and then adding the quantifier, tilde, or arrow as the last operator. Insofar as they operate on a single immediate subformula, quantifiers and tilde are *unary* operators, while $\to$ is *binary*.

Now if a formula includes an operator, that operator's *scope* is just the subformula in which the operator *first* appears. Though the notion applies generally, we shall be particularly interested in *quantifier* scope. Using underlines to indicate quantifier scope,

(R)



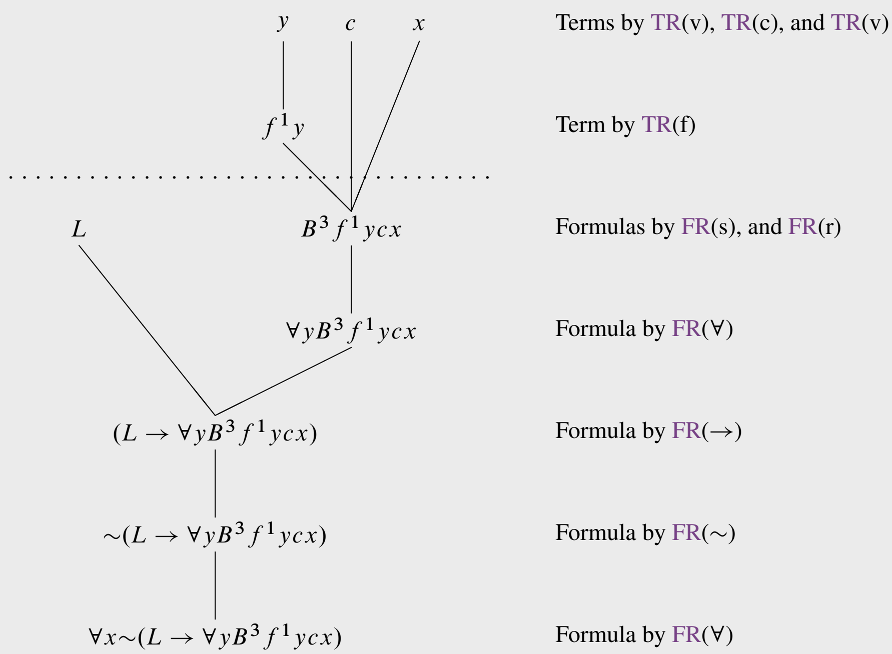
The scope of the x-quantifier is $\forall x B^2xy$

The scope of the y-quantifier is $\forall y \forall x B^2xy$

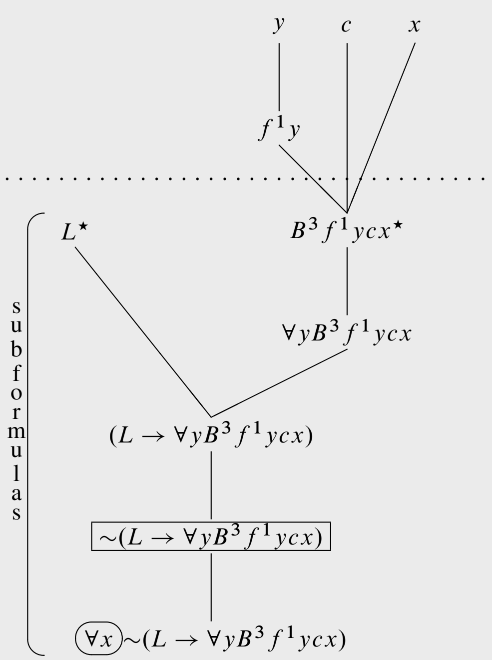
The scope of the z-quantifier is the entire formula

A variable x is *bound* iff it appears in the scope of an x-quantifier, and a variable is *free* iff it is not bound. In the above diagram, each variable is bound. The x-quantifier binds both instances of x; the y-quantifier binds both instances of y; and the z-quantifier binds both instances of z. In $\forall x R^2xy$, however, both instances of x are bound, but the y is free. An *open formula* is a formula with free variables. And

(P)

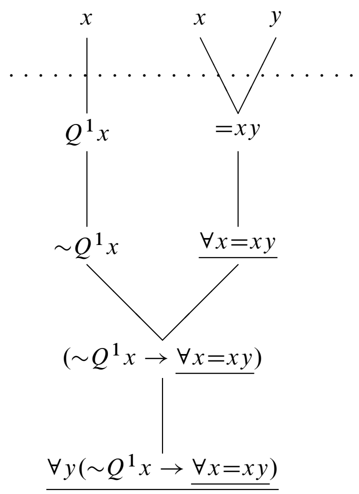

y c x
Terms by TR(v), TR(c), and TR(v)
$f^1 y$
Term by TR(f)
L $B^3 f^1 ycx$
Formulas by FR(s), and FR(r)
$\forall y B^3 f^1 ycx$
Formula by FR(∀)
$(L \to \forall y B^3 f^1 ycx)$
Formula by FR(→)
$\sim(L \to \forall y B^3 f^1 ycx)$
Formula by FR(∼)
$\forall x \sim(L \to \forall y B^3 f^1 ycx)$
Formula by FR(∀)

(Q)


y c x
$f^1 y$
$L^\star$ $B^3 f^1 ycx^\star$
$\forall y B^3 f^1 ycx$
$(L \to \forall y B^3 f^1 ycx)$
$\boxed{\sim(L \to \forall y B^3 f^1 ycx)}$
$\forall x \sim(L \to \forall y B^3 f^1 ycx)$
subformulas

finally, an expression is a *sentence* iff it is a formula and it has no free variables. To determine whether an expression is a sentence, use a tree to see if it is a formula. If it is a formula, use underlines to check whether any variable x has an instance that falls outside the scope of an x-quantifier. If it is a formula, and there is no such instance, then the expression is a sentence. From diagram (R), $\forall z(A^1 z \to \forall y \forall x B^2 xy)$ is a formula and a sentence. But as follows, $\forall y(\sim Q^1 x \to \forall x{=}xy)$ is not.

(S)



The scope of the x-quantifier is $\forall x{=}xy$

The scope of the y-quantifier is the entire formula

Recall that '$=$' is a two-place relation symbol. The expression has a tree, so it is a formula. The x-quantifier binds the last two instances of x, and the y-quantifier binds both instances of y. But the first instance of x is free. Since it has a free variable, although it is a formula, $\forall y(\sim Q^1 x \to \forall x{=}xy)$ is not a sentence. Notice that $\forall x R^2 ax$, for example, is a sentence as the only *variable* is x (a being a *constant*) and all the instances of x are bound.

E2.15. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathscr{L}_q$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. $H^1 x$

*b. $(A^1 x \to B^2 c f^1 x)$

c. $\forall x(\sim{=}xc \to A^1 g^2 ay)$

d. $\sim\forall x(B^2 xc \to \forall y \sim A^1 g^2 ay)$

e. $(S \to \sim(\forall w B^2 f^1 w h^1 a \to \sim\forall z(H^1 w \to B^2 za)))$

E2.16. Explain why the following expressions are not formulas or sentences of $\mathcal{L}_q$. Hint: You may find that an attempted tree will help you see what is wrong.

a. H^1

b. g^2ax

*c. $\forall x B^2 x g^2 a x$

d. $\sim(\sim\forall a A^1 a \to (S \to \sim B^2 z g^2 x a))$

e. $\forall x(Dax \to \forall z \sim K^2 z g^2 x a)$

E2.17. For each of the following expressions, determine whether it is a formula and a sentence of $\mathcal{L}_q$. If it is a formula, show it on a tree, and exhibit its parts as in E2.15. If it fails one or both, explain why.

a. $\sim(L \to \sim V)$

b. $\forall x(\sim L \to K^1 h^3 x b)$

c. $\forall z \forall w(\forall x R^2 w x \to \sim K^2 z w) \to \sim M^2 z z)$

*d. $\forall z(L^1 z \to (\forall w R^2 w f^3 a x w \to \forall w R^2 f^3 a z w w))$

e. $\sim((\forall w) B^2 f^1 w h^1 a \to \sim(\forall z)(H^1 w \to B^2 z a))$

2.3.4 Abbreviations

That is all there is to the official grammar. Having introduced the official grammar, though, it is nice to have in hand some abbreviated versions for official expressions. As before, abbreviated forms give us ways to manipulate official expressions without undue pain. First, for any variable x and formulas $\mathcal{P}$ and $\mathcal{Q}$,

AB ($\vee$) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \to \mathcal{Q})$

($\wedge$) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \to \sim\mathcal{Q})$

($\leftrightarrow$) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$

($\exists$) $\exists x \mathcal{P}$ abbreviates $\sim\forall x \sim\mathcal{P}$

The first three are as from AB. The last is new. For any variable x, an expression of the form $\exists x$ is an *existential* quantifier. $\exists x \mathcal{P}$ is read, 'there *exists* an x such that $\mathcal{P}$'.

As before, these abbreviations make possible derived clauses to FR. Suppose $\mathcal{P}$ is a formula and x is a variable; then by FR($\sim$), $\sim\mathcal{P}$ is a formula; so by FR($\forall$), $\forall x \sim\mathcal{P}$ is a formula; so by FR($\sim$) again, $\sim\forall x \sim\mathcal{P}$ is a formula; but this is just to say that $\exists x \mathcal{P}$ is a formula. With results from before, we are thus given, FR′ (∧) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.

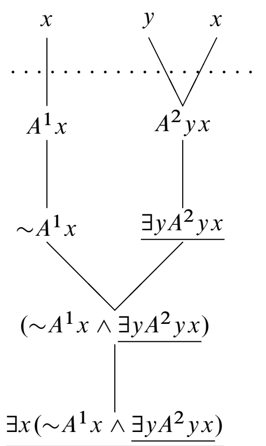
(∨) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.

(↔) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.

(∃) If $\mathcal{P}$ is a formula and x is a variable, then $\exists x\mathcal{P}$ is a *formula*.

The first three are from before. The last is new. And, as before, we can incorporate these conditions directly into trees for formulas. Thus $\exists x(\sim A^1x \wedge \exists yA^2yx)$ is a formula.

(T)



These are terms by TR(v)

These are formulas by FR(r)

These are formulas by FR(∼) and FR′(∃)

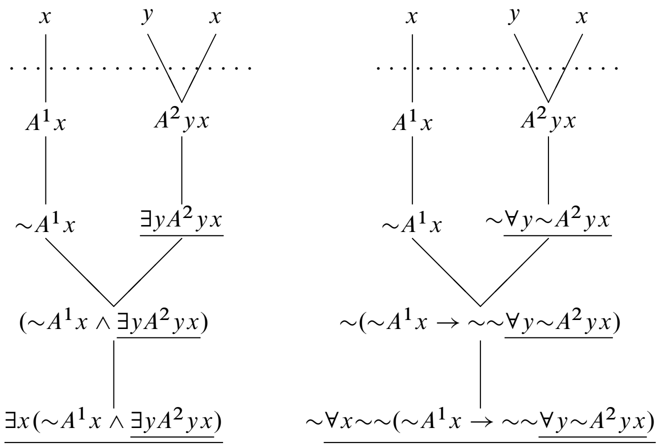
This is a formula by FR′(∧)

This is a formula by FR′(∃)

In a derived sense, we carry over additional definitions from before. Thus, where operators include the derived symbols ∧, ∨, ↔, and $\exists x$, a formula's main operator is the last operator added in its tree, subformulas are all the formulas in the formula part of the tree, atomic subformulas are the ones in the upper row of the formula part, and immediate subformulas are the one(s) to which the formula is directly connected by lines. Thus the main operator of $\exists x(\sim A^1x \wedge \exists yA^2yx)$ is the leftmost existential quantifier and the immediate subformula is $(\sim A^1x \wedge \exists yA^2yx)$. In addition, a variable is in the scope of an existential quantifier iff it would be in the scope of the unabbreviated universal one. So it is possible to discover whether an expression is a sentence directly from diagrams of this sort. Thus, as indicated by underlines, $\exists x(\sim A^1x \wedge \exists yA^2yx)$ is a sentence.

To see what it is an abbreviation for, we can reconstruct the formula on an unabbreviating tree, one operator at a time.

(U)



By AB(∃)

By AB(∧)

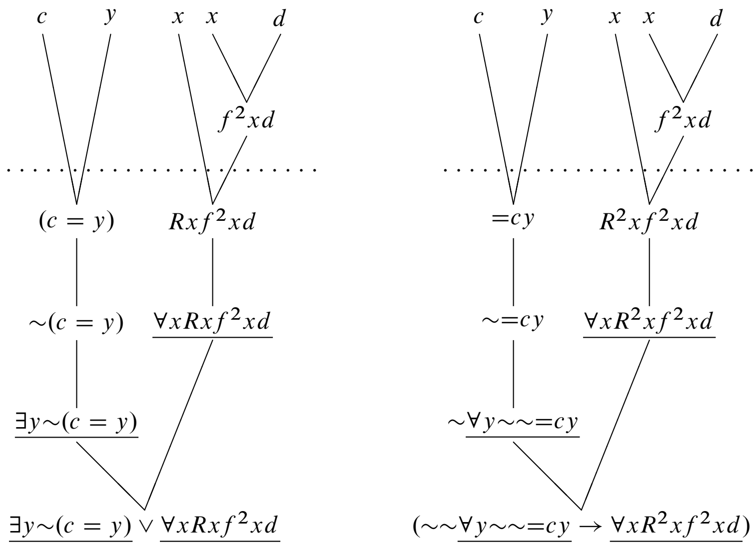
By AB(∃)

First the existential quantifier is replaced by the unabbreviated form. Then, where $\mathcal{P}$ and $\mathcal{Q}$ are joined by FR′(∧) to form $(\mathcal{P} \wedge \mathcal{Q})$, the corresponding unabbreviated expressions are combined into the unabbreviated form, $\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})$. At the last step the existential quantifier is replaced again. So $\exists x(\sim A^1x \wedge \exists y A^2yx)$ abbreviates $\sim\forall x\sim\sim(\sim A^1x \rightarrow \sim\sim\forall y\sim A^2yx)$. Again, abbreviations are nice! Notice that the resultant expression is a formula and a sentence, as it should be.

As before, it is sometimes convenient to use a pair of square brackets [ ] in place of parentheses ( ). And if the very last step of a tree for some formula is justified by FR(→), FR′(∨), FR′(∧), or FR′(↔), we may abbreviate that formula with the outermost set of parentheses or brackets dropped. In addition, for terms t_1 and t_2 we will frequently represent the formula $=t_1t_2$ as $(t_1 = t_2)$. Notice the extra parentheses. This lets us see the equality symbol in its more usual "infix" form. When there is no danger of confusion, we will sometimes omit the parentheses and write, $t_1 = t_2$. Also, where there is no potential for confusion, we sometimes omit superscripts. Thus in $\mathcal{L}_q$ we might omit superscripts on relation symbols—simply assuming that the terms following a relation symbol give its correct number of places. Thus Ax abbreviates A^1x; Axy abbreviates A^2xy; Axf^1y abbreviates A^2xf^1y; and so forth. Notice that Ax and Axy, for example, involve *different* relation symbols. In formulas of $\mathcal{L}_q$, sentence letters are distinguished from relation symbols insofar as relation symbols are followed immediately by terms, where sentence letters are not. Notice, however, that we *cannot* drop superscripts on function symbols in $\mathcal{L}_q$—thus, even given that f and g are function symbols rather than constants, apart from superscripts, there is no way to distinguish the terms in, say, $Afgxyzw$.

As a final example, $\exists y\sim(c = y) \vee \forall x Rxf^2xd$ is a formula and a sentence.

(V)



The abbreviation drops a superscript, uses the infix notation for equality, uses the existential quantifier and wedge, and drops outermost parentheses. As before, the right-hand diagram is not a direct demonstration that $(\sim\sim\forall y\sim\sim{=}cy \rightarrow \forall xR^2xf^2xd)$ is a sentence. However, it unpacks the abbreviation and we know that the result is an official sentence insofar as the left-hand tree, with its application of derived rules, tells us that $\exists y\sim(c = y) \vee \forall xRxf^2xd$ is an abbreviation of formula and a sentence, and the right-hand diagram tells us what that expression is.

E2.18. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathscr{L}_q$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. $(A \rightarrow \sim B) \leftrightarrow (A \wedge C)$

b. $\exists xFx \wedge \forall yGxy$

*c. $\exists xAf^1g^2ah^3zwf^1x \vee S$

d. $\forall x\forall y\forall z([(x = y) \wedge (y = z)] \rightarrow (x = z))$

e. $\exists y[c = y \wedge \forall xRxf^1xy]$

*E2.19. For each of the formulas in E2.18, produce an unabbreviating tree to find the unabbreviated expression it represents.

Grammar Quick Reference

VC For any quantificational language $\mathscr{L}$ the vocabulary consists of,

(p) Punctuation symbols: ( )

(o) Operator symbols: $\sim$ $\rightarrow$ $\forall$

(v) A countably infinite collection of variable symbols

(s) A countable collection of sentence letters

(c) A countable collection of constant symbols

(f) For any integer $n \geq 1$, a countable collection of n-place function symbols

(r) For any integer $n \geq 1$, a countable collection of n-place relation symbols

TR (v) If t is a variable x, then t is a *term*.

(c) If t is a constant c, then t is a *term*.

(f) If h^n is an n-place function symbol and $t_1 \dots t_n$ are n terms, then $h^n t_1 \dots t_n$ is a *term*.

(CL) Any term may be formed by repeated application of these rules.

FR (s) If $\mathcal{S}$ is a sentence letter, then $\mathcal{S}$ is a *formula*.

(r) If $\mathcal{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are n terms, $\mathcal{R}^n t_1 \dots t_n$ is a *formula*.

($\sim$) If $\mathcal{P}$ is a formula, then $\sim\mathcal{P}$ is a *formula*.

($\rightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *formula*.

($\forall$) If $\mathcal{P}$ is a formula and x is a variable, then $\forall x\mathcal{P}$ is a *formula*.

(CL) Any formula can be formed by repeated application of these rules.

Subformulas are all the formulas in the tree; *atomic* subformulas appear in the top formula row; *immediate* subformulas are the ones to which a formula is directly connected by lines; the *main operator* is the last operator added. An operator's *scope* includes just the formula in which it is introduced; a variable x is *bound* iff it is in the scope of an x-quantifier and *free* iff it is not; an *open* formula is one with free variables; an expression is a *sentence* iff it is a formula with no free variables.

AB ($\vee$) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \rightarrow \mathcal{Q})$

($\wedge$) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})$

($\leftrightarrow$) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \rightarrow \mathcal{P}))$

($\exists$) $\exists x\mathcal{P}$ abbreviates $\sim\forall x\sim\mathcal{P}$

FR′ ($\wedge$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.

($\vee$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.

($\leftrightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.

($\exists$) If $\mathcal{P}$ is a formula and x is a variable, then $\exists x\mathcal{P}$ is a *formula*.

The generic language $\mathscr{L}_q$ includes the equality symbol '$=$' along with,

Variable symbols: $i \dots z$ with or without positive integer subscripts

Sentence letters: $A \dots Z$ with or without positive integer subscripts

Constant symbols: $a \dots h$ with or without positive integer subscripts

Function symbols: for any $n \geq 1$, $a^n \dots z^n$ with or without positive integer subscripts

Relation symbols: for any $n \geq 1$, $A^n \dots Z^n$ with or without positive integer subscripts

*E2.20. For each of the unabbreviated expressions from E2.19, produce a complete tree to show by direct application of FR that it is an official formula. In each case, using underlines to indicate quantifier scope, is the expression a sentence? does this match with the result of E2.18?

2.3.5 Another Language

To emphasize the generality of our definitions VC, TR, and FR, let us introduce a language like one with which we will be much concerned later in the text. $\mathscr{L}_{NT}^{<}$ is like a minimal language we shall introduce later for *number theory*. Recall that VC leaves open what are the variable symbols, constant symbols, function symbols, sentence letters, and relation symbols of a quantificational language. So far, our generic language $\mathscr{L}_q$ fills these in by certain conventions. $\mathscr{L}_{NT}^{<}$ replaces these with the standard variables and,

Constant symbol: $\emptyset$

One-place function symbol: S

Two-place function symbols: $+, \times$

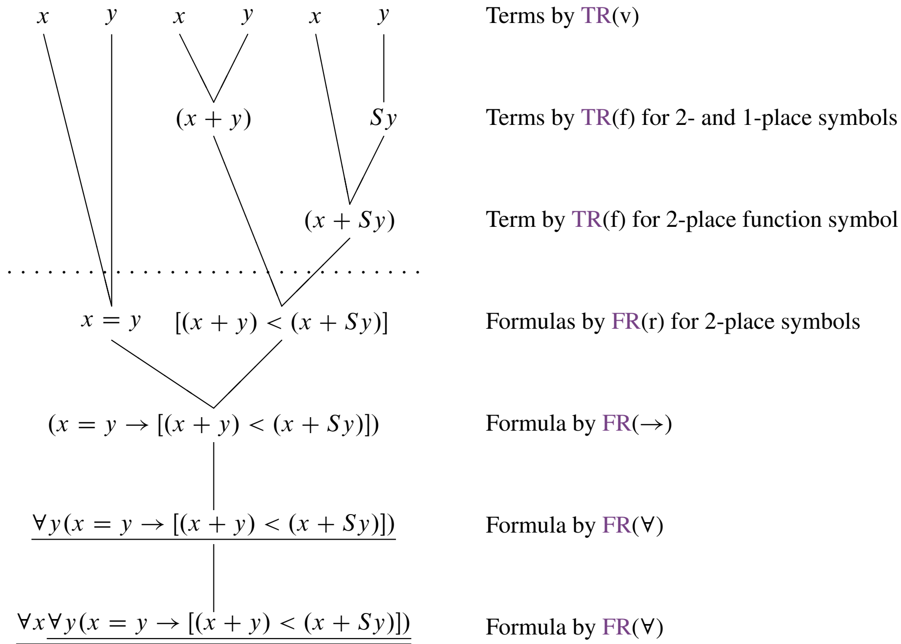
Two-place relation symbols: $=, <$

and that is all. Later we shall introduce a language like $\mathscr{L}_{NT}^{<}$ except without the $<$ symbol; for now, we leave it in. Notice that $\mathscr{L}_q$ uses capitals for sentence letters and lowercase for function symbols. But there is nothing sacred about this. Similarly, $\mathscr{L}_q$ indicates the number of places for function and relation symbols by superscripts, where in $\mathscr{L}_{NT}^{<}$ the number of places is simply built into the definition of the symbol. In fact, $\mathscr{L}_{NT}^{<}$ is an extremely simple language! Given the vocabulary, TR and FR apply in the usual way. Thus $\emptyset$, $S\emptyset$, and $SS\emptyset$ are terms—as is easy to see on a tree. And $<\emptyset SS\emptyset$ is an atomic formula.

As with our treatment for equality, for terms m and n, we often abbreviate official terms of the sort, $+mn$ and $\times mn$ as $(m + n)$ and $(m \times n)$; similarly, it is often convenient to abbreviate an atomic formula $<mn$ as $(m < n)$. And we will drop these parentheses when there is no danger of confusion. Officially, we have not said a word about what these expressions mean. It is natural, however, to think of them with their usual meanings, with S the *successor* function—so that the successor of zero, $S\emptyset$ is one, the successor of the successor of zero $SS\emptyset$ is two, and so forth. But we do not need to think about that for now.

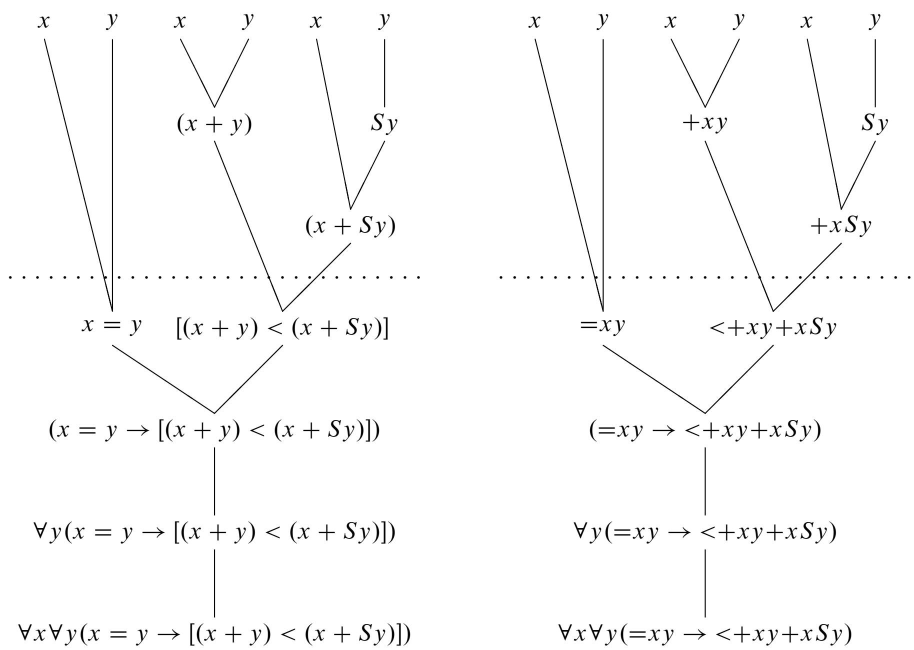
As an example, we show that $\forall x \forall y(x = y \rightarrow [(x + y) < (x + Sy)])$ is (an abbreviation of) a formula and a sentence.

(W)



And we can show what it abbreviates by unpacking the abbreviation in the usual way. This time, we need to pay attention to abbreviations in the terms as well as formulas.

(X)



The official (Polish) notation on the right may seem strange. But it follows the official definitions TR and FR. And it conveniently reduces the number of parentheses from the more typical infix presentation. (You may also be familiar with Polish notation from certain computer applications.) If you are comfortable with grammar and abbreviations for this language $\mathcal{L}_{NT}^{<}$, you are doing well with the grammar for our formal languages.

E2.21. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathcal{L}_{NT}^{<}$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. $\sim[S\emptyset = (S\emptyset \times SS\emptyset)]$

*b. $\exists x \forall y(x \times y = x)$

c. $\forall x[\sim(x = \emptyset) \rightarrow \exists y(y < x)]$

d. $\forall y[(x < y \vee x = y) \vee y < x]$

e. $\forall x \forall y \forall z[(x \times (y + z)) = ((x \times y) + (x \times z))]$

*E2.22. For each of the formulas in E2.21, produce an unabbreviating tree to find the unabbreviated expression it represents.

E2.23. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The vocabulary for a quantificational language and then for $\mathcal{L}_q$ and $\mathcal{L}_{NT}^{<}$.

b. A formula and a sentence of a quantificational language.

c. An abbreviation for an official formula and sentence of a quantificational language.

Chapter 3

Axiomatic Deduction

We have not yet said what our sentences mean. This is just what we do in the next chapter. However, just as it is possible to do grammar without reference to meaning, so it is possible to do derivations without reference to meaning. Derivations are *defined* purely in relation to the syntax of formal expressions. That is why it is crucial to *show* that derivations stand in important relations to validity and truth, as we do in Part III. And that is why it is possible to do derivations without knowing what the expressions mean. In this chapter we develop an *axiomatic* derivation system without any reference to meaning and truth. Apart from relations to meaning and truth, derivations are perfectly well-defined—counting at least as a sort of puzzle or game with, perhaps, a related "thrill of victory" and "agony of defeat." And as with a game, it is possible to build derivation skills to become a better player. Later, we will show how derivation games matter.[1]

Derivation systems are constructed for different purposes. Introductions to mathematical logic typically employ an *axiomatic* approach. We will see a *natural deduction* system in Chapter 6. The advantage of axiomatic systems is their extreme simplicity. From a practical point of view, when we want to think *about* logic, it is convenient to have a relatively simple object to think about. Axiomatic systems have this advantage, though they can be relatively difficult to apply. The axiomatic approach makes it natural to build toward increasingly complex and powerful results. However, in the beginning at least, axiomatic derivations can be challenging!

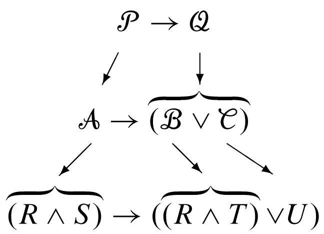
We will introduce our system in stages: After some general remarks in section 3.1 about what an axiom system is supposed to be, we will introduce the sentential component of our system (section 3.2). After that, we will turn to the full system for forms with quantifiers and equality (section 3.3), and finally to a mathematical application (section 3.4).

[1] This chapter has its place to crystallize the point about form. However it is out of order from a learning point of view. Having developed the grammar of our formal languages, a sensible course in mathematical logic will skip to Chapter 4 and return only after Chapter 6. You might attempt section 3.1 to get the basic idea. But then compare the box on page 69.

3.1 General

Before turning to the derivations themselves, it will be helpful to make a point about the metalanguage and form. We are familiar with the idea that different formulas may be of the same form. Thus, for example, where $\mathscr{P}$ and $\mathscr{Q}$ are formulas, $A \rightarrow B$ and $A \rightarrow (B \vee C)$ are both of the form $\mathscr{P} \rightarrow \mathscr{Q}$—in the one case $\mathscr{Q}$ maps to B, and in the other to $(B \vee C)$. But, similarly, one form may map to another. Thus, for example, $\mathscr{P} \rightarrow \mathscr{Q}$ maps to $\mathscr{A} \rightarrow (\mathscr{B} \vee \mathscr{C})$.

(A)



And, by a sort of derived map, any formula of the form $\mathscr{A} \rightarrow (\mathscr{B} \vee \mathscr{C})$ is of the form $\mathscr{P} \rightarrow \mathscr{Q}$ as well. In this chapter we frequently apply one form to another—depending on the fact that all formulas of one form are of another.

Given a formal language $\mathcal{L}$, an axiomatic logic *AL* consists of two parts. There is a set of *axioms* and a set of *rules*. Different axiomatic logics result from different axioms and rules. For now, the set of axioms is just some privileged collection of formulas. A rule tells us that one formula *follows* from some others. One way to specify axioms and rules is by form. Thus, for example, *modus ponens* may be included among the rules.

$$\text{MP} \qquad \frac{\mathscr{P} \rightarrow \mathscr{Q},\ \mathscr{P}}{\mathscr{Q}}$$

According to this rule, for any formulas $\mathscr{P}$ and $\mathscr{Q}$, the formula $\mathscr{Q}$ *follows* from $\mathscr{P} \rightarrow \mathscr{Q}$ together with $\mathscr{P}$. Thus, as applied to $\mathcal{L}_s$, B follows by MP from $A \rightarrow B$ and A; but also $(B \leftrightarrow D)$ follows from $(A \rightarrow B) \rightarrow (B \leftrightarrow D)$ and $(A \rightarrow B)$. And for a case put in the metalanguage, quite generally, a formula of the form $(\mathscr{B} \vee \mathscr{C})$ follows from $\mathscr{A} \rightarrow (\mathscr{B} \vee \mathscr{C})$ and $\mathscr{A}$—for any formulas of the form $\mathscr{A} \rightarrow (\mathscr{B} \vee \mathscr{C})$ and $\mathscr{A}$ are of the forms $\mathscr{P} \rightarrow \mathscr{Q}$ and $\mathscr{P}$ as well. Axioms also may be specified by form. Thus, for some language with formulas $\mathscr{P}$ and $\mathscr{Q}$, a logic might include among its axioms all formulas of the forms,

$$\wedge 1 \quad (\mathscr{P} \wedge \mathscr{Q}) \rightarrow \mathscr{P} \qquad \wedge 2 \quad (\mathscr{P} \wedge \mathscr{Q}) \rightarrow \mathscr{Q} \qquad \wedge 3 \quad \mathscr{P} \rightarrow (\mathscr{Q} \rightarrow (\mathscr{P} \wedge \mathscr{Q}))$$

Then in $\mathcal{L}_s$,

$$(A \wedge B) \rightarrow A, \qquad (A \wedge A) \rightarrow A \qquad ((A \rightarrow B) \wedge C) \rightarrow (A \rightarrow B)$$

are all axioms of form $\wedge 1$. Insofar as each has indefinitely many instances, $\wedge 1$–$\wedge 3$ are axiom *schemas* (or *schemata*). So far, for a given axiomatic logic *AL*, there are no constraints on just which formulas will be the axioms, and just which rules are included. The point is only that we specify an axiomatic logic when we specify some collection of axioms and rules.

Suppose we have specified some axioms and rules for an axiomatic logic *AL*. Then where Γ (Gamma) is a set of formulas—taken as the formal *premises* of an argument,

AV (p) If $\mathscr{P}$ is a premise (a member of Γ), then $\mathscr{P}$ is a *consequence* in *AL* of Γ.

(a) If $\mathscr{P}$ is an axiom of *AL*, then $\mathscr{P}$ is a *consequence* in *AL* of Γ.

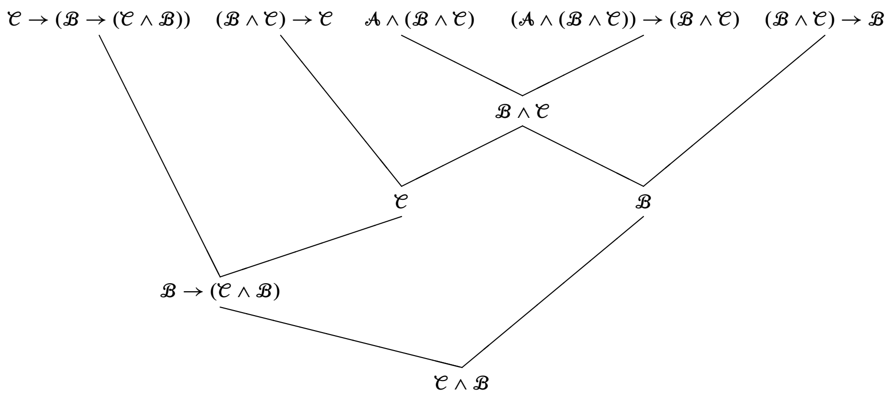
(r) If $\mathscr{Q}_1 \ldots \mathscr{Q}_n$ are consequences in *AL* of Γ, and there is a rule of *AL* such that $\mathscr{P}$ follows from $\mathscr{Q}_1 \ldots \mathscr{Q}_n$ by the rule, then $\mathscr{P}$ is a *consequence* in *AL* of Γ.

(CL) Any consequence in *AL* of Γ may be obtained by repeated application of these rules.

The first two clauses make premises and axioms consequences in *AL* of Γ. And if, say, MP is a rule of *AL* and $P \to Q$ and P are consequences in *AL* of Γ, then by AV(r), Q is a consequence in *AL* of Γ as well. If $\mathscr{P}$ is a consequence in *AL* of some premises Γ, then the premises *prove* $\mathscr{P}$ in *AL* and equivalently the argument is *valid* in *AL;* in this case we write $\Gamma \vdash_{AL} \mathscr{P}$. The $\vdash$ symbol is the *single turnstile* (to contrast with a *double* turnstile $\vDash$ from Chapter 4). If $\mathscr{Q}_1 \ldots \mathscr{Q}_n$ are the members of Γ, we sometimes write $\mathscr{Q}_1 \ldots \mathscr{Q}_n \vdash_{AL} \mathscr{P}$ in place of $\Gamma \vdash_{AL} \mathscr{P}$. If Γ has no members and $\Gamma \vdash_{AL} \mathscr{P}$, then $\mathscr{P}$ is a *theorem* of *AL*. In this case we simply write, $\vdash_{AL} \mathscr{P}$.

Before turning to our official axiomatic system *AD*, it will be helpful to consider a preliminary example. Suppose an axiomatic derivation system *AP* has MP as its only rule, and just formulas of the forms $\wedge$1, $\wedge$2, and $\wedge$3 as axioms. AV is a recursive definition like ones we have seen before. Thus nothing stops us from working out its consequences on trees. Thus we can show that $\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C}) \vdash_{AP} \mathscr{C} \wedge \mathscr{B}$ as follows:

(B)



In this case, the only member of Γ is the premise, $\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C})$. For definition AV, the basic elements are the premises and axioms. These occur across the top row. Thus, reading from the left, the first form is an instance of $\wedge$3. The second is of type $\wedge$2. The third is the premise. Any formula of the form $(\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C})) \to (\mathscr{B} \wedge \mathscr{C})$ is of the form, $(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{Q}$; so the fourth is of the type $\wedge$2. And the last is of the type $\wedge$1. So by AV(a) and AV(p) they are all consequences in *AP* of Γ. After that, all the results are by MP, and so consequences by AV(r). Thus for example, in the second row, $(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})) \rightarrow (\mathcal{B} \wedge \mathcal{C})$ and $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ are of the sort $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$, with $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ for $\mathcal{P}$ and $(\mathcal{B} \wedge \mathcal{C})$ for $\mathcal{Q}$; thus $\mathcal{B} \wedge \mathcal{C}$ follows from them by MP. So $\mathcal{B} \wedge \mathcal{C}$ is a consequence in *AP* of Γ by AV(r). And similarly for the other consequences. Notice that applications of MP and of the axiom forms are *independent* from one use to the next. The expressions that count as $\mathcal{P}$ or $\mathcal{Q}$ must be consistent within a given application of the axiom or rule, but may vary from one application of the axiom or rule to the next. If you are familiar with another derivation system, perhaps the one from Chapter 6, you may think of an axiom as a rule without inputs. Then the axiom applies to expressions of its form in the usual way.

These diagrams can get messy, and it is traditional to represent the same information as follows, using annotations to indicate relations among formulas:

(C)

1.	$\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$	prem(ise)
2.	$(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})) \rightarrow (\mathcal{B} \wedge \mathcal{C})$	$\wedge$2
3.	$\mathcal{B} \wedge \mathcal{C}$	2,1 MP
4.	$(\mathcal{B} \wedge \mathcal{C}) \rightarrow \mathcal{B}$	$\wedge$1
5.	$\mathcal{B}$	4,3 MP
6.	$(\mathcal{B} \wedge \mathcal{C}) \rightarrow \mathcal{C}$	$\wedge$2
7.	$\mathcal{C}$	6,3 MP
8.	$\mathcal{C} \rightarrow (\mathcal{B} \rightarrow (\mathcal{C} \wedge \mathcal{B}))$	$\wedge$3
9.	$\mathcal{B} \rightarrow (\mathcal{C} \wedge \mathcal{B})$	8,7 MP
10.	$\mathcal{C} \wedge \mathcal{B}$	9,5 MP

Each of the forms (1)–(10) is a consequence of $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ in *AP*. As indicated on the right, the first is a premise, and so a consequence by AV(p). The second is an axiom of the form $\wedge$2, and so a consequence by AV(a). The third follows by MP from the forms on lines (2) and (1), and so is a consequence by AV(r). And so forth. Such a demonstration is an *axiomatic derivation*. This derivation contains the very same information as the tree diagram (B), only with geometric arrangement replaced by line numbers to indicate relations between forms. Observe that we might have accomplished the same end with a different arrangement of lines. For example, we might have listed all the axioms first, with applications of MP after. The important point is that in an *axiomatic derivation*, each line is either an axiom, a premise, or follows from previous lines by a rule. Just as a tree is sufficient to demonstrate that $\Gamma \vdash_{AL} \mathcal{P}$, that $\mathcal{P}$ is a consequence of Γ in *AL*, so an axiomatic derivation is sufficient to show the same. In fact, we shall typically use derivations rather than trees to show that $\Gamma \vdash_{AL} \mathcal{P}$.

Notice that we have been reasoning with sentence *forms*. Thus we treat both general forms and particular formulas as "instances" of an axiom scheme. Correspondingly, we have shown that a formula of the form $\mathcal{C} \wedge \mathcal{B}$ follows in *AP* from one of the form $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$. Given this, we freely appeal to results of one derivation in the process of doing another. Thus, if we were to encounter a formula of the form $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ in an *AP* derivation, we might simply cite the derivation (C) completed above, and move directly to the conclusion that $\mathcal{C} \wedge \mathcal{B}$. The resultant derivation would be an *abbreviation* of an official one which includes each of the above steps to reach $\mathcal{C} \wedge \mathcal{B}$. In this way, derivations remain manageable, and we are able to build toward results of increasing complexity. (Compare the way theorems build upon one another from your high school experience of Euclidian geometry.) All of this should become more clear as we turn to the official and complete axiomatic system, *AD*.

E3.1. Where *AP* is as above with rule MP and axioms $\wedge$1–$\wedge$3, construct derivations to show each of the following.

*a. $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \vdash_{AP} \mathcal{B}$

b. $\mathcal{A}, \mathcal{B}, \mathcal{C} \vdash_{AP} \mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$

c. $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \vdash_{AP} (\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}$

d. $(\mathcal{A} \wedge \mathcal{B}) \wedge (\mathcal{C} \wedge \mathcal{D}) \vdash_{AP} \mathcal{B} \wedge \mathcal{C}$

e. $\vdash_{AP} ((\mathcal{A} \wedge \mathcal{B}) \to \mathcal{A}) \wedge ((\mathcal{A} \wedge \mathcal{B}) \to \mathcal{B})$

E3.2. Demonstrate E3.1a by a tree diagram, as for (B) above.

On a course in symbolic logic: Unless you have a special reason for studying axiomatic systems, or are just looking for some really challenging puzzles, you should pass over the rest of this chapter until you have completed Chapter 6. At that stage, you will be better prepared for this one. Chapter 3 is not required for any of chapters 4–7. It makes sense here to locate derivations in the conceptual order, and so to underline the point that derivations are defined apart from notions of validity and truth as we encounter them in Chapter 4—and thus to highlight the importance of *showing* that the same arguments come out valid on the different accounts, as we do in Part III. But this chapter is out of order from a learning point of view. After Chapter 6 you can return to this chapter, while recognizing its place in the conceptual order (see note 1 on page 65).

3.2 Sentential

We begin by focusing on sentential forms, forms involving just $\sim$ and $\to$ (and so $\wedge$, $\vee$, and $\leftrightarrow$). The sentential component *ADs* of our official axiomatic logic *AD* tells us how to manipulate such forms, whether they be forms for expressions in a sentential language like $\mathcal{L}_s$, or in a quantificational language like $\mathcal{L}_q$. *ADs* includes three axiom forms and one rule:

ADs A1. $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

A2. $(\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

A3. $(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$

MP. $\mathcal{Q}$ follows from $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$

We have already encountered MP. To take some cases to appear immediately below, the following are both of the sort A1:

$$\mathcal{A} \to (\mathcal{A} \to \mathcal{A}) \qquad (\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$$

Observe that $\mathcal{P}$ and $\mathcal{Q}$ need not be different. You should be clear about these cases. Although MP is the only rule, we allow free movement between an expression and its abbreviated forms, with justification, 'abv'. That is it! As above, $\Gamma \vdash_{ADs} \mathcal{P}$ just in case $\mathcal{P}$ is a consequence of Γ in *ADs*. $\Gamma \vdash_{ADs} \mathcal{P}$ just in case there is an *ADs* derivation of $\mathcal{P}$ from premises in Γ.

The following is a series of derivations where, as we shall see, each may depend on ones from before. At first, do not worry so much about strategy, as about the mechanics of the system.

T3.1. $\vdash_{ADs} \mathcal{A} \to \mathcal{A}$

1.	$(\mathcal{A} \to ([\mathcal{A} \to \mathcal{A}] \to \mathcal{A})) \to ((\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]) \to (\mathcal{A} \to \mathcal{A}))$	A2
2.	$\mathcal{A} \to ([\mathcal{A} \to \mathcal{A}] \to \mathcal{A})$	A1
3.	$(\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]) \to (\mathcal{A} \to \mathcal{A})$	1,2 MP
4.	$\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]$	A1
5.	$\mathcal{A} \to \mathcal{A}$	3,4 MP

Line (1) is an axiom of the form A2 with $\mathcal{A}$ for $\mathcal{O}$, $\mathcal{A} \to \mathcal{A}$ for $\mathcal{P}$, and $\mathcal{A}$ for $\mathcal{Q}$. Notice again that $\mathcal{O}$ and $\mathcal{Q}$ may be any formulas, so nothing prevents them from being the same. Line (2) is an axiom of the form A1 with $\mathcal{A} \to \mathcal{A}$ for $\mathcal{Q}$. Similarly, line (4) is an axiom of the form A1 with $\mathcal{A}$ in place of both $\mathcal{P}$ and $\mathcal{Q}$. The applications of MP should be straightforward.

T3.2. $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C} \vdash_{ADs} \mathcal{A} \to \mathcal{C}$

1.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
2.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
3.	$\mathcal{B} \to \mathcal{C}$	prem
4.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2,3 MP
5.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	1,4 MP
6.	$\mathcal{A} \to \mathcal{B}$	prem
7.	$\mathcal{A} \to \mathcal{C}$	5,6 MP

Line (1) is an instance of A2 which gives us our goal with two applications of MP—that is, from (1), $\mathcal{A} \to \mathcal{C}$ follows by MP if we have $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$ and $\mathcal{A} \to \mathcal{B}$. But the second of these is a premise, so the only real challenge is getting $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$.

But since $\mathcal{B} \to \mathcal{C}$ is a premise, we can use A1 to get *anything* arrow it—and that is just what we do on lines (2)–(4).

T3.3. $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{ADs} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$

1.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
2.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
3.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	1,2 MP
4.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$	A1
5.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	4,3 T3.2

In this case, the first four steps are very much like ones you have seen before. But the last is not. T3.2 lets us move from $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$ to $\mathcal{A} \to \mathcal{C}$; it is a sort of transitivity or "chain" principle which lets us move from a first form to a last through some middle term. We have $\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$ on line (4), and $(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$ on line (3). These are of the form to be inputs to T3.2—with $\mathcal{B}$ for $\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$ for $\mathcal{B}$, and $\mathcal{A} \to \mathcal{C}$ for $\mathcal{C}$. In this case, $\mathcal{A} \to \mathcal{B}$ is the middle term. So at line (5), we simply observe that lines (4) and (3), together with the reasoning from T3.2, give us the desired result.

T3.2 is an important principle, of significance comparable to MP for the way you think about derivations. If you have $\mathcal{X} \to \mathcal{A}$ and want $\mathcal{A}$, it makes sense to go for $\mathcal{X}$ towards an application of MP. But if you have $\mathcal{A} \to \mathcal{X}$ and want $\mathcal{A} \to \mathcal{B}$, it makes sense to go for $\mathcal{X} \to \mathcal{B}$ toward an application of T3.2. And similarly if you have $\mathcal{X} \to \mathcal{B}$ and want $\mathcal{A} \to \mathcal{B}$, it makes sense to go for $\mathcal{A} \to \mathcal{X}$ for T3.2. At (3) of the above derivation we are in a situation of this latter sort, and so obtain (4).

What we have produced above is not an official derivation where each step is a premise, an axiom, or follows from previous lines by a rule. But we have produced an abbreviation of one. And nothing prevents us from unabbreviating by including the routine from T3.2 to produce a derivation in the official form. To see this, first observe that the derivation for T3.2 has its premises at lines (3) and (6), where lines with the corresponding forms in the derivation for T3.3 appear at (3) and (4). However, it is a simple matter to reorder the derivation for T3.2 so that it takes its premises from those same lines. Thus here is another demonstration for T3.2:

	3.	$\mathcal{B} \to \mathcal{C}$	prem
	4.	$\mathcal{A} \to \mathcal{B}$	prem
	5.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
(D)	6.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
	7.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	6,3 MP
	8.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	5,7 MP
	9.	$\mathcal{A} \to \mathcal{C}$	8,4 MP

Compared to the original derivation for T3.2, all that is different is the order of a few lines, and corresponding line numbers. The *reason* for reordering the lines is for a merge of this derivation with the one for T3.3.

But now, although we are after expressions of the *form* $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$, the actual forms we want for T3.3 are $\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$ and $(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$. But we can convert derivation (D) to one with those very forms by uniform substituation of $\mathcal{B}$ for every $\mathcal{A}$; $(\mathcal{A} \to \mathcal{B})$ for every $\mathcal{B}$; and $(\mathcal{A} \to \mathcal{C})$ for every $\mathcal{C}$—that is, we apply our original map to the entire derivation (D). The result is as follows:

$$
\text{(E)}\quad
\begin{array}{lll}
3. & (\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}) & \text{prem} \\
4. & \mathcal{B} \to (\mathcal{A} \to \mathcal{B}) & \text{prem} \\
5. & [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] \to [(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))] & \text{A2} \\
6. & ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) \to [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] & \text{A1} \\
7. & \mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) & \text{6,3 MP} \\
8. & (\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C})) & \text{5,7 MP} \\
9. & \mathcal{B} \to (\mathcal{A} \to \mathcal{C}) & \text{8,4 MP}
\end{array}
$$

You should trace the parallel between derivations (D) and (E) all the way through. And you should verify that (E) is a derivation on its own. This is an application of the point that our derivation for T3.2 applies to any premises and conclusions of that form. The result is a direct demonstration that $\mathcal{B} \to (\mathcal{A} \to \mathcal{B}), (\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}) \vdash_{ADs} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$.

And now it is a simple matter to merge the lines from (E) into the derivation for T3.3 to produce a complete demonstration that $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{ADs} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$.

$$
\text{(F)}\quad
\begin{array}{lll}
1. & [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})] & \text{A2} \\
2. & \mathcal{A} \to (\mathcal{B} \to \mathcal{C}) & \text{prem} \\
3. & (\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}) & \text{1,2 MP} \\
4. & \mathcal{B} \to (\mathcal{A} \to \mathcal{B}) & \text{A1} \\
5. & [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] \to [(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))] & \text{A2} \\
6. & ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) \to [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] & \text{A1} \\
7. & \mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) & \text{6,3 MP} \\
8. & (\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C})) & \text{5,7 MP} \\
9. & \mathcal{B} \to (\mathcal{A} \to \mathcal{C}) & \text{8,4 MP}
\end{array}
$$

Lines (1)–(4) are the same as from the derivation for T3.3, and include what are the premises to (E). Lines (5)–(9) are the same as from (E). The result is a demonstration for T3.3 in which every line is a premise, an axiom, or follows from previous lines by MP. Again, you should follow each step. It is hard to believe that we could *think up* this last derivation—particularly at this early stage of our career. However, if we can produce the simpler derivation, we can be sure that this more complex one exists. Thus we can be sure that the final result is a consequence of the premise in *ADs*. That is the point of our direct appeal to T3.2 in the original derivation of T3.3. And similarly in cases that follow. In general, we are always free to appeal to prior results in any derivation—so that our toolbox gets bigger at every stage. With this in mind, you may find the *ADs* summary on page 78 helpful.

T3.4. $\vdash_{ADs} (\mathcal{B} \rightarrow \mathcal{C}) \rightarrow [(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$

1.	$[\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})] \rightarrow [(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$	A2
2.	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow [\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})]$	A1
3.	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow [(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$	2,1 T3.2

Again we have an application of T3.2. In this case, the middle term (the $\mathcal{B}$) from T3.2 maps to $\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$. Once we see that the consequent of what we want is like the consequent of A2, we should be "inspired" by T3.2 to go for (2) as a link between the antecedent of what we want and antecedent of A2. As it turns out, this is easy to get as an instance of A1. It is helpful to say to yourself in words, what the various axioms and theorems do. Thus, given some $\mathcal{P}$, A1 yields *anything* arrow it. And T3.2 is a simple transitivity principle.

T3.5. $\vdash_{ADs} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow [(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$

1.	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow [(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$	T3.4
2.	$(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow [(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$	1 T3.3

T3.5 is like T3.4 except that $\mathcal{A} \rightarrow \mathcal{B}$ and $\mathcal{B} \rightarrow \mathcal{C}$ switch places. But T3.3 precisely switches terms in those places—with $\mathcal{B} \rightarrow \mathcal{C}$ for $\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$ for $\mathcal{B}$, and $\mathcal{A} \rightarrow \mathcal{C}$ for $\mathcal{C}$. Again, often what is difficult about these derivations is "seeing" what you can do. Thus it is good to say to yourself in words what the different principles give you. Once you realize what T3.3 does, it is obvious that you have T3.5 immediately from T3.4.

T3.6. $\mathcal{B}, \mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \vdash_{ADs} \mathcal{A} \rightarrow \mathcal{C}$

Hint: You can get this in the basic system using just A1 and A2. But you can get it in just four lines if you use T3.3.

T3.7. $\vdash_{ADs} (\sim\mathcal{A} \rightarrow \mathcal{A}) \rightarrow \mathcal{A}$

Hint: This follows in just three lines from A3, with an instance of T3.1.

T3.8. $\vdash_{ADs} (\sim\mathcal{B} \rightarrow \sim\mathcal{A}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$

1.	$(\sim\mathcal{B} \rightarrow \sim\mathcal{A}) \rightarrow [(\sim\mathcal{B} \rightarrow \mathcal{A}) \rightarrow \mathcal{B}]$	A3
2.	$[\mathcal{A} \rightarrow (\sim\mathcal{B} \rightarrow \mathcal{A})] \rightarrow [((\sim\mathcal{B} \rightarrow \mathcal{A}) \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})]$	T3.5
3.	$\mathcal{A} \rightarrow (\sim\mathcal{B} \rightarrow \mathcal{A})$	A1
4.	$((\sim\mathcal{B} \rightarrow \mathcal{A}) \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$	2,3 MP
5.	$(\sim\mathcal{B} \rightarrow \sim\mathcal{A}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$	1,4 T3.2

The idea behind this derivation is that the antecedent of A3 is the antecedent of our goal. So we can get the goal by T3.2 with (1) and (4). That is, given $(\sim\mathcal{B} \rightarrow \sim\mathcal{A}) \rightarrow \mathcal{X}$, what we need to get the goal by an application of T3.2 is $\mathcal{X} \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$. But that is just what (4) is. The challenge is to get (4). Our strategy uses T3.5 with A1. This derivation is not particularly easy to see. Here is another approach, which is not all that easy either:

(G)

1.	$(\sim\mathcal{B} \to \sim\mathcal{A}) \to [(\sim\mathcal{B} \to \mathcal{A}) \to \mathcal{B}]$	A3
2.	$(\sim\mathcal{B} \to \mathcal{A}) \to [(\sim\mathcal{B} \to \sim\mathcal{A}) \to \mathcal{B}]$	1 T3.3
3.	$\mathcal{A} \to (\sim\mathcal{B} \to \mathcal{A})$	A1
4.	$\mathcal{A} \to [(\sim\mathcal{B} \to \sim\mathcal{A}) \to \mathcal{B}]$	3,2 T3.2
5.	$(\sim\mathcal{B} \to \sim\mathcal{A}) \to (\mathcal{A} \to \mathcal{B})$	4 T3.3

This derivation also begins with A3. The idea this time is to use T3.3 to "swing" $\sim\mathcal{B} \to \mathcal{A}$ out, "replace" it by $\mathcal{A}$ with T3.2 and A1, and then use T3.3 to "swing" $\mathcal{A}$ back in.

T3.9. $\vdash_{ADs} \sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$

Hint: You can do this in three lines with T3.8 and an instance of A1.

T3.10. $\vdash_{ADs} \sim\sim\mathcal{A} \to \mathcal{A}$

Hint: You can do this in three lines wih instances of T3.7 and T3.9.

T3.11. $\vdash_{ADs} \mathcal{A} \to \sim\sim\mathcal{A}$

Hint: You can do this in three lines with instances of T3.8 and T3.10.

*T3.12. $\vdash_{ADs} (\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \sim\sim\mathcal{B})$

Hint: Use T3.5 and T3.10 to get $(\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \mathcal{B})$; then use T3.4 and T3.11 to get $(\sim\sim\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \sim\sim\mathcal{B})$; the result follows easily by T3.2.

T3.13. $\vdash_{ADs} (\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \sim\mathcal{A})$

Hint: You can do this in three lines with instances of T3.8 and T3.12.

T3.14. $\vdash_{ADs} (\sim\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \mathcal{A})$

Hint: Use T3.4 and T3.10 to get $(\sim\mathcal{B} \to \sim\sim\mathcal{A}) \to (\sim\mathcal{B} \to \mathcal{A})$; the result follows easily with an instance of T3.13.

T3.15. $\vdash_{ADs} (\mathcal{A} \to \sim\mathcal{B}) \to (\mathcal{B} \to \sim\mathcal{A})$

Hint: This time you will be able to use T3.5 and T3.11 with T3.13.

T3.16. $\vdash_{ADs} (\mathcal{A} \to \mathcal{B}) \to [(\sim\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$

Hint: Use T3.13 and A3 to get $(\mathcal{A} \to \mathcal{B}) \to [(\sim\mathcal{B} \to \mathcal{A}) \to \mathcal{B}]$; then use T3.5 and T3.14 to get $[(\sim\mathcal{B} \to \mathcal{A}) \to \mathcal{B}] \to [(\sim\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$; the result follows easily by T3.2.

*T3.17. $\vdash_{ADs} \mathcal{A} \to [\sim\mathcal{B} \to \sim(\mathcal{A} \to \mathcal{B})]$

Hint: Use T3.1 and T3.3 to get $\mathcal{A} \to [(\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$; then use T3.13 to "turn around" the consequent. This idea of deriving conditionals in "reversed" form, and then using one of T3.13–T3.15 to turn them around, is frequently useful for getting tilde outside of a complex expression.

T3.18. $\vdash_{ADs} \mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$

1.	$\sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	T3.9
2.	$\mathcal{A} \to (\sim\mathcal{A} \to \mathcal{B})$	1 T3.3
3.	$\mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$	2 abv

We set as our goal the unabbreviated form. We have this at (2). Then, in the last line, simply observe that the goal abbreviates what has already been shown.

T3.19. $\vdash_{ADs} \mathcal{A} \to (\mathcal{B} \vee \mathcal{A})$

Hint: Go for $\mathcal{A} \to (\sim\mathcal{B} \to \mathcal{A})$. Then, as above, you can get the desired result in one step by abv.

T3.20. $\vdash_{ADs} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{B}$

T3.21. $\vdash_{ADs} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{A}$

*T3.22. $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{ADs} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}$

T3.23. $(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C} \vdash_{ADs} \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$

T3.24. $\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{ADs} \mathcal{B}$

Hint: $\mathcal{A} \leftrightarrow \mathcal{B}$ abbreviates the same thing as $(\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})$; you may thus move to this expression from $\mathcal{A} \leftrightarrow \mathcal{B}$ by abv.

T3.25. $\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{ADs} \mathcal{A}$

T3.26. $\sim\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{ADs} \sim\mathcal{B}$

T3.27. $\sim\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{ADs} \sim\mathcal{A}$

*E3.3. Provide derivations for T3.6–T3.7, T3.9–T3.17, and T3.19–T3.27. Again, as you are working these problems, you may find it helpful to refer to the *ADs* summary on page 78.

E3.4. For each of the following, expand derivations to include all the steps from theorems. The result should be a derivation in which each step is either a premise, an axiom, or follows from previous lines by a rule. Hint: It may be helpful to proceed in stages as for (D), (E), and then (F) above.

a. Expand your derivation for T3.7.

*b. Expand the above derivation for T3.4.

E3.5. Consider an axiomatic system A^* which takes $\wedge$ and $\sim$ as primitive operators, and treats $\mathcal{P} \rightarrow \mathcal{Q}$ as an abbreviation for $\sim(\mathcal{P} \wedge \sim\mathcal{Q})$. Forms for the axioms and rule are,

A^* A1. $\mathcal{P} \rightarrow (\mathcal{P} \wedge \mathcal{P})$
A2. $(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{P}$
A3. $(\mathcal{O} \rightarrow \mathcal{P}) \rightarrow [\sim(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \wedge \mathcal{O})]$
MP. $\sim(\mathcal{P} \wedge \sim\mathcal{Q}), \mathcal{P} \vdash_{A^*} \mathcal{Q}$ (so that $\mathcal{P} \rightarrow \mathcal{Q}, \mathcal{P} \vdash_{A^*} \mathcal{Q}$)

Provide derivations for each of the following, where derivations may appeal to any *prior* result (no matter what *you* have done).

*a. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{A^*} \sim(\sim\mathcal{C} \wedge \mathcal{A})$

*b. $\vdash_{A^*} \sim(\sim\mathcal{A} \wedge \mathcal{A})$

*c. $\vdash_{A^*} \sim\sim\mathcal{A} \rightarrow \mathcal{A}$

*d. $\vdash_{A^*} \sim(\mathcal{A} \wedge \mathcal{B}) \rightarrow (\mathcal{B} \rightarrow \sim\mathcal{A})$

e. $\vdash_{A^*} \mathcal{A} \rightarrow \sim\sim\mathcal{A}$

f. $\vdash_{A^*} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\sim\mathcal{B} \rightarrow \sim\mathcal{A})$

*g. $\sim\mathcal{A} \rightarrow \sim\mathcal{B} \vdash_{A^*} \mathcal{B} \rightarrow \mathcal{A}$

h. $\mathcal{A} \rightarrow \mathcal{B} \vdash_{A^*} (\mathcal{C} \wedge \mathcal{A}) \rightarrow (\mathcal{B} \wedge \mathcal{C})$

*i. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C}, \mathcal{C} \rightarrow \mathcal{D} \vdash_{A^*} \mathcal{A} \rightarrow \mathcal{D}$

j. $\vdash_{A^*} \mathcal{A} \rightarrow \mathcal{A}$

k. $\vdash_{A^*} (\mathcal{A} \wedge \mathcal{B}) \rightarrow (\mathcal{B} \wedge \mathcal{A})$

l. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{A^*} \mathcal{A} \rightarrow \mathcal{C}$

m. $\sim\mathcal{B} \rightarrow \mathcal{B} \vdash_{A^*} \mathcal{B}$

n. $\mathcal{B} \rightarrow \sim\mathcal{B} \vdash_{A^*} \sim\mathcal{B}$

o. $\vdash_{A^*} (\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{B}$

p. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{C} \rightarrow \mathcal{D} \vdash_{A^*} (\mathcal{A} \wedge \mathcal{C}) \rightarrow (\mathcal{B} \wedge \mathcal{D})$

q. $\mathcal{B} \to \mathcal{C} \vdash_{A*} (\mathcal{A} \wedge \mathcal{B}) \to (\mathcal{A} \wedge \mathcal{C})$

r. $\mathcal{A} \to \mathcal{B}, \mathcal{A} \to \mathcal{C} \vdash_{A*} \mathcal{A} \to (\mathcal{B} \wedge \mathcal{C})$

s. $\vdash_{A*} [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}] \to [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})]$

t. $\vdash_{A*} [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})] \to [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}]$

*u. $\vdash_{A*} [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C})]$

v. $\vdash_{A*} [(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}] \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$

w. $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{A*} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$

*x. $\mathcal{A} \to \mathcal{B}, \mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{A*} \mathcal{A} \to \mathcal{C}$

y. $\vdash_{A*} \mathcal{A} \to [\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B})]$

z. $\vdash_{A*} \mathcal{A} \to (\mathcal{B} \to \mathcal{A})$

Hints: (i): Apply (a) to the first two premises and (f) to the third; then recognize that you have the makings for an application of A3. (j): Apply A1, two instances of (h), and an instance of (i) to get $\mathcal{A} \to ((\mathcal{A} \wedge \mathcal{A}) \wedge (\mathcal{A} \wedge \mathcal{A}))$; the result follows easily with A2 and (i). (m): $\sim\mathcal{B} \to \mathcal{B}$ is equivalent to $\sim(\sim\mathcal{B} \wedge \sim\mathcal{B})$; and $\sim\mathcal{B} \to (\sim\mathcal{B} \wedge \sim\mathcal{B})$ is immediate from A1; you can turn this around by (f) to get $\sim(\sim\mathcal{B} \wedge \sim\mathcal{B}) \to \sim\sim\mathcal{B}$; then it is easy. (u): Use abv so that you are going for $\sim[\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})] \to \sim[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}]$; plan on getting to this by (f). (v): Structure your proof very much as with (u). (x): Use (u) to set up a "chain" to which you can apply transitivity.

E3.6. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The *syntactical* character of derivation systems.

b. A *consequence* of Γ in some axiomatic logic *AL*, and then a consequence of Γ in *ADs*.

3.3 Quantificational

In this section we complete the system *AD* by introducing a rule and axioms for quantifies and equality. A1–A3 and MP remain from before. There will be two axioms and one rule for manipulating quantifiers, and three axioms for features of equality. As you work through the full system *AD*, you may find it helpful to refer to the *AD* guide on page 84 (as well as the *ADs* guide on the following page).

3.3.1 Quantifiers

First, *ADq* extends *ADs* by the addition of two axioms and one rule for quantified expressions. To state the new axioms, we need a couple of definitions. First, for any formula $\mathcal{A}$, variable x, and term t, say $\mathcal{A}^x_t$ is $\mathcal{A}$ with all the free instances of x replaced by t. And say t is *free for x in $\mathcal{A}$* iff all the variables in the replacing instances of t remain free after substitution in $\mathcal{A}^x_t$. Thus, for example, where $\mathcal{A}$ is $\forall xRxy \vee Px$,

(H) $(\forall xRxy \vee Px)^x_y$ is $\forall xRxy \vee Py$

There are three instances of x in $\forall xRxy \vee Px$, but only the last is free; so y is substituted only for that instance. Since the substituted y is free in the resultant expression, y is free for x in $\forall xRxy \vee Px$. Similarly,

(I) $[\forall x(x = y) \vee Ryx]^y_{f^1x}$ is $\forall x(x = f^1x) \vee Rf^1xx$

Both instances of y in $\forall x(x = y) \vee Ryx$ are free; so our substitution replaces both. But the x in the first instance of f^1x is bound upon substitution; so f^1x is not free

ADs Quick Reference

ADs A1. $\mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

A2. $(\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

A3. $(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$

MP. $\mathcal{Q}$ follows from $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$

T3.1 $\vdash_{AD} \mathcal{A} \rightarrow \mathcal{A}$

T3.2 $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{AD} \mathcal{A} \rightarrow \mathcal{C}$

T3.3 $\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \vdash_{AD} \mathcal{B} \rightarrow (\mathcal{A} \rightarrow \mathcal{C})$

T3.4 $\vdash_{AD} (\mathcal{B} \rightarrow \mathcal{C}) \rightarrow [(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$

T3.5 $\vdash_{AD} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow [(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})]$

T3.6 $\mathcal{B}, \mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \vdash_{AD} \mathcal{A} \rightarrow \mathcal{C}$

T3.7 $\vdash_{AD} (\sim\mathcal{A} \rightarrow \mathcal{A}) \rightarrow \mathcal{A}$

T3.8 $\vdash_{AD} (\sim\mathcal{B} \rightarrow \sim\mathcal{A}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$

T3.9 $\vdash_{AD} \sim\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$

T3.10 $\vdash_{AD} \sim\sim\mathcal{A} \rightarrow \mathcal{A}$

T3.11 $\vdash_{AD} \mathcal{A} \rightarrow \sim\sim\mathcal{A}$

T3.12 $\vdash_{AD} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\sim\sim\mathcal{A} \rightarrow \sim\sim\mathcal{B})$

T3.13 $\vdash_{AD} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\sim\mathcal{B} \rightarrow \sim\mathcal{A})$

T3.14 $\vdash_{AD} (\sim\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\sim\mathcal{B} \rightarrow \mathcal{A})$

T3.15 $\vdash_{AD} (\mathcal{A} \rightarrow \sim\mathcal{B}) \rightarrow (\mathcal{B} \rightarrow \sim\mathcal{A})$

T3.16 $\vdash_{AD} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow [(\sim\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{B}]$

T3.17 $\vdash_{AD} \mathcal{A} \rightarrow [\sim\mathcal{B} \rightarrow \sim(\mathcal{A} \rightarrow \mathcal{B})]$

T3.18 $\vdash_{AD} \mathcal{A} \rightarrow (\mathcal{A} \vee \mathcal{B})$

T3.19 $\vdash_{AD} \mathcal{A} \rightarrow (\mathcal{B} \vee \mathcal{A})$

T3.20 $\vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{B}$

T3.21 $\vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{A}$

T3.22 $\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C}$

T3.23 $(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C} \vdash_{AD} \mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$

T3.24 $\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \mathcal{B}$

T3.25 $\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \mathcal{A}$

T3.26 $\sim\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \sim\mathcal{B}$

T3.27 $\sim\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \sim\mathcal{A}$

(abv) allows free movement between an expression and its abbveviated forms.

for y in $\forall x(x = y) \vee Ryx$. In contrast, f^1z goes into the same places but is free for y in $\forall x(x = y) \vee Ryx$.

Some quick applications: If x is not free in $\mathcal{A}$, then replacing every free instance of x in $\mathcal{A}$ with some term results in no change; so if x is not free in $\mathcal{A}$, then $\mathcal{A}^x_t$ is $\mathcal{A}$. Similarly, $\mathcal{A}^x_x$ is just $\mathcal{A}$ itself. Further, any variable x is sure to be free for itself in a formula $\mathcal{A}$—if every *free* instance of variable x is "replaced" with x, then the replacing instances are sure to be free. Similarly variable-free terms (like constants) are sure to be free for a variable x in a formula $\mathcal{A}$; if a term has no variables, no variable in the replacing term is bound upon substitution for free instances of x. And if $\mathcal{A}$ is quantifier-free then any t is free for variable x in $\mathcal{A}$; if $\mathcal{A}$ has no quantifiers, then no variable in t can be bound upon substitution.

Now we are ready for our axioms and rule. For the quantificational version *ADq* of our axiomatic derivation system, we add axioms A4 and A5, and a rule Gen (*Generalization*) for the universal quantifier.

ADq Includes the axioms and rule of *ADs* and,

A4. $\forall x\mathcal{P} \to \mathcal{P}^x_t$ where t is free for x in $\mathcal{P}$

A5. $\forall x(\mathcal{P} \to \mathcal{Q}) \to (\mathcal{P} \to \forall x\mathcal{Q})$ where x is not free in $\mathcal{P}$

Gen. $\forall x\mathcal{P}$ follows from $\mathcal{P}$

A1, A2, A3, and MP remain from before; then ADq adds two axioms and a rule.

A4 is a conditional whose antecedent has an x-quantifier as main operator; the consequent drops the quantifier, and substitutes term t for each resulting free instance of variable x—subject to the constraint that t is free for x in $\mathcal{P}$. Thus the first line below lists instances of A4 but the second does not.

(J)
$$\forall xRx \to Rx \quad \forall xRx \to Ry \quad \forall xRx \to Ra \quad \forall xRx \to Rf^1z \quad \forall x\forall yRxy \to \forall yRzy$$
$$\forall x\forall yRxy \to \forall yRyy \qquad \forall x\forall yRxy \to \forall yRf^1yy$$

One the first line, the consequents drop the (main) quantifier and substitute a term that is free for x. On the second line, we drop the quantifier and substitute as before; but the substituted terms *are not free;* so the constraint on A4 is violated, and those formulas do not qualify as instances of the axiom.

A5 also comes with a constraint. Instances of A5 have antecedent $\forall x(\mathcal{P} \to \mathcal{Q})$ and consequent $(\mathcal{P} \to \forall x\mathcal{Q})$ *so long as x is not free in $\mathcal{P}$*. Thus the first cases below are instances of A5, where the last is not.

(K)
$$\forall x(Ry \to Sx) \to (Ry \to \forall xSx) \qquad \forall x(Ra \to Sx) \to (Ra \to \forall xSx)$$
$$\forall x(Rx \to Sx) \to (Rx \to \forall xSx)$$

In the first cases, the variable x is not free in $\mathcal{P}$. In the last, however, x is free in $\mathcal{P}$ so that it fails to be an instance of A5.

Gen is a new rule; it lets you move from a formula to its universal quantification. So, for example, by Gen you might move from Px to $\forall xPx$ or from $Ay \to By$ to $\forall y(Ay \to By)$. Continue to move freely between an expression and its abbreviated forms with justification, abv. That is it!

Because the axioms and rule from before remain available, nothing blocks reasoning with sentential forms as before. Thus, for example, $\forall xRx \to \forall xRx$ and, more generally, $\forall x\mathcal{A} \to \forall x\mathcal{A}$ are of the form $\mathcal{A} \to \mathcal{A}$, and we might derive them by exactly the five steps for T3.1 above. Or we might just write them down with justification, T3.1. Similarly any theorem from *ADs* is a theorem of the larger *ADq*.

Here is a way to get $\forall xRx \to \forall xRx$ without either A1 or A2:

(L)

1.	$\forall xRx \to Rx$	A4
2.	$\forall x(\forall xRx \to Rx)$	1 Gen
3.	$\forall x(\forall xRx \to Rx) \to (\forall xRx \to \forall xRx)$	A5
4.	$\forall xRx \to \forall xRx$	3,2 MP

The x is sure to be free for x in Rx; so (1) is an instance of A4. And the only instances of x are bound in $\forall xRx$; so (3) satisfies the constraint on A5. The reasoning is similar in the more general case.

T3.28. $\vdash_{ADq} \forall x\mathcal{A} \to \forall v\mathcal{A}^x_v$ where v is not free in $\forall x\mathcal{A}$ but free for x in $\mathcal{A}$

1.	$\forall x\mathcal{A} \to \mathcal{A}^x_v$	A4
2.	$\forall v(\forall x\mathcal{A} \to \mathcal{A}^x_v)$	1 Gen
3.	$\forall v(\forall x\mathcal{A} \to \mathcal{A}^x_v) \to (\forall x\mathcal{A} \to \forall v\mathcal{A}^x_v)$	A5
4.	$\forall x\mathcal{A} \to \forall v\mathcal{A}^x_v$	3,2 MP

Given the constraints, this derivation works for exactly the same reasons as before. If v is free for x in $\mathcal{A}$, then (1) is a straightforward instance of A4. And if v is not free in $\forall x\mathcal{A}$, the constraint on A5 is sure to be met. The result of derivation (L) is an instance of this more general theorem. The difference is that T3.28 makes room for variable exchange. A simple instance of T3.28 in $\mathcal{L}_q$ is $\vdash_{ADq} \forall xRx \to \forall vRv$. If you are confused about restrictions on the axioms, think about the derivation as applied to this case. While our quantified instances of T3.1 could have been derived by sentential rules, T3.28 cannot; $\forall x\mathcal{A} \to \forall x\mathcal{A}$ has sentential form $\mathcal{A} \to \mathcal{A}$; but when x is not the same as v, $\forall x\mathcal{A} \to \forall v\mathcal{A}^x_v$ has sentential form, $\mathcal{A} \to \mathcal{B}$.

T3.29. $\mathcal{A} \to \mathcal{B} \vdash_{ADq} \mathcal{A} \to \forall x\mathcal{B}$ where x is not free in $\mathcal{A}$

1.	$\mathcal{A} \to \mathcal{B}$	prem
2.	$\forall x(\mathcal{A} \to \mathcal{B})$	1 Gen
3.	$\forall x(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \forall x\mathcal{B})$	A5
4.	$\mathcal{A} \to \forall x\mathcal{B}$	3,2 MP

From the restriction on the theorem, (3) is an instance of A5.

*T3.30. $\vdash_{ADq} \mathcal{A}^x_t \to \exists x\mathcal{A}$ where t is free for x in $\mathcal{A}$

Hint: As in sentential cases, show the unabbreviated form, $\mathcal{A}^x_t \to \sim\forall x{\sim}\mathcal{A}$ and get the final result by abv. You should find $\forall x{\sim}\mathcal{A} \to \sim\mathcal{A}^x_t$ to be a useful instance of A4. Notice that $[\sim\mathcal{A}]^x_t$ is the same expression as $\sim[\mathcal{A}^x_t]$, as all the replacements must go on inside the $\mathcal{A}$.

T3.31. $\vdash_{ADq} \forall x(\mathcal{A} \to \mathcal{B}) \to (\exists x\mathcal{A} \to \mathcal{B})$ where x is not free in $\mathcal{B}$

Hint: Go for an unabbreviated form, and then get the goal by abv. You will find it convenient to apply Gen and then A5 to $\forall x(\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \sim\mathcal{A})$.

T3.32. $\mathcal{A} \to \mathcal{B} \vdash_{ADq} \exists x\mathcal{A} \to \mathcal{B}$ where x is not free in $\mathcal{B}$.

This is a simple application of T3.31.

With these few examples we complete our presentation of the fragment of *AD* for both sentential operators and quantifiers. It remains to add axioms for equality.

*E3.7. Provide derivations for T3.30, T3.31, and T3.32, explaining in words for every step that has a restriction how you know the restriction is met.

E3.8. Provide derivations to show each of the following.

*a. $\forall x(Hx \to Rx), \forall yHy \vdash_{ADq} \forall zRz$

b. $\forall y(Fy \to Gy) \vdash_{ADq} \exists zFz \to \exists xGx$

*c. $\vdash_{ADq} \exists x\forall yRxy \to \forall y\exists xRxy$

d. $\forall y\forall x(Fx \to By) \vdash_{ADq} \forall y(\exists xFx \to By)$

e. $\vdash_{ADq} \exists x(Fx \to \forall yGy) \to \exists x\forall y(Fx \to Gy)$

E3.9. Some systems have a rule like T3.29 with neither A5 nor Gen. Show that this is possible by providing derivations to show $\mathcal{P} \vdash \forall x\mathcal{P}$ and, where x is not free in $\mathcal{P}$, $\vdash \forall x(\mathcal{P} \to \mathcal{Q}) \to (\mathcal{P} \to \forall x\mathcal{Q})$ with T3.29 but without A5 or Gen. Hint: For the first, where $\top$ is any theorem without free variables, you will be able to obtain $\top \to \mathcal{P}$ and apply T3.29 to it. For the second consider uses of T3.22 and T3.23.

3.3.2 Equality

The full derivation system *AD* has the axioms and rule from *ADs*, the axioms and rule from *ADq*, and three axiom forms governing equality. In this case, the axioms assert particularly simple, or basic, facts. For any variables $x_1 \dots x_n$ and y, n-place function symbol h^n, and n-place relation symbol $\mathcal{R}^n$,

AD Includes the axioms and rules of *ADs* and *ADq* and,

A6. $(y = y)$

A7. $(x_i = y) \rightarrow (h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n)$

A8. $(x_i = y) \rightarrow (\mathcal{R}^n x_1 \dots x_i \dots x_n \rightarrow \mathcal{R}^n x_1 \dots y \dots x_n)$

From A6, $x = x$ and $z = z$ are axioms. Of course, these are abbreviations for $=xx$ and $=zz$. This should be straightforward. The others are complicated only by abstract presentation. For A7, $h^n x_1 \dots x_i \dots x_n$ differs from $h^n x_1 \dots y \dots x_n$ just in that variable x_i is replaced by variable y. x_i may be any of the variables in $x_1 \dots x_n$. Thus, for example,

(M) $\quad x = y \rightarrow f^1x = f^1y \qquad x = y \rightarrow f^3wxy = f^3wyy$

are simple examples of A7. In the one case, we have a "string" of one variable and replace the only member based on the equality. In the other case, the string is of three variables, and we replace the second. Similarly, $\mathcal{R}^n x_1 \dots x_i \dots x_n$ differs from $\mathcal{R}^n x_1 \dots y \dots x_n$ just in that variable x_i is replaced by y. x_i may be any of the variables in $x_1 \dots x_n$. Thus, for example,

(N) $\quad x = z \rightarrow (A^1x \rightarrow A^1z) \qquad z = w \rightarrow (A^2xz \rightarrow A^2xw)$

are simple examples of A8.

This completes the axioms and rules of our full derivation system *AD*. As examples, let us begin with some fundamental principles of equality. Suppose that r, s, and t are arbitrary terms.

T3.33. $\vdash_{AD} t = t$ *reflexivity of equality*

1.	$y = y$	A6
2.	$\forall y(y = y)$	1 Gen
3.	$\forall y(y = y) \rightarrow (t = t)$	A4
4.	$t = t$	3,2 MP

Since $y = y$ has no quantifiers, any term t is sure to be free for y in it. So (3) is sure to be an instance of A4. This theorem strengthens A6 insofar as the axiom applies only to variables, but the theorem has application to arbitrary terms. Thus $z = z$ is an instance of the axiom; $z = z$ remains an instance of the theorem, but $f^2xy = f^2xy$ is an instance of the theorem as well. We *convert* variables to terms by Gen with A4 and MP. This pattern repeats in the following.

T3.34. $\vdash_{AD} t = s \to s = t$ *symmetry of equality*

1.	$x = y \to (x = x \to y = x)$	A8
2.	$x = x$	A6
3.	$x = y \to y = x$	2,1 T3.6
4.	$\forall y(x = y \to y = x)$	3 Gen
5.	$\forall x \forall y(x = y \to y = x)$	4 Gen
6.	$\forall x \forall y(x = y \to y = x) \to \forall y(t = y \to y = t)$	A4
7.	$\forall y(t = y \to y = t)$	6,5 MP
8.	$\forall y(t = y \to y = t) \to (t = s \to s = t)$	A4
9.	$t = s \to s = t$	8,7 MP

In (1), $x = x$ is (an abbreviation of an expression) of the form $=xx$, and $y = x$ is the same but with the first instance of x replaced by y. Thus (1) is an instance of A8. At line (3) we have symmetry expressed at the level of variables. Then the task is just to convert from variables to terms as before. (8) is sure to be an instance of A4 insofar as there is no quantifier in the consequent. For (6), to meet the restriction on A4, we require that y is not a variable in t—if y does appear in t, just uniformly replace y in this derivation with a different variable.

T3.35. $\vdash_{AD} r = s \to (s = t \to r = t)$ *transitivity of equality*

Hint: Start with $y = x \to (y = z \to x = z)$ as an instance of A8—being sure that you see how it *is* an instance of A8. Then you can use T3.34 to get $x = y \to (y = z \to x = z)$, and all you have to do is convert from variables to terms as above.

T3.36. $r = s,\ s = t \vdash_{AD} r = t$

Hint: This is a mere recasting of T3.35 and follows directly from it.

T3.37. $\vdash_{AD} t_i = s \to h^n t_1 \ldots t_i \ldots t_n = h^n t_1 \ldots s \ldots t_n$

Hint: For any given instance of this theorem, you can start with $x_i = y \to h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n$ as an instance of A7. Then it is easy to convert $x_1 \ldots x_n$ to $t_1 \ldots t_n$, and y to s.

T3.38. $\vdash_{AD} t_i = s \to (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \to \mathcal{R}^n t_1 \ldots s \ldots t_n)$

Hint: As for T3.37, for any given instance of this theorem, you can start with $x_i = y \to (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \to \mathcal{R}^n x_1 \ldots y \ldots x_n)$ as an instance of A8. Then it is easy to convert $x_1 \ldots x_n$ to $t_1 \ldots t_n$, and y to s.

We will see further examples of *AD* derivations and especially the equality axioms in the context of the extended application in the next section.

E3.10. Provide demonstrations for T3.35 and T3.36.

E3.11. Provide demonstrations for the following instances of T3.37 and T3.38. Then, in each case, say in words how you would go about showing the results for an arbitrary number of places.

a. $\vdash_{AD} f^1x = g^2xy \to h^3zf^1xf^1z = h^3zg^2xyf^1z$

*b. $\vdash_{AD} (s = t) \to (Ars \to Art)$

AD Quick Reference

AD A1. $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

A2. $(\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

A3. $(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$

A4. $\forall x\mathcal{P} \to \mathcal{P}^x_t$ where t is free for x in $\mathcal{P}$

A5. $\forall x(\mathcal{P} \to \mathcal{Q}) \to (\mathcal{P} \to \forall x\mathcal{Q})$ where x is not free in $\mathcal{P}$

A6. $(x = x)$

A7. $(x_i = y) \to (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$

A8. $(x_i = y) \to (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \to \mathcal{R}^n x_1 \ldots y \ldots x_n)$

MP. $\mathcal{Q}$ follows from $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$

Gen. $\forall x\mathcal{P}$ follows from $\mathcal{P}$

(abv) allows movement between an expression and its abbreviated forms. Then there are all the theorems listed in the ADs guide and,

T3.28	$\vdash_{AD} \forall x\mathcal{A} \to \forall v\mathcal{A}^x_v$	where v is not free in $\forall x\mathcal{A}$ but is free for x in $\mathcal{A}$
T3.29	$\mathcal{A} \to \mathcal{B} \vdash_{AD} \mathcal{A} \to \forall x\mathcal{B}$	where x is not free in $\mathcal{A}$
T3.30	$\vdash_{AD} \mathcal{A}^x_t \to \exists x\mathcal{A}$	where t is free for x in $\mathcal{A}$
T3.31	$\vdash_{AD} \forall x(\mathcal{A} \to \mathcal{B}) \to (\exists x\mathcal{A} \to \mathcal{B})$	where x is not free in $\mathcal{B}$
T3.32	$\mathcal{A} \to \mathcal{B} \vdash_{AD} \exists x\mathcal{A} \to \mathcal{B}$	where x is not free in $\mathcal{B}$
T3.33	$\vdash_{AD} t = t$	
T3.34	$\vdash_{AD} t = s \to s = t$	
T3.35	$\vdash_{AD} r = s \to (s = t \to r = t)$	
T3.36	$r = s, s = t \vdash_{AD} r = t$	
T3.37	$\vdash_{AD} t_i = s \to h^n t_1 \ldots t_i \ldots t_n = h^n t_1 \ldots s \ldots t_n$	
T3.38	$\vdash_{AD} t_i = s \to (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \to \mathcal{R}^n t_1 \ldots s \ldots t_n)$	

3.4 Application: PA

We turn now to a substantive application with which we shall be much concerned in Part IV. If you have postponed this chapter to after Chapter 6, then you have already encountered Peano Arithmetic. However, we may develop consequences of the Peano axioms directly in *AD*. For this, $\mathscr{L}_{\text{NT}}$ is a language like $\mathscr{L}^{<}_{\text{NT}}$ introduced from section 2.3.5 but without $<$. There is the constant symbol $\emptyset$, the one-place function symbol S, two-place function symbols $+$, and $\times$, and the relation symbol $=$. Variables are any of $a \dots z$ with or without positive integer subscripts. Let $\mathscr{s} \leq \mathscr{t}$ abbreviate $\exists u(u + \mathscr{s} = \mathscr{t})$ and $\mathscr{s} < \mathscr{t}$ abbreviate $\exists u(Su + \mathscr{s} = \mathscr{t})$ where u is some variable not in $\mathscr{s}$ or $\mathscr{t}$. For all this, see the language of arithmetic reference (page 301).

We will say that a formula $\mathscr{P}$ is an *AD theorem of Peano Arithmetic* just in case $\mathscr{P}$ follows in *AD* given as premises the following axioms for Peano Arithmetic:

PA
1. $\sim(Sx = \emptyset)$
2. $(Sx = Sy) \rightarrow (x = y)$
3. $(x + \emptyset) = x$
4. $(x + Sy) = S(x + y)$
5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $[\mathscr{P}^{x}_{\emptyset} \wedge \forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx})] \rightarrow \forall x\mathscr{P}$

In the ordinary case we suppress mention of PA1–PA7 as premises, and simply write PA $\vdash_{AD} \mathscr{P}$ to indicate that $\mathscr{P}$ is an *AD* theorem of Peano arithmetic—that there is an *AD* derivation of $\mathscr{P}$ which may include appeal to any of PA1–PA7. As described in Chapter 6, these axioms set up basic arithmetic on the non-negative integers. However, insofar as we are working derivations without reference to meaning and truth, we do not need to think about that for now.

PA7 represents the *principle of mathematical induction*. While PA1–PA6 are particular formulas, like A1–A8 of *AD*, PA7 is an *axiom schema* insofar as indefinitely many formulas might be of that form. Sometimes it is convenient to have the principle of mathematical induction in rule form.

T3.39. In an *AD* derivation from the axioms of PA, $\forall x\mathscr{P}$ follows from $\mathscr{P}^{x}_{\emptyset}$ and $\forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx})$. A derived rule, Ind.

1.	$\mathscr{P}^{x}_{\emptyset}$	prem
2.	$\forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx})$	prem
3.	$[\mathscr{P}^{x}_{\emptyset} \wedge \forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx})] \rightarrow \forall x\mathscr{P}$	PA7
4.	$\mathscr{P}^{x}_{\emptyset} \rightarrow [\forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx}) \rightarrow \forall x\mathscr{P}]$	3 T3.23
5.	$\forall x(\mathscr{P} \rightarrow \mathscr{P}^{x}_{Sx}) \rightarrow \forall x\mathscr{P}$	4,1 MP
6.	$\forall x\mathscr{P}$	5,2 MP

So if we encounter $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ in an *AD* derivation from the axioms of PA, we can safely move to the conclusion that $\forall x\mathcal{P}$ by this derived rule Ind.

We will have much more to say about the principle of mathematical induction in Part II. For now, it is enough to *recognize* its instances. Thus, for example, if $\mathcal{P}$ is $\sim(x = Sx)$, the corresponding instance of PA7 would be,

(O) $[\sim(\emptyset = S\emptyset) \wedge \forall x(\sim(x = Sx) \to \sim(Sx = SSx))] \to \forall x\sim(x = Sx)$

There is the formula with $\emptyset$ substituted for x, the formula itself, and the formula with Sx substituted for x. If the entire antecedent is satisfied, then the formula holds for *every* x. For the corresponding application of Ind (T3.39) you would need $\sim(\emptyset = S\emptyset)$ and $\forall x[\sim(x = Sx) \to \sim(Sx = SSx)]$ in order to move to the conclusion that $\forall x\sim(x = Sx)$. You should track these examples through. The principle of mathematical induction turns out to be essential for deriving many general results.

As before, if a theorem is derived from some premises, we use the theorem in derivations that follow. Thus we build toward increasingly complex results. As you work through these problems you may find the AD Peano reference on page 92 helpful. Let us start with some simple generalizations of the axioms for application to arbitrary terms. With a slight complication, the derivations all follow the Gen / A4 / MP pattern we have seen before. The first is trivial.

T3.40. PA $\vdash_{AD} \sim(St = \emptyset)$

1.	$\sim(Sx = \emptyset)$	PA1
2.	$\forall x\sim(Sx = \emptyset)$	1 Gen
3.	$\forall x\sim(Sx = \emptyset) \to \sim(St = \emptyset)$	A4
4.	$\sim(St = \emptyset)$	3,2 MP

As usual, because there is no quantifier in the consequent, (3) is sure to satisfy the constraint on A4, no matter what t may be. For the next theorem, let u be a variable not in t.

T3.41. PA $\vdash_{AD} (St = Ss) \to (t = s)$

1.	$Sx = Sy \to x = y$	PA2
2.	$\forall y(Sx = Sy \to x = y)$	1 Gen
3.	$\forall y(Sx = Sy \to x = y) \to (Sx = Su \to x = u)$	A4
4.	$Sx = Su \to x = u$	3,2 MP
5.	$\forall u(Sx = Su \to x = u)$	4 Gen
6.	$\forall x\forall u(Sx = Su \to x = u)$	5 Gen
7.	$\forall x\forall u(Sx = Su \to x = u) \to \forall u(St = Su \to t = u)$	A4
8.	$\forall u(St = Su \to t = u)$	7,6 MP
9.	$\forall u(St = Su \to t = u) \to (St = Ss \to t = s)$	A4
10.	$St = Ss \to t = s$	9,8 MP

Since u is not a variable in t, (7) meets the constraint on A4. PA1 – PA6 are stated in terms of the particular variables x and y. We cannot be sure that y is not a variable in t. However, t has at most finitely many variables. So we can be sure that there is *some* variable not in t. And the derivation goes through once we have switched y for it.

T3.42. PA $\vdash_{AD}$ $(t + \emptyset) = t$

corollary: PA $\vdash_{AD}$ $t = (t + \emptyset)$

*T3.43. PA $\vdash_{AD}$ $(t + S\mathcal{s}) = S(t + \mathcal{s})$

corollary: PA $\vdash_{AD}$ $S(t + \mathcal{s}) = (t + S\mathcal{s})$

T3.44. PA $\vdash_{AD}$ $(t \times \emptyset) = \emptyset$

corollary: PA $\vdash_{AD}$ $\emptyset = (t \times \emptyset)$

T3.45. PA $\vdash_{AD}$ $(t \times S\mathcal{s}) = [(t \times \mathcal{s}) + t]$

corollary: PA $\vdash_{AD}$ $[(t \times \mathcal{s}) + t] = (t \times S\mathcal{s})$

In each case, the corollary is immediate from the theorem with T3.34 and MP. We will not usually distinguish these theorems from their corollaries. And, in general, for any theorem $\mathcal{s} = t$, we will generally assume the corollary $t = \mathcal{s}$. Notice that t and $\mathcal{s}$ in these theorems may be *any* terms. Thus,

(P) $\quad x + \emptyset = x \qquad (x \times y) + \emptyset = x \times y \qquad (\emptyset + x) + \emptyset = \emptyset + x$

are all straightforward instances of T3.42.

Given this much, we are ready for a series of results which are much more interesting—for example, some general principles of commutativity and associativity. For a first application of Ind, let $\mathcal{P}$ be $(\emptyset + x) = x$; then $\mathcal{P}^x_\emptyset$ is $(\emptyset + \emptyset) = \emptyset$ and $\mathcal{P}^x_{Sx}$ is $(\emptyset + Sx) = Sx$.

T3.46. PA $\vdash_{AD}$ $(\emptyset + t) = t$

1.	$(\emptyset + \emptyset) = \emptyset$	T3.42
2.	$[(\emptyset + x) = x] \to [S(\emptyset + x) = Sx]$	T3.37
3.	$S(\emptyset + x) = (\emptyset + Sx)$	T3.43
4.	$[S(\emptyset + x) = (\emptyset + Sx)] \to [S(\emptyset + x) = Sx \to (\emptyset + Sx) = Sx]$	T3.38
5.	$S(\emptyset + x) = Sx \to (\emptyset + Sx) = Sx$	4,3 MP
6.	$[(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx]$	2,5 T3.2
7.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	6 Gen
8.	$\forall x[(\emptyset + x) = x]$	1,7 Ind
9.	$\forall x[(\emptyset + x) = x] \to [(\emptyset + t) = t]$	A4
10.	$(\emptyset + t) = t$	9,8 MP

The key to this derivation, and others like it, is bringing Ind into play. The basic strategy for the beginning and end of these arguments is always the same. In this case,

1.	$(\emptyset + \emptyset) = \emptyset$	T3.42
$\vdots$		
6.	$[(\emptyset + x) = x] \rightarrow [(\emptyset + Sx) = Sx]$	
7.	$\forall x([(\emptyset + x) = x] \rightarrow [(\emptyset + Sx) = Sx])$	6 Gen
8.	$\forall x[(\emptyset + x) = x]$	1,7 Ind
9.	$\forall x[(\emptyset + x) = x] \rightarrow [(\emptyset + t) = t]$	A4
10.	$(\emptyset + t) = t$	9,8 MP

The goal is automatic by A4 and MP once you have $\forall x[(\emptyset + x) = x]$ by Ind at (8). For this, you need $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$. We have $\mathcal{P}^x_\emptyset$ at (1) as an instance of T3.42—and $\mathcal{P}^x_\emptyset$ is almost always easy to get. $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$ is automatic by Gen from (6). So the real work is getting (6). Thus, once you see what is going on, the entire derivation for T3.46 boils down to lines (2)–(6). For this, begin by noticing that the antecedent of what we want is like the antecedent of (2), and the consequent like what we want but for the equivalence in (3). We use T3.38 to switch the one term for the equivalent one we want. The applications of T3.37 and then T3.38 in this theorem are typical.

T3.47. PA $\vdash_{AD} (St + \emptyset) = S(t + \emptyset)$

1.	$(St + \emptyset) = St$	T3.42
2.	$t = (t + \emptyset)$	T3.42
3.	$[t = (t + \emptyset)] \rightarrow [St = S(t + \emptyset)]$	T3.37
4.	$St = S(t + \emptyset)$	3,2 MP
5.	$(St + \emptyset) = S(t + \emptyset)$	1,4 T3.36

In this derivation, both (1) and (2) are instances of T3.42—where the instance on (1) has St for t, and (2) is in the "reversed" corollary form. Then the key to the derivation is that the left side of (1) is like what we want, and the right side of (1) is like what we want but for the equality on (2). The goal then is to use T3.37 to switch the one term for the equivalent one. This result forms the "zero-case" for the one that follows.

T3.48. PA $\vdash_{AD} (St + s) = S(t + s)$

See the derivation in the upper box on page 90.

The idea behind this longish derivation is to bring Ind into play, where formula $\mathcal{P}$ is $(St + x) = S(t + x)$. For now, do not worry about how we identified this formula as $\mathcal{P}$. Given that much, the following setup is automatic:

1.	$(St + \emptyset) = S(t + \emptyset)$	T3.47
$\vdots$		
12.	$[(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)]$	
13.	$\forall x([(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)])$	12 Gen
14.	$\forall x[(St + x) = S(t + x)]$	1,13 Ind
15.	$\forall x[(St + x) = S(t + x)] \rightarrow [(St + s) = S(t + s)]$	A4
16.	$(St + s) = S(t + s)$	15,14 MP

We have the zero-case from T3.47 on (1); the goal is automatic once we have the result on (12). For (12), the antecedent at (2) is what we want, and the consequent is right but for the equivalences on (3) and (9). We use T3.38 to substitute terms into the consequent. The equivalence on (3) is a straightforward instance of T3.43. We had to work (just a bit) starting again with T3.43 to get the equivalence on (9).

T3.49. PA $\vdash_{AD}$ $t + s = s + t$ *commutativity of addition*

See the derivation in the lower box on the following page.

The pattern of this derivation is very much like ones we have seen before. Where $\mathscr{P}$ is $t + x = x + t$ we have the zero-case at (3), and the derivation effectively reduces to getting (12). We get this by substituting into the consequent of (4) by means of the equivalences on (5) and (9).

T3.50. PA $\vdash_{AD}$ $(r + s) + \emptyset = r + (s + \emptyset)$

Hint: Begin with $(r + s) + \emptyset = r + s$ as an instance of T3.42. The derivation is then a matter of using T3.42 to replace s in the right-hand side with $s + \emptyset$.

*T3.51. PA $\vdash_{AD}$ $(r + s) + t = r + (s + t)$ *associativity of addition*

Hint: For an application of Ind, let $\mathscr{P}$ be $(r + s) + x = r + (s + x)$. Start with $[(r + s) + x = r + (s + x)] \rightarrow [S((r + s) + x) = S(r + (s + x))]$ as an instance of T3.37, and substitute into the consequent as necessary by T3.43 to reach $[(r + s) + x = r + (s + x)] \rightarrow [(r + s) + Sx = r + (s + Sx)]$. The derivation is longish, but straightforward.

T3.52. PA $\vdash_{AD}$ $\emptyset \times t = \emptyset$

Hint: For an application of Ind, let $\mathscr{P}$ be $\emptyset \times x = \emptyset$; then the derivation reduces to showing $[\emptyset \times x = \emptyset] \rightarrow [\emptyset \times Sx = \emptyset]$. This is easy enough if you use T3.42 and T3.45 to show that $\emptyset \times x = \emptyset \times Sx$.

T3.53. PA $\vdash_{AD}$ $St \times \emptyset = (t \times \emptyset) + \emptyset$

Hint: This does not require application of Ind.

T3.48

1.	$(St + \emptyset) = S(t + \emptyset)$	T3.47
2.	$[(St + x) = S(t + x)] \rightarrow [S(St + x) = SS(t + x)]$	T3.37
3.	$S(St + x) = (St + Sx)$	T3.43
4.	$[S(St + x) = (St + Sx)] \rightarrow$	
	$([S(St + x) = SS(t + x)] \rightarrow [(St + Sx) = SS(t + x)])$	T3.38
5.	$[S(St + x) = SS(t + x)] \rightarrow [(St + Sx) = SS(t + x)]$	4,3 MP
6.	$[(St + x) = S(t + x)] \rightarrow [(St + Sx) = SS(t + x)]$	2,5 T3.2
7.	$S(t + x) = (t + Sx)$	T3.43
8.	$[S(t + x) = (t + Sx)] \rightarrow [SS(t + x) = S(t + Sx)]$	T3.37
9.	$SS(t + x) = S(t + Sx)$	8,7 MP
10.	$[SS(t + x) = S(t + Sx)] \rightarrow$	
	$([(St + Sx) = SS(t + x)] \rightarrow [(St + Sx) = S(t + Sx)])$	T3.38
11.	$[(St + Sx) = SS(t + x)] \rightarrow [(St + Sx) = S(t + Sx)]$	10,9 MP
12.	$[(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)]$	6,11 T3.2
13.	$\forall x([(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)])$	12 Gen
14.	$\forall x[(St + x) = S(t + x)]$	1,13 Ind
15.	$\forall x[(St + x) = S(t + x)] \rightarrow [(St + s) = S(t + s)]$	A4
16.	$(St + s) = S(t + s)$	15,14 MP

T3.49

1.	$t + \emptyset = t$	T3.42
2.	$t = \emptyset + t$	T3.46
3.	$t + \emptyset = \emptyset + t$	1,2 T3.36
4.	$[t + x = x + t] \rightarrow [S(t + x) = S(x + t)]$	T3.37
5.	$S(t + x) = (t + Sx)$	T3.43
6.	$[S(t + x) = (t + Sx)] \rightarrow$	
	$([S(t + x) = S(x + t)] \rightarrow [(t + Sx) = S(x + t)])$	T3.38
7.	$[S(t + x) = S(x + t)] \rightarrow [(t + Sx) = S(x + t)]$	6,5 MP
8.	$[t + x = x + t] \rightarrow [(t + Sx) = S(x + t)]$	4,7 T3.2
9.	$S(x + t) = (Sx + t)$	T3.48
10.	$[S(x + t) = (Sx + t)] \rightarrow$	
	$([t + Sx = S(x + t)] \rightarrow [t + Sx = Sx + t])$	T3.38
11.	$[t + Sx = S(x + t)] \rightarrow [t + Sx = Sx + t]$	10,9 MP
12.	$[t + x = x + t] \rightarrow [t + Sx = Sx + t]$	8,11 T3.2
13.	$\forall x([t + x = x + t] \rightarrow [t + Sx = Sx + t])$	12 Gen
14.	$\forall x[t + x = x + t]$	3,13 Ind
15.	$\forall x[t + x = x + t] \rightarrow [t + s = s + t]$	A4
16.	$t + s = s + t$	15,14 MP

T3.54. PA $\vdash_{AD} (t \times x) + (x + St) = (t \times Sx) + Sx$

Hint: Set $x + St = t + Sx$ as a prelininary goal. This does not require application of Ind.

*T3.55. PA $\vdash_{AD} St \times s = (t \times s) + s$

Hint: For an application of Ind, let $\mathscr{P}$ be $St \times x = (t \times x) + x$. The derivation reduces to getting $[St \times x = (t \times x) + x] \rightarrow [St \times Sx = (t \times Sx) + Sx]$. For this, you can start with $[St \times x = (t \times x) + x] \rightarrow [(St \times x) + St = ((t \times x) + x) + St]$ as an instance of T3.37, and substitute into the consequent.

T3.56. PA $\vdash_{AD} t \times s = s \times t$ *commutativity of multiplication*

Hint: For an application of Ind, let $\mathscr{P}$ be $t \times x = x \times t$. You can start with $[t \times x = x \times t] \rightarrow [(t \times x) + t = (x \times t) + t]$ as an instance of T3.37, and substitute into the consequent.

We will stop here. With the derivation system *ND* of Chapter 6, we obtain all these results and more. But that system is easier to manipulate than what we have so far in *AD*. Still, we have obtained some significant results! Perhaps you have heard from your mother's knee that $a + b = b + a$. But this is a sweeping general claim of the sort that cannot ever have all its instances checked. We have derived it from the Peano axioms.

*E3.12. Provide derivations to show each of T3.42–T3.45, and T3.50–T3.56.

E3.13. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Term t being *free for* variable x in formula $\mathcal{A}$ along with the restrictions on A4 and A5.

b. An *AD* theorem of Peano arithmetic.

Peano Arithmetic (*AD*)

PA
1. $\sim(Sx = \emptyset)$
2. $(Sx = Sy) \rightarrow (x = y)$
3. $(x + \emptyset) = x$
4. $(x + Sy) = S(x + y)$
5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})] \rightarrow \forall x\mathcal{P}$

T3.39 In an *AD* derivation from the axioms of PA, $\forall x\mathcal{P}$ follows from $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$. (Ind)

T3.40 PA $\vdash_{AD} \sim(St = \emptyset)$

T3.41 PA $\vdash_{AD} (St = Ss) \rightarrow (t = s)$

T3.42 PA $\vdash_{AD} (t + \emptyset) = t$

T3.43 PA $\vdash_{AD} (t + Ss) = S(t + s)$

T3.44 PA $\vdash_{AD} (t \times \emptyset) = \emptyset$

T3.45 PA $\vdash_{AD} (t \times Ss) = [(t \times s) + t]$

T3.46 PA $\vdash_{AD} (\emptyset + t) = t$

T3.47 PA $\vdash_{AD} (St + \emptyset) = S(t + \emptyset)$

T3.48 PA $\vdash_{AD} (St + s) = S(t + s)$

T3.49 PA $\vdash_{AD} t + s = s + t$ *commutativity of addition*

T3.50 PA $\vdash_{AD} (r + s) + \emptyset = r + (s + \emptyset)$

T3.51 PA $\vdash_{AD} (r + s) + t = r + (s + t)$ *associativity of addition*

T3.52 PA $\vdash_{AD} \emptyset \times t = \emptyset$

T3.53 PA $\vdash_{AD} St \times \emptyset = (t \times \emptyset) + \emptyset$

T3.54 PA $\vdash_{AD} (t \times x) + (x + St) = (t \times Sx) + Sx$

T3.55 PA $\vdash_{AD} St \times s = (t \times s) + s$

T3.56 PA $\vdash_{AD} t \times s = s \times t$ *commutativity of multiplication*

Any theorem $t = s$ has corollary $s = t$.

Chapter 4

Semantics

Having introduced the grammar for our formal languages and even (if you did not skip the last chapter) done derivations in them, we need to say something about *semantics*—about the conditions under which their expressions are true and false. In addition to *logical validity* from Chapter 1 and *validity in AD* from Chapter 3, this will lead to a third, *semantic* notion of validity. Again, the discussion divides into the relatively simple sentential case (section 4.1), and then the full quantificational version (section 4.2). Recall that we are introducing formal languages in their "pure" form, apart from associations with ordinary language. Having discussed, in this chapter, conditions under which formal expressions are true and not, in the next chapter, we will finally turn to translation, and so to ways formal expressions are associated with ordinary ones.

4.1 Sentential

For any sentential or quantificational language, starting with a sentence and working up its tree, let us say that its *basic* sentences are the first sentences that do not have an operator from the sentential language ($\sim$, $\rightarrow$, $\vee$, $\wedge$, $\leftrightarrow$) as main operator. For a sentential language, basic sentences are the sentence letters, as the atomics are the first and only sentences without a main operator from the sentential language. In the quantificational case, basic sentences may be more complex.[1] In this section, we treat basic sentences as atomic. Our initial focus is on forms with just operators $\sim$ and $\rightarrow$. We begin with an account of the conditions under which sentences are true and not true, learn to apply that account in arbitrary conditions, and turn to validity. The section concludes with applications to our abbreviations, $\wedge$, $\vee$, and $\leftrightarrow$.

[1] Thus the basic sentences of $A \wedge B$ are just the atomic subformulas A and B. However $Fa \wedge \exists x Gx$ has atomic subformulas Fa and Gx, but basic sentences Fa and $\exists x Gx$ since the latter does not have an operator from the sentential language as its main operator.

4.1.1 Interpretations and Truth

Sentences are true and false relative to an *interpretation* of basic sentences. In the sentential case, the notion of an interpretation is particularly simple. For any formal language $\mathcal{L}$, a *sentential interpretation* assigns a truth value *true* or *false*, T or F, to each of its basic sentences. Thus, for $\mathcal{L}_s$ we might have interpretations I and J,

(A)

	A	B	C	D	E	F	G	H	
I	T	T	T	T	T	T	T	T	...

	A	B	C	D	E	F	G	H	
J	T	T	F	F	T	T	F	F	...

These assignments may be made in arbitrary ways. *Any* assignment of truth values to the basic sentences counts as a sentential interpretation. When a sentence $\mathcal{A}$ is T on an interpretation I, we write I[$\mathcal{A}$] = T, and when it is F, we write, I[$\mathcal{A}$] = F. Thus, in the above case, J[B] = T and J[C] = F.

Truth for complex sentences depends on truth and falsity for their parts. In particular, for any interpretation I,

ST (∼) For any sentence $\mathcal{P}$, I[$\sim\mathcal{P}$] = T iff I[$\mathcal{P}$] = F; otherwise I[$\sim\mathcal{P}$] = F.

(→) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, I[$(\mathcal{P} \rightarrow \mathcal{Q})$] = T iff I[$\mathcal{P}$] = F or I[$\mathcal{Q}$] = T (or both); otherwise I[$(\mathcal{P} \rightarrow \mathcal{Q})$] = F.

Thus a basic sentence is true or false depending on the interpretation. For complex sentences, $\sim\mathcal{P}$ is true iff $\mathcal{P}$ is not true; and $(\mathcal{P} \rightarrow \mathcal{Q})$ is true iff $\mathcal{P}$ is not true or $\mathcal{Q}$ is. It is traditional to represent the information from ST(∼) and ST(→) in the following *truth tables:*

T(∼)

$\mathcal{P}$	$\sim\mathcal{P}$
T	***F***
F	***T***

T(→)

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \rightarrow \mathcal{Q}$
T	T	***T***
T	F	***F***
F	T	***T***
F	F	***T***

From ST(∼), we have that if $\mathcal{P}$ is F then $\sim\mathcal{P}$ is T; and if $\mathcal{P}$ is T then $\sim\mathcal{P}$ is F. This is just the way to read table T(∼) from left to right in the bottom row, and then the top row. Similarly, from ST(→), we have that $\mathcal{P} \rightarrow \mathcal{Q}$ is T in conditions represented by the first, third, and fourth rows of T(→). The only way for $\mathcal{P} \rightarrow \mathcal{Q}$ to be F is when $\mathcal{P}$ is T and $\mathcal{Q}$ is F as in the second row.

ST works recursively. Whether a basic sentence is true comes directly from the interpretation; truth for other sentences depends on truth for their immediate subformulas—and can be read directly off the tables. As usual, we can use trees to see how it works. As we build a formula from its parts to the whole, so now we calculate truth from parts to the whole. Suppose I[A] = T, I[B] = F, and I[C] = F. Then I[$\sim(A \rightarrow \sim B) \rightarrow C$] = T.

(B)

$A^{(\mathsf{T})}$	$B^{(\mathsf{F})}$	$C^{(\mathsf{F})}$	From I
	$\sim B^{(\mathsf{T})}$		By T($\sim$), row 2
$(A \to \sim B)^{(\mathsf{T})}$			By T($\to$), row 1
$\sim(A \to \sim B)^{(\mathsf{F})}$			By T($\sim$), row 1
$\sim(A \to \sim B) \to C^{(\mathsf{T})}$			By T($\to$), row 4

The basic tree is the same as the one that shows $\sim(A \to \sim B) \to C$ is a formula. From the interpretation, A is T, B is F, and C is F. These are across the top. Since B is F, from the bottom row of table T($\sim$), $\sim B$ is T. Since A is T and $\sim B$ is T, reading across the top row of the table T($\to$), $A \to \sim B$ is T. And similarly, according to the tree, for the rest. You should carefully follow each step.

Here is the same formula considered on another interpretation. With the interpretation J on the previous page, J[$\sim(A \to \sim B) \to C$] = F.

(C)

$A^{(\mathsf{T})}$	$B^{(\mathsf{T})}$	$C^{(\mathsf{F})}$	From J
	$\sim B^{(\mathsf{F})}$		By T($\sim$), row 1
$(A \to \sim B)^{(\mathsf{F})}$			By T($\to$), row 2
$\sim(A \to \sim B)^{(\mathsf{T})}$			By T($\sim$), row 2
$\sim(A \to \sim B) \to C^{(\mathsf{F})}$			By T($\to$), row 2

This time, for both applications of ST($\to$), the antecedent is T and the consequent is F; thus we are working on the second row of table T($\to$), and the conditionals evaluate to F. Again, you should follow each step in the tree.

E4.1. Where the interpretation is J on the preceding page, with J[A] = T, J[B] = T and J[C] = F, use trees to decide whether the following sentences of $\mathcal{L}_s$ are T or F.

*a. $\sim A$

b. $\sim\sim C$

c. $A \rightarrow C$

d. $C \rightarrow A$

*e. $\sim(A \rightarrow A)$

*f. $(\sim A \rightarrow A)$

g. $\sim(A \rightarrow \sim C) \rightarrow C$

h. $(\sim A \rightarrow C) \rightarrow C$

*i. $(A \rightarrow \sim B) \rightarrow \sim(B \rightarrow \sim A)$

j. $\sim(B \rightarrow \sim A) \rightarrow (A \rightarrow \sim B)$

4.1.2 Arbitrary Interpretations

Sentences are true and false relative to an interpretation. But whether an argument is *semantically valid* depends on truth and falsity relative to *every* interpretation. As a first step toward determining semantic validity, in this section, we generalize the method of the last section to calculate truth values relative to arbitrary interpretations.

First, any sentence has a *finite* number of basic sentences as components. It is thus possible simply to *list* all the possible interpretations of those basic sentences. If an expression has just one basic sentence $\mathscr{A}$, then on any interpretation whatsoever, that basic sentence must be T or F.

(D)

$\mathscr{A}$
T
F

If an expression has basic sentences $\mathscr{A}$ and $\mathscr{B}$, then the possible interpretations of its basic sentences are,

(E)

$\mathscr{A}$	$\mathscr{B}$
T	T
T	F
F	T
F	F

$\mathscr{B}$ can take its possible values, T and F when $\mathscr{A}$ is true, and $\mathscr{B}$ can take its possible values, T and F when $\mathscr{A}$ is false. And similarly, every time we add a basic sentence, we double the number of possible interpretations, so that n basic sentences always have 2^n possible interpretations. Thus the possible interpretations for three and four basic sentences are,

(F)

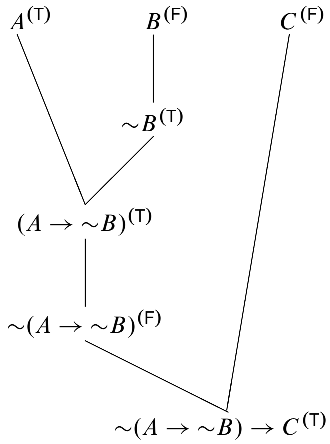
$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

(G)

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$
T	T	T	T
T	T	T	F
T	T	F	T
T	T	F	F
T	F	T	T
T	F	T	F
T	F	F	T
T	F	F	F
F	T	T	T
F	T	T	F
F	T	F	T
F	T	F	F
F	F	T	T
F	F	T	F
F	F	F	T
F	F	F	F

Extra horizontal lines are added purely for visual convenience. There are $8 = 2^3$ combinations with three basic sentences and $16 = 2^4$ combinations with four. In general, to write down all the possible combinations for n basic sentences, begin by finding the total number $r = 2^n$ of combinations or rows. Then write down a column with half that many $(r/2)$ Ts and half that many $(r/2)$ Fs; then a column alternating half again as many $(r/4)$ Ts and Fs; and a column alternating half again as many $(r/8)$ Ts and Fs—continuing to the n^{th} column alternating groups of just one T and one F. Thus, for example, with four basic sentences, $r = 2^4 = 16$; so we begin with a column consisting of $r/2 = 8$ Ts and $r/2 = 8$ Fs; this is followed by a column alternating groups of 4 Ts and 4 Fs, a column alternating groups of 2 Ts and 2 Fs, and a column alternating groups of 1 T and 1 F. The result lists all the possible interpretations of the basic sentences. And similarly in other cases.

Given an expression involving, say, four basic sentences, we could imagine doing trees for each of the 16 possible interpretations. But, to exhibit truth values for each of the possible interpretations, we can reduce the amount of work a bit—or at least represent it in a relatively compact form. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$, and $\mathsf{I}[C] = \mathsf{F}$, and consider the tree from (B) above, along with a "compressed" version of the same information.

(H) 

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	F	F	F	T	T	T	F	***T***	F

In the table on the right, we begin by simply listing the interpretation we will consider in its left-hand part: A is T, B is F, and C is F. Then, under each basic sentence we put its truth value, and for a non-basic sentence place its truth value *under its main operator*. Notice that the calculation must proceed *precisely* as it does in the tree. It is because B is F, that we put T under the second $\sim$. It is because A is T and $\sim B$ is T that we put a T under the first $\to$. It is because $(A \to \sim B)$ is T that we put F under the first $\sim$. And it is because $\sim(A \to \sim B)$ is F and C is F that we put a T under the second $\to$. In effect, then, we work "down" through the tree, only in this compressed form. Or we might think of truth values from the tree as "squished" up into the one row. Because there is a T under its main operator, we conclude that the whole formula, $\sim(A \to \sim B) \to C$ is T when $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$, and $\mathsf{I}[C] = \mathsf{F}$. In this way, we might conveniently calculate and represent the truth value of $\sim(A \to \sim B) \to C$ for all eight of the possible interpretations of its basic sentences.

(I)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T	T	T	F	F	T	***T***	T
T	T	F	T	T	F	F	T	***F***	F
T	F	T	F	T	T	T	F	***T***	T
T	F	F	F	T	T	T	F	***T***	F
F	T	T	F	F	T	F	T	***T***	T
F	T	F	F	F	T	F	T	***T***	F
F	F	T	F	F	T	T	F	***T***	T
F	F	F	F	F	T	T	F	***T***	F

The emphasized column under the second $\to$ indicates the truth value of $\sim(A \to \sim B) \to C$ for each of the interpretations on the left—which is to say, for every possible interpretation of the three basic sentences. So the only way for $\sim(A \to \sim B) \to C$ to be F is for C to be F, and A and B to be T. Our above tree (H) represents just the fourth row of this table.

In practice, it is easiest to work these *truth tables* "vertically." For this, begin with the basic sentences in some standard order along with all their possible interpretations in the left-hand column. For $\mathcal{L}_s$ let the standard order be alphanumeric $(A, A_1, A_2, \ldots, B, B_1, B_2, \ldots, C, \ldots)$. And repeat truth values for basic sentences under their occurrences in the formula (this is not crucial, since truth values for basic sentences are already listed on the left; it will be up to you whether to repeat values for basic sentences). This is done in table (J) below.

(J)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T		T			T		T
T	T	F		T			T		F
T	F	T		T			F		T
T	F	F		T			F		F
F	T	T		F			T		T
F	T	F		F			T		F
F	F	T		F			F		T
F	F	F		F			F		F

(K)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T		T		F	T		T
T	T	F		T		F	T		F
T	F	T		T		T	F		T
T	F	F		T		T	F		F
F	T	T		F		F	T		T
F	T	F		F		F	T		F
F	F	T		F		T	F		T
F	F	F		F		T	F		F

Now, given the values for B as in (J), we are in a position to calculate the values for $\sim B$; so get the T($\sim$) table in you mind, put your eye on the column under B in the formula (or on the left if you have decided not to repeat the values for B under its occurrence in the formula). Then fill in the column under the second $\sim$, reversing the values from under B. This is accomplished in (K). Given the values for A and $\sim B$, we are now in a position to calculate values for $A \to \sim B$; so get the T($\to$) table in your head, and put your eye on the columns under A and $\sim B$. Then fill in the column under the first $\to$, going with F only when A is T and $\sim B$ is F. This is accomplished in (L).

On alphanumeric order: It is worth asking what happens if basic sentences are listed in some order other than alphanumeric.

A	B
T	T
T	F
F	T
F	F

B	A
T	T
T	F
F	T
F	F

All the combinations are still listed, but their locations in a table change.

Each of the above tables lists all of the combinations for the basic sentences. But the first table has the interpretation I with I[A] = T and I[B] = F in the second row, where the second table has this combination in the third. Similarly, the tables exchange rows for the interpretation J with J[A] = F and J[B] = T. As it turns out, the only real consequence of switching rows is that it becomes difficult to compare tables as, for example, with the *Answers to Selected Exercises*. And it may matter as part of the standard of correctness for exercises!

(L)

A	B	C	$\sim$	$(A$	$\rightarrow$	$\sim$	$B)$	$\rightarrow$	C
T	T	T		T	F	F	T		T
T	T	F		T	F	F	T		F
T	F	T		T	T	T	F		T
T	F	F		T	T	T	F		F
F	T	T		F	T	F	T		T
F	T	F		F	T	F	T		F
F	F	T		F	T	T	F		T
F	F	F		F	T	T	F		F

(M)

A	B	C	$\sim$	$(A$	$\rightarrow$	$\sim$	$B)$	$\rightarrow$	C
T	T	T	T	T	F	F	T		T
T	T	F	T	T	F	F	T		F
T	F	T	F	T	T	T	F		T
T	F	F	F	T	T	T	F		F
F	T	T	F	F	T	F	T		T
F	T	F	F	F	T	F	T		F
F	F	T	F	F	T	T	F		T
F	F	F	F	F	T	T	F		F

Now we are ready to fill in the column under the first $\sim$. So get the T($\sim$) table in your head, and put your eye on the column under the first $\rightarrow$. The column is completed in table (M). And the table is finished as in (I) by completing the column under the last $\rightarrow$, based on the columns under the first $\sim$ and under the C. Notice again that the order in which you work the columns exactly parallels the order from the tree.

As another example, consider these tables for $\sim(B \rightarrow A)$, the first with truth values repeated under basic sentences, the second without.

(N)

A	B	$\sim$	$(B$	$\rightarrow$	$A)$
T	T	***F***	T	T	T
T	F	***F***	F	T	T
F	T	***T***	T	F	F
F	F	***F***	F	T	F

(O)

A	B	$\sim$	$(B$	$\rightarrow$	$A)$
T	T	***F***		T	
T	F	***F***		T	
F	T	***T***		F	
F	F	***F***		T	

We complete the table as before. First, with our eye on the columns under B and A, we fill in the column under $\rightarrow$. Then, with our eye on that column, we complete the one under $\sim$. For this, first, notice that $\sim$ is the *main* operator. You would *not* calculate $\sim B$ and then the arrow! Rather, your calculations move from the smaller parts to the larger; so the arrow comes first and then the tilde. Again, the order is the same as on a tree. Second, if you do not repeat values for basic formulas, be careful about $B \rightarrow A$; the leftmost column of table (O), under A, is the column for the *consequent* and the column immediately to its right, under B, is for the *antecedent;* in this case, then, the second row under arrow is T and the *third* is F. Though it is fine to omit columns under basic sentences, as they are already filled in on the left side, you should *not* skip other columns, as they are essential building blocks for the final result.

E4.2. For each of the following sentences of $\mathcal{L}_s$ construct a truth table to determine its truth value for each of the possible interpretations of its basic sentences.

*a. $\sim\sim A$

b. $\sim(A \rightarrow A)$

c. $(\sim A \rightarrow A)$

*d. $(\sim B \rightarrow A) \rightarrow B$

e. $\sim(B \rightarrow \sim A) \rightarrow B$

f. $(A \rightarrow \sim B) \rightarrow \sim(B \rightarrow \sim A)$

*g. $C \rightarrow (A \rightarrow B)$

h. $[A \rightarrow (C \rightarrow B)] \rightarrow [(A \rightarrow C) \rightarrow (A \rightarrow B)]$

*i. $(\sim A \rightarrow B) \rightarrow (\sim C \rightarrow D)$

j. $\sim(A \rightarrow \sim B) \rightarrow \sim(C \rightarrow \sim D)$

4.1.3 Validity

As we have seen, sentences are true and false relative to an interpretation. For any interpretation, a sentence has some definite value. Now consider an *argument* whose premises and conclusion are some formal sentences. So, for example, perhaps the premises are $A \rightarrow B$ and A and the conclusion is B. A formal argument is *sententially valid* depending on *all* the interpretations of the sentences that are its premises and conclusion. Suppose a formal argument has premises $\mathcal{P}_1 \dots \mathcal{P}_n$ and conclusion $\mathcal{Q}$. Then,

> $\mathcal{P}_1 \dots \mathcal{P}_n$ *sententially entail* $\mathcal{Q}$ ($\mathcal{P}_1 \dots \mathcal{P}_n \vDash_s \mathcal{Q}$) iff there is no sentential interpretation I such that $\mathsf{I}[\mathcal{P}_1] = \mathsf{T}$ and ... and $\mathsf{I}[\mathcal{P}_n] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$.

Premises entail a conclusion when no interpretation makes all the premises true and the conclusion false (or, equivalently, when every interpretation is such that it does not make the premises true and conclusion false). We can put the definition more generally as follows: Suppose Γ (Gamma) is a set of formulas—these are the premises. Say $\mathsf{I}[\Gamma] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ for each $\mathcal{P}$ in Γ. Then,

SV Γ *sententially entails* $\mathcal{Q}$ ($\Gamma \vDash_s \mathcal{Q}$) iff there is no sentential interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$.

Where the members of Γ are $\mathcal{P}_1 \dots \mathcal{P}_n$, this says the same as before. Γ sententially entails $\mathcal{Q}$ when there is no sentential interpretation that makes each member of Γ true and $\mathcal{Q}$ false. Γ does not sententially entail $\mathcal{Q}$ ($\Gamma \nvDash_s \mathcal{Q}$) when there is some sentential interpretation on which all the members of Γ are true, but $\mathcal{Q}$ is false.[2]

[2]Definition SV allows any collection of premises, and so relaxes the supposition that an argument has *finitely* many premises. However, having made this observation, for the time being we set it to the side: Any ordinary argument has finitely many premises—and methods from this chapter are restricted to the finite case.

Greek Characters

Greek characters frequently appear in logical contexts. In order to read them (as something besides "funny squiggle") unique characters and their names are listed here.

α	alpha	ι	iota	σ, Σ	sigma, Sigma
β	beta	κ	kappa	τ	tau
γ, Γ	gamma, Gamma	λ, Λ	lambda, Lambda	υ, Υ	upsilon, Upsilon
δ, Δ	delta, Delta	μ	mu	ϕ, Φ	phi, Phi
ϵ	epsilon	ν	nu	χ	chi
ζ	zeta	ξ, Ξ	xi, Xi	ψ, Ψ	psi, Psi
η	eta	π, Π	pi, Pi	ω, Ω	omega, Omega
θ, Θ	theta, Theta	ρ	rho		

If Γ sententially entails $\mathcal{Q}$ we say the argument whose premises are the members of Γ and conclusion is $\mathcal{Q}$ is *sententially valid*. To say that an argument is sententially valid and that its premises sententially entail its conclusion is to say the same thing only with a different grammatical subject: an *argument* is sententially valid just in case its *premises* sententially entail the conclusion. We can think of the premises as *constraining* the interpretations that matter: For validity it is just the interpretations where the members of Γ are all true on which the conclusion $\mathcal{Q}$ cannot be false. If Γ has no members then there are no constraints on relevant interpretations, and the conclusion is valid iff it is true on every interpretation. In the case where there are no premises, we simply write $\vDash_s \mathcal{Q}$, and if $\mathcal{Q}$ is valid it is a *tautology*. Notice the new *double turnstile* $\vDash$ for this semantic notion, in contrast to the *single* turnstile $\vdash$ for derivations.

Given that we are already in a position to exhibit truth values for arbitrary interpretations, it is a simple matter to determine whether an argument is sententially valid. Where the premises and conclusion of an argument include basic sentences $\mathcal{B}_1 \ldots \mathcal{B}_n$, begin by calculating the truth values of the premises and conclusion for each of the possible interpretations for $\mathcal{B}_1 \ldots \mathcal{B}_n$. Then *look* to see if any interpretation makes all the premises true but the conclusion false. If no interpretation makes the premises true and the conclusion not, then by SV the argument is sententially valid. If some interpretation does make the premises true and the conclusion false, then it is not valid.

Thus, for example, suppose we want to know whether the following argument is sententially valid.

$$\text{(P)} \quad \begin{array}{l} (\sim A \to B) \to C \\ B \\ \hline C \end{array}$$

By SV, the question is whether there is an interpretation that makes the premises true and the conclusion not. So we begin by calculating the values of the premises and conclusion for each of the possible interpretations of the basic sentences in the premises and conclusion.

A	B	C	$($ $\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	C	B	/ C
T	T	T	F	T	T	T	***T***	T	***T***	***T***
T	T	F	F	T	T	T	***F***	F	***T***	***F***
T	F	T	F	T	T	F	***T***	T	***F***	***T***
T	F	F	F	T	T	F	***F***	F	***F***	***F***
F	T	T	T	F	T	T	***T***	T	***T***	***T***
F	T	F	T	F	T	T	***F***	F	***T***	***F***
F	F	T	T	F	F	F	***T***	T	***F***	***T***
F	F	F	T	F	F	F	***T***	F	***F***	***F***

Now we simply look to see whether any interpretation makes all the premises true but the conclusion not. Interpretations represented by the top row, ones that make A, B, and C all T, do not make the premises true and the conclusion not, because both the premises and the conclusion come out true. In the second row, the conclusion is false, but the first premise is false as well; so not *all* the premises are true *and* the conclusion is false. In the third row, we do not have either all the premises true or the conclusion false. In the fourth row, though the conclusion is false, the premises are not true. In the fifth row, the premises are true, but the conclusion is not false. In the sixth row, the first premise is not true, and in the seventh and eighth rows, the second premise is not true. So no interpretation makes the premises true and the conclusion false. So by SV, $(\sim A \rightarrow B) \rightarrow C$, $B \vDash_s C$. Notice that the only column that matters for a complex formula is the one under its main operator—the one that gives the value of *the sentence* for each of the interpretations; the other columns exist only to support the calculation of the value of the whole.

In contrast, $\sim[(B \rightarrow A) \rightarrow B] \nvDash_s \sim(A \rightarrow B)$. That is, an argument with premise, $\sim[(B \rightarrow A) \rightarrow B]$ and conclusion $\sim(A \rightarrow B)$ is not sententially valid.

(Q)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/ $\sim$	$(A$	$\rightarrow$	$B)$	
T	T	***F***	T	T	T	T	T	***F***	T	T	T	
T	F	***T***	F	T	T	F	F	***T***	T	F	F	
F	T	***F***	T	F	F	T	T	***F***	F	T	T	
F	F	***T***	F	T	F	F	F	***F***	F	T	F	$\Leftarrow$

In the first row, the premise is F. In the second, the conclusion is T. In the third, the premise is F. However, in the last, the premise is T and the conclusion is F. So there are interpretations (any interpretation that makes A and B both F) that make the premise T and the conclusion F. So by SV, $\sim[(B \rightarrow A) \rightarrow B] \nvDash_s \sim(A \rightarrow B)$, and the argument is not sententially valid. All it takes is *one* interpretation that makes all the premises T and the conclusion F to render an argument not sententially valid. Of course, there might be more than one, but one is enough!

As a final example, consider table (I) for $\sim(A \rightarrow \sim B) \rightarrow C$ on page 98 above. From the table, there is an interpretation where the sentence is not true. Thus, by SV, $\nvDash_s \sim(A \rightarrow \sim B) \rightarrow C$. A sentence is valid only when it is true on every interpretation.

Since there is an interpretation on which it is not true, the sentence is not valid (not a tautology).

Since all it takes to demonstrate invalidity is *one* interpretation on which all the premises are true and the conclusion is false, we do not actually need an entire table to demonstrate invalidity. You may decide to produce a whole truth table in order to *find* an interpretation to demonstrate invalidity. But we can sometimes work "backward" from what we are trying to show to an interpretation that does the job. Thus, for example, to find the result from table (Q), we need an interpretation on which the premise is T and the conclusion is F. That is, we need a row like this:

(R)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
		T							***F***			

In order for the premise to be T, the conditional in the brackets must be F. And in order for the conclusion to be F, the conditional must be T. So we can fill in this much:

(S)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
		T				F			***F***		T	

Since there are three ways for an arrow to be T, there is not much to be done with the conclusion. But since the conditional in the premise is F, we know that its antecedent is T and consequent is F. So we have:

(T)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
		T		T		F	F		***F***		T	

That is, $(B \rightarrow A)$ is T and B is F. But now we can fill in the information about B wherever it occurs. The result is as follows:

(U)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
	F	***T***	F	T		F	F		***F***		T	F

Since the first B in the premise is F, the first conditional in the premise is T irrespective of the assignment to A. But, with B false, the only way for the conditional in the argument's conclusion to be T is for A to be false as well. The result is our completed row:

(V)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
F	F	***T***	F	T	F	F	F		***F***	F	T	F

And we have recovered the row that demonstrates invalidity—without doing the entire table. In this case, the full table had only four rows, and we might just as well have done the whole thing. However, when there are many rows, this "shortcut" approach can be attractive. A disadvantage is that sometimes it is not obvious just *how* to proceed. In this example, each stage led to the next. At stage (S), there were three ways to make the conditional subformula in the conclusion true. We were able to proceed insofar as the premise forced the next step. But it might have been that neither the premise nor the conclusion forced a definite next stage. In this sort of case, you might decide to do the whole table, just so that you can can grapple with all the different combinations in an orderly way.

Notice what happens when we try this approach with an argument that is not invalid. Returning to argument (P) above, suppose we try to find a row where the premises are T and the conclusion is F. That is, we set out to find a row like this:

(W)
$$\begin{array}{ccc|cccccccccc}
A & B & C & (\sim & A & \rightarrow & B) & \rightarrow & C & & B & / & C \\ \hline
 & & & & & & & \textbf{T} & & & \textbf{T} & & \textbf{F}
\end{array}$$

Immediately, we are in a position to fill in values for B and C:

(X)
$$\begin{array}{ccc|cccccccccc}
A & B & C & (\sim & A & \rightarrow & B) & \rightarrow & C & & B & / & C \\ \hline
 & \mathsf{T} & \mathsf{F} & & & & \mathsf{T} & \textbf{T} & \mathsf{F} & & \textbf{T} & & \textbf{F}
\end{array}$$

Since the first premise is a true arrow with a false consequent, its antecedent $(\sim A \rightarrow B)$ must be F. But this requires that $\sim A$ be T and that B be F:

(Y)
$$\begin{array}{ccc|cccccccccc}
A & B & C & (\sim & A & \rightarrow & B) & \rightarrow & C & & B & / & C \\ \hline
 & \mathsf{T} & \mathsf{F} & \mathsf{T} & & \mathsf{F} & \mathsf{F/T} & \textbf{T} & \mathsf{F} & & \textbf{T} & & \textbf{F}
\end{array}$$

And there is no way to set B to F, as we have already seen that it has to be T in order to keep the second premise true—and no interpretation makes B *both* T and F. At this stage, we know, in our hearts, that there is no way to make both of the premises true and the conclusion false. In Part II we will turn this knowledge into an official mode of reasoning for validity. However, for now, let us consider a single row of a truth table (or a marked row of a full table) sufficient to demonstrate *invalidity*, but require a full table, exhibiting all the options, to show that an argument is sententially valid.

You may encounter odd situations where premises are never T, where conclusions are never F, or whatever. But if you stick to the definition, always asking whether there is any interpretation of the basic sentences that makes all the premises T and the conclusion F, all will be well.

E4.3. For each of the following, use truth tables to decide whether the entailment claims hold. Notice that a couple of the tables are already done from E4.2.

*a. $A \rightarrow \sim A \vDash_s \sim A$

b. $\sim A \rightarrow A \vDash_s \sim A$

*c. $A \rightarrow B, \sim A \vDash_s \sim B$

d. $A \rightarrow B, \sim B \vDash_s \sim A$

e. $\sim(A \rightarrow \sim B) \vDash_s B$

f. $\vDash_s C \rightarrow (A \rightarrow B)$

*g. $\vDash_s [A \rightarrow (C \rightarrow B)] \rightarrow [(A \rightarrow C) \rightarrow (A \rightarrow B)]$

h. $(A \rightarrow B) \rightarrow \sim(B \rightarrow A), \sim A, \sim B \vDash_s \sim(C \rightarrow C)$

i. $A \rightarrow \sim(B \rightarrow \sim C), B \rightarrow (\sim C \rightarrow D) \vDash_s A \rightarrow \sim(B \rightarrow \sim D)$

j. $\sim[(A \rightarrow \sim(B \rightarrow \sim C)) \rightarrow D], \sim D \rightarrow A \vDash_s C$

4.1.4 Abbreviations

We turn finally to applications for our abbreviations. Consider, first, a truth table for $\mathscr{P} \vee \mathscr{Q}$, that is for $\sim\mathscr{P} \rightarrow \mathscr{Q}$:

T′(∨)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$\mathscr{P}$	$\rightarrow$	$\mathscr{Q}$
T	T	F	T	***T***	T
T	F	F	T	***T***	F
F	T	T	F	***T***	T
F	F	T	F	***F***	F

so that

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \vee \mathscr{Q}$
T	T	***T***
T	F	***T***
F	T	***T***
F	F	***F***

When $\mathscr{P}$ is T and $\mathscr{Q}$ is T, $\mathscr{P} \vee \mathscr{Q}$ is T; when $\mathscr{P}$ is T and $\mathscr{Q}$ is F, $\mathscr{P} \vee \mathscr{Q}$ is T; and so forth. Thus, when $\mathscr{P}$ is T and $\mathscr{Q}$ is T, we *know* that $\mathscr{P} \vee \mathscr{Q}$ is T, without going through all the steps to get there in the unabbreviated form. Just as when $\mathscr{P}$ is a formula and $\mathscr{Q}$ is a formula, we move directly to the conclusion that $\mathscr{P} \vee \mathscr{Q}$ is a formula without explicitly working all the intervening steps, so if we know the truth value of $\mathscr{P}$ and the truth value of $\mathscr{Q}$, we can move in a tree by the above table to the truth value of $\mathscr{P} \vee \mathscr{Q}$ without all the intervening steps. And similarly for the other abbreviating sentential operators. For $\wedge$:

T′(∧)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$(\mathscr{P}$	$\rightarrow$	$\sim$	$\mathscr{Q})$
T	T	***T***	T	F	F	T
T	F	***F***	T	T	T	F
F	T	***F***	F	T	F	T
F	F	***F***	F	T	T	F

so that

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \wedge \mathscr{Q}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***F***

And for ($\leftrightarrow$):

T′(↔)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$[(\mathscr{P}$	$\rightarrow$	$\mathscr{Q})$	$\rightarrow$	$\sim$	$(\mathscr{Q}$	$\rightarrow$	$\mathscr{P})]$
T	T	***T***	T	T	T	F	F	T	T	T
T	F	***F***	T	F	F	T	F	F	T	T
F	T	***F***	F	T	T	T	T	T	F	F
F	F	***T***	F	T	F	F	F	F	T	F

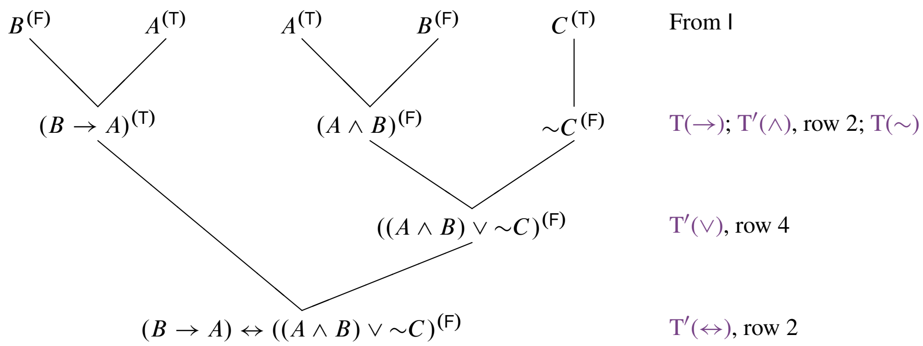
so that

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \leftrightarrow \mathscr{Q}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***T***

As a help toward remembering these tables, notice that $\mathscr{P} \vee \mathscr{Q}$ is F only when $\mathscr{P}$ is F and $\mathscr{Q}$ is F; $\mathscr{P} \wedge \mathscr{Q}$ is T only when $\mathscr{P}$ is T and $\mathscr{Q}$ is T; and $\mathscr{P} \leftrightarrow \mathscr{Q}$ is T only when $\mathscr{P}$ and $\mathscr{Q}$ are the same, and F when $\mathscr{P}$ and $\mathscr{Q}$ are different. The tables T′(∨), T′(∧), and T′(↔) represent derived additions to the definition for truth.

And nothing prevents direct application of the derived tables in trees. Suppose, for example, I$[A]$ = T, I$[B]$ = F, and I$[C]$ = T. Then I$[(B \rightarrow A) \leftrightarrow ((A \wedge B) \vee \sim C)]$ = F.

(Z)



We might get the same result by working through the full tree for the unabbreviated form. But there is no need. When A is T and B is F, we *know* that $(A \wedge B)$ is F; when $(A \wedge B)$ is F and $\sim C$ is F, we *know* that $((A \wedge B) \vee \sim C)$ is F; and so forth. Thus we move through the tree directly by the derived tables.

Similarly, we can work directly with abbreviated forms in truth tables.

(AA)

A	B	C	$(B$	$\rightarrow$	$A)$	$\leftrightarrow$	$((A$	$\wedge$	$B)$	$\vee$	$\sim$	$C)$
T	T	T	T	T	T	***T***	T	T	T	T	F	T
T	T	F	T	T	T	***T***	T	T	T	T	T	F
T	F	T	F	T	T	***F***	T	F	F	F	F	T
T	F	F	F	T	T	***T***	T	F	F	T	T	F
F	T	T	T	F	F	***T***	F	F	T	F	F	T
F	T	F	T	F	F	***F***	F	F	T	T	T	F
F	F	T	F	T	F	***F***	F	F	F	F	F	T
F	F	F	F	T	F	***T***	F	F	F	T	T	F

Tree (Z) represents just the third row of this table. As before, we construct the table "vertically," with tables for abbreviating operators in mind as appropriate.

Finally, given that we have tables for abbreviated forms, we can use them for evaluation of *arguments* with abbreviated forms. Thus, for example, $A \leftrightarrow B, A \models_s A \wedge B$.

Some perspective: There are different ways to understand tables for these new operators: We have understood them as derived from basic tables T($\sim$) and T($\rightarrow$). However, as we shall see in Chapter 11, it is possible to take tables for operators other than $\sim$ and $\rightarrow$ as basic—say just T($\sim$) and T$'$($\vee$), or just T($\sim$) and T$'$($\wedge$)—and then to abbreviate $\rightarrow$ in terms of them. Challenge: Find an expression involving just $\sim$ and $\vee$ that has the same table as $\rightarrow$; find one involving just $\sim$ and $\wedge$. Another option introduces all five as basic. Then the task is not *showing* that the table for $\vee$ is TTTF—that is given; rather we simply notice that $\mathcal{P} \vee \mathcal{Q}$, say, is redundant with $\sim\mathcal{P} \rightarrow \mathcal{Q}$. The latter approach avoids abbreviation. The former options abbreviate non-basic operators but preserve relative simplicity in the basic language.

(AB)

A	B	$(A$	$\leftrightarrow$	$B)$	A	/	$(A$	$\wedge$	$B)$
T	T	T	***T***	T	***T***		T	***T***	T
T	F	T	***F***	F	***T***		T	***F***	F
F	T	F	***F***	T	***F***		F	***F***	T
F	F	F	***T***	F	***F***		F	***F***	F

There is no row where each of the premises is true and the conclusion is false. So the argument is sententially valid. And, from either of the following rows,

(AC)

A	B	C	D	$(B$	$\rightarrow$	$A)$	$\wedge$	$(\sim$	C	$\vee$	$D)$	$(A$	$\leftrightarrow$	$\sim D)$	$\wedge$	$(\sim D$	$\rightarrow$	$B)$	/	B
F	F	T	T	F	T	F	***T***	F	T	T	T	F	T	FT	***T***	FT	T	F		***F***
F	F	F	T	F	T	F	***T***	T	F	T	T	F	T	FT	***T***	FT	T	F		***F***

we may conclude that $(B \rightarrow A) \wedge (\sim C \vee D), (A \leftrightarrow \sim D) \wedge (\sim D \rightarrow B) \nvDash_s B$. In this case, the shortcut table is attractive relative to the full version with sixteen rows!

E4.4. For each of the following, use truth tables to decide whether the entailment claims hold.

a. $\vDash_s A \vee \sim A$

b. $A \leftrightarrow [\sim A \leftrightarrow (A \wedge \sim A)], A \rightarrow \sim(A \leftrightarrow A) \vDash_s \sim A \rightarrow A$

*c. $B \vee \sim C \vDash_s B \rightarrow C$

d. $\sim(A \wedge \sim B) \vDash_s \sim A \vee B$

e. $\vDash_s \sim(A \leftrightarrow B) \leftrightarrow (A \wedge \sim B)$

*f. $A \vee B, \sim C \rightarrow \sim A, \sim(B \wedge \sim C) \vDash_s C$

g. $A \rightarrow (B \vee C), C \leftrightarrow B, \sim C \vDash_s \sim A$

h. $A \wedge (B \rightarrow C) \vDash_s (A \wedge B) \vee (A \wedge C)$

i. $A \vee (B \wedge \sim C), \sim(\sim B \vee C) \rightarrow \sim A \vDash_s \sim A \leftrightarrow \sim(C \vee \sim B)$

j. $A \vee B, \sim D \rightarrow (C \vee A) \vDash_s B \leftrightarrow \sim C$

E4.5. Complete the chart below to exhibit and explain step by step how to construct one or both rows from table (AC).

	A	B	C	D	$(B$	$\rightarrow$	$A)$	$\wedge$	$(\sim$	C	$\vee$	$D)$	$(A$	$\leftrightarrow$	$\sim D)$	$\wedge$	$(\sim D$	$\rightarrow$	$B)$	/	B	
1.								***T***								***T***					***F***	- premises T, conclusion F
2.		F			F			***T***								***T***			F		***F***	- fill in values for B

Semantics Quick Reference (sentential)

For any formal language $\mathcal{L}$, starting with a sentence and working up its tree, the *basic* sentences are the first sentences that do not have an operator from the sentential language as main operator. A *sentential interpretation* assigns a truth value *true* or *false*, T or F, to each basic sentence. Then for any interpretation I,

ST (∼) For any sentence $\mathcal{P}$, I[$\sim\mathcal{P}$] = T iff I[$\mathcal{P}$] = F; otherwise I[$\sim\mathcal{P}$] = F.

(→) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, I[$(\mathcal{P} \to \mathcal{Q})$] = T iff I[$\mathcal{P}$] = F or I[$\mathcal{Q}$] = T (or both); otherwise I[$(\mathcal{P} \to \mathcal{Q})$] = F.

And for abbreviated expressions,

ST′ (∧) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, I[$(\mathcal{P} \wedge \mathcal{Q})$] = T iff I[$\mathcal{P}$] = T and I[$\mathcal{Q}$] = T; otherwise I[$(\mathcal{P} \wedge \mathcal{Q})$] = F.

(∨) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, I[$(\mathcal{P} \vee \mathcal{Q})$] = T iff I[$\mathcal{P}$] = T or I[$\mathcal{Q}$] = T (or both); otherwise I[$(\mathcal{P} \vee \mathcal{Q})$] = F.

(↔) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, I[$(\mathcal{P} \leftrightarrow \mathcal{Q})$] = T iff I[$\mathcal{P}$] = I[$\mathcal{Q}$]; otherwise I[$(\mathcal{P} \leftrightarrow \mathcal{Q})$] = F.

These conditions result in tables as follows:

$\mathcal{P}$	$\mathcal{Q}$	$\sim\mathcal{P}$	$\mathcal{P} \to \mathcal{Q}$	$\mathcal{P} \vee \mathcal{Q}$	$\mathcal{P} \wedge \mathcal{Q}$	$\mathcal{P} \leftrightarrow \mathcal{Q}$
T	T	***F***	***T***	***T***	***T***	***T***
T	F	***F***	***F***	***T***	***F***	***F***
F	T	***T***	***T***	***T***	***F***	***F***
F	F	***T***	***T***	***F***	***F***	***T***

If Γ is a set of formulas, I[Γ] = T iff I[$\mathcal{P}$] = T for each $\mathcal{P}$ in Γ. Then, where the members of Γ are the formal premises of an argument, and sentence $\mathcal{Q}$ is its conclusion,

SV Γ *sententially entails* $\mathcal{Q}$ ($\Gamma \vDash_s \mathcal{Q}$) iff there is no sentential interpretation I such that I[Γ] = T but I[$\mathcal{Q}$] = F.

When Γ sententially entails $\mathcal{Q}$, the argument with premises Γ and conclusion $\mathcal{Q}$ is *sententially valid*. If Γ has no members and $\vDash_s \mathcal{Q}$, then $\mathcal{Q}$ is a *tautology*.

We treat a single row of a truth table (or a marked row of a full table) as sufficient to demonstrate *invalidity*, but require a full table, exhibiting all the options, to show that an argument is sententially valid.

E4.6. For each of the following, use truth tables to decide whether the entailment claims hold. Hint: The trick here is to identify the *basic* sentences; after that, everything proceeds in the usual way.

*a. $\exists x Ax \rightarrow \exists x Bx, \sim\exists x Ax \vDash_s \exists x Bx$

b. $\forall x Ax \rightarrow \sim\exists x(Ax \wedge \forall y By), \exists x(Ax \wedge \forall y By) \vDash_s \sim\forall x Ax$

E4.7. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Sentential interpretations and truth for complex sentences.

b. Sentential validity.

4.2 Quantificational

Semantics for the quantificational case work along the same lines as the sentential one. Sentences are true or false relative to an interpretation; arguments are semantically valid when there is no interpretation on which the premises are true and the conclusion is not. But, corresponding to differences between sentential and quantificational languages, the notion of an interpretation differs. And we introduce a preliminary notion of a *term* assignment, along with a preliminary notion of *satisfaction* distinct from truth, before we get to truth and validity. Certain issues are put off for Chapter 7 at the start of Part II. However, we should be able to do enough to see *how* the definitions work. This time, we will say a bit more about connections to English, though it remains important to see the definitions for what they are, and we leave official discussion of translation to the next chapter.

4.2.1 Interpretations

Given a quantificational language $\mathscr{L}$, formulas are true relative to a *quantificational interpretation*. As in the sentential case, languages do not *come* associated with any interpretation. Rather, a language consists of symbols which may be interpreted in different ways. In the sentential case, interpretations assigned T or F to basic sentences—and the assignments were made in arbitrary ways. Now assignments are more complex, but remain arbitrary. In general, QI A *quantificational interpretation* I of language $\mathcal{L}$ consists of a nonempty set U, the universe (or domain) of the interpretation, along with,

(s) An assignment of a truth value $\mathsf{I}[\mathcal{S}]$ to each sentence letter $\mathcal{S}$ of $\mathcal{L}$.

(c) An assignment of a member $\mathsf{I}[c]$ of U to each constant symbol c of $\mathcal{L}$.

(r) An assignment of an n-place relation $\mathsf{I}[\mathcal{R}^n]$ on U to each n-place relation symbol $\mathcal{R}^n$ of $\mathcal{L}$, where $\mathsf{I}[=]$ is always assigned $\{\langle \mathsf{o}, \mathsf{o}\rangle \mid \mathsf{o} \in \mathsf{U}\}$.

(f) An assignment of a total n-place function $\mathsf{I}[h^n]$ from U^{n} to U to each n-place function symbol h^n of $\mathcal{L}$.

The notions of a *relation* and a *function* for clauses (r) and (f) come from set theory—if these are in any way unfamiliar, you should refer now to the set theory reference on the following page. Conceived literally and mathematically, these assignments are themselves *functions* from symbols in the language $\mathcal{L}$ to objects. Each sentence letter is associated with a truth value, T or F—this is no different than before. Each constant symbol is associated with some element of U. Each n-place relation symbol is associated with a subset of U^{n}—with a set whose members are of the sort $\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle$ where $\mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}$ are elements of U. Each n-place function symbol is associated with a set whose members are of the sort $\langle\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle, \mathsf{b}\rangle$, where every $\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle \in \mathsf{U}^{\mathsf{n}}$ is matched to a single $\mathsf{b} \in \mathsf{U}$. And where $\mathsf{U} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots\}$, $\mathsf{I}[=] = \{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{b}, \mathsf{b}\rangle, \langle \mathsf{c}, \mathsf{c}\rangle, \ldots\}$. Note the (slight) typographical difference between '$=$' in the object language and '$=$' we use to express the relation. U may be any non-empty set, and so need not be countable. *Any* such assignments count as a quantificational interpretation.

Intuitively, the universe contains whatever objects are under consideration in a given context. Thus one may ask whether "everyone" wants anchovies on their pizza, and have in mind some limited collection of individuals—not literally everyone in the world. Constant symbols work like proper names: Constant symbol a names the object $\mathsf{I}[a]$ with which it is associated. So, for example, in $\mathcal{L}_q$ we might set $\mathsf{I}[b]$ to Barack, and $\mathsf{I}[c]$ to Michelle. Relation symbols are interpreted like predicates: Relation symbol $\mathcal{R}^n$ *applies* to the n-tuples with which it is associated. Thus in $\mathcal{L}_q$, where U is the set of all people, we might set $\mathsf{I}[H^1]$ to $\{\mathsf{o} \mid \mathsf{o}$ is happy$\}$,[3] and $\mathsf{I}[L^2]$ to $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ loves $\mathsf{n}\}$. Then if Barack is happy, H applies to Barack, and if Barack loves Michelle, L applies to $\langle$Barack, Michelle$\rangle$—though if she happens to be upset with him, L might not apply to $\langle$Michelle, Barack$\rangle$. Function symbols are used to pick out one object by means of other(s). Thus, when we say that *Bill's father* is happy, we pick out an object (the father) by means of another (Bill). Similarly, function symbols are like "oblique" names which pick out objects in response to inputs. Such behavior is commonplace in mathematics when we say, for example that $3 + 3$ is even—and we

[3]Or $\{\langle \mathsf{o}\rangle \mid \mathsf{o}$ is happy $\}$. As from the set theory reference, one-tuples are collapsed into their members.

Basic Notions of Set Theory

I. A *set* is a thing that may have other things as elements or members. If m is a member of set s we write $\mathsf{m} \in \mathsf{s}$. One set is identical to another iff their members are the same—so order is irrelevant. The members of a set may be specified by list: {Sally, Bob, Jim}, or by membership condition: $\{\mathsf{o} \mid \mathsf{o}$ is a student at CSUSB$\}$; read, 'the set of all objects o such that o is a student at CSUSB'. Since sets are things, one set may have other sets as members.

II. Like a set, an *n-tuple* is a thing with other things as elements or members. For any positive integer n, an n-tuple has n elements, where order matters. 2-tuples are frequently referred to as "pairs." An n-tuple may be specified by list: ⟨Sally, Bob, Jim⟩, or by membership condition, 'the first 5 people (taken in order) in line at the Bursar's window'. Nothing prevents sets of n-tuples, as $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ loves $\mathsf{n}\}$; read, 'the set of all m/n pairs such that the first member loves the second'. 1-tuples are frequently equated with their members. So, depending on context, {Sally, Bob, Jim} may be {⟨Sally⟩, ⟨Bob⟩, ⟨Jim⟩}.

III. Set r is a *subset* of set s iff every member of r is also a member of s. If r is a subset of s we write $\mathsf{r} \subseteq \mathsf{s}$. r is a *proper subset* of s ($\mathsf{r} \subset \mathsf{s}$) iff $\mathsf{r} \subseteq \mathsf{s}$ but $\mathsf{r} \neq \mathsf{s}$. Thus, for example, the subsets of $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$ are $\{\,\}$, $\{\mathsf{m}\}$, $\{\mathsf{n}\}$, $\{\mathsf{o}\}$, $\{\mathsf{m}, \mathsf{n}\}$, $\{\mathsf{m}, \mathsf{o}\}$, $\{\mathsf{n}, \mathsf{o}\}$, and $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. All but $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$ are proper subsets of $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. Notice that the *empty set* $\{\,\}$ (or $\varnothing$) is a subset of any set s, for it is sure to be the case that any member of it is also a member of s.

IV. The *union* of sets r and s is the set of all objects that are members of r *or* s. Thus, if $\mathsf{r} = \{\mathsf{m}, \mathsf{n}\}$ and $\mathsf{s} = \{\mathsf{n}, \mathsf{o}\}$, then the union of r and s, $(\mathsf{r} \cup \mathsf{s}) = \{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. Given a larger collection of sets, $\mathsf{s}_1, \mathsf{s}_2, \ldots$ the union of them all, $\bigcup \mathsf{s}_1, \mathsf{s}_2, \ldots$ is the set of all objects that are members of s_1, or s_2, or.... Similarly, the *intersection* of sets r and s is the set of all objects that are members of r *and* s. Thus the intersection of r and s, $(\mathsf{r} \cap \mathsf{s}) = \{\mathsf{n}\}$, and $\bigcap \mathsf{s}_1, \mathsf{s}_2, \ldots$ is the set of all objects that are members of s_1, and s_2, and....

V. Let s^{n} be the set of all n-tuples formed from members of s. Then an n-place *relation on set* s is any subset of s^{n}. Thus, for example, $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ is married to $\mathsf{n}\}$ is a subset of the pairs of people, and so is a 2-place relation on the set of people. An *n-place function from* r^{n} *to* s is a set of pairs whose first member is an element of r^{n} and whose second member is an element of s—restricted so that if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$ and $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}\rangle, \mathsf{b}\rangle \in \mathsf{f}$ then $\mathsf{a} = \mathsf{b}$; so no member of r^{n} is paired with more than one member of s. Thus $\langle\langle 1, 1\rangle, 2\rangle$ and $\langle\langle 1, 2\rangle, 3\rangle$ might be members of an addition function. $\langle\langle 1, 1\rangle, 2\rangle$ and $\langle\langle 1, 1\rangle, 3\rangle$ could not be members of the *same* function. A *total* function from r^{n} to s is one that pairs *each* member of r^{n} with some member of s. We think of the first element of these pairs as an *input*, and the second as the function's *output* for that input. Thus if $\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \in \mathsf{f}$ we say $\mathsf{f}(\mathsf{m}, \mathsf{n}) = \mathsf{o}$.

are talking about 6. Thus we might assign $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{n}$ is the father of $\mathsf{m}\}$ to one-place function symbol f and $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}$ plus $\mathsf{n} = \mathsf{o}\}$ to two-place function symbol p.

For some examples of interpretations, let us return to the language $\mathscr{L}_{NT}^{<}$ from section 2.3.5. Recall that $\mathscr{L}_{NT}^{<}$ includes just constant symbol $\emptyset$; two-place relation symbols $<, =$; one-place function symbol S; and two-place function symbols $\times$ and $+$. Given these symbols, terms and formulas are generated in the usual way. Where $\mathbb{N}$ is the set $\{0, 1, 2, \ldots\}$ of *natural numbers* and the *successor* of any natural number is the number after it, the *standard* interpretation $\bar{\mathsf{N}}$ for $\mathscr{L}_{NT}^{<}$ has universe $\mathbb{N}$ with,

$\bar{\mathsf{N}}$

$\bar{\mathsf{N}}[\emptyset] = 0$

$\bar{\mathsf{N}}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

$\bar{\mathsf{N}}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$\bar{\mathsf{N}}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\bar{\mathsf{N}}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

where it is automatic from QI that $\bar{\mathsf{N}}[=]$ is $\{\langle 0, 0\rangle, \langle 1, 1\rangle, \langle 2, 2\rangle, \ldots\}$. These definitions work just as we expect. Thus,

(AD)

$$\begin{aligned}
&\bar{\mathsf{N}}[<] = \{\langle 0, 1\rangle, \langle 0, 2\rangle, \langle 0, 3\rangle, \ldots, \langle 1, 2\rangle, \langle 1, 3\rangle, \ldots\}\\
&\bar{\mathsf{N}}[S] = \{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \ldots\}\\
&\bar{\mathsf{N}}[+] = \{\langle\langle 0, 0\rangle, 0\rangle, \langle\langle 0, 1\rangle, 1\rangle, \langle\langle 0, 2\rangle, 2\rangle, \ldots, \langle\langle 1, 0\rangle, 1\rangle, \langle\langle 1, 1\rangle, 2\rangle, \ldots\}\\
&\bar{\mathsf{N}}[\times] = \{\langle\langle 0, 0\rangle, 0\rangle, \langle\langle 0, 1\rangle, 0\rangle, \langle\langle 0, 2\rangle, 0\rangle, \ldots, \langle\langle 1, 0\rangle, 0\rangle, \langle\langle 1, 1\rangle, 1\rangle, \ldots\}
\end{aligned}$$

So $<$ is assigned a set of pairs; S a one-place total function, that is $\{\langle\langle 0\rangle, 1\rangle, \langle\langle 1\rangle, 2\rangle, \langle\langle 2\rangle, 3\rangle, \ldots\}$ but with 1-tuples reduced to their members; and $+$ and $\times$ are assigned two-place total functions. The standard interpretation represents the way you have understood these symbols since grade school.

But there is nothing sacred about this interpretation. Abbreviating, let *Bar(ack)* and *Mic(helle)* be Barack and Michelle. Then, for example, we might introduce a J with $\mathsf{U} = \{$Bar, Mic$\}$ and,

J

$\mathsf{J}[\emptyset] =$ Bar

$\mathsf{J}[<] = \{\langle$Mic, Mic$\rangle, \langle$Mic, Bar$\rangle\}$

$\mathsf{J}[S] = \{\langle$Bar, Bar$\rangle, \langle$Mic, Mic$\rangle\}$

$\mathsf{J}[+] = \{\langle\langle$Bar, Bar$\rangle$, Mic$\rangle, \langle\langle$Bar, Mic$\rangle$, Mic$\rangle, \langle\langle$Mic, Bar$\rangle$, Mic$\rangle, \langle\langle$Mic, Mic$\rangle$, Mic$\rangle\}$

$\mathsf{J}[\times] = \{\langle\langle$Bar, Bar$\rangle$, Mic$\rangle, \langle\langle$Bar, Mic$\rangle$, Bar$\rangle, \langle\langle$Mic, Bar$\rangle$, Bar$\rangle, \langle\langle$Mic, Mic$\rangle$, Bar$\rangle\}$

This assigns a member of the universe to the constant symbol; a set of pairs to the two-place relation symbol (where the interpretation of $=$ is automatic); a total 1-place function to S, and total 2-place functions to $+$ and $\times$. So it counts as an interpretation of $\mathscr{L}_{NT}^{<}$. Observe that a total n-place function on an m-membered universe has m^n members—so our 1-place function has $2^1 = 2$ members, and 2-place functions $2^2 = 4$ members.

It is frequently convenient to link assignments with bits of (relatively) ordinary language. This is a key to translation, as explored in the next chapter. But there is no requirement that we link up with ordinary language. All that is required is that we assign a member of U to each constant symbol, a subset of U^{n} to each n-place relation symbol, and a total function from U^{n} to U to each n-place function symbol. That is all that is required—and nothing beyond that is required in order to say what the function and predicate symbols "mean." So J counts as a legitimate (though non-standard) interpretation of $\mathscr{L}_{NT}^{<}$. With a language like $\mathscr{L}_q$ it is not always possible to specify assignments for *all* the symbols in the language. Even so, we can specify a *partial* interpretation—an interpretation for the symbols that matter in a given context.[4]

E4.8. Suppose Barack and Michelle have another child and name her Ama. Where U = {Bar, Mic, Ama}, give another interpretation K for $\mathscr{L}_{NT}^{<}$. Arrange your interpretation so that: (i) K[∅] ≠ Bar; (ii) there are exactly five pairs in K[<]; and (iii) for any m, ⟨⟨m, Bar⟩, Ama⟩ and ⟨⟨Bar, m⟩, Ama⟩ are in K[+]. Include K[=] in your account.

4.2.2 Term Assignments

In the sentential case, interpretations make assignments to basic sentences; assignments to further expressions derive from them. And similarly here: An interpretation (supplemented by a "variable assignment") makes assignments to basic vocabulary; assignments to complex expressions derive from basic assignments. We begin with *terms*.

For some language $\mathscr{L}$, say U = {o | o is a person}, one-place predicate H is assigned the set of happy people, and constant b is assigned Barack. Perhaps H applies to Barack. In this case, Hb comes out true. Intuitively, however, we cannot say that Hx is either true or false on this interpretation, precisely because there is no particular individual that x picks out—we do not know *who* is supposed to be happy. However we will be able to say that Hx is *satisfied* or not when the interpretation is supplemented with a *variable (designation) assignment* d associating each variable with some individual in U.

Given a language $\mathscr{L}$ and interpretation I, a *variable assignment* d associates each variable of $\mathscr{L}$ with some member of the universe U—a variable assignment is a total function from the variables of $\mathscr{L}$ to objects in U. Conceived pictorially, where U = $\{\mathsf{o}_1, \mathsf{o}_2, \ldots\}$, d and e are variable assignments:

[4]There are alternatives to the (classical) notion of an interpretation developed here. So, for example, it is possible to drop the assumptions that U is nonempty and that all assignments are to members of U. *Free logic* does just this: It sets up "inner" and "outer" domains, allowing that an inner domain U might be empty, and that not all assignments are to members of it. With our classical approach as background, free logics are introduced in Priest, *Non-Classical Logics*. A potential application is to *possible worlds* where not every object exists in the universe of every world.

$$\mathsf{d} \quad \begin{array}{ccccccccc} i & j & k & l & m & n & o & p & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \dots \\ \mathsf{o}_1 & \mathsf{o}_2 & \mathsf{o}_3 & \mathsf{o}_4 & \mathsf{o}_5 & \mathsf{o}_6 & \mathsf{o}_7 & \mathsf{o}_8 & \end{array}$$

$$\mathsf{e} \quad \begin{array}{cccccccccc} i & j & & k & l & m & n & o & p & \\ \downarrow & \downarrow & \swarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \dots \\ \mathsf{o}_1 & \mathsf{o}_2 & & \mathsf{o}_3 & \mathsf{o}_4 & \mathsf{o}_5 & \mathsf{o}_6 & \mathsf{o}_7 & \mathsf{o}_8 & \end{array}$$

Observe that the total function from variables to things assigns some element of U to every variable of $\mathcal{L}$. But this leaves room for one thing assigned to different variables, and things assigned to no variable at all. All that is required is that every variable is associated with some thing. If d assigns o to x we write $\mathsf{d}[x] = \mathsf{o}$. So $\mathsf{d}[k] = \mathsf{o}_3$ and $\mathsf{e}[k] = \mathsf{o}_2$. For any assignment d, $\mathsf{d}(x|\mathsf{o})$ is the assignment that is just like d except that o is assigned to x. Thus, $\mathsf{d}(k|\mathsf{o}_2) = \mathsf{e}$. Similarly,

$$\mathsf{f} \quad \begin{array}{ccccccccccc} i & j & & k & l & & m & n & o & p & \\ \downarrow & \downarrow & \swarrow & & & \searrow & \downarrow & \downarrow & \downarrow & \downarrow & \dots \\ \mathsf{o}_1 & \mathsf{o}_2 & & \mathsf{o}_3 & \mathsf{o}_4 & & \mathsf{o}_5 & \mathsf{o}_6 & \mathsf{o}_7 & \mathsf{o}_8 & \end{array}$$

$\mathsf{d}(k|\mathsf{o}_2, l|\mathsf{o}_5) = \mathsf{e}(l|\mathsf{o}_5) = \mathsf{f}$. Of course, if some d already has o assigned to x, then $\mathsf{d}(x|\mathsf{o})$ is just d. Thus, for example, $\mathsf{f}(i|\mathsf{o}_1)$ is just f itself. We will be willing to say that Hx is *satisfied* or not satisfied relative to an interpretation supplemented by a variable assignment.

But before we get to satisfaction, we need the general notion of a *term* assignment. In general, a term contributes to a formula by picking out some member of the universe U—terms act something like names. We have seen that an interpretation I assigns a member $\mathsf{I}[c]$ of U to each constant symbol c. And a variable assignment d assigns a member $\mathsf{d}[x]$ to each variable x. But these are assignments just to "basic" terms. For function symbols an interpretation assigns, not individual members of U, but certain complex sets. Still an interpretation I supplemented by a variable assignment d is sufficient to associate a member $\mathsf{I}_\mathsf{d}[t]$ of U with any term t of $\mathcal{L}$. Where $\langle\langle \mathsf{a}_1 \dots \mathsf{a}_\mathsf{n}\rangle, \mathsf{b}\rangle \in \mathsf{I}[h^n]$, let $\mathsf{I}[h^n]\langle \mathsf{a}_1 \dots \mathsf{a}_\mathsf{n}\rangle = \mathsf{b}$; that is, $\mathsf{I}[h^n]\langle \mathsf{a}_1 \dots \mathsf{a}_\mathsf{n}\rangle$ is the thing the function $\mathsf{I}[h^n]$ associates with input $\langle \mathsf{a}_1 \dots \mathsf{a}_\mathsf{n}\rangle$. Thus, for example, from the interpretations on page 113, $\bar{\mathsf{N}}[+]\langle 1, 1\rangle = 2$ and $\mathsf{J}[+]\langle \text{Bar, Mic}\rangle = \text{Mic}$. Then for any interpretation I and variable assignment d,

TA (c) If c is a constant, then $\mathsf{I}_\mathsf{d}[c] = \mathsf{I}[c]$.

(v) If x is a variable, then $\mathsf{I}_\mathsf{d}[x] = \mathsf{d}[x]$.

(f) If h^n is a function symbol and $t_1 \dots t_n$ are terms, then $\mathsf{I}_\mathsf{d}[h^n t_1 \dots t_n] = \mathsf{I}[h^n]\langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle$.

The first two clauses take over assignments to constants and variables from I and d. The last clause is parallel to the one by which terms are formed. The assignment to a complex term $h^n t_1 \dots t_n$ depends on the interpretation of h^n, together with assignments to $t_1 \dots t_n$.

Again the definition is recursive, and we can see how it works on a tree—in this case, one with the very same shape as the one by which we see that an expression is in fact a term. Say the interpretation of $\mathscr{L}_{NT}^{<}$ is J from page 113, and $\mathsf{d}[x]$ = Mic; then $\mathsf{J_d}[S(Sx \times \emptyset)]$ = Bar.

(AE)

$$
\begin{array}{lll}
x^{[\text{Mic}]} & \emptyset^{[\text{Bar}]} & \text{By TA(v) and TA(c)} \\
Sx^{[\text{Mic}]} & & \text{With the input, since } \langle \text{Mic, Mic} \rangle \in \mathsf{J}[S] \text{, by TA(f)} \\
(Sx \times \emptyset)^{[\text{Bar}]} & & \text{With the inputs, since } \langle\langle \text{Mic, Bar} \rangle, \text{Bar} \rangle \in \mathsf{J}[\times] \text{, by TA(f)} \\
S(Sx \times \emptyset)^{[\text{Bar}]} & & \text{With the input, since } \langle \text{Bar, Bar} \rangle \in \mathsf{J}[S] \text{, by TA(f)}
\end{array}
$$

As usual, basic elements occur in the top row. After that, given the interpretation of the parts we *look* to see the interpretation of the whole. In the simplest case, the assignment to a term $h^1 t$ is whatever object the interpretation of h^1 pairs with the object assigned to t; so from $\mathsf{J_d}[x]$ = Mic and $\langle$Mic, Mic$\rangle \in \mathsf{J}[S]$, $\mathsf{J_d}[Sx]$ = Mic. For $h^n t_1 \dots t_n$ we find the object paired with whatever objects are assigned to $t_1 \dots t_n$ taken in that order; so given $\mathsf{J_d}[Sx]$ = Mic and $\mathsf{J_d}[\emptyset]$ = Bar, with $\langle\langle$Mic, Bar$\rangle$, Bar$\rangle \in \mathsf{J}[\times]$, we get $\mathsf{J_d}[Sx \times \emptyset]$ = Bar. Perhaps the hard part about definition TA is just reading clause (f)—it may be easier to apply in practice than to read. For a complex term, assignments to terms that are the parts together with the assignment to the function symbol, determine the assignment to the whole. And this is just what clause (f) says. For practice, convince yourself that $\mathsf{J}_{\mathsf{d}(x|\text{Bar})}[S(Sx \times \emptyset)]$ = Mic; and where $\bar{\mathsf{N}}$ is as above and $\mathsf{d}[x] = 1$, that $\bar{\mathsf{N}}_\mathsf{d}[S(Sx \times \emptyset)] = 1$.

E4.9. For $\mathscr{L}_{NT}^{<}$ and interpretation $\bar{\mathsf{N}}$ from page 113, let d include,

$$
\mathsf{d} \quad
\begin{array}{cccc}
w & x & y & z \\
\downarrow & \downarrow & \downarrow & \downarrow \\
1 & 2 & 3 & 4
\end{array}
$$

and use trees to determine each of the following.

*a. $\bar{\mathsf{N}}_\mathsf{d}[+xS\emptyset]$

b. $\bar{\mathsf{N}}_\mathsf{d}[x + (SS\emptyset \times x)]$

c. $\bar{\mathsf{N}}_\mathsf{d}[w \times S(\emptyset + (y \times SSSz))]$

*d. $\bar{\mathsf{N}}_{\mathsf{d}(x|4)}[x + (SS\emptyset \times x)]$

e. $\bar{\mathsf{N}}_{\mathsf{d}(x|1,w|2)}[S(x \times (S\emptyset + Sw))]$

E4.10. For $\mathcal{L}_{NT}^{<}$ and interpretation J from page 113, let d include,

$$\mathsf{d} \quad \begin{array}{cccc} w & x & y & z \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \text{Bar} & \text{Mic} & \text{Mic} & \text{Mic} \end{array}$$

and use trees to determine each of the following.

*a. $\mathsf{J}_{\mathsf{d}}[+xS\emptyset]$

b. $\mathsf{J}_{\mathsf{d}}[x + (SS\emptyset \times x)]$

c. $\mathsf{J}_{\mathsf{d}}[w \times S(\emptyset + (y \times SSSz))]$

*d. $\mathsf{J}_{\mathsf{d}(x|\text{Bar})}[x + (SS\emptyset \times x)]$

e. $\mathsf{J}_{\mathsf{d}(x|\text{Bar},w|\text{Mic})}[S(x \times (S\emptyset + Sw))]$

E4.11. Consider your interpretation K for $\mathcal{L}_{NT}^{<}$ from E4.8. Supposing that $\mathsf{d}[w]$ = Bar, $\mathsf{d}[y]$ = Mic, and $\mathsf{d}[z]$ = Ama, determine $\mathsf{K}_{\mathsf{d}}[w \times S(\emptyset + (y \times SSSz))]$.

E4.12. For $\mathcal{L}_q$ and an interpretation L with universe U = {Amy, Bob, Chris} with,

L	$\mathsf{L}[a]$ = Amy
$\mathsf{L}[c]$ = Chris
$\mathsf{L}[f^1]$ = {⟨Amy, Bob⟩, ⟨Bob, Chris⟩, ⟨Chris, Amy⟩}
$\mathsf{L}[g^2]$ = {⟨⟨Amy, Amy⟩, Amy⟩, ⟨⟨Amy, Bob⟩, Chris⟩, ⟨⟨Amy, Chris⟩, Bob⟩,
⟨⟨Bob, Amy⟩, Chris⟩, ⟨⟨Bob, Bob⟩, Bob⟩, ⟨⟨Bob, Chris⟩, Amy⟩,
⟨⟨Chris, Amy⟩, Bob⟩, ⟨⟨Chris, Bob⟩, Amy⟩, ⟨⟨Chris, Chris⟩, Chris⟩}

where $\mathsf{d}[x]$ = Bob, $\mathsf{d}[y]$ = Amy and $\mathsf{d}[z]$ = Bob, use trees to determine each of the following.

a. $\mathsf{L}_{\mathsf{d}}[f^1c]$

*b. $\mathsf{L}_{\mathsf{d}}[g^2yf^1c]$

c. $\mathsf{L}_{\mathsf{d}}[g^2g^2axf^1c]$

d. $\mathsf{L}_{\mathsf{d}(x|\text{Chris})}[g^2g^2axf^1c]$

e. $\mathsf{L}_{\mathsf{d}(x|\text{Amy})}[g^2g^2g^2xyzg^2f^1af^1c]$

4.2.3 Satisfaction

A term's assignment depends on an interpretation supplemented by an assignment for variables, that is, on some $\mathsf{I_d}$. Similarly, a formula's *satisfaction* depends on both the interpretation and variable assignment. If a formula $\mathscr{P}$ is satisfied on I supplemented with d, we write $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$; if $\mathscr{P}$ is not satisfied on I with d, $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$. For any interpretation I with variable assignment d,

SF (s) If $\mathscr{S}$ is a sentence letter, then $\mathsf{I_d}[\mathscr{S}] = \mathsf{S}$ iff $\mathsf{I}[\mathscr{S}] = \mathsf{T}$; otherwise $\mathsf{I_d}[\mathscr{S}] = \mathsf{N}$.

(r) If $\mathscr{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are terms, $\mathsf{I_d}[\mathscr{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n] \rangle \in \mathsf{I}[\mathscr{R}^n]$; otherwise $\mathsf{I_d}[\mathscr{R}^n t_1 \dots t_n] = \mathsf{N}$.

($\sim$) If $\mathscr{P}$ is a formula, then $\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$; otherwise $\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{N}$.

($\rightarrow$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathsf{I_d}[(\mathscr{P} \rightarrow \mathscr{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I_d}[(\mathscr{P} \rightarrow \mathscr{Q})] = \mathsf{N}$.

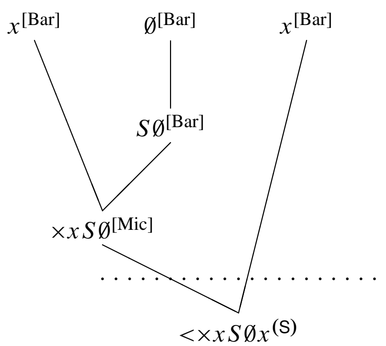
($\forall$) If $\mathscr{P}$ is a formula and x is a variable, then $\mathsf{I_d}[\forall x \mathscr{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$; otherwise $\mathsf{I_d}[\forall x \mathscr{P}] = \mathsf{N}$.

SF(s) and SF(r) determine satisfaction for atomic formulas. Satisfaction for other formulas depends on satisfaction of their immediate subformulas. SF(s), SF($\sim$), and SF($\rightarrow$) are closely related to ST from before, though satisfaction applies now to any *formulas* and not only to sentences. SF(r) and SF($\forall$) are new.

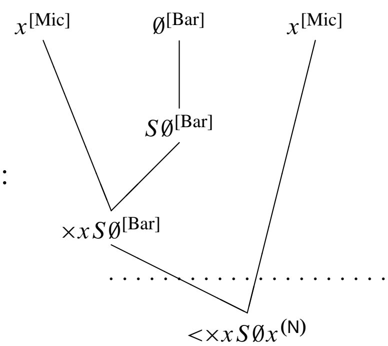
First, the satisfaction of a sentence letter works just like truth before: a sentence letter is satisfied on some $\mathsf{I_d}$ iff it is true on the interpretation I. Thus satisfaction for sentence letters depends only on the interpretation, and not at all on the variable assignment.

In contrast, to see if $\mathscr{R}^n t_1 \dots t_n$ is satisfied, we find out which things are assigned to the terms, and then see if those objects, taken in order, are in the interpretation of the relation symbol. It is natural to think about this on a tree like the one by which we show that the expression is a formula. Thus given interpretation J for $\mathscr{L}_{NT}^{<}$ from page 113, consider $(x \times S\emptyset) < x$; and compare cases with $\mathsf{d}[x] = \text{Bar}$, and $\mathsf{h}[x] = \text{Mic}$. It will be convenient to think about the expression in its unabbreviated form, $<\times x S\emptyset x$.

$\mathsf{J_d}$:



$\mathsf{J_h}$:



Above the dotted line, we calculate term assignments in the usual way. But $<\times x S\emptyset x$ is a formula of the sort $<t_1 t_2$. From the left-hand tree, $\mathsf{J_d}[\times x S\emptyset]$ = Mic, and $\mathsf{J_d}[x]$ = Bar. So the assignments to t_1 and t_2 are Mic and Bar. Since ⟨Mic, Bar⟩ ∈ $\mathsf{J}[<]$, by SF(r), $\mathsf{J_d}[<\times x S\emptyset x] = \mathsf{S}$. But from the right-hand tree, $\mathsf{J_h}[\times x S\emptyset]$ = Bar, and $\mathsf{J_h}[x]$ = Mic. And ⟨Bar, Mic⟩ ∉ $\mathsf{J}[<]$, so by SF(r), $\mathsf{J_h}[<\times x S\emptyset x] = \mathsf{N}$. $\mathcal{R}^n t_1 \ldots t_n$ is satisfied just in case the n-tuple of the thing assigned to t_1 and ... and the thing assigned to t_n is in the set assigned to the relation symbol. To decide if $\mathcal{R}^n t_1 \ldots t_n$ is satisfied, we find out what things are assigned to the term or terms, and then look to see whether the relevant ordered sequence is in the interpretation. The simplest sort of case is when there is just one term. $\mathsf{I_d}[\mathcal{R}^1 t] = \mathsf{S}$ just in case $\mathsf{I_d}[t] \in \mathsf{I}[\mathcal{R}^1]$. When there is more than one term, we look for the objects taken in order.

SF(∼) and SF(→) correspond to ST(∼) and ST(→). And we could work out their consequences on trees or tables for satisfaction as before. In this case though, to accomodate quantifiers, it will be convenient to turn the "trees" on their sides. For this, we begin by constructing the tree in the "forward direction," from left to right, and then determine satisfaction the other way—from the branch tips back to the trunk. Where the members of U are {m, n, ...}, the branch conditions are as follows:

			forward	*backward*
B(s)	$\mathsf{I_d}[\mathcal{S}]$		no branching	the tip is S iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$
B(r)	$\mathsf{I_d}[\mathcal{R}^n t_1 \ldots t_n]$		branches only for terms	the tip is S iff $\langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n] \rangle \in \mathsf{I}[\mathcal{R}^n]$
B(∼)	$\mathsf{I_d}[\sim\mathcal{P}]$	∼	$\mathsf{I_d}[\mathcal{P}]$	the trunk is S iff the branch is N
B(→)	$\mathsf{I_d}[(\mathcal{P} \rightarrow \mathcal{Q})]$	→	$\mathsf{I_d}[\mathcal{P}]$ $\mathsf{I_d}[\mathcal{Q}]$	the trunk is S iff the top branch is N or the bottom branch is S (or both)
B(∀)	$\mathsf{I_d}[\forall x \mathcal{P}]$	$\forall x$	$\mathsf{I_{d(x\mid m)}}[\mathcal{P}]$ $\mathsf{I_{d(x\mid n)}}[\mathcal{P}]$ ⋮ one branch for each member of U	the trunk is S iff every branch is S

A formula branches according to its main operator. If it is atomic, it does not branch (or branches only for its terms). The trees $\mathsf{J_d}$ and $\mathsf{J_h}$ on the preceding page are examples of branching for terms, only oriented vertically. If the main operator is ∼, a formula has just one branch; if its main operator is →, it has two branches; and if its main operator is ∀ it has as many branches as there are members of U. This last condition makes it impractical to construct these trees in all but the most simple cases—and impossible when U is infinite. Still, we can use them to see how the definitions work.

When there are no quantifiers, we should be able to recognize these trees as a mere "sideways" variant of ones we have seen before. Thus, consider an interpretation M with $\mathsf{U} = \{\text{Bob, Sue, Jim}\}$ and,

$$
\begin{array}{ll}
\mathsf{M} & \mathsf{M}[A] = \mathsf{T} \\
 & \mathsf{M}[B^1] = \{\text{Sue}\} \\
 & \mathsf{M}[C^2] = \{\langle \text{Bob, Sue} \rangle, \langle \text{Sue, Jim} \rangle\}
\end{array}
$$

and variable assignment d such that $\mathsf{d}[x] = \text{Bob}$. Then,

$$
\begin{array}{llllll}
 & 1 & & 2 & & 3 \\
 & & & \underline{\mathsf{M_d}[\sim A]^{(\mathsf{N})}} & \sim & \underline{\mathsf{M_d}[A]^{(\mathsf{S})}} \\
\text{(AF)} & \underline{\mathsf{M_d}[\sim A \rightarrow Bx]^{(\mathsf{S})}} & \rightarrow & & & \\
 & & & \underline{\mathsf{M_d}[Bx]^{(\mathsf{N})}} & \vdots & \underline{\quad} \; x^{[\text{Bob}]}
\end{array}
$$

The main operator at stage (1) is $\rightarrow$; so there are two branches. Bx on the bottom is atomic, so the formula branches no further—though we use TA to calculate the term assignment. On the top at (2), $\sim A$ has main operator $\sim$. So there is one branch. And we are done with the forward part of the tree. Given this, we can calculate satisfaction from the tips back toward the trunk. Since $\mathsf{M}[A] = \mathsf{T}$, by B(s), the top at (3) is S. And since this is S, by B($\sim$), the top at (2) is N. But since $\mathsf{M_d}[x] = \text{Bob}$, and $\text{Bob} \notin \mathsf{M}[B]$, by B(r), the bottom at (2) is N. And with both the top and bottom at (2) N, by B($\rightarrow$), the formula at (1) is S. So $\mathsf{M_d}[\sim A \rightarrow Bx] = \mathsf{S}$. You should be able to recognize that the diagram (AF) rotated counterclockwise by 90 degrees would be a mere variant of diagrams we have seen before. And the branch conditions merely implement the corresponding conditions from SF.

Things are more interesting when there are quantifiers. For a quantifier, there are as many branches as there are members of U. First, working with a "stripped down" version of M that has $\mathsf{U} = \{\text{Bob}\}$, consider $\mathsf{M_d}[\forall y \sim Cxy]$. With just one thing in the universe, the tree branches as follows:

$$
\begin{array}{llllllll}
 & 1 & & 2 & & 3 & & 4 \\
 & & & & & & & x^{[\text{Bob}]} \\
\text{(AG)} & \underline{\mathsf{M_d}[\forall y \sim Cxy]^{(\mathsf{S})}} & \forall y & \underline{\mathsf{M_{d(y|\text{Bob})}}[\sim Cxy]^{(\mathsf{S})}} & \sim & \underline{\mathsf{M_{d(y|\text{Bob})}}[Cxy]^{(\mathsf{N})}} & \vdots < & \\
 & & & & & & & y^{[\text{Bob}]}
\end{array}
$$

The main operator at (1) is the universal quantifier. With one thing in U, there is the one branch. Notice that the variable assignment d becomes $\mathsf{d}(y|\text{Bob})$. The main operator at (2) is $\sim$. So there is the one branch, carrying forward the assignment $\mathsf{d}(y|\text{Bob})$. The formula at (3) is atomic, so the only branching is for the term assignment. Then, in the backward direction, $\mathsf{M_{d(y|\text{Bob})}}$ still assigns Bob to x; and $\mathsf{M_{d(y|\text{Bob})}}$ assigns Bob to y. Since $\langle \text{Bob, Bob} \rangle \notin \mathsf{M}[C^2]$, the branch at (3) is N; so the branch at (2) is S. And since *all* the branches for the universal quantifier are S, by B(∀), the formula at (1) is S.

But M was originally defined with U = {Bob, Sue, Jim}. In this case the quantifier requires not one but three branches, and the tree is as follows:

$$\begin{array}{llllllll}
 & 1 & & 2 & & 3 & & 4 \\
 & & & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Bob})}[\sim Cxy]}^{(\mathsf{S})} & \sim & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Bob})}[Cxy]}^{(\mathsf{N})} & \vdots & x^{[\text{Bob}]} \quad y^{[\text{Bob}]} \\
\text{(AH)} & \underline{\mathsf{M}_{\mathsf{d}}[\forall y{\sim}Cxy]}^{(\mathsf{N})} & \forall y & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Sue})}[\sim Cxy]}^{(\mathsf{N})} & \sim & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Sue})}[Cxy]}^{(\mathsf{S})} & \vdots & x^{[\text{Bob}]} \quad y^{[\text{Sue}]} \\
 & & & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Jim})}[\sim Cxy]}^{(\mathsf{S})} & \sim & \underline{\mathsf{M}_{\mathsf{d}(y|\text{Jim})}[Cxy]}^{(\mathsf{N})} & \vdots & x^{[\text{Bob}]} \quad y^{[\text{Jim}]}
\end{array}$$

The quantifier has one branch for each member of U. Note the modification of d on each branch, and the way the modified assignments carry forward and are used for evaluation at the tips. $\mathsf{d}(y|\text{Sue})$, say, has the same assignment to x as d, but assigns Sue to y. And similarly for the rest. This time, not all the branches for the universal quantifier are S. So the formula at (1) is N. You should convince yourself that it is S on M_{h} where $\mathsf{h}[x]$ = Jim. And it would be S with assignment d as above, but formula $\forall y{\sim}Cyx$.

(AI) on page 123 is an example for $\forall x[(Sx < x) \rightarrow \forall y((Sy + \emptyset) = x)]$ using interpretation J from page 113 and $\mathcal{L}^{<}_{NT}$. This case should help you to see how all the parts fit together in a reasonably complex example. It turns out to be helpful to think about the formula in its unabbreviated form, $\forall x({<}Sxx \rightarrow \forall y{=}{+}Sy\emptyset x)$. For this case notice especially how when multiple quantifiers come off, a variable assignment once modified is simply modified again for the new variable. If you follow through the details of this case by the definitions, you are doing well.

A word of advice: Once you have the idea, constructing these trees to determine satisfaction is a mechanical (and tedious) process. About the only way to go wrong or become confused is by skipping steps or modifying the form of trees. But, very often, skipping steps or modifying form does correlate with confusion. So it is best to stick with the official pattern—and so to follow the way it forces you through definitions SF and TA.

E4.13. Supplement interpretation L for E4.12 so that U = {Amy, Bob, Chris} and,

L $\mathsf{L}[a]$ = Amy
$\mathsf{L}[c]$ = Chris
$\mathsf{L}[f^1]$ = {⟨Amy, Bob⟩, ⟨Bob, Chris⟩, ⟨Chris, Amy⟩}
$\mathsf{L}[g^2]$ = {⟨⟨Amy, Amy⟩, Amy⟩, ⟨⟨Amy, Bob⟩, Chris⟩, ⟨⟨Amy, Chris⟩, Bob⟩,
⟨⟨Bob, Amy⟩, Chris⟩, ⟨⟨Bob, Bob⟩, Bob⟩, ⟨⟨Bob, Chris⟩, Amy⟩,
⟨⟨Chris, Amy⟩, Bob⟩, ⟨⟨Chris, Bob⟩, Amy⟩, ⟨⟨Chris, Chris⟩, Chris⟩}

$\mathsf{L}[S] = \mathsf{T}$
$\mathsf{L}[H^1] = \{\text{Amy, Bob}\}$
$\mathsf{L}[L^2] = \{\langle\text{Amy, Amy}\rangle, \langle\text{Amy, Bob}\rangle, \langle\text{Amy, Chris}\rangle, \langle\text{Bob, Bob}\rangle, \langle\text{Bob, Chris}\rangle\}$

Where $\mathsf{d}[x]$ = Amy, and $\mathsf{d}[y]$ = Bob, use trees to determine whether the following formulas are satisfied on L with d.

*a. Hx b. Lxa

c. Hf^1y d. $\forall xLyx$

e. $\forall xLxg^2cx$ *f. $\sim\forall x(Hx \to \sim S)$

*g. $\forall y\sim\forall xLxy$ h. $\forall y\sim\forall xLyx$

i. $\forall x(Hf^1x \to Lxx)$ j. $\forall x(Hx \to \sim\forall y\sim Lyx)$

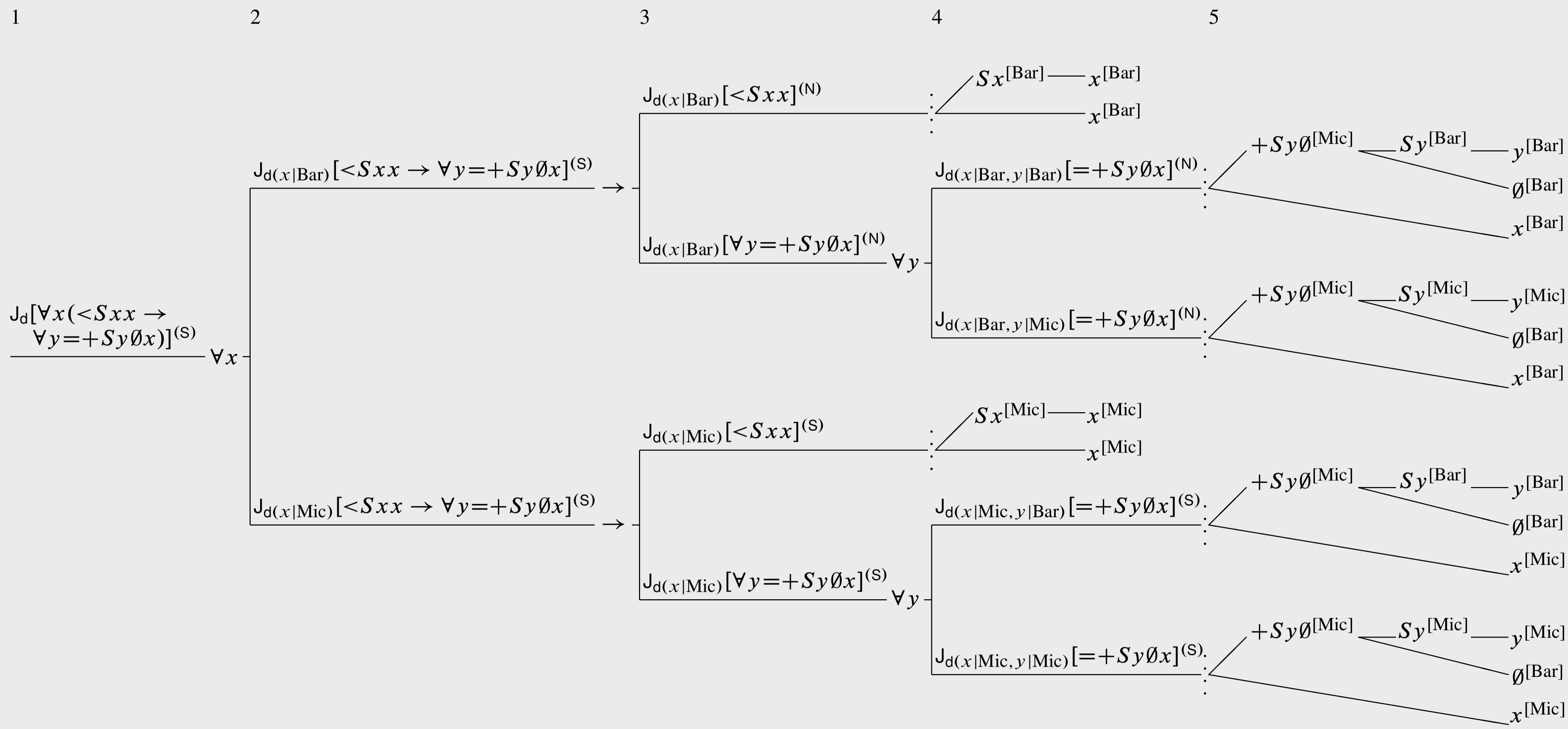
E4.14. For the previous problem, what if anything changes with the variable assignment h where $\mathsf{h}[x]$ = Chris and $\mathsf{h}[y]$ = Amy? Challenge: Explain why differences in the initial variable assignment *cannot* matter for the evaluation of (e)–(j).

4.2.4 Truth and Validity

It is a short step from satisfaction to definitions for *truth* and *validity*. As we have seen, formulas are satisfied or not on an interpretation I together with a variable assignment d. After that, truth runs through satisfaction: a formula is true on an interpretation when it is satisfied relative to *every* variable assignment. A consequence is that truth does not depend on the details of any particular assignment—and formulas are *true* and *false* relative just to an interpretation I.

TI A formula $\mathscr{P}$ is *true* on an interpretation I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$. $\mathscr{P}$ is *false* on I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$.

A formula is true on I just in case it is satisfied with every variable assignment for I. From (AH), then, we are already in a position to see that $\forall y\sim Cxy$ is not true on M—for there is a variable assignment d on which it is N; since there is an assignment on which it is N, it is not satisfied on *every* assignment, and so is not true. Neither is $\forall y\sim Cxy$ false on M, insofar as it is satisfied on the h that assigns Jim to x; since there is an assignment on which it is S, it is not N on *every* assignment, and so is not false. In contrast, from (AI), $\forall x[(Sx < x) \to \forall y((Sy + \emptyset) = x)]$ is true on J. For some variable assignment d, the tree shows directly that $\mathsf{J_d}[\forall x[(Sx < x) \to \forall y((Sy + \emptyset) = x)]] = \mathsf{S}$. But the reasoning for the tree *makes no assumptions whatsoever about* d. That is, with any variable assignment, we might have reasoned in just the same way to reach the conclusion that the formula is satisfied. Since it comes out satisfied no matter what the variable assignment may be, by TI, it is true.



(AI)

Forward: Since there are two objects in U, there are two branches for each quantifier. At stage (2), for the x-quantifier, d is modified for assignments to x, and at lower sections of (4) for the y-quantifier those assignments are modified again. $<Sxx$ and $=+Sy\emptyset x$ are atomic. Branching for terms continues at stages (4) and (5) in the usual way.

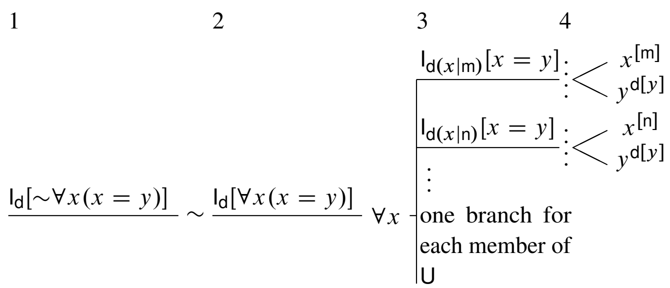
Backward: For terms, apply the variable assignment from the corresponding atomic formula. So in the top at (5) with $\mathsf{d}(x|\text{Bar}, y|\text{Bar})$, both x and y are assigned to Bar. The assignment to $\emptyset$ comes from the interpretation. Then terms and formulas are calculated in the usual way. At (4), recall that $\mathsf{J}[=]$ is automatically $\{\langle \text{Bar}, \text{Bar}\rangle, \langle \text{Mic}, \text{Mic}\rangle\}$.

In general, if a *sentence* is satisfied on some d for I, then it is satisfied on every d for I. We shall demonstrate this more formally in Chapter 8. However, we are already in a position to see the basic idea: In a sentence, every variable is bound; so by the time you get to formulas without quantifiers at the tips of a tree, assignments are of the sort, $\mathsf{d}(x|\mathsf{m}, y|\mathsf{n}, \ldots)$ for every variable in the formula; so satisfaction depends just on assignments that are set on the branch itself, and the initial d is irrelevant to satisfaction at the tips—and thus to evaluation of the formula as a whole. Adjustments to the assignment that occur within the tree *override* the original assignment so that every starting d gives the same result. So if a *sentence* is satisfied on some d for I, it is satisfied on every d for I, and therefore true on I. Similarly, if a sentence is N on some d for I, it is N on every d for I, and therefore false on I.

In contrast, a formula with free variables may be sensitive to the initial variable assignment. If variable x is free in formula $\mathcal{P}$, then the value for x at a branch tip results from the original $\mathsf{d}[x]$ rather than by adjustments to the assignment that are set within the branch. Thus, in the ordinary case, Hx is not true and not false: There may be an assignment d on which x is assigned an object in the interpretation of H so that Hx is satisfied, and an assignment h on which x is assigned an object not in the interpretation of H so that Hx is not satisfied; in this case, Hx is neither true nor false. We have seen this pattern so far in examples and exercises: For formulas with free variables, there may be variable assignments where they are satisfied, and variable assignments where they are not. Therefore the formulas fail to be either true or false by TI. Sentences, on the other hand, are satisfied on every variable assignment if they are satisfied on any, and not satisfied on every assignment if they are not satisfied on any. Therefore the sentences from our examples and exercises come out either true or false.

But a word of caution is in order: Sentences are always true or false on an interpretation. And, in the ordinary case, formulas with free variables are neither true nor false. But this is not always so. Thus $x = x$ is true on any I: given the fixed interpretation of '$=$', for any d and object $\mathsf{I_d}[x]$, $\langle \mathsf{I_d}[x], \mathsf{I_d}[x] \rangle$ is sure to be an element of $\mathsf{I}[=]$, so that $\mathsf{I_d}[x = x] = \mathsf{S}$ and $\mathsf{I}[x = x] = \mathsf{T}$. Similarly, $\mathsf{I}[Hx] = \mathsf{T}$ if $\mathsf{I}[H] = \mathsf{U}$ and F if $\mathsf{I}[H] = \{\,\}$. And $\sim\forall x(x = y)$ is true on any I with a U that has more than one member. To see this, suppose for some I, $\mathsf{U} = \{\mathsf{m}, \mathsf{n}, \ldots\}$; then for an arbitrary d the tree is as follows:

(AJ)



No matter what d is like, exactly one branch at (3) is S. If $\mathsf{d}[y] = \mathsf{m}$ then the top branch at (3) is S and the rest are N. If $\mathsf{d}[y] = \mathsf{n}$ then the second branch at (3) is S and the others are N. And so forth. So in this case where U has more than one member, at least one branch is N for any d. So the universally quantified expression is N for any d, and the negation at (1) is S for any d. So by TI it is true. So satisfaction for an open formula may but need not be sensitive to the particular variable assignment under consideration. Again, though, a *sentence* is always true or false depending only on the interpretation. To show that a sentence is true, it is enough to show that it is satisfied on some d, from which it follows that it is satisfied on any. For a formula with free variables, the matter is more complex—though you can show that such a formula is *not* true by finding an assignment that makes it N, and *not* false by finding an assignment that makes it S.

Given the notion of truth, *quantificational validity* works very much as before. Where Γ is a set of formulas, say $\mathsf{I}[\Gamma] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ for each formula $\mathcal{P} \in \Gamma$. Then for any formula $\mathcal{P}$,

QV Γ *quantificationally entails* $\mathcal{P}$ iff there is no quantificational interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$.

Γ quantificationally entails $\mathcal{P}$ when there is no quantificational interpretation that makes the premises true and the conclusion not. If Γ quantificationally entails $\mathcal{P}$ we write, $\Gamma \vDash \mathcal{P}$, and say an argument whose premises are the members of Γ and conclusion is $\mathcal{P}$ is *quantificationally valid*. Γ does not quantificationally entail $\mathcal{P}$ ($\Gamma \nvDash \mathcal{P}$) when there is some quantificational interpretation on which all the premises are true but the conclusion is not true (notice that there is a difference between being not true, and being false). As before, if $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are the members of Γ, we sometimes write $\mathcal{Q}_1 \ldots \mathcal{Q}_n \vDash \mathcal{P}$ in place of $\Gamma \vDash \mathcal{P}$. In the case where Γ is empty and there are no premises, we simply write $\vDash \mathcal{P}$. If $\vDash \mathcal{P}$, then $\mathcal{P}$ is a *tautology*.[5] Notice again the double turnstile $\vDash$, in contrast to the single turnstile $\vdash$ for derivations.

In the quantificational case, *demonstrating* semantic validity is problematic. In the sentential case, we could simply *list* all the ways a sentential interpretation could make basic sentences T or F. In the quantificational case, it is not possible to list all interpretations. Consider just interpretations with universe $\mathbb{N}$: the interpretation of a one-place relation symbol $\mathcal{R}$ might be $\{1\}$ or $\{2\}$ or $\{3\}$ or…; it might be $\{1, 2\}$ or $\{1, 3\}$ or $\{1, 3, 5, \ldots\}$, or whatever. There are infinitely many options for this one relation symbol—and so at least as many for quantificational interpretations in general. Similarly, when the universe is so large, by our methods, we cannot calculate even *satisfaction* and *truth* in arbitrary cases—for quantifiers would have an infinite number

[5] In the quantificational case, 'tautology' may be differently defined. Many authors restrict tautologies to formulas whose form is *sententially* valid. On this account, $Fx \to Fx$ with sentential form $\mathcal{P} \to \mathcal{P}$ is a tautology, while $\forall x Fx \to Fx$ with form $\mathcal{P} \to \mathcal{Q}$ is not. As we shall see, however, both $\vDash Fx \to Fx$ and $\vDash \forall x Fx \to Fx$—so that, on this alternative account, tautologies are a proper subset of quantificationally valid formulas.

of branches. One might begin to suspect that there is no way to demonstrate semantic validity in the quantificational case. There is a way. And we respond to this concern in Chapter 7.

For now, though, we rest content with demonstrating *invalidity*. To show that an argument is invalid, we do not need to consider all possible interpretations; it is enough to find one interpretation on which the premises are true and the conclusion is not. (Compare the invalidity test from Chapter 1 and "shortcut" truth tables in this chapter.) An argument is quantificationally valid just in case there is no I on which its premises are true and its conclusion is not true. So to show that an argument is not quantificationally valid, it is sufficient to produce an interpretation that violates this condition—an interpretation on which its premises are true and conclusion is not. And in some cases, including ones considered below, this can be done by very simple interpretations. This should be enough at least to let us see *how* the definitions work, and we postpone the larger question about showing quantificational validity to later.

For now, then, our idea is to produce an interpretation, and then to use trees in order to show that the interpretation makes premises true but the conclusion not. Thus, for example, for $\mathcal{L}_q$ we can show that $\sim\forall x Px \nvDash \sim Pa$—that an argument with premise $\sim\forall x Px$ and conclusion $\sim Pa$ is not quantificationally valid. To see this, consider an I with $\mathsf{U} = \{1, 2\}$, $\mathsf{I}[P] = \{1\}$, and $\mathsf{I}[a] = 1$. Then $\sim\forall x Px$ is T on I.

(AK)

$$\begin{array}{llllll}
1 & & 2 & & 3 & 4 \\
 & & & & \mathsf{I}_{\mathsf{d}(x|1)}[Px]^{(\mathsf{S})} & x^{[1]} \\
\mathsf{I}_{\mathsf{d}}[\sim\forall x Px]^{(\mathsf{S})} & \sim & \mathsf{I}_{\mathsf{d}}[\forall x Px]^{(\mathsf{N})} & \forall x & & \\
 & & & & \mathsf{I}_{\mathsf{d}(x|2)}[Px]^{(\mathsf{N})} & x^{[2]}
\end{array}$$

$\sim\forall x Px$ is satisfied with this d for I; and since it is a sentence it is satisfied with any d for I. So by TI it is true on I. But $\sim Pa$ is not true on this I.

$$\begin{array}{llll}
1 & & 2 & 3 \\
\mathsf{I}_{\mathsf{d}}[\sim Pa]^{(\mathsf{N})} & \sim & \mathsf{I}_{\mathsf{d}}[Pa]^{(\mathsf{S})} & a^{[1]}
\end{array}$$

From the tree, $\mathsf{I}_{\mathsf{d}}[\sim Pa] = \mathsf{N}$; and since there an assignment on which it is not satisfied, by TI, $\mathsf{I}[\sim Pa] \neq \mathsf{T}$. So there is an interpretation on which the premise is true and the conclusion is not. So by QV, $\sim\forall x Px \nvDash \sim Pa$, and the argument is not quantificationally valid. Notice that it is sufficient to show that the conclusion is not true—which is not always the same as showing that the conclusion is false.

Here is another example. We show that $\sim\forall x \sim Px, \sim\forall x \sim Qx \nvDash \forall y(Py \to Qy)$. In general, to show that an argument is not quantificationally valid, you want to think "backward" to see what kind of interpretation you need to make the premises true but the conclusion not true. In this case, to make the conclusion false, we need something that is P but not Q; the premises are true if something is P and something Q. One way to do this is with an I that has $\mathsf{U} = \{1, 2\}$ where $\mathsf{I}[P] = \{1\}$ and $\mathsf{I}[Q] = \{2\}$. Then the premises are true.

(AL)

$$\begin{array}{lllll}
1 & 2 & 3 & 4 & 5 \\
 & & \mathsf{I}_{\mathsf{d}(x|1)}[\sim Px]^{(\mathsf{N})} \;\sim & \mathsf{I}_{\mathsf{d}(x|1)}[Px]^{(\mathsf{S})} & x^{[1]} \\
\mathsf{I}_{\mathsf{d}}[\sim\forall x\sim Px]^{(\mathsf{S})} \;\sim & \mathsf{I}_{\mathsf{d}}[\forall x\sim Px]^{(\mathsf{N})} \;\forall x & & & \\
 & & \mathsf{I}_{\mathsf{d}(x|2)}[\sim Px]^{(\mathsf{S})} \;\sim & \mathsf{I}_{\mathsf{d}(x|2)}[Px]^{(\mathsf{N})} & x^{[2]}
\end{array}$$

$$\begin{array}{lllll}
 & & \mathsf{I}_{\mathsf{d}(x|1)}[\sim Qx]^{(\mathsf{S})} \;\sim & \mathsf{I}_{\mathsf{d}(x|1)}[Qx]^{(\mathsf{N})} & x^{[1]} \\
\mathsf{I}_{\mathsf{d}}[\sim\forall x\sim Qx]^{(\mathsf{S})} \;\sim & \mathsf{I}_{\mathsf{d}}[\forall x\sim Qx]^{(\mathsf{N})} \;\forall x & & & \\
 & & \mathsf{I}_{\mathsf{d}(x|2)}[\sim Qx]^{(\mathsf{N})} \;\sim & \mathsf{I}_{\mathsf{d}(x|2)}[Qx]^{(\mathsf{S})} & x^{[2]}
\end{array}$$

To make $\sim\forall x\sim Px$ true, we require that there is at least one thing in $\mathsf{I}[P]$. We accomplish this by putting 1 in its interpretation. This makes the top branch at stage (4) S; this makes the top branch at (3) N; so the quantifier at (2) is N and the formula at (1) comes out S. Since it is a sentence and satisfied on the arbitrary assignment, it is true. $\sim\forall x\sim Qx$ is true for related reasons. For it to be true, we require at least one thing in $\mathsf{I}[Q]$. This is accomplished by putting 2 in its interpretation. But this interpretation does not make the conclusion true.

$$\begin{array}{llll}
1 & 2 & 3 & 4 \\
 & & \mathsf{I}_{\mathsf{d}(y|1)}[Py]^{(\mathsf{S})} & y^{[1]} \\
 & \mathsf{I}_{\mathsf{d}(y|1)}[Py \to Qy]^{(\mathsf{N})} \;\to & & \\
 & & \mathsf{I}_{\mathsf{d}(y|1)}[Qy]^{(\mathsf{N})} & y^{[1]} \\
\mathsf{I}_{\mathsf{d}}[\forall y(Py \to Qy)]^{(\mathsf{N})} \;\forall y & & & \\
 & & \mathsf{I}_{\mathsf{d}(y|2)}[Py]^{(\mathsf{N})} & y^{[2]} \\
 & \mathsf{I}_{\mathsf{d}(y|2)}[Py \to Qy]^{(\mathsf{S})} \;\to & & \\
 & & \mathsf{I}_{\mathsf{d}(y|2)}[Qy]^{(\mathsf{S})} & y^{[2]}
\end{array}$$

The conclusion is not satisfied so long as something is in $\mathsf{I}[P]$ but not in $\mathsf{I}[Q]$. We accomplish this by making the thing in the interpretation of P *different* from the thing in the interpretation of Q. Since 1 is in $\mathsf{I}[P]$ but not in $\mathsf{I}[Q]$, there is an S/N pair at (3), so that the top branch at (2) is N and the formula at (1) is N. Since the formula is not satisfied, by TI it is not true. And since there is an interpretation on which the premises are true and the conclusion is not, by QV, the argument is not quantificationally valid.

To show that an argument is not quantificationally valid it is to your advantage to think of simple interpretations. Remember that U need only be non-empty. So it will often do to work with universes that have just one or two members. And the interpretation of a relation symbol might even be empty. It is often convenient to let the universe be some set of integers. If there is any interpretation that demonstrates invalidity, there is sure to be one whose universe is some set of integers—but we will get to this in Part III.

E4.15. For language $\mathcal{L}_q$ consider an interpretation I such that U = {1, 2}, and

$$\mathsf{I} \quad \begin{array}{l} \mathsf{I}[a] = 1 \\ \mathsf{I}[b] = 2 \\ \mathsf{I}[A] = \mathrm{T} \\ \mathsf{I}[P^1] = \{1\} \\ \mathsf{I}[f^1] = \{\langle 1, 2\rangle, \langle 2, 1\rangle\} \end{array}$$

Use interpretation I and trees to show that (a) below is not quantificationally valid. Then demonstrate that each of the others is invalid by an interpretation I* that modifies just one assignment (line) from interpretation I. Hint: If you are having trouble finding the appropriate modified interpretation, try working out the trees on I, and think about a change to the interpretation that would have the result you want.

a. $Pa \nvDash \forall x Px$

b. $\forall x Px \nvDash \sim Pa$

c. $\sim(Pa \to \sim Pb) \nvDash \forall x Px$

*d. $\forall x Pf^1x \nvDash \forall x Px$

e. $\forall x Px \to A \nvDash \forall x(Px \to A)$

E4.16. Find interpretations and use trees to demonstrate each of the following. Be sure to explain why your interpretations and trees have the desired result.

*a. $\forall x(Qx \to Px) \nvDash \forall x(Px \to Qx)$

b. $\forall x(Px \to Qx), \forall x(Rx \to \sim Px) \nvDash \forall y(Ry \to Qy)$

c. $\sim\forall x Px \nvDash \sim\forall x \sim Px$

*d. $\sim\forall x Px \nvDash \forall x \sim Px$

e. $\forall x Px \to \forall x Qx, Qb \nvDash Pa \to \forall x Qx$

f. $\sim(A \to \forall x Px) \nvDash \forall x(A \to \sim Px)$

g. $\forall x(Px \to Qx), \sim Qa \nvDash \forall x \sim Px$

*h. $\sim\forall y \forall x Rxy \nvDash \forall x \sim\forall y Rxy$

i. $\forall x \forall y(Rxy \to Ryx), \forall x \sim\forall y \sim Rxy \nvDash \forall x Rxx$

j. $\forall x \forall y[y = f^1x \to \sim(x = f^1y)] \nvDash \forall x(Px \to Pf^1x)$

4.2.5 Abbreviations

Finally, we turn to applications for abbreviations. Consider first a tree for $(\mathscr{P} \wedge \mathscr{Q})$, that is for $\sim(\mathscr{P} \rightarrow \sim\mathscr{Q})$.

$$
\text{(AM)}\quad
\begin{array}{ccccccc}
1 & & 2 & & 3 & & 4 \\
 & & & & \mathsf{I_d}[\mathscr{P}] & & \\
\underline{\mathsf{I_d}[\sim(\mathscr{P} \rightarrow \sim\mathscr{Q})]} & \sim & \underline{\mathsf{I_d}[\mathscr{P} \rightarrow \sim\mathscr{Q}]} & \rightarrow & & & \\
 & & & & \mathsf{I_d}[\sim\mathscr{Q}] & \sim & \underline{\mathsf{I_d}[\mathscr{Q}]}
\end{array}
$$

The formula at (1) is satisfied iff the formula at (2) is not. But the formula at (2) is not satisfied iff the top at (3) is satisfied and the bottom is not satisfied. And the bottom at (3) is not satisfied iff the formula at (4) is satisfied. So the formula at (1) is satisfied iff $\mathscr{P}$ is satisfied and $\mathscr{Q}$ is satisfied. The only way for $(\mathscr{P} \wedge \mathscr{Q})$ to be satisfied on some I and d is for $\mathscr{P}$ and $\mathscr{Q}$ both to be satisfied on that I and d. If either $\mathscr{P}$ or $\mathscr{Q}$ is not satisfied, then $(\mathscr{P} \wedge \mathscr{Q})$ is not satisfied. Reasoning similarly for $\vee$, $\leftrightarrow$, and $\exists$, we get the following derived branch conditions:

$$
\text{B}'(\wedge)\quad \underline{\mathsf{I_d}[(\mathscr{P} \wedge \mathscr{Q})]} \;\wedge\; \begin{array}{|l} \mathsf{I_d}[\mathscr{P}] \\ \hline \mathsf{I_d}[\mathscr{Q}] \\ \hline \end{array}
$$

the trunk is S iff both branches are S

$$
\text{B}'(\vee)\quad \underline{\mathsf{I_d}[(\mathscr{P} \vee \mathscr{Q})]} \;\vee\; \begin{array}{|l} \mathsf{I_d}[\mathscr{P}] \\ \hline \mathsf{I_d}[\mathscr{Q}] \\ \hline \end{array}
$$

the trunk is S iff at least one branch is S

$$
\text{B}'(\leftrightarrow)\quad \underline{\mathsf{I_d}[(\mathscr{P} \leftrightarrow \mathscr{Q})]} \;\leftrightarrow\; \begin{array}{|l} \mathsf{I_d}[\mathscr{P}] \\ \hline \mathsf{I_d}[\mathscr{Q}] \\ \hline \end{array}
$$

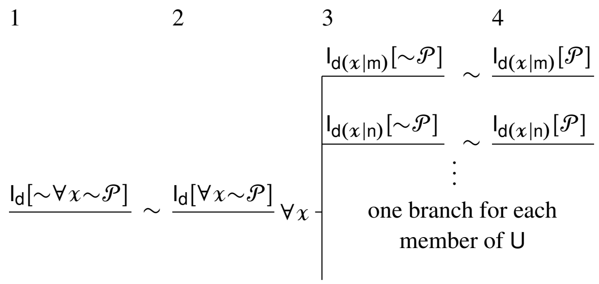
the trunk is S iff both branches are S or both are N

$$
\text{B}'(\exists)\quad \underline{\mathsf{I_d}[\exists x\mathscr{P}]} \;\exists x\; \begin{array}{|l} \mathsf{I_{d(x|m)}}[\mathscr{P}] \\ \hline \mathsf{I_{d(x|n)}}[\mathscr{P}] \\ \hline \vdots \\ \text{one branch for} \\ \text{each member} \\ \text{of } \mathsf{U} \end{array}
$$

the trunk is S iff at least one branch is S

The conditions for $\wedge$, $\vee$, and $\leftrightarrow$ work like ones from the sentential case. For $\exists$, consider a tree for $\sim\forall x\sim\mathscr{P}$, that is for $\exists x\mathscr{P}$.

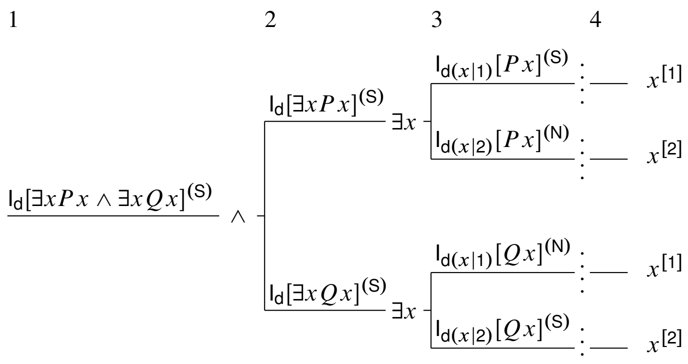
(AN)



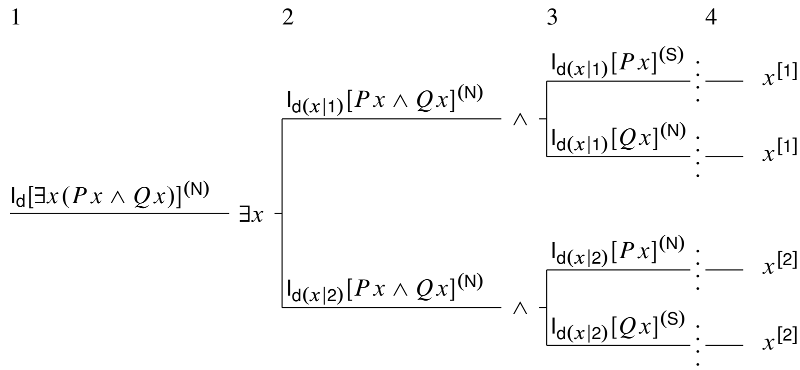
The formula at (1) is satisfied iff the formula at (2) is not. But the formula at (2) is not satisfied iff at least one of the branches at (3) is not satisfied. And for a branch at (3) to be not satisfied, the corresponding branch at (4) has to be satisfied. So $\exists x\mathscr{P}$ is satisfied on I with assignment d iff for some $\mathsf{o} \in \mathsf{U}$, $\mathscr{P}$ is satisfied on I with $\mathsf{d}(x|\mathsf{o})$; if there is no such $\mathsf{o} \in \mathsf{U}$, then $\exists x\mathscr{P}$ is N on I with d.

Given derived branch conditions, we can work directly with abbreviations in trees for determining satisfaction and truth. And the definition of validity applies in the usual way. Thus, for example, $\exists xPx \wedge \exists xQx \not\models \exists x(Px \wedge Qx)$. To see this, consider an I with $\mathsf{U} = \{1, 2\}$, $\mathsf{I}[P] = \{1\}$, and $\mathsf{I}[Q] = \{2\}$. The premise, $\exists xPx \wedge \exists xQx$ is true on I. To see this, we construct a tree, making use of derived clauses as necessary.

(AO)



The existentials are satisfied because at least one branch is satisfied, and the conjunction because both branches are satisfied, according to derived conditions B$'(\exists)$ and B$'(\wedge)$. So the formula is satisfied, and because it is a sentence, is true. But the conclusion, $\exists x(Px \wedge Qx)$ is not true.



The conjunctions at (2) are not satisfied, in each case because not both branches at (3) are satisfied. And the existential at (1) requires that at least one branch at (2) be satisfied; since none is satisfied, the main formula $\exists x(Px \wedge Qx)$ is not satisfied, and so by TI not true. Since there is an interpretation on which the premise is true and the conclusion is not, by QV, $\exists xPx \wedge \exists xQx \nvDash \exists x(Px \wedge Qx)$. As we will see in the next chapter, the intuitive point is simple: just because something is P and something is Q, it does not follow that something is both P and Q. And this is just what our interpretation I illustrates.

E4.17. Produce interpretations to demonstrate each of the following. Use trees, with derived clauses as necessary, to demonstrate your results. Be sure to explain why your interpretations and trees have the results they do. Hint: In some cases, it may be convenient to produce only that part of the tree which is necessary for the result.

*a. $\exists xPx \nvDash \exists y(Py \wedge Qy)$

*b. $\exists xPx \nvDash \forall yPy$

c. $\exists xPx \nvDash \exists yPf^1y$

d. $Pa \to \forall xQx \nvDash \exists xPx \to \forall xQx$

e. $\forall x\exists yRxy \nvDash \exists y\forall xRxy$

f. $\forall xPx \leftrightarrow \forall xQx, \exists x\exists y(Px \wedge Qy) \nvDash \exists y(Py \leftrightarrow Qy)$

*g. $\forall x(\exists yRxy \leftrightarrow {\sim}A) \nvDash \exists xRxx \vee A$

h. $\exists x(Px \wedge \exists yQy) \nvDash \exists x\forall y(Px \wedge Qy)$

i. $\forall x\exists y(Px \leftrightarrow Qxy), \exists xPx \nvDash \forall x\exists yQxy$

j. $\exists x\exists y{\sim}(x = y) \nvDash \forall x\forall y\exists z({\sim}(x = z) \wedge {\sim}(y = z))$

Semantics Quick Reference (quantificational)

For a quantificational language $\mathcal{L}$, a *quantificational interpretation* I consists of a nonempty set U, the *universe* of the interpretation, along with,

QI (s) An assignment of a truth value $\mathsf{I}[\mathcal{S}]$ to each sentence letter $\mathcal{S}$ of $\mathcal{L}$.

(c) An assignment of a member $\mathsf{I}[c]$ of U to each constant symbol c of $\mathcal{L}$.

(r) An assignment of an n-place relation $\mathsf{I}[\mathcal{R}^n]$ on U to each n-place relation symbol $\mathcal{R}^n$ of $\mathcal{L}$, where $\mathsf{I}[=]$ is always assigned $\{\langle \mathsf{o}, \mathsf{o}\rangle \mid \mathsf{o} \in \mathsf{U}\}$.

(f) An assignment of a total n-place function $\mathsf{I}[h^n]$ from U^n to U to each n-place function symbol h^n of $\mathcal{L}$.

Given a language $\mathcal{L}$ and interpretation I, a *variable assignment* d is a total function from the variables of $\mathcal{L}$ to objects in the universe U. Then for any interpretation I and variable assignment d,

TA (c) If c is a constant, then $\mathsf{I}_\mathsf{d}[c] = \mathsf{I}[c]$.

(v) If x is a variable, then $\mathsf{I}_\mathsf{d}[x] = \mathsf{d}[x]$.

(f) If h^n is a function symbol and $t_1 \dots t_n$ are terms, then $\mathsf{I}_\mathsf{d}[h^n t_1 \dots t_n] = \mathsf{I}[h^n]\langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle$.

SF (s) If $\mathcal{S}$ is a sentence letter, then $\mathsf{I}_\mathsf{d}[\mathcal{S}] = \mathsf{S}$ iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; otherwise $\mathsf{I}_\mathsf{d}[\mathcal{S}] = \mathsf{N}$.

(r) If $\mathcal{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are terms, then $\mathsf{I}_\mathsf{d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; otherwise $\mathsf{I}_\mathsf{d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{N}$.

($\sim$) If $\mathcal{P}$ is a formula, then $\mathsf{I}_\mathsf{d}[\sim\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{N}$; otherwise $\mathsf{I}_\mathsf{d}[\sim\mathcal{P}] = \mathsf{N}$.

($\to$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $\mathsf{I}_\mathsf{d}[(\mathcal{P} \to \mathcal{Q})] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{N}$ or $\mathsf{I}_\mathsf{d}[\mathcal{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I}_\mathsf{d}[(\mathcal{P} \to \mathcal{Q})] = \mathsf{N}$.

($\forall$) If $\mathcal{P}$ is a formula and x is a variable, then $\mathsf{I}_\mathsf{d}[\forall x\mathcal{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$; otherwise $\mathsf{I}_\mathsf{d}[\forall x\mathcal{P}] = \mathsf{N}$.

SF′ ($\wedge$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $\mathsf{I}_\mathsf{d}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{S}$ and $\mathsf{I}_\mathsf{d}[\mathcal{Q}] = \mathsf{S}$; otherwise $\mathsf{I}_\mathsf{d}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{N}$.

($\vee$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $\mathsf{I}_\mathsf{d}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{S}$ or $\mathsf{I}_\mathsf{d}[\mathcal{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I}_\mathsf{d}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{N}$.

($\leftrightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $\mathsf{I}_\mathsf{d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{I}_\mathsf{d}[\mathcal{Q}]$; otherwise $\mathsf{I}_\mathsf{d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{N}$.

($\exists$) If $\mathcal{P}$ is a formula and x is a variable, then $\mathsf{I}_\mathsf{d}[\exists x\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$; otherwise $\mathsf{I}_\mathsf{d}[\exists x\mathcal{P}] = \mathsf{N}$.

TI A formula $\mathcal{P}$ is *true* on an interpretation I iff with any d for I, $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{S}$. $\mathcal{P}$ is *false* on I iff with any d for I, $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{N}$.

QV Γ *quantificationally entails* $\mathcal{P}$ ($\Gamma \vDash \mathcal{P}$) iff there is no quantificational interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$.

If $\Gamma \vDash \mathcal{P}$, an argument whose premises are the members of Γ and conclusion is $\mathcal{P}$ is *quantificationally valid*.

E4.18. Produce interpretations to demonstrate each of the following (now in $\mathcal{L}_{NT}^{<}$). Use trees to demonstrate your results. Be sure to explain why your interpretations and trees have the results they do. Hint: When there are no premises, all you need is an interpretation where the expression is not true. You need not use the *standard* interpretation. Again, in some cases, it may be convenient to produce only that part of the tree which is necessary for the result.

a. $\not\models \forall x(x < Sx)$

b. $\not\models (S\emptyset + S\emptyset) = SS\emptyset$

c. $\not\models \exists x{\sim}((x \times x) = x)$

*d. $\not\models \forall x \forall y({\sim}(x = y) \to (x < y \vee y < x))$

e. $\not\models \forall x \forall y \forall z((x < y \wedge y < z) \to x < z)$

E4.19. On page 129 we say that reasoning similar to that for $\wedge$ results in other branch conditions. Give the reasoning similar to that for $\wedge$ and $\exists$ to demonstrate from trees the conditions B($\vee$) and B($\leftrightarrow$).

E4.20. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Quantificational interpretations.

b. Term assignments, satisfaction, and truth.

c. Quantificational validity.

Chapter 5

Translation

We have introduced logical validity from Chapter 1, along with validity in an axiomatic derivation system from Chapter 3, and semantic validity from Chapter 4. But logical validity applies to arguments expressed in ordinary language, where the other notions apply to arguments expressed in a formal language. Our guiding idea has been to *use* the formal notions with application to ordinary arguments via *translation* from ordinary language to formal language. It is to the translation task that we now turn. After some general remarks in section 5.1, we will take up issues specific to the sentential (section 5.2), and then the quantificational case (section 5.3).

5.1 General

As speakers of ordinary languages (at least English for those reading this book) we presumably have some understanding of the conditions under which ordinary language sentences are true and false. Similarly, we now have an understanding of the conditions under which sentences of our formal languages are true and false. This puts us in a position to recognize when the conditions under which ordinary sentences are true are the *same* as the conditions under which formal sentences are true. And that is what we want: Our goal is to translate the premises and conclusion of ordinary arguments into formal expressions that are true when the ordinary sentences are true, and false when the ordinary sentences are false. Insofar as validity has to do with conditions under which sentences are true and false, our translations should thus be an adequate basis for evaluations of validity.

We can put this point with greater precision. Formal sentences are true and false relative to interpretations. As we have seen, many different interpretations of a formal language are possible. In the sentential case, any sentence letter can be true or false—so that there are 2^n ways to interpret any n sentence letters. When we specify an interpretation, we select just one of the many available options. Thus, for example, we might set $\mathsf{I}[B] = \mathsf{T}$ and $\mathsf{I}[M] = \mathsf{F}$. But we might also specify an interpretation as follows:

(A)
B: Barack is happy
M: Michelle is happy

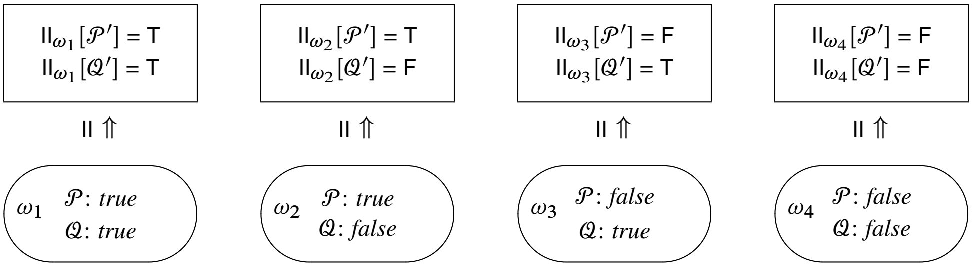
intending B to take the same truth value as 'Barack is happy' and M the same as 'Michelle is happy'. In this case, the single specification might result in different interpretations, depending on how the world is: depending on how Barack and Michelle are, the interpretation of B might be true or false, and similarly for M. That is, specification (A) is really a *function* from ways the world could be (from maximal and consistent stories) to interpretations of the sentence letters. It results in a specific or *intended* interpretation relative to any way the world could be. Thus, where ω ranges over ways the world could be, (A) is a function II which results in an intended interpretation II_ω corresponding to any such way—thus $\mathsf{II}_\omega[B]$ is T if Barack is happy at ω and F if he is not.

When we set out to translate some ordinary sentences into a formal language, we always begin by specifying an interpretation function. In the sentential case, this typically takes the form of a specification like (A). Then for ω any way the world can be, there is an intended interpretation II_ω of the formal language. Given this, for an ordinary sentence $\mathcal{A}$, the aim is to produce a formal counterpart $\mathcal{A}'$ such that for any ω, $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff $\mathcal{A}$ is true at world ω. This is the content of saying we want to produce formal expressions that "are true when the ordinary sentences are true, and false when the ordinary sentences are false." In fact, we can turn this into a *criterion of goodness* for translation:

CG Given some ordinary sentence $\mathcal{A}$, a translation consisting of an interpretation function II and formal sentence $\mathcal{A}'$ is *good* iff it captures available sentential/quantificational structure and, where ω is any way the world can be, $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff $\mathcal{A}$ is true at ω.

If there is a collection of sentences, a translation consisting of an II and some formal sentences is *good* only if each ordinary $\mathcal{A}$ of the collection has a formal $\mathcal{A}'$ where for any ω, $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff $\mathcal{A}$ is true at ω. Set aside the question of what it is to capture "available" sentential/quantificational structure, this will emerge as we proceed. For now, the point is simply that we want formal sentences to be true on intended interpretations when originals are true at corresponding worlds, and false on intended interpretations when originals are false. CG says that this correspondence is necessary for goodness. And, supposing that sufficient structure is reflected, according to CG such correspondence is sufficient as well.

The situation might be pictured as follows. There is a specification II which results in an intended interpretation corresponding to any way the world can be. And corresponding to ordinary sentences $\mathcal{P}$ and $\mathcal{Q}$ there are formal sentences $\mathcal{P}'$ and $\mathcal{Q}'$. Then with oval for worlds and box for interpretations built on them,



The interpretation function results in an intended interpretation corresponding to each world. The intended interpretations make assignments to basic vocabulary (in the sentential case, to sentence letters). Then a translation is good only if no matter how the world is, the values of $\mathcal{P}'$ and $\mathcal{Q}'$ on the intended interpretations match the values of the ordinary $\mathcal{P}$ and $\mathcal{Q}$ at the corresponding worlds or stories.

The premises and conclusion of an argument are some sentences. So the translation of an argument is *good* iff the translation of the sentences that are its premises and conclusion is good. And good translations of arguments put us in a position to *use* our machinery to evaluate questions of validity. Of course, so far, this is an abstract description of what we are about to do. But it should give some orientation, and help you understand what is accomplished as we proceed.

5.2 Sentential

We begin with the sentential case. Again, the general idea is to *recognize* when the conditions under which ordinary sentences are true are the *same* as the conditions under which formal ones are true. Surprisingly perhaps, the hardest part is on the side of recognizing truth conditions in ordinary language. With this in mind, let us begin with some definitions whose application is to expressions of *ordinary* language; after that, we will turn to a procedure for translation, and to discussion of particular operators.

5.2.1 Some Definitions

In this section, we introduce a series of definitions whose application is to ordinary language. These definitions are not meant to compete with anything you have learned in English class. Rather they are specific to our purposes. With the definitions under our belt, we will be able to say with some precision what we want to do.

First, a *declarative sentence* is a sentence which has a truth value—a sentence that is either true or false. ‘Snow is white’ and ‘Snow is green’ are declarative sentences—the first true and the second false. ‘Study harder!’ and ‘Why study?’ are sentences, but not declarative sentences. Given this, a *sentential operator* is an expression containing “blanks” such that when the blanks are filled with declarative sentences, the result is a declarative sentence. In ordinary speech and writing, such blanks do not typically appear (!) however punctuation and expression typically fill the same role. Examples are,

John believes that ____

John heard that ____

it is not the case that ____

____ and ____

'John believes that snow is white', 'John believes that snow is green', and 'John believes that dogs fly' are all sentences—some more plausibly true than others. Still, 'Snow is white', 'Snow is green', and 'Dogs fly' are all declarative sentences, and when we put them in the blank of 'John believes that ____' the result is a declarative sentence, where the same would be so for any declarative sentence in the blank; so 'John believes that ____' is a sentential operator. Similarly, 'Snow is white and dogs fly' is a declarative sentence—a false one, since dogs do not fly. And, so long as we put declarative sentences in the blanks of '____ and ____' the result is always a declarative sentence. So '____ and ____' is a sentential operator. In contrast,

when ____

____ is white ____

are not sentential operators. Though 'Snow is white' is a declarative sentence, 'when snow is white' is an adverbial clause, not a declarative sentence. And, though 'Dogs fly' and 'Snow is green' are declarative sentences, 'dogs fly is white snow is green' is ungrammatical nonsense. If you can think of even one case where putting declarative sentences in the blanks of an expression does not result in a declarative sentence, then the expression is not a sentential operator. So these are not sentential operators.

Now, as in these examples, we can think of some declarative sentences as generated by the combination of sentential operators with other declarative sentences. Declarative sentences generated from other sentences by means of sentential operators are *compound;* all others are *simple*. Thus, for example, 'Bob likes Mary' and 'Socrates is wise' are simple sentences, they do not have a declarative sentence in the blank of any operator. In contrast, 'John believes that Bob likes Mary' and 'Jim heard that John believes that Bob likes Mary' are compound. The first has a simple sentence in the blank of 'John believes that ____'. The second puts a compound in the blank of 'Jim heard that ____'.

For cases like these, the *main operator* of a compound sentence is that operator not in the blank of any other operator. The main operator of 'John believes that Bob likes Mary' is 'John believes that ____'. And the main operator of 'Jim heard that John believes that Bob likes Mary' is 'Jim heard that ____'. The main operator of

'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob' is '____ and ____', for that is the operator not in the blank of any other. Notice that the main operator of a sentence need not be the *first* operator in the sentence. Observe also that operator structure may not be obvious. Thus, for example, 'Jim heard that Bob likes Sue and Sue likes Jim' is capable of different interpretations. It might be, 'Jim heard that Bob likes Sue and Sue likes Jim' with main operator, 'Jim heard that ____' and the compound, 'Bob likes Sue and Sue likes Jim' in its blank. But it might be 'Jim heard that Bob likes Sue and Sue likes Jim' with main operator, '____ and ____'. The question is what Jim heard, and what the 'and' joins. As suggested above, punctuation and expression often serve in ordinary language to disambiguate confusing cases. These questions of interpretation are not peculiar to our purposes! Rather they are the ordinary questions that might be asked about meaning. The underline structure serves to disambiguate claims, to make it very clear how the operators apply.

We shall want to identify the operator structure of sentences. When faced with a compound sentence, the best approach is start with the whole, rather than the parts. So begin with blank(s) for the main operator. Thus, as we have seen, the main operator of 'It is not the case that Bob likes Sue, and it is not the case that Sue likes Bob' is '____ and ____'. So begin with lines for that operator, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob' (leaving space for lines above). Now focus on the sentence in one of the blanks, say the left; that sentence, 'It is not the case that Bob likes Sue' is a compound with main operator, 'it is not the case that ____'. So add the underline for that operator, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob'. The sentence in the blank of 'it is not the case that ____' is simple. So turn to the sentence in the right blank of the main operator. That sentence has main operator 'it is not the case that ____'. So add an underline. In this way we end up with, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob'. Thus a complex problem is reduced to ones that are progressively simpler. Perhaps this problem was obvious from the start. But this approach will serve you well as problems get more complex!

We come finally to the key notion of a *truth functional* operator. A sentential operator is *truth functional* iff any compound generated by it has its truth value wholly determined by the truth values of the sentences in its blanks. We will say that the truth value of a compound is "determined" by the truth values of sentences in blanks just in case there is no way to switch the truth value of the whole while keeping truth values of sentences in the blanks constant.

This leads to a test for truth functionality: We show that an operator is *not* truth functional, if we come up with some situation(s) where truth values of sentences in the blanks are the same, but the truth value of the resulting compounds are not. To take a simple case, consider 'John believes that ____'. If things are pretty much as in the actual world, 'There is a Santa' and 'Dogs fly' are both false. But if John is a small child it may be that,

(B)	John believes that	there is a Santa dogs fly
	T/F	***F***

John believes there is a Santa, but knows perfectly well that dogs do not fly. So the compound is true with one in the blank, and false with the other. Thus the truth value of the compound is not wholly determined by the truth value of the sentence in the blank. We have found a situation where sentences with the same truth value in the blank result in a different truth value for the whole. Thus 'John believes that ____' is not truth functional. We might make the same point with a pair of sentences that are true, say 'Dogs bark' and 'There are infinitely many prime numbers' (be clear in your mind about how this works).

As a second example, consider, '____ because ____'. Suppose 'You are happy', 'You understand the material', 'There are fish in the sea', and 'You woke up this morning' are all true.

(C)	You are happy There are fish in the sea	because	you understand the material you woke up this morning
	T	***T/F***	***T***

Still, it is natural to think that the truth value of the compound, 'You are happy because you understand the material' may be true, while 'There are fish in the sea because you woke up this morning' is false. For perhaps understanding the material makes you happy, but the fish in the sea have nothing to do with your waking up. Thus there are consistent situations or stories where sentences in the blanks have the same truth values, but the compounds do not. Thus, by the definition, '____ because ____' is not a truth functional operator. To show that an operator is not truth functional it is sufficient to produce some situation of this sort: where truth values for sentences in the blanks match, but truth values for the compounds do not. Observe that in order to meet this condition it would be sufficient to find, say, a case where sentences in the first blank remain T, sentences in the second remain F but the value of the whole flips from T to F. To show that an operator is not truth functional, any combination on which the blanks remain constant but the whole flips value will do.

To show that an operator is truth functional, we need to show that no such cases are possible. For this, we show *how* the truth value of what is in the blank determines the truth value of the whole. As an example, consider first,

	it is not the case that	____
(D)	***F***	T
	T	F

In this table, we represent the truth value of whatever is in the blank by the column under the blank, and the truth value for the whole by the column under the operator. If we put something true according to a consistent story into the blank, the resultant compound is sure to be false according to that story. Thus, for example, in the true story, 'Snow is white', '$2+2=4$', and 'Dogs bark' are all true; correspondingly, 'It is not the case that snow is white', 'It is not the case that $2+2=4$', and 'It is not the case that dogs bark' are all false. Similarly, if we put something false according to a story into the blank, the resultant compound is sure to be true according to the story. Thus, for example, in the true story, 'Snow is green' and '$2+2=3$' are both false. Correspondingly, 'It is not the case that snow is green' and 'It is not the case that $2+2=3$' are both true. It is no coincidence that the above table for 'it is not the case that ____' looks like the table for $\sim$. We will return to this point shortly.

For a second example of a truth functional operator, consider '____ and ____'. This seems to have table,

(E)

____	and	____
T	***T***	T
T	***F***	F
F	***F***	T
F	***F***	F

Consider a situation where Bob and Sue each love themselves, but hate each other. Then 'Bob loves Bob and Sue loves Sue' is true. But if at least one blank has a sentence that is false, the compound is false. Thus in that situation, 'Bob loves Bob and Sue loves Bob' is false; 'Bob loves Sue and Sue loves Sue' is false; and 'Bob loves Sue and Sue loves Bob' is false. For a compound, '____ and ____' to be true, the sentences in both blanks have to be true. And if they are both true, the compound is itself true. So the operator is truth functional. Again, it is no coincidence that the table looks so much like the table for $\wedge$. To show that an operator is truth functional, it is sufficient to produce the table that shows how the truth values of the compound are fixed by the truth values of the sentences in the blanks.

For an interesting sort of case, consider the operator 'according to every consistent story ____', and the following attempted table:

(F)

according to every consistent story	____
?	T
F	F

Say we put some sentence $\mathcal{P}$ that is false according to a consistent story into the blank. Then since $\mathcal{P}$ is false according to that very story, it is not the case that $\mathcal{P}$ according to every consistent story—and the compound is sure to be false. So we fill in the bottom row under the operator as above. So far, so good. But consider 'Dogs bark' and '$2+2=4$'. Both are true according to the true story. But only the second is true according to *every* consistent story—we can tell stories where 'Dogs bark' is true and where it is false, but '$2+2=4$' is true in every consistent story. So the compound is false with the first in the blank, true with the second. So 'according to every consistent story ____' is therefore *not* a truth functional operator. The truth value of the compound is not *wholly* determined by the truth value of the sentence in the blank. Similarly, it is natural to think that '____ because ____' is false whenever one of the sentences in its blanks is false. It cannot be true that $\mathcal{P}$ because $\mathcal{Q}$ if not-$\mathcal{P}$, and it cannot be true that $\mathscr{P}$ because $\mathscr{Q}$ if not-$\mathscr{Q}$. If you are not happy, then it cannot be that you are happy because you understand the material; and if you do not understand the material, it cannot be that you are happy because you understand the material. So far, then, the table for '____ because ____' is like the table for '____ and ____'.

(G)

____	because	____
T	?	T
T	***F***	F
F	***F***	T
F	***F***	F

However, as we saw at (C) above, in contrast to '____ and ____', compounds generated by '____ because ____' may or may not be true when sentences in the blanks are both true. So, although '____ and ____' is truth functional, '____ because ____' is not.

Thus the question is whether we can complete a table of the above sort: If there is a way to complete the table, the operator is truth functional. The test to show an operator is not truth functional simply finds some case to show that such a table cannot be completed.

E5.1. For each of the following, (i) say whether it is simple or compound. If the sentence is compound, (ii) use underlines to exhibit its operator structure, and (iii) say what is its main operator.

*a. Bob likes Mary.

*b. Jim believes that Bob likes Mary.

c. It is not the case that Bob likes Mary.

*d. Jane heard that it is not the case that Bob likes Mary.

e. Jane heard that Jim believes that it is not the case that Bob likes Mary.

f. Iron Man is strong, but it is not the case that Tony Stark is strong.

g. Iron Man fights for justice and Thor fights for justice, but it is not the case that Thanos fights for justice.

*h. Iron Man believes that Iron Man is stronger than Hulk and Iron Man is stronger than Hulk, but Hulk believes that Hulk is stronger than Iron Man and it is not the case that Hulk is stronger than Iron Man.

i. Thanos believes that genocide is good, but it is not the case that genocide is good; and Thanos is an evil being.

j. Iron Man believes that justice is good and Thor believes that justice is good, but it is not the case that Thanos believes that justice is good.

E5.2. Which of the following operators are truth functional and which are not? If the operator is truth functional, display the relevant table; if it is not, give cases that flip the value of the compound, with the value in the blanks constant. Explain your response.

*a. it is a fact that ____

b. Elmore believes that ____

*c. ____ but ____

d. according to some consistent story ____

e. although ____ , ____

*f. it is always the case that ____

g. sometimes it is the case that ____

h. ____ therefore ____

i. ____ however ____

j. either ____ or ____ (or both)

Definitions for Translation

DC A *declarative sentence* is a sentence which has a truth value.

SO A *sentential operator* is an expression containing "blanks" such that when the blanks are filled with declarative sentences, the result is a declarative sentence.

CS Declarative sentences generated from other sentences by means of sentential operators are *compound;* all others are *simple*.

MO The *main operator* of a compound sentence is that operator not in the blank of any other operator.

TF A sentential operator is *truth functional* iff any compound generated by it has its truth value wholly determined by the truth values of the sentences in its blanks.

To show that an operator is not truth functional it is sufficient to produce some situations where truth values for sentences in the blanks are constant, but truth values for the compounds are not.

To show that an operator is truth functional, it is sufficient to produce the table that shows how the truth values of the compound are fixed by truth values of the sentences in the blanks.

5.2.2 Parse Trees

We are now ready to outline a procedure for translation into our formal sentential language. In the end, you will often be able to see how translations should go and to write them down without going through all the official steps. However, the procedure should get you thinking in the right direction, and remain useful for complex cases. To translate some ordinary sentences $\mathcal{P}_1 \ldots \mathcal{P}_n$ the basic translation procedure is,

TP (1) Convert the ordinary $\mathcal{P}_1 \ldots \mathcal{P}_n$ into corresponding ordinary equivalents exposing truth functional and operator structure.

(2) Generate a "parse tree" for each of $\mathcal{P}_1 \ldots \mathcal{P}_n$ and specify the interpretation function II by assigning sentence letters to sentences at the bottom nodes.

(3) Using sentence letters from II and equivalent formal expressions, for each parse tree construct a parallel tree to generate formal $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ corresponding to $\mathcal{P}_1 \ldots \mathcal{P}_n$.

For now at least, the idea behind step (1) is simple: Sometimes all you need to do is expose operator structure by introducing underlines. In complex cases, this can be difficult! But we know how to do it. Sometimes, however, truth functional structure does not lie on the surface. Ordinary sentences are *equivalent* when they are true and false in exactly the same consistent stories. And we want ordinary equivalents exposing truth functional structure. Suppose $\mathcal{P}$ is a sentence of the sort,

(H) Bob is not happy

Is this a truth functional compound? Not officially. There is no declarative sentence in the blank of a sentential operator; so it is not compound; so it is not a truth functional compound. But one might think that (H) is short for,

(I) It is not the case that <u>Bob is happy</u>

which is a truth functional compound. At least (H) and (I) are equivalent in the sense that they are true and false in the same consistent stories. Similarly, 'Bob and Carol are happy' is not a compound of the sort we have described, with declarative sentences in the blanks of a sentential operator. However, it is a short step from this sentence to the equivalent, '<u>Bob is happy</u> and <u>Carol is happy</u>' which is an official truth functional compound. As we shall see, in some cases, this step can be more complex. But let us leave it at that for now.

Moving to step (2), in a *parse tree* we begin with sentences constructed as in step (1). If a sentence has a *truth functional* main operator, then it branches downward for the sentence(s) in its blanks. If these have truth functional main operators, they branch for the sentences in *their* blanks; and so forth, until sentences are simple or have non-truth functional main operators. Then given trees for each of $\mathcal{P}_1 \ldots \mathcal{P}_n$, construct the interpretation function II by assigning a distinct sentence letter to each distinct sentence at a bottom node.

Some simple examples should make this clear. Say we want to translate a collection of four sentences.

1. Bob is happy
2. Carol is not happy
3. Bob is healthy and Carol is not
4. Bob is happy and John believes that Carol is not healthy

The first is a simple sentence. Thus there is nothing to be done at step (1). And since there is no main operator, there is no branching and the sentence itself is a completed parse tree. The tree is just,

(J) Bob is happy

Insofar as the simple sentence is a complete branch of the tree, it counts as a bottom node of its tree. It is not yet assigned a sentence letter, so we assign it one. B_1: Bob is happy. We select this letter to remind us of the assignment.

As it stands, the second sentence is not a truth functional compound. Thus in the first stage, 'Carol is not happy' is expanded to the equivalent, 'It is not the case that Carol is happy'. In this case, there is a main operator; since it is truth functional, the tree has some structure.

(K)

It is not the case that Carol is happy

Carol is happy

The bottom node is simple, so the tree ends. 'Carol is happy' is not assigned a letter; so we assign it one. C_1: Carol is happy.

The third sentence is equivalent to, 'Bob is healthy and it is not the case that Carol is healthy'. Again, the operators are truth functional, and the result is a structured tree.

(L)

Bob is healthy and it is not the case that Carol is healthy

Bob is healthy | it is not the case that Carol is healthy

Carol is healthy

The main operator is truth functional. So there is a branch for each of the sentences in its blanks. Observe that underlines continue to reflect the structure of *these* sentences (so we "lift" the sentences from their blanks with structure intact). On the left, 'Bob is healthy' has no main operator, so it does not branch. On the right, 'it is not the case that Carol is healthy' has a truth functional main operator, and so branches. At bottom, we end up with 'Bob is healthy' and 'Carol is healthy'. Neither has a letter, so we assign them ones. B_2: Bob is healthy; C_2: Carol is healthy.

The final sentence is equivalent to, 'Bob is happy and John believes it is not the case that Carol is healthy'. It has a truth functional main operator. So there is a structured tree.

(M)

Bob is happy and John believes it is not the case that Carol is healthy

Bob is happy   John believes it is not the case that Carol is healthy

On the left, 'Bob is happy' is simple. On the right, 'John believes it is not the case that Carol is healthy' is compound. But its main operator is not truth functional. So *it does not branch*. We only branch for sentences in the blanks of truth functional main operators. Given this, we proceed in the usual way. 'Bob is happy' already has a letter. The other does not; so we give it one. J: John believes it is not the case that Carol is healthy.

And that is all. We have now compiled an interpretation function,

II B_1: Bob is happy

C_1: Carol is happy

B_2: Bob is healthy

C_2: Carol is healthy

J: John believes it is not the case that Carol is healthy

Of course, we might have chosen different letters. All that matters is that we have a distinct letter for each distinct sentence. For any way the world can be, our interpretation function yields an interpretation on which a sentence letter is true when its assigned sentence is true in that world, and false when its assigned sentence is false. In the last case, there is a compulsion to think that we can somehow get down to the simple sentence 'Carol is healthy'. But resist temptation! A non-truth functional operator "seals off" that upon which it operates, and forces us to treat the compound as a unit. We do not automatically assign sentence letters to simple sentences, but rather to parts that are *not* truth functional compounds. Simple sentences fit this description. But so do compounds with non-truth functional main operators.

E5.3. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences. Hint: Pay attention to punctuation as a guide to structure.

a. Bingo is spotted, and Spot can play bingo.

b. Bingo is not spotted, and Spot cannot play bingo.

c. Bingo is spotted, and believes that Spot cannot play bingo.

*d. It is not the case that: Bingo is spotted and Spot can play bingo.

e. It is not the case that: Bingo is not spotted and Spot cannot play bingo.

E5.4. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences.

*a. People have rights and dogs have rights, but rocks do not.

b. It is not the case that: rocks have rights, but people do not.

c. Aliens believe that rocks have rights, but it is not the case that people believe it.

d. Aliens landed in Roswell NM in 1947, and live underground but not in my backyard.

e. Rocks do not have rights and aliens do not have rights, but people and dogs do.

5.2.3 Formal Sentences

Now we are ready for step (3) of the translation procedure TP. Corresponding to each parse tree we construct a parallel tree using the interpretation function and then equivalent formal expressions to capture the force of ordinary truth functional operators. An ordinary truth functional operator has a table. Similarly, our formal expressions have tables. An ordinary truth functional operator is *equivalent* to some formal expression containing blanks just in case their tables are the same. Thus '$\sim$____' is equivalent to 'it is not the case that ____'. They are equivalent insofar as in each case, the whole has the opposite truth value of what is in the blank. Similarly, '____ $\wedge$ ____' is equivalent to '____ and ____'. In either case, when sentences in the blanks are both T the whole is T, and in other cases, the whole is F. Of course, the complex '$\sim$(____ $\rightarrow$ $\sim$____)' takes the same values as the '____ $\wedge$ ____' that abbreviates it. So different formal expressions may be equivalent to a given ordinary one.

To see how this works, let us return to the sample sentences from above. Again, the idea is to generate a parallel tree. The parallel tree has exactly one node corresponding to each node in the parse tree. We begin by *using* the sentence letters from our interpretation function for the bottom nodes. The case is particularly simple when the tree has no structure. 'Bob is happy' has a simple unstructured tree, and we assigned it a sentence letter directly. Thus our original and parallel trees are,

(N) Bob is happy B_1

So for a simple sentence, we simply read off the final translation from the interpretation function. So much for the first sentence.

As we have seen, the second sentence is equivalent to 'It is not the case that Carol is happy' with a parse tree as on the left below. We begin the parallel tree on the other side.

(O)

It is not the case that Carol is happy

|

Carol is happy C_1

We know how to translate the bottom node. But now we want to capture the force of the truth functional operator with some equivalent formal expression. For this, we need a formal expression containing blanks whose table mirrors the table for the sentential operator in question. In this case, '$\sim$____' works fine. That is, we have,

it is not the case that	____		$\sim$	____
F	T		***F***	T
T	F		***T***	F

In each case, when the expression in the blank is T, the whole is F, and when the expression in the blank is F, the whole is T. So '$\sim$____' is sufficient as a translation of 'it is not the case that ____'. Other formal expressions might do just as well. Thus, for example, we might go with, '$\sim\sim\sim$____'. The table for this is the same as the table for '$\sim$____'. But it is hard to see why we would do this with $\sim$ so close at hand. Now the idea is to apply the equivalent expression *to* the already translated expression from the blank. But this is easy to do. Thus we complete the parallel tree as follows:

It is not the case that Carol is happy $\sim C_1$

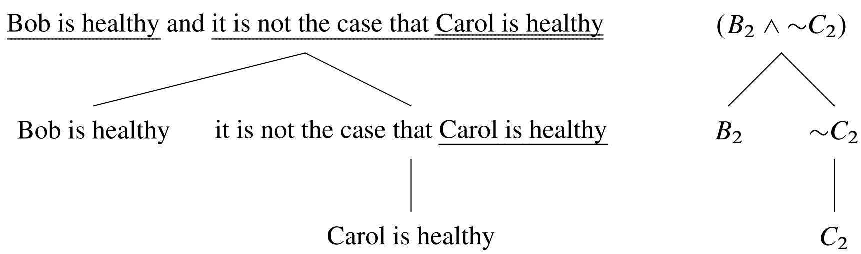
| |

Carol is happy C_1

The result is the completed translation, $\sim C_1$.

The third sentence has a parse tree as on the left below, and resultant parallel tree as on the right. As usual, we begin with sentence letters from the interpretation function for the bottom nodes.

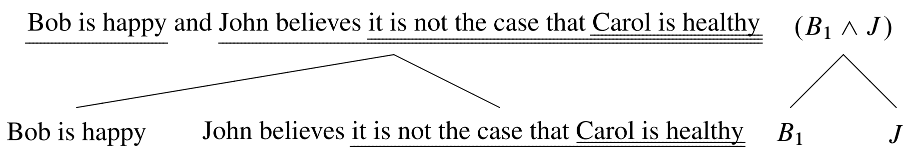
(P)



Given translations for the bottom nodes, we work our way up through the tree, applying equivalent expressions to translations already obtained. As we have seen, a natural translation of 'it is not the case that ____' is '$\sim$____'. Thus, working up from 'Carol is healthy', our parallel to 'it is not the case that Carol is healthy' is $\sim C_2$. But now we have translations for both of the blanks of '____ and ____'. As we have seen, this has the same table as '(____ $\wedge$ ____)'. So that is our translation. Again, other expressions might do. In particular, $\wedge$ is an abbreviation with the same table as '$\sim$(____ $\rightarrow$ $\sim$____)'. In each case, the whole is true when the sentences in both blanks are true, and otherwise false. Since this is the same as for '____ and ____', either would do as a translation. But again, the simplest thing is to go with '(____ $\wedge$ ____)'. Thus the final result is $(B_2 \wedge \sim C_2)$. With the alternate translation for the main operator, the result would have been $\sim(B_2 \rightarrow \sim\sim C_2)$.

Our last sentence is equivalent to 'Bob is happy and John believes it is not the case that Carol is healthy'. Given what we have done, the parallel tree should be easy to construct.

(Q)



Given that the tree "bottoms out" on both 'Bob is happy' and 'John believes it is not the case that Carol is healthy' the only operator to translate is the main operator '____ and ____'. And we have just seen how to deal with that. The result is the completed translation, $(B_1 \wedge J)$.

Again, once you become familiar with this procedure the full method with trees may become tedious—and we will often want to set it to the side. But notice: the method breeds good habits! And the method puts us in a position to translate complex expressions, even ones that are so complex that we can barely grasp what they are saying. Beginning with the main operator, we break expressions down from complex parts to ones that are simpler. Then we construct translations, one operator at a time, where each step is manageable.

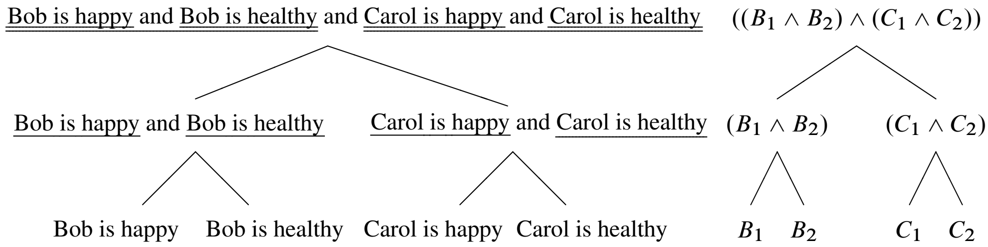
Also, we should be able to see *why* the method results in good translations: Consider some situation with its corresponding intended interpretation. Truth values for *basic* parts are the same just by the specification of the interpretation function.

And with equivalent tables, parts built out of them must be the same as well, all the way up to the truth value of the whole. We satisfy the first part of our criterion CG insofar as the way we break down sentences in parse trees forces us to capture all the sentential structure there is to be captured.

For a last example, consider, ‘Bob is happy and Bob is healthy and Carol is happy and Carol is healthy’. This is true only if ‘Bob is happy’, ‘Bob is healthy’, ‘Carol is happy’, and ‘Carol is healthy’ are all true. But the method may apply in different ways. We might, at step one, treat the sentence as a complex expression involving multiple uses of ‘____ and ____’; perhaps something like,

(R) Bob is happy and Bob is healthy and Carol is happy and Carol is healthy

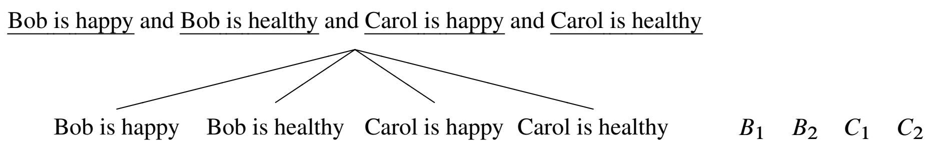
In this case, there is a straightforward move from the ordinary operators to formal ones in the final step. That is, the situation is as follows:



So we use multiple applications of our standard caret operator. But we might have treated the sentence as something like,

(S) Bob is happy and Bob is healthy and Carol is happy and Carol is healthy

involving a single four-blank operator, ‘____ and ____ and ____ and ____’, which yields true only when sentences in all its blanks are true. We have not seen anything like this before, but nothing stops a tree with four branches all at once. In this case, we would begin,



But now we need an equivalent formal expression with *four* blanks that is true when sentences in all the blanks are true and otherwise false. Here is something that would do: ‘((____ ∧ ____) ∧ (____ ∧ ____))’. On either of these approaches, then, the result is $((B_1 \wedge B_2) \wedge (C_1 \wedge C_2))$. Other options might result in something like $(((B_1 \wedge B_2) \wedge C_1) \wedge C_2)$. In this way, there is room for shifting burden between steps one and three. Such shifting explains how step (1) can be more complex than it was initially represented to be. Choices about expanding truth functional structure in the initial stage may matter for what are the equivalent expressions at the end. And the case exhibits how there are options for different, equally good, translations of the same ordinary expressions. What matters for CG is that resultant expressions capture available structure and be true when the originals are true and false when the originals are false. In most cases, one translation will be more *natural* than others, and it is good form to strive for natural translations. If there had been a comma so that the original sentence was, ‘Bob is happy and Bob is healthy, and Carol is happy and Carol is healthy’ it would have been most natural to go for an account along the lines of (R). And it is crazy to use, say, ‘$\sim\sim\sim$____’ when ‘$\sim$____’ will do as well.

*E5.5. Construct parallel trees to complete the translation of the sentences from E5.3 and E5.4. Hint: You will not need any operators other than $\sim$ and $\wedge$.

E5.6. Use our method to translate each of the following. That is, for each sentence, generate a parse tree and interpretation function, and then a parallel tree to produce a formal equivalent.

a. Plato and Aristotle were great philosophers, but Ayn Rand was not.

b. Plato was a great philosopher and everything Plato said was true, but Ayn Rand was not a great philosopher and not everything she said was true.

*c. It is not the case that: everything Plato, and Aristotle, and Ayn Rand said was true.

d. Plato was a great philosopher but not everything he said was true, and Aristotle was a great philosopher but not everything he said was true.

e. Not everyone agrees that Ayn Rand was not a great philosopher, and not everyone thinks that not everything she said was true.

E5.7. Use our method to translate each of the following. That is, for each sentence, generate a parse tree and interpretation function, and then a parallel tree to produce a formal equivalent.

a. Bob and Sue and Jim will pass the class.

b. Sue will pass the class, but it is not the case that: Bob will pass and Jim will pass.

c. It is not the case that: Bob will pass the class and Sue will not.

d. Jim will not pass the class, but it is not the case that: Bob will not pass and Sue will not pass.

e. It is not the case that: Jim will pass and not pass; and it is not the case that: Sue will pass and not pass.

5.2.4 Not, And, Or

Our idea has been to recognize when truth conditions for ordinary and formal sentences are the same. As we have seen, this turns out to require recognizing when *tables* for ordinary operators are equivalent to ones for formal expressions. We have had a lot to say about 'it is not the case that ____' and '____ and ____'. We now turn to a more general treatment. We will not be able to provide a complete menu of ordinary operators. Rather, we will see that some uses of some ordinary operators can appropriately be translated by our symbols. We should be able to discuss enough cases for you to see how to approach others on a case-by-case basis. The discussion is organized around our operators, $\sim$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$, taken in that order.

First, as we have seen, 'it is not the case that ____' has the same table as $\sim$. And various ordinary expressions may be equivalent to expressions involving this operator. Thus, 'Bob is not married' and 'Bob is unmarried' might be understood as equivalent to 'It is not the case that Bob is married'. Given this, we might assign a sentence letter, say, M to 'Bob is married' and translate $\sim M$. But the second case calls for comment. By comparison, consider, 'Bob is unlucky'. Given what we have done, it is natural to treat 'Bob is unlucky' as equivalent to 'It is not the case that Bob is lucky'; assign L to 'Bob is lucky'; and translate $\sim L$. But this is not obviously right. Consider three situations: (i) Bob goes to Las Vegas with \$1,000, and comes away with \$1,000,000. (ii) Bob goes to Las Vegas with \$1,000, and comes away with \$100, having seen a show and had a good time. (iii) Bob goes to Las Vegas with \$1,000, falls into a manhole on his way into the casino, and has his money stolen by a light-fingered thief on the way down. In the first case he is lucky; in the third, unlucky. But, in the second, one might want to say that he was neither lucky nor unlucky.

$$\begin{array}{ll} \text{(i)} & \text{Bob is lucky} \\ \text{(ii)} & \text{Bob is neither lucky nor unlucky} \\ \text{(iii)} & \text{Bob is unlucky} \end{array} \qquad \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{It is not the case that Bob is lucky}$$

If this is right, 'Bob is unlucky' is *not* equivalent to 'It is not the case that Bob is lucky'—for it is not the case that Bob is lucky in *both* situations (ii) and (iii). Thus we might have to assign 'Bob is lucky' one letter, and 'Bob is unlucky' another.[1] Decisions about this sort of thing may depend heavily on context, and assumptions which are in the background of conversation. We will ordinarily *assume* contexts where there is no "neutral" state—so that being unlucky is not being lucky, and similarly in other cases.

Second, as we have seen, '____ and ____' has the same table as $\wedge$. As you may recall from E5.2, another common operator that works this way is '____ but ____'. Consider, for example, 'Bob likes Mary but Mary likes Jim'. Suppose Bob does like

[1] Or so we have to do in the context of our logic where T and F are the only truth values. Another option is to allow three values so that the one letter might be T, F, or neither. It is possible to proceed on this basis—though the two valued (classical) approach has the virtue of relative simplicity. With the classical approach as background, some such alternatives are developed in Priest, *Non-Classical Logics*.

Mary and Mary does like Jim; then the compound sentence is true. Suppose one of the simples is false, Bob does not like Mary or Mary does not like Jim; then the compound is false. Thus '____ but ____' has the table,

	____	but	____
	T	***T***	T
(T)	T	***F***	F
	F	***F***	T
	F	***F***	F

and so has the same table as $\wedge$. So, in this case, we might assign B to 'Bob likes Mary' M to 'Mary likes Jim', and translate, $(B \wedge M)$. Of course, the ordinary expression 'but' carries a sense of opposition that 'and' does not. Our point is not that 'and' and 'but' somehow *mean* the same, but rather that compounds formed by means of them have the same truth function. Another common operator with this table is 'although ____ , ____'. You should convince yourself that this is so, and be able to find other ordinary terms that work just the same way.

Once again, however, there is room for caution in some cases. Consider, for example, 'Bob took a shower and got dressed'. Given what we have done, it is natural to treat this as equivalent to 'Bob took a shower and Bob got dressed'; assign letters S and D; and translate $(S \wedge D)$. But this is not obviously right. Suppose Bob gets dressed, but then realizes that he is late for a date and forgot to shower, so he jumps in the shower fully clothed, and air-dries on the way. Then it is true that Bob took a shower, and true that Bob got dressed. But is it true that Bob took a shower and got dressed? If not—because the order is wrong—our translation $(S \wedge D)$ might be true when the original sentence is not. Again, decisions about this sort of thing depend heavily upon context and background assumptions. And there may be a distinction between what is *said* and what is conversationally *implied* in a given context. Perhaps what was said corresponds to the table, so that our translation is right, though there are certain assumptions typically made in conversation that go beyond. But we need not get into this. Our point is not that the ordinary 'and' *always* works like our operator $\wedge$; rather the point is that some (indeed, many) ordinary uses are rightly regarded as having the same table.[2]

[2]The ability to make this point is an important byproduct of our having introduced the formal operators "as themselves." Where $\wedge$ and the like are introduced as *being* direct translations of ordinary operators, a natural reaction to cases of this sort—a reaction had even by some professional logicians and philosophers—is that "the table is wrong." But this is mistaken! Our $\wedge$ operator has its own significance, which may or may not agree with the shifting meaning of ordinary terms. The situation is no different than for translation across ordinary languages, where terms may or may not have uniform equivalents.

But now one may feel a certain tension with our account of what it is for an operator to be truth functional—for there seem to be contexts where the truth values of sentences in the blanks do not determine the truth value of the whole, even for a purportedly truth functional operator like '____ and ____'. However, we want to distinguish different *senses* in which an operator may be used (or an ambiguity as between a *bank* of a river and a *bank* where you deposit money)—in this case between '____ and ____' and '____ and (then) ____'. The first of these has the usual table, but the second is not

The operator which is most naturally associated with ∨ is '____ or ____'. In this case, there is room for caution from the start. Consider first a restaurant menu which says that you will get soup or you will get salad with your dinner. This is naturally understood as 'You will get soup or you will get salad' where the sentential operator is '____ or ____'. In this case, the table would seem to be,

(U)

____	or	____
T	***F***	T
T	***T***	F
F	***T***	T
F	***F***	F

The compound is true if you get soup, true if you get salad, but not if you get neither or both. None of our operators has this table.

But contrast this case with one where a professor promises either to give you an 'A' on a paper, or to give you very good comments so that you will know what went wrong. Suppose the professor gets excited about your paper, giving you both an 'A' and comments. Presumably, she did not break her promise! That is, in this case, we seem to have, 'I will give you an 'A' or I will give you comments' with the table,

(V)

____	or	____
T	***T***	T
T	***T***	F
F	***T***	T
F	***F***	F

The professor breaks her word just in case she gives you a low grade without comments. This table is identical to the table for ∨. For another case, suppose you set out to buy a power saw, and say to your friend 'I will go to Home Depot or I will go Lowe's'. You go to Home Depot, do not find what you want, so go to Lowe's and make your purchase. When your friend later asks where you went, and you say you went to both, he or she will not say you lied (!) when you said where you were going—for your statement required only that you would try at least one of those places.

The grading and shopping cases represent the so-called "inclusive" use of 'or'—including the case when both components are T; the menu uses the "exclusive" use of 'or'—excluding the case when both are T. Ordinarily, we will *assume* that 'or' is used in its inclusive sense, and so is translated directly by ∨.[3] Another operator that works this way is '____ unless ____'. Again, there are exclusive and inclusive senses—which you should be able to see by considering restaurant and shopping examples: 'You will get soup unless you will get salad' and 'I will go to Home Depot

truth functional at all. Again, we will ordinarily *assume* a context where 'and', 'but', and the like have tables that correspond to ∧.

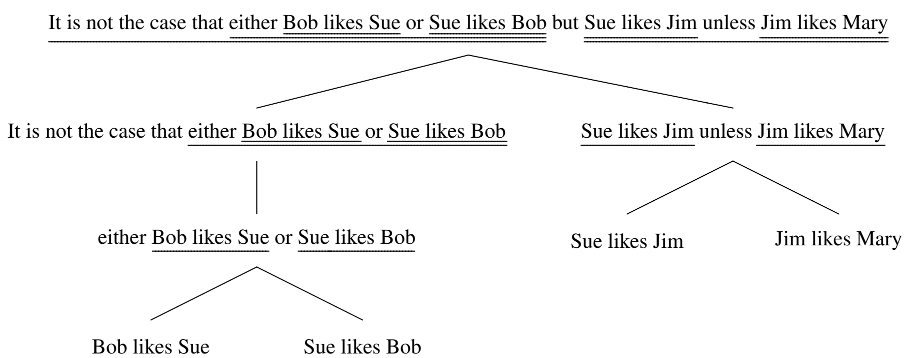
[3]Again, there may be a distinction between what is *said* and what is conversationally *implied* in a given context. Perhaps what is said generally corresponds to the inclusive table, though many uses are against background assumptions which automatically exclude the case when both are T. But we need not get into this. It is enough that some uses are according to the inclusive table.

unless I will go to Lowe's'. And again, we will ordinarily assume that the inclusive sense is intended. For the exclusive cases, we can generate the table by means of complex expressions. Thus, for example $\sim(\mathcal{P} \leftrightarrow \mathcal{Q})$ does the job. You should convince yourself that this is so.

Observe that 'either ____ or ____' has the same table as '____ or ____'; and 'both ____ and ____' the same as '____ and ____'. So one might think that 'either' and 'both' play no real role. They do however serve a sort of "bracketing" function: Consider 'Neither Bob likes Sue nor Sue likes Bob'. This is naturally understood as, 'It is not the case that either Bob likes Sue or Sue likes Bob' with translation $\sim(B \vee S)$. Observe that this division is required: An attempt to parse it to 'It is not the case that either Bob likes Sue or Sue like Bob' results in the fragment 'either Bob likes Sue' in the blank for 'it is not the case that ____'. There would be an ambiguity about the main operator if 'either' were missing; but with it there, the only way to keep complete sentences in the blanks is to make 'it is not the case that ____' the main operator. Similarly, 'Not both Bob likes Sue and Sue likes Bob' comes to 'It is not the case that both Bob likes Sue and Sue likes Bob' with translation $\sim(B \wedge S)$. There would be an ambiguity about the main operator if 'both' were missing; but with it there, the only way to keep complete sentences in the blanks is to make 'it is not the case that ____' the main operator.

And we continue to work with complex forms on trees. Thus, for example, consider 'Neither Bob likes Sue nor Sue likes Bob, but Sue likes Jim unless Jim likes Mary'. This is a mouthful, but we can deal with it in the usual way. The hard part, perhaps, is just exposing the operator structure.

(Y)



Given this, with what we have said above, generate the interpretation function and then the parallel tree as follows:

‘Neither nor’ and ‘Not both’

We have given accounts of ‘neither ____ nor ____’ and ‘not both ____ and ____’ which treat them as combining ordinary negation with conjunction or disjunction. However, it is possible to see them as unstructured ordinary operators.

So, for example, we might treat ‘neither ____ nor ____’ as an unstructured sentential operator with a table as in (W) below.

(W)

neither	____ nor	____
F	T	T
F	T	F
F	F	T
T	F	F

(X)

$\mathcal{P}$	$\mathcal{Q}$	$\sim$	($\mathcal{P}$	$\vee$	$\mathcal{Q}$)
T	T	***F***	T	T	T
T	F	***F***	T	T	F
F	T	***F***	F	T	T
F	F	***T***	F	F	F

Thus ‘Neither Bob likes Sue nor Sue likes Bob’ is true just when ‘Bob likes Sue’ and ‘Sue likes Bob’ are both false, and otherwise the compound is false. No operator of our formal language has a table which is T just when the components are both F. Still, we may form complex expressions which work this way. So from (X), $\sim(\mathcal{P} \vee \mathcal{Q})$ has the same table. In this case, with the natural interpretation function, the parse and parallel trees are,

Neither Bob likes Sue nor Sue likes Bob

Bob likes Sue   Sue likes Bob

$\sim(B \vee S)$

B S

As usual, there is one node in the parallel tree for each node in the parse tree. Effectively, this strategy unpacks ‘neither ____ nor ____’ in the third stage of TP rather than the first. Though the resultant tree has a different shape than a tree from the account of the main text, the result is the same. Another expression with the same table is $\sim\mathcal{P} \wedge \sim\mathcal{Q}$. Either is a good translation of ‘neither ____ nor ____’ conceived as an unstructured operator.

Similarly we might treat ‘not both ____ and ____’ as an unstructured sentential operator whose table is F just when the components are both T. Again, no operator of our formal language works this way. But we may form complex expressions that do the job. So, as from the main discussion, $\sim(\mathcal{P} \wedge \mathcal{Q})$ has the same table. Another expression that works this way is $\sim\mathcal{P} \vee \sim\mathcal{Q}$.

Observe that $\sim(\mathcal{P} \vee \mathcal{Q})$ for ‘neither nor’ has the same table as $\sim\mathcal{P} \wedge \sim\mathcal{Q}$; and $\sim(\mathcal{P} \wedge \mathcal{Q})$ for ‘not both’ the same as $\sim\mathcal{P} \vee \sim\mathcal{Q}$. It is thus a *mistake* to “distribute” the tilde of $\sim(\mathcal{P} \vee \mathcal{Q})$ to $\sim\mathcal{P} \vee \sim\mathcal{Q}$—this changes from ‘neither nor’ to ‘not both’. Similarly it is a mistake to distribute the tilde of $\sim(\mathcal{P} \wedge \mathcal{Q})$ to $\sim\mathcal{P} \wedge \sim\mathcal{Q}$—this changes from ‘not both’ to ‘neither nor’. Rather, to preserve equivalence, when $\sim$ goes into a disjunction, $\vee$ flips to $\wedge$; and when $\sim$ goes into a conjunction, $\wedge$ flips to $\vee$.

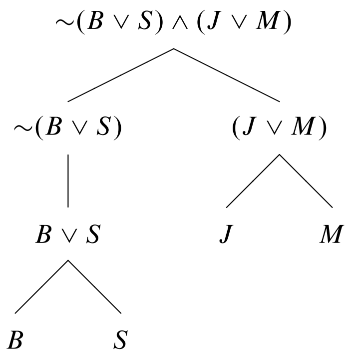
Choices among structured and unstructured approaches to ‘not both’ and ‘neither nor’ are a matter of taste rather than correctness.

B: Bob likes Sue

S: Sue likes Bob

J: Sue likes Jim

M: Jim likes Mary



Given that 'or' and 'unless' are equivalent to '____ $\vee$ ____', everything works as before. Again, the complex problem is rendered simple if we attack it one operator at a time.

E5.8. Using the interpretation function below, produce parse trees and then parallel ones to complete the translation for each of the following.

B: Bob likes Sue

S: Sue likes Bob

B_1: Bob is cool

S_1: Sue is cool

a. Bob likes Sue.

b. Sue does not like Bob.

c. Bob likes Sue and Sue likes Bob.

d. Bob likes Sue or Sue likes Bob.

e. Bob likes Sue unless she is not cool.

*f. Either Bob does not like Sue or Sue does not like Bob.

g. Neither Bob likes Sue, nor Sue likes Bob.

*h. Not both Bob and Sue are cool.

*i. Bob and Sue are cool, and Bob likes Sue but Sue does not like Bob.

j. Although neither Bob nor Sue are cool, either Bob likes Sue or Sue likes Bob.

E5.9. Use our method to translate each of the following. That is, for each sentence, generate a parse tree and interpretation function, and then a parallel tree to produce a formal equivalent.

a. Charlie is not good at baseball.

b. Either Snoopy or Patty is good at baseball.

c. Neither Charlie nor Lucy is good at baseball.

*d. Neither Charlie, nor Lucy, nor Woodstock is good at baseball.

e. Not both Charlie and Snoopy are good at baseball.

f. The team will lose unless Patty plays for them.

g. Charlie is not the best baseball player, however he wishes that he was.

*h. Although guns and knives are illegal in baseball, sliding is not.

i. Either Schroeder wears his mask or his face is not protected, and a pitch to the face hurts.

j. The Boston Red Sox won the World Series in 2018, but not in 2019, 2020, 2021, or 2022.

5.2.5 If, Iff

The operator which is most naturally associated with → is ‘if ____ then ____’. Consider some fellow, perhaps of less than sterling character, of whom we assert, ‘If he loves her, then she is rich’—that is, ‘If he loves her, then she is rich’. In this case, the table begins,

	if ____	then	____
	T	***T***	T
(Z)	T	***F***	F
	F	?	T
	F	***T***	F

If ‘He loves her’ and ‘She is rich’ are both true, then what we said about him is true. If he loves her, but she is not rich, what we said was wrong. If he does not love her, and she is poor, then we are also fine, for all we said was that *if* he loves her, then she is rich. But what about the other case? Suppose he does not love her, but she is rich. There is a temptation to say that our conditional assertion is false. But do not give in! Notice: we did not say that he loves all the rich girls. All we said was that *if* he loves this particular girl, then she is rich. So the existence of rich girls he does not love does not undercut our claim. For another case, say you are trying to find the car he is driving and say ‘If he is in his own car, then it is a Corvette’—that is, ‘If he is in his own car then he is in a Corvette’. You would be mistaken if he has traded his Corvette for a Yugo. But say the Corvette is in the shop and he is driving a loaner that also happens to be a Corvette. Then ‘He is in his own car’ is F and ‘He is in a Corvette’ is T. Still, there is nothing wrong with your claim—*if* he is in his own car, then he is in a Corvette. Given this, we are left with the completed table,

(AA)

if ___	then	___
T	***T***	T
T	***F***	F
F	***T***	T
F	***T***	F

which is identical to the table for $\rightarrow$. With L for 'He loves her' and R for 'She is rich', for 'If he loves her then she is rich' the natural translation is $(L \rightarrow R)$. Another operator which works this way is '____ only if ____'. You should be able to see this with examples as above: 'He loves her only if she is rich' and 'He is in his own car only if he is in a Corvette'. So far, perhaps, so good.

But the conditional calls for special comment. First, notice that the table shifts with the position of 'if'. Suppose he loves her if she is rich. Intuitively, 'He loves her if she is rich' says the same as 'If she is rich then he loves her'. Thus, with the above table and assignments, we end up with translation $(R \rightarrow L)$. Notice that the order is switched around the arrow. This time, we are mistaken if she is rich and he does not love her. We can make this point directly from the original claim.

(AB)

He loves her	if	she is rich
T	***T***	T
T	***T***	F
F	***F***	T
F	***T***	F

The claim is false just in the case where she is rich but he does not love her. The result is *not* the same as the table for $\rightarrow$. What we need is an expression that is F in the case when R is T and L is F, and otherwise T. We get just this with $(R \rightarrow L)$. Of course, this is just the same result as by intuitively reversing the operator into the regular 'if ____ then ____' form.

In the formal language, the *order* of the components is crucial. In a true material conditional, the truth of the antecedent guarantees the truth of the consequent. In ordinary language this role is played, not by the order of the components, but by operator placement. In general, *if* by itself is an *antecedent* indicator; and *only if* is a *consequent* indicator. That is, we get,

(AC)
$$\begin{array}{lll} \text{if } \mathcal{P} \text{ then } \mathcal{Q} & \Longrightarrow & (\mathcal{P} \rightarrow \mathcal{Q}) \\ \mathcal{P} \text{ if } \mathcal{Q} & \Longrightarrow & (\mathcal{Q} \rightarrow \mathcal{P}) \\ \mathcal{P} \text{ only if } \mathcal{Q} & \Longrightarrow & (\mathcal{P} \rightarrow \mathcal{Q}) \\ \text{only if } \mathcal{P}, \mathcal{Q} & \Longrightarrow & (\mathcal{Q} \rightarrow \mathcal{P}) \end{array}$$

'If', taken alone, identifies what does the guaranteeing, and so the antecedent of our material conditional; 'only if' identifies what is guaranteed, and so the consequent.[4]

[4]It may feel natural to convert '$\mathcal{P}$ unless $\mathcal{Q}$' to '$\mathcal{P}$ if not $\mathcal{Q}$' and translate $(\sim\mathcal{Q} \rightarrow \mathcal{P})$. This is fine and, as is clear from the abbreviated form, equivalent to $(\mathcal{Q} \vee \mathcal{P})$. However, with the extra negation and concern about direction of the arrow, it is easy to get confused on this approach—so the simple wedge is less likely to go wrong.

Cause and Conditional

It is important that the material conditional does *not* directly indicate causal connection. Suppose we have sentences,

S: You strike the match
L: The match will light.

And consider,

(i)	If you strike the match then it will light	$S \to L$
(ii)	The match will light only if you strike it	$L \to S$

with natural translations by our method on the right. Good. But clearly the *cause* of the lighting is the striking. So the first arrow runs from cause to effect, and the second from effect to cause. Why? In (i) we represent the cause as *sufficient* for the effect: striking the match guarantees that it will light. In (ii) we represent the cause as *necessary* for the effect—the only way to get the match to light, is to strike it—so if the match lights, it was struck.

There may be a certain *tendency* to associate the ordinary 'if' and 'only if' with cause, so that we say, 'if $\mathcal{P}$ then $\mathcal{Q}$' when we think of $\mathcal{P}$ as a (sufficient) cause of $\mathcal{Q}$, and say '$\mathcal{P}$ only if $\mathcal{Q}$' when we think of $\mathcal{Q}$ as a (necessary) cause of $\mathcal{P}$. But causal direction is not reflected by the arrow, which comes out $(\mathcal{P} \to \mathcal{Q})$ either way. The material conditional indicates *guarantee*.

This point is important insofar as certain ordinary conditionals seem inextricably tied to causation. This is particularly the case with "subjunctive" conditionals (conditionals about what *would* have been). Suppose after a game of one-on-one basketball I brag, 'If I had played LeBron, I would have won' where this is,

'If it were the case that I played LeBron then it would have been that I won the game'.

Intuitively, this is false, LeBron would wipe the floor with me. But contrast,

'If it were the case that I played Lassie then it would have been that I won the game'.

Now, intuitively, this is true; Lassie has many talents but, presumably, basketball is not among them—and I could take her. But I have never played LeBron or Lassie, so both 'I played LeBron' and 'I played Lassie' are false. Thus the truth value of the whole conditional changes from false to true though the values of sentences in the blanks remain the same; and 'if it were the case that ____ then it would have been that ____' is not even truth functional. Subjunctive conditionals do offer a sort of guarantee, but the guarantee is for situations alternate to the way things actually are. So actual truth values do not determine the truth of the conditional.

Conditionals other than the material conditional are a central theme of Priest, *Non-Classical Logics*. As usual, we simply assume that 'if' and 'only if' are used in their truth functional sense, and so are given a good translation by $\to$.

As we have just seen, the natural translation of '$\mathcal{P}$ if $\mathcal{Q}$' is $\mathcal{Q} \rightarrow \mathcal{P}$, and the translation of '$\mathcal{P}$ only if $\mathcal{Q}$' is $\mathcal{P} \rightarrow \mathcal{Q}$. Thus it should come as no surprise that the translation of '$\mathcal{P}$ if *and* only if $\mathcal{Q}$' is $(\mathcal{P} \rightarrow \mathcal{Q}) \wedge (\mathcal{Q} \rightarrow \mathcal{P})$, where this is precisely what is abbreviated by $(\mathcal{P} \leftrightarrow \mathcal{Q})$. We can also make this point directly. Consider, 'He loves her if and only if she is rich'. The operator is truth functional with the table,

(AD)

He loves her	if and only if	she is rich
T	***T***	T
T	***F***	F
F	***F***	T
F	***T***	F

It cannot be that he loves her and she is not rich, because he loves her *only if* she is rich; so the second row is F. And it cannot be that she is rich and he does not love her, because he loves her *if* she is rich; so the third row is F. The biconditional is true just when both she is rich and he loves her, or neither. Another operator that works this way is '____ just in case ____'. You should convince yourself that this is so. Notice that 'if', 'only if', and 'if and only if' play very different roles for translation—you almost want to think of them as completely different words: *if*, *onlyif*, and *ifandonlyif*, each with its own distinctive logical role. Do not get the different roles confused!

For an example that puts some of this together consider, 'She is rich if he loves her, if and only if he is a cad or very generous'. This comes to the following:

(AE)

She is rich if he loves her if and only if he is a cad or he is very generous

She is rich if he loves her   he is a cad or he is very generous

She is rich   he loves her   he is a cad   he is very generous

We begin by assigning sentence letters to the simple sentences at the bottom. Then the parallel tree is constructed as follows:

R: She is rich

L: He loves her

C: He is a cad

G: He is very generous

$((L \rightarrow R) \leftrightarrow (C \vee G))$

$(L \rightarrow R)$ $(C \vee G)$

R L C G

Observe that 'She is rich if he loves her' is equivalent to $(L \rightarrow R)$, not the other way around. Then the wedge translates '____ or ____', and the main operator has the same table as $\leftrightarrow$.

Notice again that our procedure for translating, one operator or part at a time, lets us translate even where the original is so complex that it is difficult to comprehend.

The method forces us to capture all available sentential structure, and the resultant translation is good insofar as, given its interpretation function, a formal sentence comes out true on precisely the intended interpretations that correspond to stories on which the original is true. It does this because the formal and informal sentences *work* the same way. Eventually, you want to be able to work translations without the trees. (And maybe you have already begun to do so.) In fact, it will be natural to generate translations simultaneously with a (mental) parse tree. The result produces translations from the *top down*, rather than from the bottom up, building the translation operator-by-operator as you take the sentence apart from the main operator down. But, of course, the result should be the same no matter how you do it.

From definition AR on page 5, an argument is some sentences, one of which (the conclusion) is taken to be supported by the remaining sentences (the premises). In some courses on logic or critical reasoning, one might spend a great deal of time learning to identify premises and conclusions in ordinary discourse. However, we have taken this much as given, representing arguments in standard form, with premises listed as complete sentences above a line, and the conclusion under. Thus, for example,

(AF)

If you strike the match, then it will light
The match will not light
———
You did not strike the match

is a simple argument of the sort we might have encountered in Chapter 1. By the Chapter 1 validity test VT, this argument is logically valid.

We get the same result by our formal methods: To translate the argument, we produce a translation for the premises and conclusion, retaining the "standard-form" structure. Thus we might end up with an interpretation function and translation as below,

(AG)

S: You strike the match

L: The match will light

$$\begin{array}{l} S \to L \\ \sim L \\ \hline \sim S \end{array}$$

The result is an object to which we can apply truth tables and derivations in a straightforward way. Thus by a truth table and (Chapter 3) derivation,

(AH)

L	S	$S \to L$	$\sim L$	/	$\sim S$
T	T	***T***	***F***		***F***
T	F	***T***	***F***		***T***
F	T	***F***	***T***		***F***
F	F	***T***	***T***		***T***

1.	$S \to L$	prem
2.	$\sim L$	prem
3.	$(S \to L) \to (\sim L \to \sim S)$	T3.13
4.	$\sim L \to \sim S$	3,1 MP
5.	$\sim S$	4,2 MP

both $S \to L, \sim L \vDash_s \sim S$ and $S \to L, \sim L \vdash_{ADs} \sim S$. If you have not yet seen derivations, do not worry about it for now.

And these results are just what we want. For the table, recall that (i) for any way a world (consistent story) can be, an interpretation function results in an intended interpretation; and (ii) on a good translation, the truth value of an ordinary sentence at an arbitrary world is the same as its formal counterpart on the corresponding intended interpretation. For some good formal translation of premises and conclusion: Suppose an argument is sententially valid; then by SV there is *no* interpretation on which the premises are true and the conclusion is false; so no *intended* interpretation from (i) makes the premises true and the conclusion is false; so with (ii) no *consistent story* makes the premises true and conclusion false; so by LV the original argument is logically valid. So if an argument is sententially valid, then it is logically valid. We will make this point again, in some detail, in Part III.[5] For now, notice that our formal methods, derivations and truth tables, apply to arguments of arbitrary complexity. So we are in a position to demonstrate validity for arguments that would have set us on our heels in Chapter 1. With this in mind, consider again the butler case (B) from page 2. Demonstration that the argument is logically valid is entirely straightforward by a good translation and then a truth table to demonstrate semantic validity.

E5.10. Using the interpretation function below, produce parse trees and then parallel ones to complete the translation for each of the following.

L: Lassie barks

T: Timmy is in trouble

P: Pa will help

H: Lassie is healthy

a. If Timmy is in trouble, then Lassie barks.

b. Timmy is in trouble if Lassie barks.

*c. Lassie barks only if Timmy is in trouble.

d. If Timmy is in trouble and Lassie barks, then Pa will help.

*e. If Timmy is in trouble, then if Lassie barks Pa will help.

f. If Pa will help only if Lassie barks, then Pa will help if and only if Timmy is in trouble.

g. Pa will help if Lassie barks, just in case Lassie barks only if Timmy is in trouble.

h. If Timmy is in trouble and Pa will not help, then Lassie is not healthy or does not bark.

*i. If Timmy is in trouble, then either Lassie is not healthy or if Lassie barks then Pa will help.

[5] And it remains for Part III to show how *derivations* matter for logical validity.

j. If Lassie neither barks nor is healthy, then Timmy is in trouble if Pa will not help.

E5.11. Use our method, with or without parse trees, to produce a translation, including interpretation function for the following.

a. If animals feel pain, then animals have intrinsic value.

b. Animals have intrinsic value only if they feel pain.

c. Although animals feel pain, vegetarianism is not right.

d. Animals do not have intrinsic value unless vegetarianism is not right.

e. Vegetarianism is not right only if animals do not feel pain or do not have intrinsic value.

f. If you think animals feel pain, then vegetarianism is right.

*g. If you think animals do not feel pain, then vegetarianism is not right.

h. If animals feel pain, then if animals have intrinsic value if they feel pain, then animals have intrinsic value.

*i. Vegetarianism is right only if both animals feel pain, and animals have intrinsic value just in case they feel pain; but it is not the case that animals have intrinsic value just in case they feel pain.

j. If animals do not feel pain if and only if you think animals do not feel pain, but you do think animals feel pain, then you do not think that animals feel pain.

E5.12. For each of the following arguments: (i) Produce a good translation, including interpretation function and translations for the premises and conclusion. Then (ii) use truth tables to determine whether the argument is sententially valid.

*a. Our car will not run unless it has gasoline
Our car has gasoline
———
Our car will run

b. If Barack is president, then Michelle is first lady
Michelle is not first lady
———
Barack is not president

c. Snow is white and snow is not white
———
Dogs can fly

d. If Mustard murdered Boddy, then it happened in the library.
The weapon was the pipe if and only if it did not happen in the library, and the weapon was not the pipe only if Mustard murdered him.

Mustard murdered Boddy.

e. There is evil.
If god is good, there is no evil unless god has morally sufficient reasons for allowing it.
If god is omnipotent, then god does not have morally sufficient reasons for allowing evil.

God is not both good and omnipotent.

E5.13. For those who have studied derivations from at least the sentential portion of Chapter 3 or Chapter 6: For each of the arguments of E5.12 that is sententially valid, show that it is also valid in *ADs* or *NDs+*, whichever is appropriate.

E5.14. Use a translation and truth table to show that the butler argument (B) from page 2 is semantically valid.

E5.15. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Good translations.

b. Truth functional operators.

c. Parse trees, interpretation functions and parallel trees.

5.3 Quantificational

It is not surprising that our goals for the quantificational case remain very much as in the sentential one. We still want to produce translations—consisting of interpretation functions and formal sentences—which capture available structure, making a formal $\mathcal{P}'$ true at intended interpretation I_ω just when the corresponding ordinary $\mathcal{P}$ is true at story ω. We do this as before, by assuring that the various parts of the ordinary and formal languages work the same way. Of course, now we are interested in capturing *quantificational* structure, and the interpretation and formal sentences are for *quantificational* languages.

In the last section, we developed a recipe for translating from ordinary language into sentential expressions, associating particular bits of ordinary language with various formal symbols. We might proceed in very much the same way here, moving from our notion of *truth functional* operators, to that of *extensional* terms, relation symbols, and operators. Roughly, an ordinary term is *extensional* when the satisfaction of a formula in which it appears depends just on the object to which it refers; an ordinary relation symbol is *extensional* when the satisfaction of a formula in which it appears depends just on the objects to which it applies; and an ordinary operator is *extensional* when the satisfaction of a formula in which it appears depends just on the satisfaction of expressions which appear in its blanks. Clearly the notion of an extensional operator at least is closely related to that of a truth functional operator. Extensional terms, relation symbols, and operators in ordinary language work very much like corresponding ones in a formal quantificational language—where, again, the idea would be to identify bits of ordinary language which contribute to truth values in the same way as corresponding parts of the formal language.

However, in the quantificational case, there is no simple *recipe* for translation. It is best to work directly with the fundamental goal of producing formal translations that are true in the same situations as ordinary expressions. To be sure, certain patterns and strategies will emerge but, again, we should think of what we are doing less as applying a recipe than as directly using our understanding of what makes ordinary and formal sentences true to produce good translations. With this in mind, let us move directly to sample cases, beginning with those that are relatively simple, and advancing to ones that are more complex.

5.3.1 Elementary Sentences

First, sentences without quantifiers work very much as in the sentential case. Consider a simple example. Say we are confronted with 'Bob is happy'. We might begin, as in the sentential case, with the interpretation function,

B: Bob is happy

and use B for 'Bob is happy', $\sim B$ for 'Bob is not happy', and so forth. But this is to ignore structure we are now capable of capturing. Thus, in our standard quantificational language $\mathcal{L}_q$, we might let U be the set of all people, and set,

b: Bob

H^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is happy$\}$

Then we can use Hb for 'Bob is happy', $\sim Hb$ for 'Bob is not happy', and so forth. If I_ω assigns Bob to b, and the set of happy people to H, then Hb is satisfied and true on II_ω just in case Bob is happy at ω—which is just what we want. Similarly suppose we are confronted with 'Bob's father is happy'. In the sentential case, we might have tried, F: Bob's father is happy. But this is to miss structure available to us now. So we might consider assigning a constant d to Bob's father and going with Hd as above. But this also misses available structure. In this case, we can expand the interpretation function to include,

f^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the father of $\mathsf{m}\}$

Then for any variable assignment d, $\mathsf{I_d}[b]$ = Bob and $\mathsf{I_d}[f^1 b]$ is Bob's father. So Hf^1b is satisfied and true just in case Bob's father is happy. $\sim Hf^1b$ is satisfied just in case Bob's father is not happy, and so forth—which is just what we want. In these cases without quantifiers, once we have translated simple sentences, everything else proceeds as in the sentential case. Thus, for example, for 'Neither Bob nor his father is happy' we might offer, $\sim(Hb \vee Hf^1b)$.

The situation gets more interesting when we add quantifiers. We will begin with cases where a quantifier's scope includes neither binary operators nor other quantifiers, and gradually increase complexity. Consider the following interpretation function:

II $\quad$ U: $\{\mathsf{o} \mid \mathsf{o}$ is a dog$\}$
$\quad f^1$: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the father of $\mathsf{m}\}$
$\quad W^1$: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o will have its day$\}$

We assume that there is some definite content to a dog's having its day, and that every dog *has* a father—if a dog "Adam" has no father at all, we will not have specified a legitimate interpretation. (Why?) Say we want to translate the following sentences:

(1) Every dog will have its day

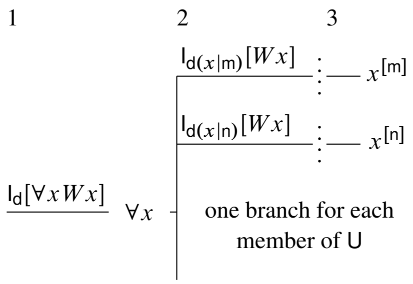
(2) Some dog will have its day

(3) Some dog will not have its day

(4) No dog will have its day

Assume 'some' means 'at least one'. The first sentence is straightforward. $\forall x Wx$ is read, 'for any x, Wx'; it is true just in case every dog will have its day. Suppose II_ω is an interpretation I where the elements of U are m, n, and so forth. Then the tree is as follows:

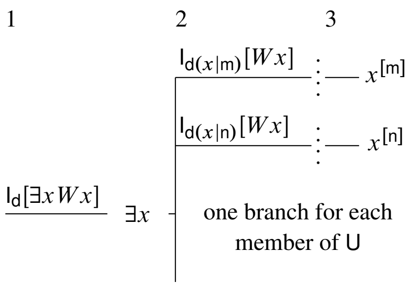
(AI)



The formula at (1) is satisfied just in case each of the branches at (2) is satisfied. But this can be the case only if each member of U is in the interpretation of W—which given our interpretation function, can only be the case if each dog will have its day. If even one dog does not have its day, then $\forall xWx$ is not satisfied, and is not true.

The second case is also straightforward. $\exists xWx$ is read, 'there is an x such that Wx'; it is true just in case some dog will have its day.

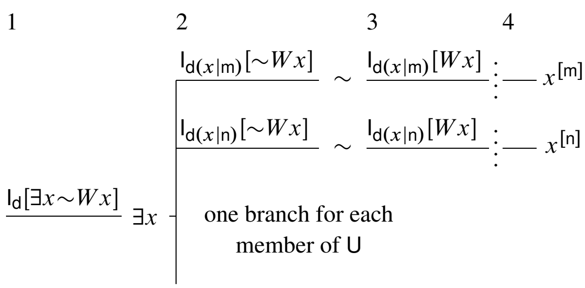
(AJ)



The formula at (1) is satisfied just in case at least one of the branches at (2) is satisfied. But this can be the case only if some member of U is in the interpretation of W—which, given the interpretation function, is to say that some dog will have its day.

The next two cases are only slightly more difficult. $\exists x{\sim}Wx$ is read, 'there is an x such that not Wx'; it is true just in case some dog will not have its day.

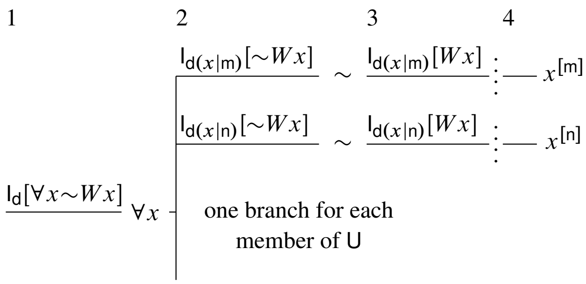
(AK)



The formula at (1) is satisfied just in case at least one of the branches at (2) is satisfied. And a branch at (2) is satisfied just in case the corresponding branch at (3) is not satisfied. So $\exists x{\sim}Wx$ is satisfied and true just in case some member of U is not in the interpretation of W—just in case some dog does not have its day.

The last case is similar. $\forall x{\sim}Wx$ is read, 'for any x, not Wx'; it is true just in case every dog does not have its day.

(AL)



The formula at (1) is satisfied just in case all of the branches at (2) are satisfied. And this is so just in case none of the branches at (3) are satisfied. So $\forall x \sim Wx$ is satisfied and true just in case none of the members of U are in the interpretation of W—just in case no dog has its day.

Perhaps it has already occurred to you that there are other ways to translate these sentences. The following lists what we have done, with "quantifier switching" alternatives on the right:

(AM)

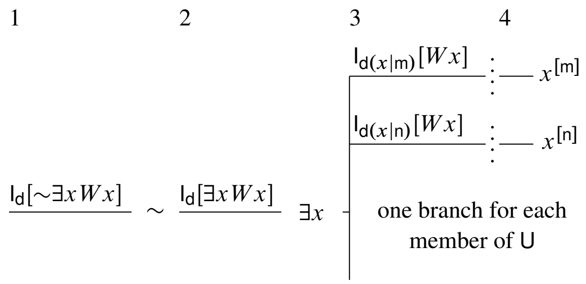
Every dog will have its day	$\forall x Wx$	$\sim\exists x \sim Wx$
Some dog will have its day	$\exists x Wx$	$\sim\forall x \sim Wx$
Some dog will not have its day	$\exists x \sim Wx$	$\sim\forall x Wx$
No dog will have its day	$\forall x \sim Wx$	$\sim\exists x Wx$

There are different ways to think about these alternatives. First, in ordinary language, beginning from the bottom, no dog will have its day just in case not even one dog does. Similarly, moving up the list, some dog will not have its day just in case not every dog does. Some dog will have its day just in case not every dog does not. And every dog will have its day iff not even one dog does not. These equivalences may be difficult to absorb at first but, if you think about them, each should make sense.

Next, we might think about the alternatives purely in terms of abbreviations. Notice that, in a tree, $\mathsf{I}_{\mathsf{d}}[\sim\sim\mathcal{P}]$ is always the same as $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$—the tildes "cancel each other out." But then, in the top case, $\sim\exists x \sim Wx$ abbreviates $\sim\sim\forall x \sim\sim Wx$ which is satisfied just in case $\forall x Wx$ is satisfied. In the second case, $\exists x Wx$ directly abbreviates $\sim\forall x \sim Wx$. In the third, $\exists x \sim Wx$ abbreviates $\sim\forall x \sim\sim Wx$ which is satisfied just in case $\sim\forall x Wx$ is satisfied. And, in the last case, $\sim\exists x Wx$ abbreviates $\sim\sim\forall x \sim Wx$, which is satisfied just in case $\forall x \sim Wx$ is satisfied. So, again, the alternatives are true under just the same conditions.

Finally, we might think about the alternatives directly, based on their branch conditions. Taking just the last case,

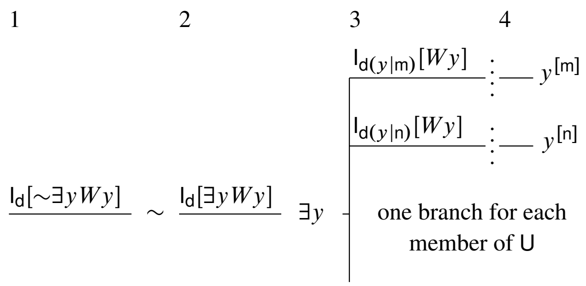
(AN)



The formula at (1) is satisfied just in case the formula at (2) is not. But the formula at (2) is not satisfied just in case none of the branches at (3) is satisfied—and this can only happen if no dog is in the interpretation of W, where this is as it should be for 'No dog will have its day'. In practice, there is no reason to prefer $\forall x{\sim}\mathcal{P}$ over ${\sim}\exists x\mathcal{P}$—the choice is purely a matter of taste. It would be less natural to use, say, ${\sim}\exists x{\sim}\mathcal{P}$ in place of $\forall x\mathcal{P}$, or ${\sim}\forall x{\sim}\mathcal{P}$ in place of $\exists x\mathcal{P}$. And it is a matter of good form to pursue translations that are natural. At any rate, all of the options satisfy CG. (But notice that we leave further room for alternatives among good answers, thus complicating comparisons with, for example, the *Answers to Selected Exercises*.)

Observe that variables are *mere placeholders* for these expressions so that choice of variables also does not matter. Thus, in tree (AN) immediately above, the formula is true just in case no dog is in the interpretation of W. But we get the exact same result if the variable is y.

(AO)



In either case, what matters in the end is whether the objects are in the interpretation of the relation symbol: whether $\mathsf{m} \in \mathsf{I}[W]$, and so forth. If none are, then the formulas are satisfied. Thus the formulas are satisfied under *exactly* the same conditions. And since one is satisfied iff the other is satisfied, one is a good translation iff the other is. So the choice of variables is up to you.

Given all this, we continue to treat truth functional operators as before—and we can continue to use underlines to expose truth functional structure. The difference is that what we would have seen as "simple" sentences have structure we were not able to expose before. So, for example, 'Either every dog will have its day or no dog will have its day' gets translation, $\forall xWx \vee \forall x{\sim}Wx$; 'Some dog will have its day and some dog will not have its day', gets, $\exists xWx \wedge \exists x{\sim}Wx$; and so forth. If we want to say that some dog is such that its father will have his day, we might try $\exists xWf^1x$—there is an x such that the *father of it* will have its day.

E5.16. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. Assume Phil 300 is a logic class with Ninfa and Harold as members in which each student is associated with a unique homework partner.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a student in Phil 300$\}$

a: Ninfa

h: Harold

p^1: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the homework partner of $\mathsf{m}\}$

G^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o gets a good grade$\}$

H^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m gets a higher grade than $\mathsf{n}\}$

*a. Ninfa and Harold both get a good grade.

b. Ninfa gets a good grade, but her homework partner does not.

c. Ninfa gets a good grade only if both her homework partner and Harold do.

d. Harold gets a higher grade than Ninfa.

*e. If Harold gets a higher grade than Ninfa, then he gets a higher grade than her homework partner.

f. Nobody gets a good grade.

*g. If someone gets a good grade, then Ninfa's homework partner does.

h. If Ninfa does not get a good grade, then nobody does.

*i. Nobody gets a grade higher than their own grade.

j. If no one gets a higher grade than Harold, then no one gets a good grade.

E5.17. Produce a good quantificational translation for each of the following. In this case you should provide a single interpretation function with application to all the sentences. Let U be the set of famous philosophers and, assuming that each has a unique successor, implement a *successor* function.

a. Plato is a good philosopher.

*b. Plato is better than Aristotle.

c. Neither Plato is better than Aristotle, nor Aristotle is better than Plato.

*d. If Plato is good, then his successor and successor's successor are good.

e. No philosopher is better than his successor.

f. Not every philosopher is better than Plato.

g. If all philosophers are good, then Plato and Aristotle are good.

h. If neither Plato nor his successor are good, then no philosopher is good.

*i. If some philosopher is better than Plato, then Aristotle is.

j. If every philosopher is better than his successor, then no philosopher is better than Plato.

E5.18. On page 168 we say that we may show directly, based on branch conditions, that the alternatives of table (AM) have the same truth conditions, but show it only for the last case. Use trees to demonstrate that the other alternatives are true under the same conditions. Be sure to explain how your trees have the desired results.

5.3.2 Complex Quantifications

With a small change to our interpretation function, we introduce a new sort of complexity into our translations. Suppose U includes not just all dogs, but all physical objects, so that our interpretation function II has,

$$\begin{array}{lrl} \mathsf{II} & \mathsf{U}\text{:} & \{\mathsf{o} \mid \mathsf{o} \text{ is a physical object}\} \\ & W^1\text{:} & \{\mathsf{o} \mid \mathsf{o} \in \mathsf{U} \text{ and } \mathsf{o} \text{ will have its day}\} \\ & D^1\text{:} & \{\mathsf{o} \mid \mathsf{o} \in \mathsf{U} \text{ and } \mathsf{o} \text{ is a dog}\} \end{array}$$

Thus the universe includes more than dogs, and D is a relation symbol with application to dogs. We set out to translate the same sentences as before.[6]

(1) Every dog will have its day

(2) Some dog will have its day

(3) Some dog will not have its day

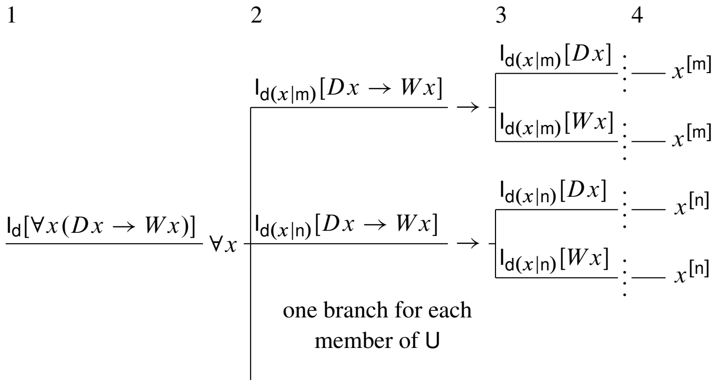
(4) No dog will have its day

This time, $\forall x Wx$ does *not* say that every dog will have its day. $\forall x Wx$ is true just in case everything in U, dogs along with everything else, will have its day. So it might be that every *dog* will have its day even though something else, for example my left sock, does not. So $\forall x Wx$ is not a good translation of 'Every dog will have its day'.

[6]Sentences of the sort, 'all $\mathcal{P}$ are $\mathcal{Q}$', 'no $\mathcal{P}$ are $\mathcal{Q}$', 'some $\mathcal{P}$ are $\mathcal{Q}$', and 'some $\mathcal{P}$ are not $\mathcal{Q}$' are, in a tradition reaching back to Aristotle, often associated with a "square of opposition" and called A, E, I, and O sentences (see, for example, Chapter 4 of Hurley, *A Concise Introduction to Logic*). In a context with the full flexibility of quantifier languages, there is little point to the special treatment, insofar as our methods apply to these as well as to ones that are more complex.

We do better with $\forall x(Dx \to Wx)$. $\forall x(Dx \to Wx)$ is read, 'for any x if x is a dog, then x will have its day'; it is true just in case every dog will have its day. Again, suppose $\mathsf{I}\mathsf{I}_\omega$ is an interpretation I such that the elements of U are $\mathsf{m}, \mathsf{n}, \ldots$.

(AP)

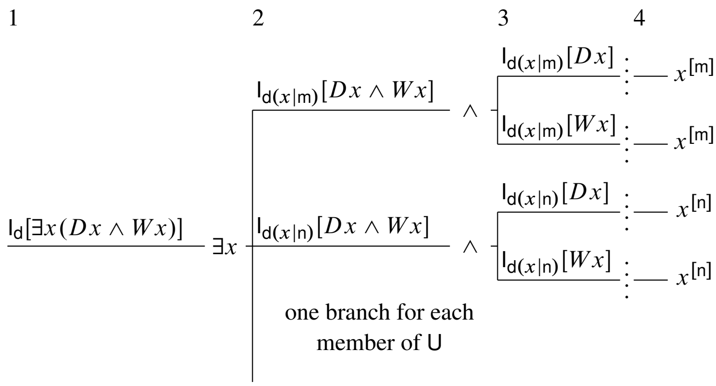


The formula at (1) is satisfied just in case each of the branches at (2) is satisfied. And all the branches at (2) are satisfied just in case there is no S/N pair at (3). This is so just in case nothing in U is a dog that does not have its day; that is, just in case every dog has its day. It is important to see how this works: There is a branch at (2) for *each* thing in U. The key is that branches for things that are not dogs are "vacuously" satisfied *just because the things are not dogs*. If $\forall x(Dx \to Wx)$ is true, however, whenever a branch is for a thing that is a dog—so that a top branch of a pair at (3) is satisfied—that thing must be one that will have its day. If anything is a dog that does not have its day, there is a S/N pair at (3), and $\forall x(Dx \to Wx)$ is not satisfied and not true.

It is worth noting some expressions that do not result in a good translation. $\forall x(Dx \wedge Wx)$ is true just in case everything is a dog that will have its day. To make it false, all it takes is one thing that is not a dog, or one thing that will not have its day—but this is not what we want. If this is not clear, work it out on a tree. Similarly, $\forall x Dx \to \forall x Wx$ is true just in case *if* everything is a dog, then everything will have its day. To make it true, all it takes is one thing that is not a dog—then the antecedent is false, and the conditional is true; but again, this is not what we want. In the good translation, $\forall x(Dx \to Wx)$, the quantifier picks out each thing in U, the antecedent of the conditional identifies the ones we want to talk about, and the consequent says what we want to say about them.

Moving on to the second sentence, $\exists x(Dx \wedge Wx)$ is read, 'there is an x such that x is a dog, and x will have its day'; it is true just in case some dog will have its day.

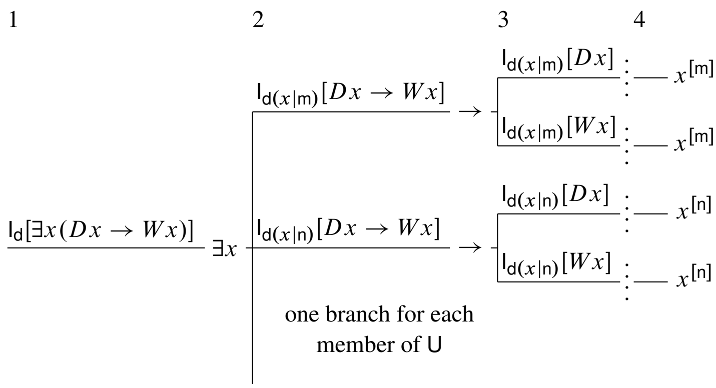
(AQ)



The formula at (1) is satisfied just in case one of the branches at (2) is satisfied. A branch at (2) is satisfied just in cases both branches in the corresponding pair at (3) are satisfied. And this is so just in case something is a dog that will have its day.

Again, it is worth noting expressions that do not result in good translation. $\exists x Dx \wedge \exists x Wx$ is true just in case something is a dog, and something will have its day—where these need not be the same; so $\exists x Dx \wedge \exists x Wx$ might be true even though no *dog* has its day. $\exists x(Dx \rightarrow Wx)$ is true just in case something is such that *if* it is a dog, then it will have its day.

(AR)

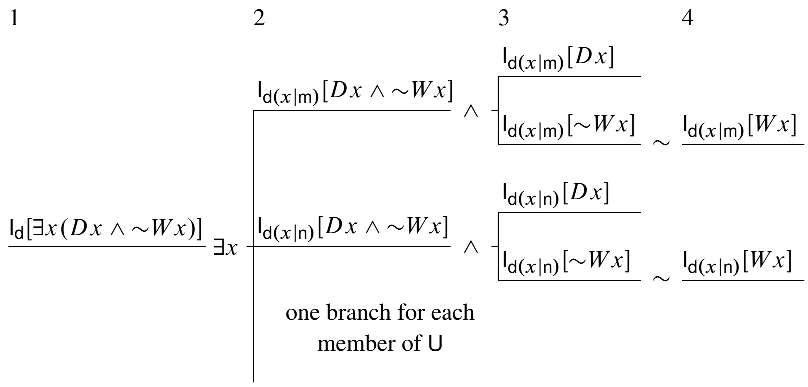


The formula at (1) is satisfied just in case one of the branches at (2) is satisfied; and a branch at (2) is satisfied just in case there is a pair at (3) in which the top is N or the bottom is S. So all we need for $\exists x(Dx \rightarrow Wx)$ to be true is for there to be even one thing that is not a dog—for example, my sock—or one thing that will have its day. So $\exists x(Dx \rightarrow Wx)$ can be true though no dog has its day.

The cases we have just seen are typical. Ordinarily, the existential quantifier operates on expressions with main operator $\wedge$. If it operates on an expression with main operator $\rightarrow$, the resultant expression is satisfied just by virtue of something that does not satisfy the antecedent. And, ordinarily, the universal quantifier operates on expressions with main operator $\rightarrow$. If it operates on an expression with main operator $\wedge$, the expression is satisfied only if *everything* in U has features from both parts of the conjunction—and it is uncommon to say something about everything in U, as opposed to all the objects of a certain sort. Again, when the universal quantifier operates on an expression with main operator $\rightarrow$, the antecedent of the conditional identifies the objects we want to talk about, and the consequent says what we want to say about them.

Once we understand these two cases, the next two are relatively straightforward. $\exists x(Dx \wedge \sim Wx)$ is read, 'there is an x such that x is a dog and x will not have its day'; it is true just in case some dog will not have its day. Here is the tree without branches for the (by now obvious) term assignments:

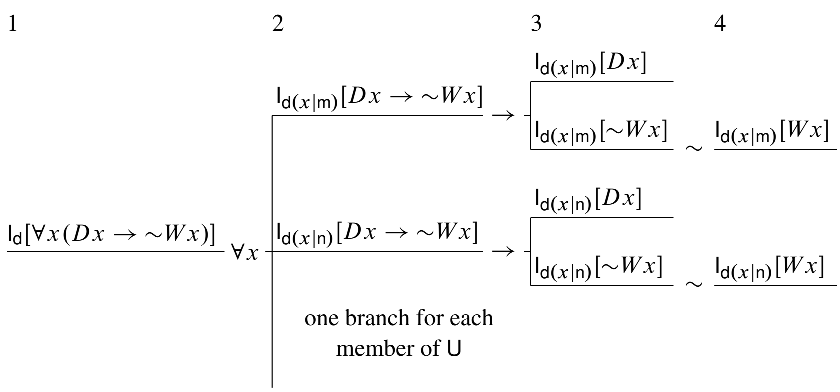
(AS)



The formula at (1) is satisfied just in case some branch at (2) is satisfied. A branch at (2) is satisfied just in case the corresponding pair of branches at (3) is satisfied. And for a lower branch at (3) to be satisfied, the corresponding branch at (4) has to be unsatisfied. So for $\exists x(Dx \wedge \sim Wx)$ to be satisfied, there has to be something that is a dog and does not have its day. In principle, this is just like 'Some dog will have its day'. We set out to say that some object of sort $\mathcal{P}$ has feature $\mathcal{Q}$. For this, we say that there is an x that is of type $\mathcal{P}$, and has feature $\mathcal{Q}$. In 'Some dog will have its day', $\mathcal{Q}$ is the simple Wx. In this case, $\mathcal{Q}$ is the slightly more complex $\sim Wx$.

Finally, $\forall x(Dx \rightarrow \sim Wx)$ is read, 'for any x, if x is a dog, then x will not have its day'; it is true just in case every dog will not have its day—that is, just in case no dog will have its day.

(AT)



The formula at (1) is satisfied just in case every branch at (2) is satisfied. Every branch at (2) is satisfied just in case there is no S/N pair at (3); and for this to be so there cannot be a case where a top at (3) is satisfied, and the corresponding bottom at (4) is satisfied as well. So $\forall x(Dx \rightarrow \sim Wx)$ is satisfied and true just in case nothing is a dog that will have its day. Again, in principle, this is like 'Every dog will have its day'. Using the universal quantifier, we pick out the class of things we want to talk about in the antecedent, and say what we want to say about the members of the class in the consequent. In this case, what we want to say is that things in the class will not have their day.

As before, quantifier-switching alternatives are possible. In the table below, alternatives to what we have done are listed on the right.

(AU)	Every dog will have its day	$\forall x(Dx \rightarrow Wx)$	$\sim\exists x(Dx \wedge \sim Wx)$
	Some dog will have its day	$\exists x(Dx \wedge Wx)$	$\sim\forall x(Dx \rightarrow \sim Wx)$
	Some dog will not have its day	$\exists x(Dx \wedge \sim Wx)$	$\sim\forall x(Dx \rightarrow Wx)$
	No dog will have its day	$\forall x(Dx \rightarrow \sim Wx)$	$\sim\exists x(Dx \wedge Wx)$

Beginning from the bottom, if not even one thing is a dog that will have its day, then no dog will have its day. Moving up, if it is not the case that everything that is a dog will have its day, then some dog will not. Similarly, if it is not the case that everything that is a dog will not have its day, then some dog does. And if not even one thing is a dog that does not have its day, then every dog will have its day. Again, choices among the alternatives are a matter of taste, though the bottom alternatives may be more natural than ones above. If you have any questions about how the alternatives work, work them through on trees.

Before turning to some exercises, let us generalize what we have done a bit. Include in our interpretation function,

H^1: {o | o $\in$ U and o is happy}

C^1: {o | o $\in$ U and o is a cat}

Suppose we want to say, not that every dog will have its day, but that every happy dog will have its day. Again, in principle this is like what we have done. With the universal quantifier, we pick out the class of things we want to talk about in the antecedent—in this case happy dogs—and say what we want about them in the consequent. Thus $\forall x[(Dx \wedge Hx) \rightarrow Wx]$ is true just in case everything that is both happy and a dog will have its day, which is to say, every happy dog will have its day. Similarly, if we want to say that every dog will or will not have its day, we might try, $\forall x[Dx \rightarrow (Wx \vee \sim Wx)]$. Or putting these together, for 'Every happy dog will or will not have its day', $\forall x[(Dx \wedge Hx) \rightarrow (Wx \vee \sim Wx)]$. We consistently pick out the things we want to talk about in the antecedent, and say what we want about them with the consequent. Similar points apply to the existential quantifier. Thus 'Some happy dog will have its day' has natural translation, $\exists x[(Dx \wedge Hx) \wedge Wx]$—something is a happy dog and will have its day. ‘Some happy dog will or will not have its day’ gets, $\exists x[(Dx \wedge Hx) \wedge (Wx \vee \sim Wx)]$. And so forth.

It is tempting to treat ‘All dogs and cats will have their day’ similarly with translation $\forall x[(Dx \wedge Cx) \rightarrow Wx]$. But this would be a mistake. We do not want to say that everything which is a dog *and* a cat will have its day—for nothing is both a dog and a cat! Rather, good translations are $\forall x(Dx \rightarrow Wx) \wedge \forall x(Cx \rightarrow Wx)$—all dogs will have their day *and* all cats will have their day, or the more elegant $\forall x[(Dx \vee Cx) \rightarrow Wx]$—each thing that is either a dog *or* a cat will have its day. In the happy dog case, we needed to *restrict* the class under consideration to include just happy dogs; in this dog and cat case, we are not restricting the class, but rather expanding it to include both dogs and cats. The disjunction $Dx \vee Cx$ applies to things in the broader class which includes both dogs and cats.

This dog and cat case brings out the point that we do not merely "cookbook" from ordinary language to formal translations, but rather want truth conditions to match. And we can make the conditions match for expressions where standard language does not lie directly on the surface. Thus consider ‘Only dogs will have their day’. This does *not* say that all dogs will have their day. Rather it tells us that anything that has its day is a dog, $\forall x(Wx \rightarrow Dx)$. Similarly, ‘Leaving out the happy ones, no dogs will have their day’, tells us that dogs other than the happy ones do not have their day, $\forall x[(Dx \wedge \sim Hx) \rightarrow \sim Wx]$. ‘Except’ has a similar effect as in, ‘Excepting the happy ones, no dogs will have their day’. It is tempting to add that the happy dogs will have their day, but it is not clear that this is part of what we have actually *said;* ‘except’ seems precisely to *except* members of the specified class from what is said.[7]

Further, as in the dog and cat case, sometimes surface language is positively misleading compared to standard readings. Consider, for example, ‘If some dog is happy, it will have its day’. It is tempting to translate, $\exists x[(Dx \wedge Hx) \rightarrow Wx]$—but this is not right. All it takes to make this expression true is something that is not a happy dog (for example, my sock); if something is not a happy dog, then a conditional branch is satisfied, so that the existentially quantified expression is satisfied. But we want rather to say something about *all* dogs—if some (*arbitrary*) dog is happy it will have its day—so that no matter what dog you pick, if it is happy then it will have its day; thus the correct translation is $\forall x[(Dx \wedge Hx) \rightarrow Wx]$. Or again, consider ‘If any dog is happy, then they all are’. It is tempting to translate by the universal quantifier. But the correct translation is rather, $\exists x(Dx \wedge Hx) \rightarrow \forall x(Dx \rightarrow Hx)$—if *some* dog is happy, then every dog is happy. The best way to approach these cases is to think directly about the *conditions* under which the ordinary expressions are true and false, and to produce formal translations that are true and false under the same conditions. For these last cases however, it is worth noting that when there is "pronominal" cross reference as, ‘if some/any $\mathcal{P}$ is $\mathcal{Q}$ then *it* has such-and-such features’ the statement

[7] It may be that we conventionally use ‘except’ in contexts where the consequent is reversed for the excepted class, for example, ‘I like all foods except brussels sprouts’—where I say it *because* I do not like brussels sprouts. But, again, it is not clear that I have actually said whether I like them or not.

translates most naturally with the universal quantifier. But when such cross-reference is absent as, 'if some/any $\mathscr{P}$ is $\mathscr{Q}$ then so-and-so is such-and-such' the statement translates naturally as a conditional with an existential antecedent. The point is not that there are no grammatical cues! But cues are not so simple that we can always simply read from 'some' to the existential quantifier, and from 'any' to the universal. Perhaps this is sufficient for us to move to the following exercises.

E5.19. Given the following partial interpretation function for $\mathscr{L}_q$, complete the translation for each of the following. (Perhaps these sentences reflect residual frustration over a Mustang the author owned in graduate school.)

U: $\{\mathsf{o} \mid \mathsf{o}$ is a car$\}$

T^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a Toyota$\}$

F^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a Ford$\}$

E^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o was built in the eighties$\}$

J^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a piece of junk$\}$

R^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is reliable$\}$

a. Some Ford is a piece of junk.

*b. Some Ford is an unreliable piece of junk.

c. Some Ford built in the eighties is a piece of junk.

d. Some Ford built in the eighties is an unreliable piece of junk.

e. Any Ford is a piece of junk.

f. Any Ford is an unreliable piece of junk.

*g. Any Ford built in the eighties is a piece of junk.

h. Any Ford built in the eighties is an unreliable piece of junk.

i. No reliable car is a piece of junk.

j. No Toyota is an unreliable piece of junk.

*k. If a car is unreliable, then it is a piece of junk.

l. If some Toyota is unreliable, then every Ford is.

m. Only Toyotas are reliable.

n. Not all Toyotas and Fords are reliable.

o. Any car, except for a Ford, is reliable.

E5.20. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. Assume that Bob is married, and that each married person has a unique "primary" spouse in case of more than one.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a person who is married$\}$

b: Bob

s^1: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the (primary) spouse of $\mathsf{m}\}$

A^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is having an affair$\}$

E^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is employed$\}$

H^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is happy$\}$

L^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m loves $\mathsf{n}\}$

M^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m is married to $\mathsf{n}\}$

a. Bob's spouse is happy.

*b. Someone is married to Bob.

c. Anyone who loves their spouse is happy.

d. Nobody who is happy and loves their spouse is having an affair.

e. Someone is happy just in case they are employed.

f. Someone is happy just in case someone is employed.

g. Some happy people are having an affair, and some are not.

*h. Anyone who loves and is loved by their spouse is happy, though some are not employed.

i. Only someone who loves their spouse and is employed is happy.

j. Anyone who is unemployed and whose spouse is having an affair is unhappy.

k. If someone is both unemployed and unhappy, then their spouse is having an affair.

*l. Anyone married to Bob is happy if Bob is not having an affair.

m. If anyone married to Bob is happy then Bob is employed and is not having an affair.

n. If Bob is having an affair, then everyone married to him is unhappy, and nobody married to him loves him.

o. Only unemployed people and unhappy people have affairs, but if someone loves and is loved by their spouse, then they are happy unless they are unemployed.

E5.21. Produce a good quantificational translation for each of the following. You should produce a single interpretation function with application to all of the sentences. Let U be the set of all animals.

a. Not all animals make good pets.

b. Dogs and cats make good pets.

c. Some dogs are ferocious and make good pets, but no cat is both.

d. No ferocious animal makes a good pet, unless it is a dog.

e. No ferocious animal makes a good pet, unless Lassie is both.

f. Some, but not all good pets are dogs.

g. Only dogs and cats make good pets.

h. Not all dogs and cats make good pets, but some of them do.

i. If Lassie does not make a good pet, then the only good pet is a cat that is ferocious, or a dog that is not.

j. A dog or cat makes a good pet if and only if it is not ferocious.

E5.22. Use trees to show that the quantifier-switching alternatives from (AU) are true and false under the same conditions as their counterparts. Be sure to explain how your trees have the desired results.

5.3.3 Overlapping Quantifiers

The full power of our quantificational languages emerges only when we allow one quantifier to appear in the scope of another.[8] So let us turn to some cases of this sort. First, let U be the set of all people, and suppose the intended interpretation of L^2 is $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$, and m loves $\mathsf{n}\}$. Say we want to translate,

(1) Everyone loves everyone.

(2) Someone loves someone.

[8] Aristotle's categorical logic is capable of handling simple A, E, I, and O sentences—consider experience you may have had with "Venn diagrams." But you will not be able to make his logic, or such diagrams, apply to the full range of cases that follow (see note 6 on page 171).

(3) Everyone loves someone.

(4) Everyone is loved by someone.

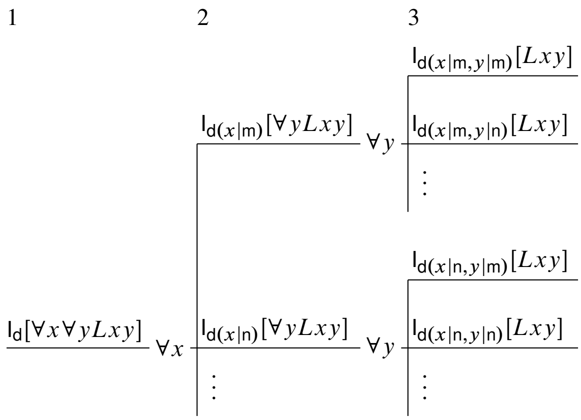
(5) Someone loves everyone.

(6) Someone is loved by everyone.

First, you should be clear how each of these differs from the others. In particular, it is enough for (3) ‘Everyone loves someone’ that each person loves some person—perhaps their mother (or themselves); but for (6) ‘Someone is loved by everyone’ we need some one person, say Elvis, that everyone loves. Similarly, it is enough for (4) ‘Everyone is loved by someone’ that for each person there is a lover of them—perhaps their mother (or themselves); but for (5) ‘Someone loves everyone’ we need some particularly loving individual, say Mother Theresa, who loves everyone.

The first two are straightforward. $\forall x \forall y Lxy$ is read, ‘for any x and any y, x loves y’; it is true just in case everyone loves everyone.

(AV)

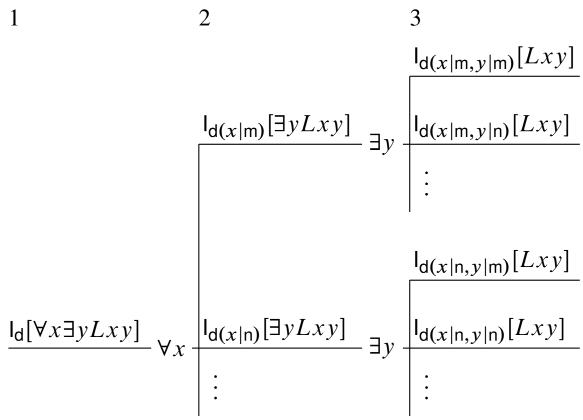


The branch at (1) is satisfied just in case all of the branches at (2) are satisfied. And all of the branches at (2) are satisfied just in case all of the branches at (3) are satisfied. But every combination of objects appears at the branch tips. So $\forall x \forall y Lxy$ is satisfied and true just in case for any pair $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{U}^2$, $\langle \mathsf{m}, \mathsf{n} \rangle$ is in the interpretation of L. Notice that the order of the quantifiers and variables makes no difference: for a given interpretation I, $\forall x \forall y Lxy$, $\forall x \forall y Lyx$, $\forall y \forall x Lxy$, and $\forall y \forall x Lyx$ are all satisfied and true under the same condition—just when every $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{U}^2$ is a member of $\mathsf{I}[L]$.

The case for the second sentence is similar. $\exists x \exists y Lxy$ is read, ‘there is an x and there is a y such that x loves y’; it is true just in case some $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{U}^2$ is a member of $\mathsf{I}[L]$—just in case someone loves someone. The tree is like (AV) above, but with $\exists$ uniformly substituted for $\forall$. Then the formula at (1) is satisfied iff a branch at (2) is satisfied; iff a branch at (3) is satisfied; iff someone loves someone. Again the order of the quantifiers does not matter.

The next cases are more interesting. $\forall x \exists y Lxy$ is read, ‘for any x there is a y such that x loves y’; it is true just in case everyone loves someone.

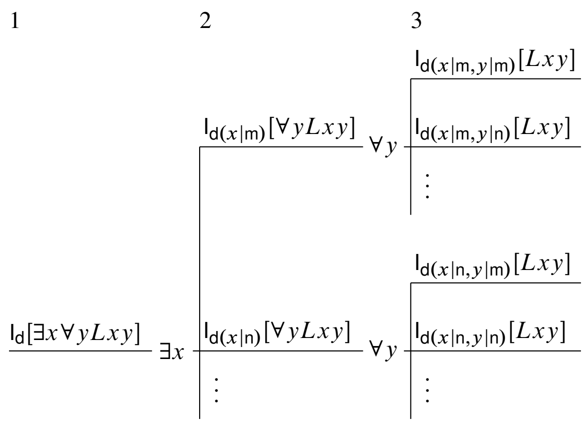
(AW)



The branch at (1) is satisfied just in case each of the branches at (2) is satisfied. And a branch at (2) is satisfied just in case at least one of the corresponding branches at (3) is satisfied. So $\forall x\exists yLxy$ is satisfied just in case, no matter which o you pick, there is some p such that such that o loves p—so that everyone loves someone. This time, the order of the of the variables makes a difference: Thus $\forall x\exists yLyx$ translates sentence (4), 'Everyone is loved by someone'. The picture is like the one above, with Lyx uniformly replacing Lxy. This expression is satisfied just in case no matter which o you pick, there is some p such that such that p loves o—so that everyone is loved by someone.

Finally, $\exists x\forall yLxy$ is read, 'there is an x such that for any y, x loves y'; it is satisfied and true just in case someone loves everyone.

(AX)



The branch at (1) is satisfied just in case some branch at (2) is satisfied. And a branch at (2) is satisfied just in case *each* of the corresponding branches at (3) is satisfied. So $\exists x\forall yLxy$ is satisfied and true just in case there is some $\mathsf{o} \in \mathsf{U}$ such that, no matter what $\mathsf{p} \in \mathsf{U}$ you pick, $\langle \mathsf{o}, \mathsf{p} \rangle \in \mathsf{I}[L]$—just when there is someone who loves everyone. If we switch Lyx for Lxy, we get a tree for $\exists x\forall yLyx$; this formula is true just when someone is loved by everyone. Switching the order of the quantifiers and variables makes no difference when quantifiers are the same. But it matters crucially when quantifiers are different!

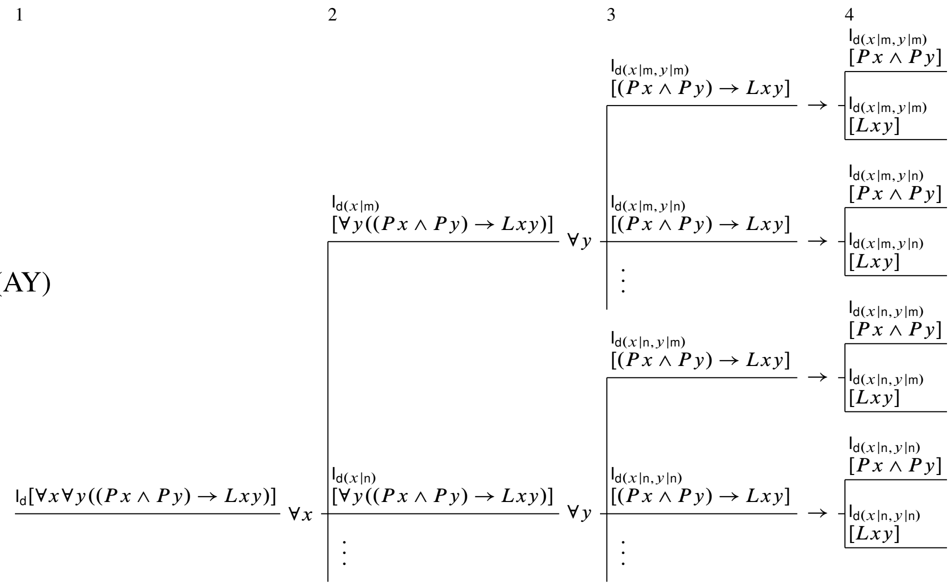
Let us see what happens when, as before, we broaden the interpretation function so that U includes all physical objects.

$$\begin{array}{lrl} \mathsf{II} & \mathsf{U}\text{:} & \{\mathsf{o} \mid \mathsf{o} \text{ is a physical object}\} \\ & P^1\text{:} & \{\mathsf{o} \mid \mathsf{o} \in \mathsf{U} \text{ and } \mathsf{o} \text{ is a person}\} \\ & L^2\text{:} & \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}, \text{ and } \mathsf{m} \text{ loves } \mathsf{n}\} \end{array}$$

Let us set out to translate the same sentences as before.

For 'Everyone loves everyone', where we are talking about *people*, $\forall x \forall y Lxy$ will not do. $\forall x \forall y Lxy$ requires that each member of U love all the other members of U—but then we are requiring that my left sock love my computer, and so forth. What we need is rather, $\forall x \forall y[(Px \wedge Py) \to Lxy]$. With the last branch tips omitted, the tree is as follows:

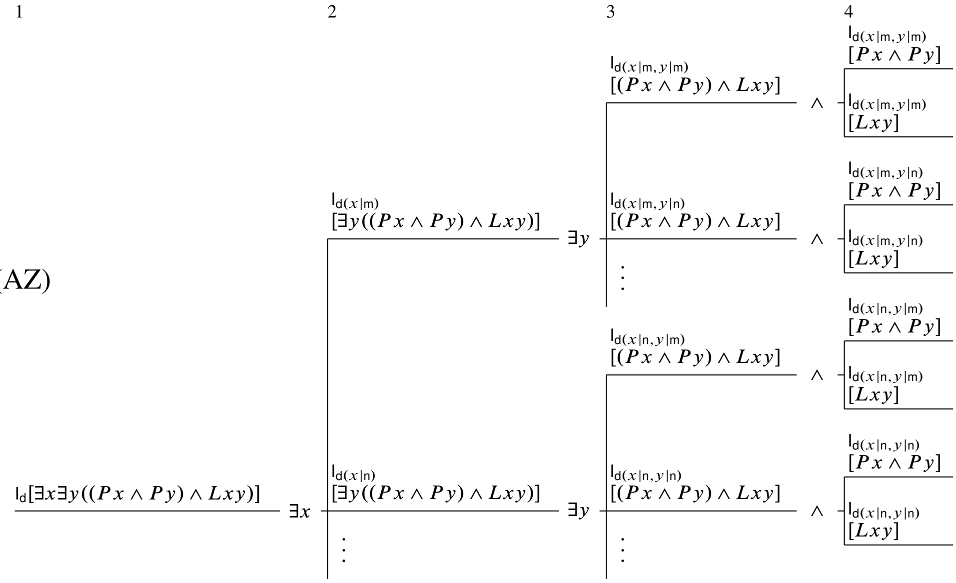
(AY)



The formula at (1) is satisfied iff all the branches at (2) are satisfied; all the branches at (2) are satisfied just in case all the branches at (3) are satisfied. And, for this to be the case, there can be no pair at (4) where the top is satisfied and the bottom is not. That is, there can be no o and p such that o and p are people, $\mathsf{o}, \mathsf{p} \in \mathsf{I}[P]$, but o does not love p, $\langle \mathsf{o}, \mathsf{p}\rangle \notin \mathsf{I}[L]$. The idea is very much as before: With the universal quantifiers, we select the things we want to talk about in the antecedent, we make sure that x and y pick out *people*, and then say what we want to say about the things in the consequent.

The case for 'Someone loves someone' also works on close analogy with what has gone before. In this case, we do not use the conditional. If the quantifiers in (AY) were existential, all we would need is *one* branch at (2) to be satisfied, and *one* branch at (3) satisfied. And, for this, all we would need is one thing that is not a person—so that the top branch for the conditional is N, and the conditional is therefore S. On the analogy with what we have seen before, what we want is something like, $\exists x \exists y[(Px \wedge Py) \wedge Lxy]$. There are some *people* x and y such that x loves y.

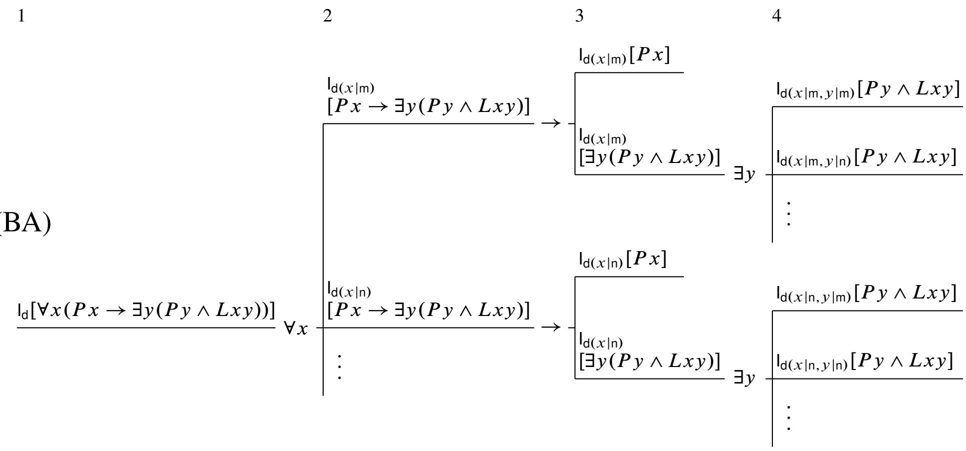
(AZ)



The formula at (1) is satisfied iff at least one branch at (2) is satisfied. At least one branch at (2) is satisfied just in case at least one branch at (3) is satisfied. And for this to be the case, we need some branch pair at (4) where both the top and the bottom are satisfied—some o and p such that o and p are people, $\mathsf{o}, \mathsf{p} \in \mathsf{I}[P]$, and o loves p, $\langle \mathsf{o}, \mathsf{p} \rangle \in \mathsf{I}[L]$.

In these cases, the order of the quantifiers and variables does not matter. But order matters when quantifiers are mixed. Thus for 'Everyone loves someone', $\forall x[Px \rightarrow \exists y(Py \wedge Lxy)]$ is good—if any thing x is a person, then there is some y such that y is a person and x loves y.

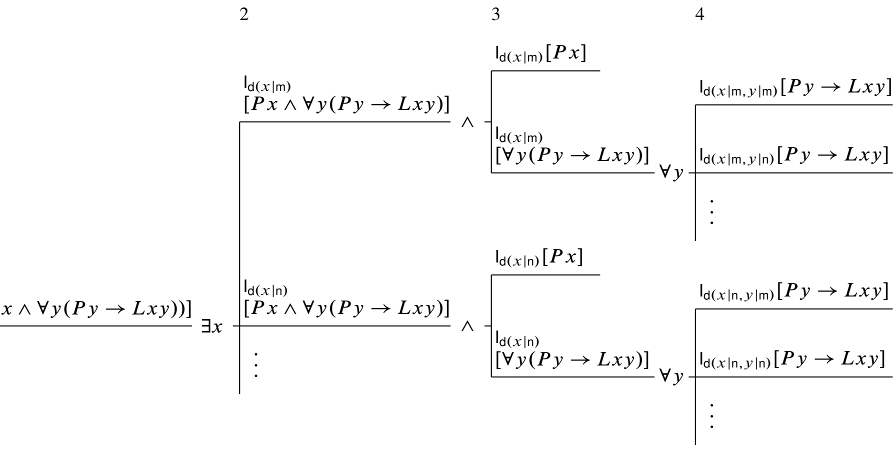
(BA)



The formula at (1) is satisfied just in case all the branches at (2) are satisfied. All the branches at (2) are satisfied just in case no pair at (3) has the top satisfied and the bottom not. If x is assigned to something that is not a person, the branch at (2) is satisfied trivially. But where the assignment to x is some o that is a person, a bottom branch at (3) is satisfied just in case at least one of the corresponding branches at (4) is satisfied—just in case there is some p such that p is a person and o loves p. Notice, again, that the universal quantifier is associated with a conditional, and the existential with a conjunction. Similarly we translate ‘Everyone is loved by someone’, $\forall x[Px \rightarrow \exists y(Py \wedge Lyx)]$. The tree is as above, with Lxy uniformly replaced by Lyx.

For ‘Someone loves everyone’, $\exists x[Px \wedge \forall y(Py \rightarrow Lxy)]$ is good—there is an x such that x is a person, and for any y, if y is a person, then x loves y.

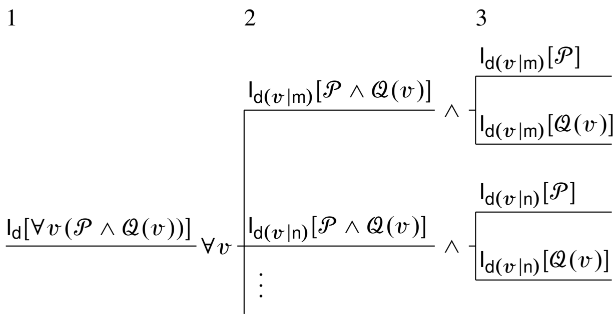
(BB)



The formula at (1) is satisfied just in case some branch at (2) is satisfied. A branch at (2) is satisfied just in case the corresponding pair at (3) is satisfied. The top of such a pair is satisfied when the assignment to x is some $\mathsf{o} \in \mathsf{I}[P]$; the bottom is satisfied just in case all of the corresponding branches at (4) are satisfied—just in case any p is such that *if* it is a person, then o loves it. So there has to be a person o that loves every person p. Similarly, you should be able to see that $\exists x[Px \wedge \forall y(Py \rightarrow Lyx)]$ is good for ‘Someone is loved by everyone’.

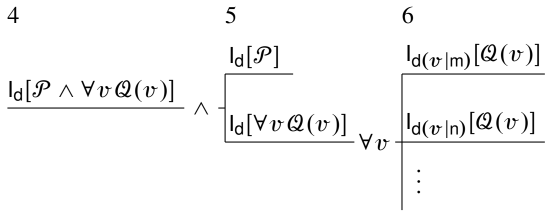
Again, it may have occurred to you already that there are other options for these sentences. This time natural alternatives are not for quantifier switching, but for quantifier *placement*. For ‘Someone loves everyone’ we have given, $\exists x[Px \wedge \forall y(Py \rightarrow Lxy)]$ with the universal quantifier on the inside. However, $\exists x \forall y[Px \wedge (Py \rightarrow Lxy)]$ would do as well. As a matter of strategy, it is best to keep quantifiers as close as possible to that which they modify. However, we can show that, in this case, pushing the quantifier across that which it does not bind leaves the truth condition unchanged. Let us make the point generally. Say $\mathcal{Q}(v)$ is a formula with variable v free, but $\mathcal{P}$ is one in which v is not free. We are interested in the relation between $\mathcal{P} \wedge \forall v \mathcal{Q}(v)$ and $\forall v(\mathcal{P} \wedge \mathcal{Q}(v))$. Here are the trees:

(BC)



and,

(BD)



The key is this: Since $\mathscr{P}$ has no free instances of v, for any $o \in U$, $I_d[\mathscr{P}]$ is satisfied just in case $I_{d(v|o)}[\mathscr{P}]$ is satisfied; for if v is not free in $\mathscr{P}$, then d's assignment to v makes no difference to the evaluation of $\mathscr{P}$. In (BC), the formula at (1) is satisfied iff each of the branches at (2) is satisfied; and each of the branches at (2) is satisfied iff each of the branches at (3) is satisfied. In (BD) the formula at (4) is satisfied iff both branches at (5) are satisfied; and the bottom at (5) requires that all the branches at (6) are satisfied. But the branches at (6) are just like the bottom branches from (3) in (BC); and given the equivalence between $I_d[\mathscr{P}]$ and $I_{d(v|o)}[\mathscr{P}]$, the top at (5) is satisfied iff each of the tops at (3) is satisfied. So all the branches at (3) are satisfied iff the top at (5) and all the branches at (6) are satisfied; so the one formula is satisfied iff the other is as well. Notice that this only works because v is not free in $\mathscr{P}$ and $I_d[\mathscr{P}] = I_{d(v|o)}[\mathscr{P}]$. You can move the quantifier past the $\mathscr{P}$ only if it does not bind a variable free in $\mathscr{P}$!

Parallel reasoning would work for any combination of $\forall$ and $\exists$, with $\wedge$, $\vee$, and $\rightarrow$. That is, supposing that v is not free in $\mathscr{P}$, each of the following pairs is equivalent.

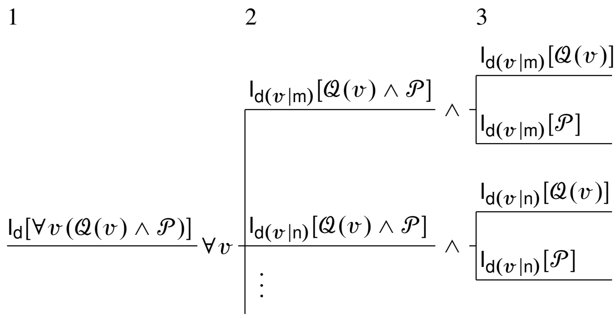
(BE)

$$\begin{array}{lcl}
\forall v(\mathscr{P} \wedge \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \wedge \forall v \mathscr{Q}(v) \\
\exists v(\mathscr{P} \wedge \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \wedge \exists v \mathscr{Q}(v) \\
\forall v(\mathscr{P} \vee \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \vee \forall v \mathscr{Q}(v) \\
\exists v(\mathscr{P} \vee \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \vee \exists v \mathscr{Q}(v) \\
\forall v(\mathscr{P} \rightarrow \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \rightarrow \forall v \mathscr{Q}(v) \\
\exists v(\mathscr{P} \rightarrow \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \rightarrow \exists v \mathscr{Q}(v)
\end{array}$$

The comparison between $\forall y[Px \wedge (Py \rightarrow Lxy)]$ and $[Px \wedge \forall y(Py \rightarrow Lxy)]$ is an instance of the first pair. In effect, then, we can "push" the quantifier into the parentheses across a formula to which the quantifier does not apply, and "pull" it out across a formula to which the quantifier does not apply—without changing the conditions under which the formula is satisfied.

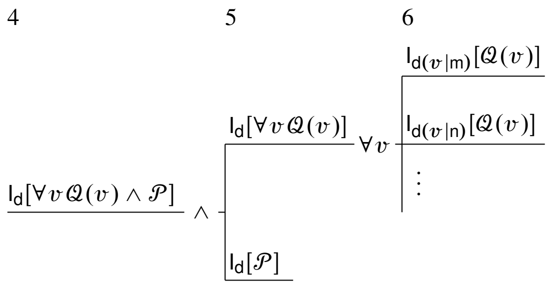
But we need to be more careful when the order of $\mathcal{P}$ and $\mathcal{Q}(v)$ is reversed. Some cases work the way we expect. Consider $\forall v(\mathcal{Q}(v) \wedge \mathcal{P})$ and $\forall v\mathcal{Q}(v) \wedge \mathcal{P}$.

(BF)



and,

(BG)



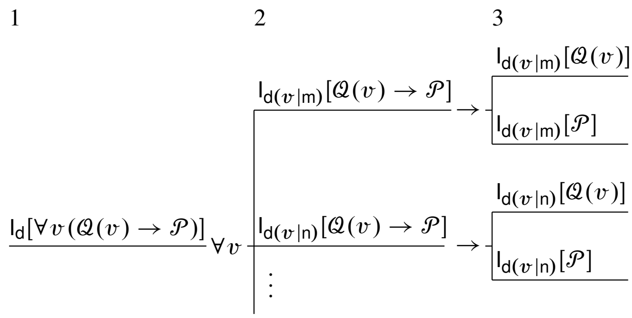
In this case, the reasoning is as before. In (BF), the formula at (1) is satisfied iff all the branches at (2) are satisfied; and all the branches at (2) are satisfied iff all the branches at (3) are satisfied. In (BG), the formula at (4) is satisfied iff both branches at (5) are satisfied; and the top at (5) is satisfied iff all the branches at (6) are satisfied. But the branches at (6) are like the tops at (3); and given the equivalence between $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ and $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{P}]$, the bottom at (5) is satisfied iff the bottoms at (3) are satisfied. So all the branches at (3) are satisfied iff the bottom at (5) and all the branches at (6) are satisfied; so, again, the formulas are satisfied under the same conditions. And similarly for different combinations of the quantifiers $\forall$ or $\exists$ and the operators $\wedge$ or $\vee$. Thus our table extends as follows:

(BH)
$$\begin{array}{lcl}
\forall v(\mathcal{Q}(v) \wedge \mathcal{P}) & \Longleftrightarrow & \forall v\mathcal{Q}(v) \wedge \mathcal{P} \\
\exists v(\mathcal{Q}(v) \wedge \mathcal{P}) & \Longleftrightarrow & \exists v\mathcal{Q}(v) \wedge \mathcal{P} \\
\forall v(\mathcal{Q}(v) \vee \mathcal{P}) & \Longleftrightarrow & \forall v\mathcal{Q}(v) \vee \mathcal{P} \\
\exists v(\mathcal{Q}(v) \vee \mathcal{P}) & \Longleftrightarrow & \exists v\mathcal{Q}(v) \vee \mathcal{P}
\end{array}$$

We can push a quantifier "into" the front part of a parenthesis or pull it out as above.

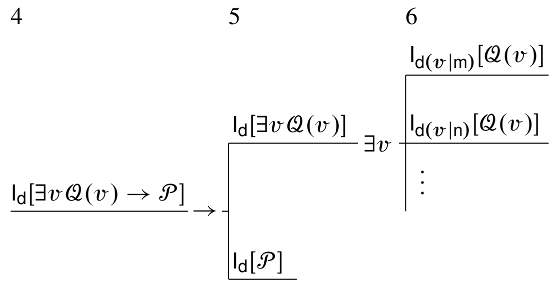
But the case is different when the inner operator is $\rightarrow$. Consider trees for $\forall v(\mathcal{Q}(v) \rightarrow \mathcal{P})$ and, noting the quantifier switch, for $\exists v\mathcal{Q}(v) \rightarrow \mathcal{P}$.

(BI)



and

(BJ)



Starting with (BJ), the formula at (4) is satisfied so long as at (5) the upper branch is N or bottom is S; and the top is N iff no branch at (6) is S; thus the formula at (4) is satisfied so long as none of the branches at (6) are S or the bottom at (5) is S; or, put the other way around, the formula at (4) is N iff one of the branches at (6) is S and the bottom at (5) is N. The formula at (1) is satisfied iff all the branches at (2) are satisfied; and all the branches at (2) are satisfied iff there is no S/N pair at (3); so the formula at (1) is N iff there is an S/N pair at (3). But, as before, the tops at (3) are the same as the branches at (6); and given the match between $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ and $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{P}]$, the bottoms at (3) are the same as the bottom at (5). So there is an S/N pair at (3) iff some branch at (6) is S and the bottom at (5) is N. So $\forall v(\mathcal{Q}(v) \to \mathcal{P})$ and $\exists v\,\mathcal{Q}(v) \to \mathcal{P}$ are (not) satisfied under the same conditions. By similar reasoning, we are left with the following equivalences to complete our table:

(BK)
$$\begin{array}{lcl} \forall v(\mathcal{Q}(v) \to \mathcal{P}) & \Longleftrightarrow & \exists v\,\mathcal{Q}(v) \to \mathcal{P} \\ \exists v(\mathcal{Q}(v) \to \mathcal{P}) & \Longleftrightarrow & \forall v\,\mathcal{Q}(v) \to \mathcal{P} \end{array}$$

When a universal goes into the antecedent of a conditional, it flips to an existential. And when an existentitial quantifier goes in to the antecedent of a conditional, it flips to a universal.

Here is an explanation for what is happening: The universal quantifier of $\forall v(\mathcal{Q}(v) \to \mathcal{P})$ requires that each inner conditional branch be satisfied; with tips for $\mathcal{P}$ the same, this requires either that every antecedent tip be N or the consequent be S. But once the quantifier is pushed in, the resultant conditional $\mathcal{A} \to \mathcal{P}$ is satisfied only when the antecedent is N or the consequent is S; so the original requirement that all the antecedent tips be N or $\mathcal{P}$ be S is matched by the requirement that an *existential*

$\mathcal{A}$ be N or $\mathcal{P}$ be S. Similarly, the existential quantifier of $\exists v(\mathcal{Q}(v) \to \mathcal{P})$ requires that some inner conditional branch be satisfied; with tips for $\mathcal{P}$ the same, this requires either that some antecedent tip be N or the consequent be S. But once the quantifier is pushed in, the resultant conditional $\mathcal{A} \to \mathcal{P}$ is satisfied when the antecedent is N or the consequent is S; so the original requirement that some antecedent tip be N or $\mathcal{P}$ be S is matched by the requirement that a *universal* $\mathcal{A}$ be N or $\mathcal{P}$ be S. These cases differ from others insofar as an inner conditional branch is S when its antecedent tip is N. In the standard cases, a branch is S when the tip remains S—and the quantifiers go in as one would expect. The place for caution is when a quantifier comes from or goes into the antecedent of a conditional.[9]

Return to 'Everybody loves somebody'. We gave as a translation, $\forall x[Px \to \exists y(Py \wedge Lxy)]$. But $\forall x\exists y[Px \to (Py \wedge Lxy)]$ does as well. To see this, notice that the immediate subformula, $[Px \to \exists y(Py \wedge Lxy)]$ is of the form $[\mathcal{P} \to \exists v\mathcal{Q}(v)]$ where $\mathcal{P}$ has no free instance of the quantified variable v. The quantifier is in the consequent of the conditional, so $[Px \to \exists y(Py \wedge Lxy)]$ is equivalent to $\exists y[Px \to (Py \wedge Lxy)]$. So the larger formula $\forall x[Px \to \exists y(Py \wedge Lxy)]$ is equivalent to $\forall x\exists y[Px \to (Py \wedge Lxy)]$. And similarly in other cases. Officially, there is no reason to prefer one option over the other. Informally, however, there is less room for confusion when we keep quantifiers relatively close to the expressions they modify. One reason is that we continue to associate $\forall$ with $\to$ and $\exists$ with $\wedge$. In so doing, we avoid unexpected results from quantifier flipping. On this basis, $\forall x[Px \to \exists y(Py \wedge Lxy)]$ is to be preferred. To illustrate the point, consider 'Everyone is such that if someone loves them then they love themselves'. The natural translation is $\forall x[Px \to (\exists y(Py \wedge Lyx) \to Lxx)]$. By our principles, this is equivalent to $\forall x[Px \to \forall y((Py \wedge Lyx) \to Lxx)]$ and then $\forall x\forall y[Px \to ((Py \wedge Lyx) \to Lxx)]$. Again, the first is preferable relative to the others, with their unintuitive use of the universal y-quantifier outside parentheses.[10]

If you have followed this discussion, you are doing well—and should be in a good position to think about the following exercises.

E5.23. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. (The last generates a famous paradox—can a barber shave himself?)

[9]By similar reasoning, we should expect quantifier flipping when pushing into expressions $\forall v(\mathcal{P} \downarrow \mathcal{Q}(v))$ or $\forall v(\mathcal{Q}(v) \downarrow \mathcal{P})$ with a *neither-nor* operator true only when both sides are false. And this is just so: The universal expression is satisfied only when all the inner branches are satisfied; and all the inner branches are satisfied just when all the tips are not. And this is like the condition from the existential quantifier in $\mathcal{P} \downarrow \exists v\mathcal{Q}$ or $\exists v\mathcal{Q} \downarrow \mathcal{P}$. Observe also that we get results as above by previously established equivalences: $\forall v(\mathcal{Q}(v) \to \mathcal{P}) = \forall v(\sim\mathcal{Q}(v) \vee \mathcal{P}) = \forall v\sim\mathcal{Q}(v) \vee \mathcal{P} = \sim\exists v\mathcal{Q}(v) \vee \mathcal{P} = \exists v\mathcal{Q} \to \mathcal{P}$. The universal goes into the disjunction as we expect, but the negation flips it to existential. And similarly for other cases.

[10]And $\forall x\exists y[Px \to ((Py \wedge Lyx) \to Lxx)]$ is a mistake: It goes to $\forall x[Px \to \exists y((Py \wedge Lyx) \to Lxx)]$ and then $\forall x[Px \to (\forall y(Py \wedge Lyx) \to Lxx)]$—'Everyone is such that if *everything is a person that loves them* then they love themselves'.

U: $\{o \mid o \text{ is a person}\}$

b: Bob

B^1: $\{o \mid o \in U \text{ and } o \text{ is a barber}\}$

M^1: $\{o \mid o \in U \text{ and } o \text{ is a man}\}$

S^2: $\{\langle m, n\rangle \mid m, n \in U \text{ and } m \text{ shaves } n\}$

a. Bob shaves himself.

b. Everyone shaves everyone.

c. Someone shaves everyone.

d. Everyone is shaved by someone.

e. Someone is shaved by everyone.

f. Not everyone shaves themselves.

*g. Any man is shaved by someone.

h. Some man shaves everyone.

i. No man is shaved by all barbers.

*j. Any man who shaves everyone is a barber.

k. If someone shaves all men, then they are a barber.

l. If someone shaves everyone, then they shave themselves.

*m. A barber shaves anyone who does not shave themselves.

*n. A barber shaves only people who do not shave themselves.

o. A barber shaves all and only people who do not shave themselves.

E5.24. Produce a good quantificational translation for each of the following. In this case you should provide an interpretation function for the sentences. Let U be the set of people and, assuming that each has a unique best friend, implement a *best friend of* function.

a. Bob's best friend likes all New Yorkers.

b. Some New Yorker likes all Californians.

c. No Californian likes all New Yorkers.

d. Any Californian likes some New Yorker.

e. Californians who like themselves, like at least some people who do not.

f. New Yorkers who do not like themselves, do not like anybody.

g. Nobody likes someone who does not like them.

h. There is someone who dislikes every New Yorker, and is liked by every Californian.

i. Anyone who likes themselves and dislikes every New Yorker, is liked by every Californian.

j. Everybody who likes Bob's best friend likes some New Yorker who does not like Bob.

E5.25. (i) Use trees to explain the fourth ($\exists$ / $\vee$) equivalence in table (BE). (ii) Use trees to explain an equivalence in (BH) for an operator other than $\wedge$. Then (iii) use trees to explain the second equivalence in (BK). Be sure to explain how your trees justify the results.

E5.26. Explain why we have not listed quantifier placement equivalences matching $\forall v(\mathcal{P} \leftrightarrow \mathcal{Q}(v))$ with $(\mathcal{P} \leftrightarrow \forall v\mathcal{Q}(v))$. Hint: Consider $\forall v(\mathcal{P} \leftrightarrow \mathcal{Q}(v))$ as an abbreviation of $\forall v[(\mathcal{P} \rightarrow \mathcal{Q}(v)) \wedge (\mathcal{Q}(v) \rightarrow \mathcal{P})]$; from trees, you can see that this is equivalent to $[\forall v(\mathcal{P} \rightarrow \mathcal{Q}(v)) \wedge \forall v(\mathcal{Q}(v) \rightarrow \mathcal{P})]$. Now, what is the consequence of quantifier placement difficulties for $\rightarrow$? Would it work if the quantifier did not flip?

5.3.4 Equality

We complete our discussion of translation by turning to some important applications for equality. Adopt as an interpretation function,

II $\quad$ U: $\{\mathsf{o} \mid \mathsf{o}$ is a person$\}$

b: Bob

c: Bob

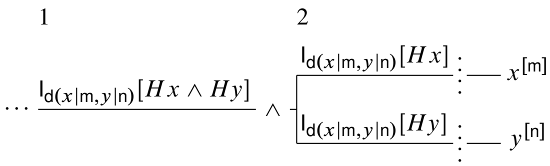
f^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$, and n is the father of $\mathsf{m}\}$

H^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a happy person$\}$

(Maybe Bob's friends call him "Cronk.") The simplest applications for $=$ assert the identity of individuals. Thus, for example, on any intended interpretation I, $b = c$ is satisfied insofar as $\langle \mathsf{I_d}[b], \mathsf{I_d}[c]\rangle \in \mathsf{I}[=]$. Similarly, $\exists x(b = f^1x)$ is satisfied just in case Bob is someone's father. And, on the standard interpretation of $\mathcal{L}_{NT}^{<}$, $\exists x[(x+x) = (x \times x)]$ is satisfied insofar as, say, $\langle \bar{\mathsf{N}}_{\mathsf{d}(x|2)}[x + x], \bar{\mathsf{N}}_{\mathsf{d}(x|2)}[x \times x]\rangle \in \bar{\mathsf{N}}[=]$—that is, $\langle 4, 4\rangle \in \bar{\mathsf{N}}[=]$. If this last case is not clear, think about it on a tree.

We get to an interesting class of cases when we turn to *quantity* expressions. Thus, for example, we can easily say 'At least one person is happy', $\exists x Hx$. But notice that neither $\exists x Hx \wedge \exists y Hy$ nor $\exists x \exists y(Hx \wedge Hy)$ work for 'At least two people are happy'. For the first, it should be clear that each conjunct is satisfied, so that the conjunction is satisfied, so long as there is at least one happy person. And similarly for the second. To see this in a simple case, suppose Bob, Sue, and Jim are the only people in U. Then the existentials for $\exists x \exists y(Hx \wedge Hy)$ result in nine branches of the following sort:

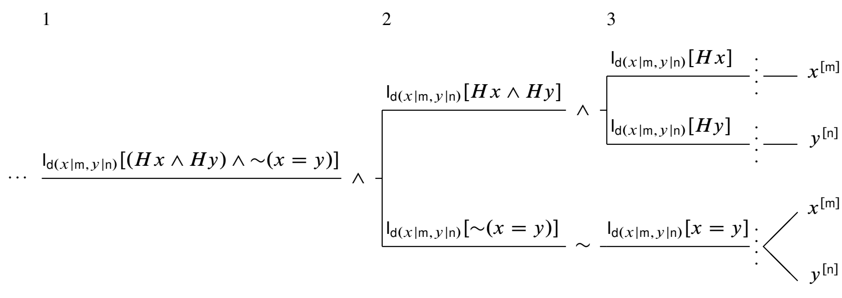
(BL)



for some individuals m and n. Just one of these branches has to be satisfied in order for the main sentence to be satisfied and true. Clearly none of the tips are satisfied if none of Bob, Sue, or Jim is happy; then the branches are N and $\exists x \exists y(Hx \wedge Hy)$ is N as well. But suppose just one of them, say Sue, is happy. Then on the branch for $\mathsf{d}_{(x|\text{Sue},y|\text{Sue})}$ *both* Hx and Hy are satisfied. Thus the conjunction is satisfied, and the existential is satisfied as well. So $\exists x \exists y(Hx \wedge Hy)$ does not require that at least two people are happy. The problem, again, is that the same person might satisfy both conjuncts at once.

But this case points the way to a good translation for 'At least two people are happy'. We get the right result with, $\exists x \exists y[(Hx \wedge Hy) \wedge {\sim}(x = y)]$. Now, in our simple example, the existentials result in nine branches as follows:

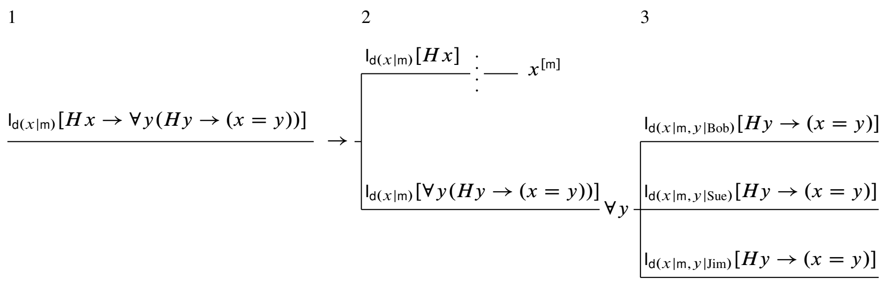
(BM)



The sentence is satisfied and true if at least one such branch is satisfied. Now in the case where just Sue is happy, on the branch with $\mathsf{d}_{(x|\text{Sue},y|\text{Sue})}$ both Hx and Hy are satisfied as before, so that the top at (2) is satisfied. But $x = y$ is satisfied; so ${\sim}(x = y)$ is not, and the branch as a whole fails. But suppose both Bob and Sue are happy. Then on the branch with $\mathsf{d}_{(x|\text{Bob},y|\text{Sue})}$ both Hx and Hy are satisfied; but this time, $x = y$ is not satisfied; so ${\sim}(x = y)$ is satisfied, and the branch is satisfied, so that the whole sentence, $\exists x\exists y[(Hx \wedge Hy) \wedge \sim(x = y)]$ is satisfied and true. That is, the sentence is satisfied and true just when the happy people assigned to x and y are distinct—just when there are at least two happy people. On this pattern, you should be able to see how to say there are at least three happy people, and so forth.

Now suppose we want to say, ‘At most one person is happy’. We have, of course, learned a couple of ways to say nobody is happy, $\forall x{\sim}Hx$ and $\sim\exists xHx$. But for ‘at most one’ we need something like, $\forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$. For this, in our simplified case, the universal quantifier yields three branches of the sort, $\mathsf{I}_{\mathsf{d}(x|\mathsf{m})}[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$. The beginning of the branch is as follows:

(BN)



The universal $\forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$ is satisfied and true if and only if all the conditional branches at (1) are satisfied. And the branches at (1) are satisfied so long as there is no S/N pair at (2). This is, of course, true if nobody is happy so that the top at (2) is never satisfied. But suppose m is a happy person, say Sue, and the top at (2) is satisfied. Then the bottom comes out S so long as Sue is the only happy person. If Sue is the only happy person, when y is assigned to objects other than Sue, Hy is N and so the conditionals are S; and when y is assigned to Sue, the equality is S and so the conditional is S. So there is no S/N pair. But suppose Jim, say, is also happy; then on the very bottom branch at (3), Hy is S and $x = y$ is N; so the conditional is N; so the universal at (2) is N; so the conditional at (1) is N; and the entire sentence is N. Suppose x is assigned to a happy person; in effect, $\forall y(Hy \rightarrow (x = y))$ limits the range of happy things, telling us that *anything happy is it*. We get ‘At most two people are happy’ with $\forall x\forall y[(Hx \wedge Hy) \rightarrow \forall z(Hz \rightarrow (x = z \vee y = z))]$—if some things are happy, then anything that is happy is one of them. And similarly in other cases.

To say ‘Exactly one person is happy’, it is enough to say at least one person is happy, and at most one person is happy. Thus, using what we have already done, $\exists xHx \wedge \forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$ does the job. But we can use the “limiting” strategy with the universal quantifier more efficiently. Thus, for example, if we want to say ‘Bob is the only happy person’ we might try $Hb \wedge \forall y[Hy \rightarrow (b = y)]$—Bob is happy, and every happy person *is* Bob. Similarly, for ‘Exactly one person is happy’, $\exists x[Hx \wedge \forall y(Hy \rightarrow (x = y))]$ is good. We say that there is a happy person, and that all the happy people are identical to it. For ‘Exactly two people are happy’,

$\exists x \exists y[((Hx \wedge Hy) \wedge \sim(x = y)) \wedge \forall z(Hz \rightarrow [(x = z) \vee (y = z)])]$ does the job—there are at least two happy people, and anything that is a happy person is identical to one of them.

Phrases of the sort "the such-and-such" are *definite descriptions*. Perhaps it is natural to think "the such-and-such is so-and-so" *fails* when there is more than one such-and-such. Similarly, phrases of the sort "the such-and-such is so-and-so" seem to fail when nothing is such-and-such. Thus, for example, neither 'The desk at CSUSB is wobbly' nor 'The present king of France is bald' seem to be true—the first because the description fails to pick out just one object, and the second because the description does not pick out any object. Of course, if a description does pick out just one object, then the predicate must apply. So, for example, as I write, 'The president of the USA is a woman' is not true. There is exactly one object which is the president of the USA, but it is not a woman. And 'The president of the USA is a man' is true. In this case, exactly one object is picked out by the description, and the predicate does apply. Thus, in "On Denoting," Bertrand Russell famously proposes that a statement of the sort 'the $\mathcal{P}$ is $\mathcal{Q}$' is true just in case there is exactly one $\mathcal{P}$ and it is $\mathcal{Q}$. On Russell's account, then, where $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ have variable x free, and $\mathcal{P}(v)$ is like $\mathcal{P}(x)$ but with free instances of x replaced by a new variable v, $\exists x[(\mathcal{P}(x) \wedge \forall v(\mathcal{P}(v) \rightarrow x = v)) \wedge \mathcal{Q}(x)]$ is good—there is a $\mathcal{P}$, it is the only $\mathcal{P}$, and it is $\mathcal{Q}$. Thus, for example, with the natural interpretation function, $\exists x[(Px \wedge \forall y(Py \rightarrow x = y)) \wedge Wx]$ translates 'The president is a woman'. In a course on philosophy of language, one might spend a great deal of time discussing definite descriptions. But in ordinary cases we will simply assume Russell's account for translating expressions of the sort, 'the $\mathcal{P}$ is $\mathcal{Q}$'.

Finally, notice that equality can play a role in *exception* clauses. This is particularly important when making general comparisons. Thus, for example, if we want to say that zero is least of the natural numbers, with the standard interpretation $\bar{\mathsf{N}}$ of $\mathcal{L}_{NT}^{<}$, $\forall x(\emptyset < x)$ is a mistake. This formula is satisfied only if zero is less than zero! What we want is rather, $\forall x[\sim(x = \emptyset) \rightarrow (\emptyset < x)]$. Similarly, if we want to say that there is a tallest person, we would not use $\exists x \forall y Txy$ where Txy when x is taller than y. This would require that the tallest person be taller than herself. What we want is rather, $\exists x \forall y[\sim(x = y) \rightarrow Txy]$.

Observe that relations of this sort may play a role in definite descriptions. Thus it seems natural to talk about *the* least natural number, or *the* tallest person. We might therefore additionally assert uniqueness with something like, $\exists x[x$ is taller than every other $\wedge \forall z(z$ is taller than every other $\rightarrow x = z)]$.[11] However, we will not usually add the second clause, insofar as uniqueness follows automatically in these cases from the initial claim, $\exists x \forall y[\sim(x = y) \rightarrow Txy]$ together with the premise that *taller than* (*less than*) is asymmetric, that $\forall x \forall y(Txy \rightarrow \sim Tyx)$. For arbitrary relation $\mathcal{R}$, $\exists x \forall y[\sim(x = y) \rightarrow \mathcal{R}xy]$ does not require uniqueness—it says only that there is an

[11] $\exists x[\forall y(\sim(x = y) \rightarrow Txy) \wedge \forall z(\forall y(\sim(z = y) \rightarrow Tzy) \rightarrow x = z)]$.

object that stands in relation $\mathscr{R}$ to every other. Given the additional premise that $\mathscr{R}$ is asymmetric, however, it follows that just one thing has $\mathscr{R}$ to all the others: If m has $\mathscr{R}$ to everything other than itself, and n has $\mathscr{R}$ to everything other than itself, but $m \neq n$, then $\mathscr{R}mn$ and $\mathscr{R}nm$, so that $\mathscr{R}$ is not asymmetric; thus, put the other way around, if $\mathscr{R}$ *is* asymmetric, no distinct objects m, n are such that each has $\mathscr{R}$ to all the others. Thus for 'The tallest person is happy' it is sufficient to conjoin 'An object with T to every other is happy' with asymmetry,

$$\exists x[\forall y(\sim(x = y) \to Txy) \wedge Hx] \wedge \forall x \forall y(Txy \to \sim Tyx)$$

Taken together, these imply all the elements of Russell's account. And similarly in other cases.

E5.27. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a snake in my yard$\}$

a: Aaalph

G^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is in the grass$\}$

D^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is deadly$\}$

B^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m is bigger than $\mathsf{n}\}$

a. There is at least one snake in the grass.

b. There are at least two snakes in the grass.

*c. There are at least three snakes in the grass.

d. There are no snakes in the grass.

e. There is at most one snake in the grass.

*f. There are at most two snakes in the grass.

g. There are at most three snakes in the grass.

h. There is exactly one snake in the grass.

*i. There are exactly two snakes in the grass.

j. There are exactly three snakes in the grass.

*k. The snake in the grass is deadly.

l. The deadly snake is in the grass.

*m. Aaalph is the biggest snake.

n. The biggest snake is in the grass.

o. The biggest snake in the grass is deadly.

E5.28. Given $\mathscr{L}_{NT}^{<}$ and the standard interpretation $\bar{\mathsf{N}}$ as below, complete the translation for each of the following.[12]

U: $\mathbb{N}$

$\emptyset$: zero

S: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$+$: $\{\langle\langle \mathsf{m}, \mathsf{n} \rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\times$: $\{\langle\langle \mathsf{m}, \mathsf{n} \rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

$<$: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

a. Any number is equal to itself (identity is *reflexive*).

b. If a number a is equal to a number b, then b is equal to a (identity is *symmetric*).

c. If a number a is equal to a number b and b is equal to c, then a is equal to c (identity is *transitive*).

d. No number is less than itself (less-than is *irreflexive*).

*e. If a number a is less than a number b, then b is not less then a (less-than is *asymmetric*).

f. If a number a is less than a number b and b is less than c, then a is less than c (less-than is *transitive*).

g. There is no largest number.

*h. Four is even (a number such that two times something is equal to it).

i. Three is odd (such that two times something plus one is equal to it).

*j. Any odd number is the sum of an odd and an even.

k. Any even number other than zero is the sum of one odd with another.

[12]This exercise translates some truths of arithmetic. Notice that these are *necessary* truths. It is easy enough to cook up stories where nobody loves anybody, where everybody loves everybody, and anything between. However there is no consistent story where one plus one is other than two—and, as translations, any tautology would seem to satisfy CG. Still, as a sort of addendum to our criterion of goodness, it is natural to proceed *as though* 'plus', 'times' and the like might apply in arbitrary ways. In fact, this will be the way you naturally approach these exercises.

l. The sum of one odd with another odd is even.

m. There is no even number greater than every other even number.

*n. Three is prime (a number divided by no number other than one and itself—though you will have to put this in terms of multipliers).

o. Every prime except two is odd.

E5.29. For each of the following arguments: (i) Produce a good translation, including interpretation function and translations for the premises and conclusion. Then (ii) for each argument that is not quantificationally valid, produce an interpretation (trees optional) to show that the argument is not quantificationally valid.

a. Only citizens can vote
Hannah is a citizen
―――――
Hannah can vote

b. All citizens can vote
If someone is a citizen, then their father is a citizen
Hannah is a citizen
―――――
Hannah's father can vote

*c. Alice is taller than everyone else
―――――
Only Alice is taller than everyone else

d. Alice is taller than everyone else
The *taller than* relation is asymmetric
―――――
Only Alice is taller than everyone else

e. There is a dog
At most one dog is pursuing a cat
At least one cat is being pursued (by some animal)
―――――
Some dog is pursuing a cat

E5.30. For those who have studied derivations from Chapter 3 or Chapter 6: For each of the arguments of E5.29 that is not quantificationally invalid, show that it is valid in *AD* or *ND+*, whichever is appropriate.

E5.31. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Quantifier switching.

b. Quantifier placement.

c. Quantity expressions and definite descriptions.

Chapter 6

Natural Deduction

Natural deduction systems are so-called because their rules formalize patterns of reasoning that occur in relatively ordinary "natural" contexts. Thus, initially at least, the rules of natural deduction systems are easier to motivate than the axioms and rules of axiomatic systems. By itself, this is sufficient to give natural deduction a special interest. As we shall see, natural deduction is also susceptible to proof *strategies* in a way that (primitive) axiomatic systems are not. If you have had another course in formal logic, you have probably been exposed to natural deduction. So, again, it may seem important to bring what we have done into contact what you have encountered in other contexts. After some general remarks about natural deduction in section 6.1, we turn to the sentential part of our natural derivation system *NDs* (section 6.2), then the full version with quantifiers and equality *ND* (section 6.3), and finally consider some applications to arithmetic (section 6.4).

6.1 General

This section develops some concepts required for *NDs* and *ND*. The first part develops a "toy" system to introduce the very idea of a derivation and a derivation rule. We then turn to some concepts required for the particular rules of *ND*.[1]

6.1.1 Derivations as Games

Derivations can be seen as a kind of game—with the aim of getting from a starting point to a goal by rules. In their essential nature, these rules are defined in terms of form: the forms of expressions authorize "moves" in the game. Given this, there is no immediate or obvious connection between derivations and semantic validity or truth. All the same, even though the rules are not *defined* by a relation to validity and

[1]Parts of this section are reminiscent of section 3.1 and, especially if you skipped over that section, you may want to look it over now as additional background.

truth, ultimately we shall be able to establish relations between the derivation rules and these notions.

We begin introducing natural derivations purely in their essential nature as games. Thus, for example, consider a preliminary system *NP* with the following rules:

$$NP \qquad \text{R1}\ \frac{\mathcal{P} \to \mathcal{Q},\ \mathcal{P}}{\mathcal{Q}} \qquad \text{R2}\ \frac{\mathcal{P} \vee \mathcal{Q}}{\mathcal{Q}} \qquad \text{R3}\ \frac{\mathcal{P} \wedge \mathcal{Q}}{\mathcal{P}} \qquad \text{R4}\ \frac{\mathcal{P}}{\mathcal{P} \vee \mathcal{Q}}$$

In this system, R1: given formulas of the form $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$, you may move to $\mathcal{Q}$; R2: given a formula of the form $\mathcal{P} \vee \mathcal{Q}$, you may move to $\mathcal{Q}$; R3: given a formula of the form $\mathcal{P} \wedge \mathcal{Q}$, you may move to $\mathcal{P}$; and R4: given a formula $\mathcal{P}$ you may move to $\mathcal{P} \vee \mathcal{Q}$ for any $\mathcal{Q}$. For now, at least, the game is played as follows: You begin with some starting formulas and a goal. The starting formulas are like "cards" in your hand. You then apply the rules to obtain more formulas, to which the rules may be applied again and again. You win if you eventually obtain the goal formula.

Let us consider some examples. At this stage, do not worry about strategy, about why we do what we do, as much as about how the rules work and the way the game is played. A game always begins with starting premises at the top, and goal on the bottom.

$$\text{(A)} \qquad \begin{array}{r|ll} 1. & A \to (B \wedge C) & \text{P(remise)} \\ 2. & \underline{A} & \text{P(remise)} \\ & & \\ & & \\ & B \vee D & \text{(goal)} \end{array}$$

The formulas on lines (1) and (2) are of the form $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$, where $\mathcal{P}$ maps to A and $\mathcal{Q}$ to $(B \wedge C)$; so we are in a position to apply rule R1 to get the $\mathcal{Q}$.

$$\begin{array}{r|ll} 1. & A \to (B \wedge C) & \text{P(remise)} \\ 2. & \underline{A} & \text{P(remise)} \\ 3. & B \wedge C & \text{1,2 R1} \\ & & \\ & B \vee D & \text{(goal)} \end{array}$$

The justification for our move—the way the rules apply—is listed on the right; in this case, we use the formulas on lines (1) and (2) according to rule R1 to get $B \wedge C$; so that is indicated by the notation. Now $B \wedge C$ is of the form $\mathcal{P} \wedge \mathcal{Q}$. So we can apply R3 to it in order to obtain the $\mathcal{P}$, namely B.

$$\begin{array}{r|ll} 1. & A \to (B \wedge C) & \text{P(remise)} \\ 2. & \underline{A} & \text{P(remise)} \\ 3. & B \wedge C & \text{1,2 R1} \\ 4. & B & \text{3 R3} \\ & B \vee D & \text{(goal)} \end{array}$$

Notice that one application of a rule is independent of another. It does not matter what formula was $\mathcal{P}$ or $\mathcal{Q}$ in a previous move for evaluation of this one. Finally, where $\mathcal{P}$ is B, $B \vee D$ is of the form $\mathcal{P} \vee \mathcal{Q}$. So we can apply R4 to get the final result.

1.	$A \to (B \wedge C)$	P(remise)
2.	A	P(remise)
3.	$B \wedge C$	1,2 R1
4.	B	3 R3
5.	$B \vee D$	4 R4 Win!

Notice that R4 leaves the $\mathcal{Q}$ unrestricted: Given some $\mathcal{P}$, we can move to $\mathcal{P} \vee \mathcal{Q}$ for *any* $\mathcal{Q}$. Since we reached the goal from the starting sentences, we win! In this simple derivation system, any line of a successful derivation is either given as a premise, or justified from lines before it by the rules.

Here are a couple more examples, this time of completed derivations. First:

(B)

1.	$A \wedge C$	P
2.	$(A \vee B) \to D$	P
3.	A	1 R3
4.	$A \vee B$	3 R4
5.	D	2,4 R1
6.	$D \vee (R \to S)$	5 R4 Win!

$A \wedge C$ is of the form $\mathcal{P} \wedge \mathcal{Q}$. So we can apply R3 to obtain the $\mathcal{P}$, in this case A. Then where $\mathcal{P}$ is A, we use R4 to add on a B to get $A \vee B$. $(A \vee B) \to D$ and $A \vee B$ are of the form $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$; so we apply R1 to get the $\mathcal{Q}$, that is D. Finally, where D is $\mathcal{P}$, $D \vee (R \to S)$ is of the form $\mathcal{P} \vee \mathcal{Q}$; so we apply R4 to get the final result. Notice again that the $\mathcal{Q}$ may be any formula whatsoever.

Here is another example:

(C)

1.	$(A \wedge B) \wedge D$	P
2.	$(A \wedge B) \to C$	P
3.	$A \to (C \to (B \wedge D))$	P
4.	$A \wedge B$	1 R3
5.	C	2,4 R1
6.	A	4 R3
7.	$C \to (B \wedge D)$	3,6 R1
8.	$B \wedge D$	7,5 R1
9.	B	8 R3 Win!

You should be able to follow the steps. In this case, we use $A \wedge B$ on line (4) twice; once as part of an application of R1 to get C, and again in an application of R3 to get the A. Once you have a formula in your "hand" you can use it as many times and whatever way the rules will allow. Also, the order in which we worked might have been different. Thus, for example, we might have obtained A on line (5) and then C after. You win if you get to the goal by the rules; how you get there is up to you. Finally, it is tempting to think we could get B from, say, $A \wedge B$ on line (4). We will able to do this in our official system. But the rules we have so far do not let us do so. R3 lets us move just to the left conjunct of a formula of the form $\mathcal{P} \wedge \mathcal{Q}$.

When there is a way to get from the premises of some argument to its conclusion by the rules of derivation system *N*, the premises *prove* the conclusion in system *N*. In this case, where Γ is the set of premises and $\mathcal{P}$ the conclusion, we write $\Gamma \vdash_N \mathcal{P}$. If $\Gamma \vdash_N \mathcal{P}$ the argument is *valid* in derivation system *N*. Notice the distinction between this "single turnstile" $\vdash$ and the double turnstile $\vDash$ associated with semantic validity. As usual, if $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are the members of Γ, we sometimes write $\mathcal{Q}_1 \ldots \mathcal{Q}_n \vdash_N \mathcal{P}$ in place of $\Gamma \vdash_N \mathcal{P}$. If Γ has no members then we simply write $\vdash_N \mathcal{P}$. In this case, $\mathcal{P}$ is a *theorem* of derivation system *N*.

One can imagine setting up many different rule sets, and so many different games of this kind. In the end, we want our game to serve a specific purpose. That is, we want to use the game in the identification of valid arguments. In order for our games to be an indicator of validity, we would like it to be the case that $\Gamma \vdash_N \mathcal{P}$ iff $\Gamma \vDash \mathcal{P}$, that Γ *proves* $\mathcal{P}$ iff Γ *entails* $\mathcal{P}$. In Part III we will show that our official derivation games have this property. For now, we can at least see how this might be: Roughly, we impose the following condition on rules: We require of our rules that *the inputs always semantically entail the outputs*. Then if some premises are true, and we make a move to a formula, the formula we move to must be true; and if the formulas in our "hand" are all true, and we add some formula by another move, the formula we add must be true; and so forth for each formula we add until we get to the goal, which will have to be true as well. So if the premises are true, the goal must be true as well.

Notice that our rules R1, R3, and R4 each meet the proposed requirement on rules, but R2 does not.

(D)

		R1			R2		R3		R4	
$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \to \mathcal{Q}$	$\mathcal{P}$	/ $\mathcal{Q}$	$\mathcal{P} \vee \mathcal{Q}$	/ $\mathcal{Q}$	$\mathcal{P} \wedge \mathcal{Q}$	/ $\mathcal{P}$	$\mathcal{P}$	/ $\mathcal{P} \vee \mathcal{Q}$
T	T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	F	**F**	**T**	**F**	**T**	**F**	**F**	**T**	**T**	**T**
F	T	**T**	**F**	**T**	**T**	**T**	**F**	**F**	**F**	**T**
F	F	**T**	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**

R1, R3, and R4 have no row where the input(s) are T and the output is F. But for R2, the second row has input T and output F. So R2 does not meet our condition. This does not mean that one cannot construct a *game* with R2 as a part. Rather, the point is that R2 will not help us accomplish what we want to accomplish with our games. So long as rules meet the condition, a win in the game always corresponds to an argument that is semantically valid.

Thus for example, from table (F) on the following page, derivation (C), in which R2 does not appear, corresponds to the result that $(A \wedge B) \wedge D$, $(A \wedge B) \to C$, $A \to (C \to (B \wedge D)) \vDash_s B$. The table has no row where the premises are T and the conclusion is F. So the argument is sententially valid. As the number of rows goes up, we may decide that the games are dramatically easier to complete than the tables. And similarly for the quantificational case, where we have not yet been able to demonstrate semantic validity at all.

E6.1. Show that each of the following is valid in *NP*. Complete (a)–(d) using just rules R1, R3, and R4. You will need an application of R2 for (e).

*a. $(A \wedge B) \wedge C \vdash_{NP} A$

b. $(A \wedge B) \wedge C, A \rightarrow (B \wedge C) \vdash_{NP} B$

c. $(A \wedge B) \rightarrow (B \wedge A), A \wedge B \vdash_{NP} B \vee A$

d. $R, [R \vee (S \vee T)] \rightarrow S \vdash_{NP} S \vee T$

e. $A \vdash_{NP} A \rightarrow C$

*E6.2. (i) For each of the arguments in E6.1, use a truth table to decide if the argument is sententially valid. (ii) To what do you attribute the fact that a win in *NP* is not a sure indicator of semantic validity?

6.1.2 Auxiliary Assumptions

Having introduced the idea of a derivation by our little system *NP*, we now turn to some additional concepts that are background to the rules of our official derivation system *ND*. So far, our derivations have had the following form:

(E)

$$\begin{array}{r|ll} \text{a.} & \mathcal{A} & \text{P(remise)} \\ & \vdots & \\ \text{b.} & \mathcal{B} & \text{P(remise)} \\ \hline & \vdots & \\ \text{c.} & \mathcal{G} & \text{(goal)} \end{array}$$

We have some premise(s) at the top, and a conclusion at the bottom. The premises are against a line which indicates the range or *scope* over which the premises apply. In

(F)

A	B	C	D	$(A \wedge B)$	$\wedge\ D$	$(A \wedge B)$	$\rightarrow C$	$A \rightarrow$	$(C \rightarrow$	$(B \wedge D))$	/ B
T	T	T	T	T	***T***	T	***T***	***T***	T	T	***T***
T	T	T	F	T	***F***	T	***T***	***F***	F	F	***T***
T	T	F	T	T	***T***	T	***F***	***T***	T	T	***T***
T	T	F	F	T	***F***	T	***F***	***T***	T	F	***T***
T	F	T	T	F	***F***	F	***T***	***F***	F	F	***F***
T	F	T	F	F	***F***	F	***T***	***F***	F	F	***F***
T	F	F	T	F	***F***	F	***T***	***T***	T	F	***F***
T	F	F	F	F	***F***	F	***T***	***T***	T	F	***F***
F	T	T	T	F	***F***	F	***T***	***T***	T	T	***T***
F	T	T	F	F	***F***	F	***T***	***T***	F	F	***T***
F	T	F	T	F	***F***	F	***T***	***T***	T	T	***T***
F	T	F	F	F	***F***	F	***T***	***T***	T	F	***T***
F	F	T	T	F	***F***	F	***T***	***T***	F	F	***F***
F	F	T	F	F	***F***	F	***T***	***T***	F	F	***F***
F	F	F	T	F	***F***	F	***T***	***T***	T	F	***F***
F	F	F	F	F	***F***	F	***T***	***T***	T	F	***F***

each case, the line extends from the premises to the conclusion, indicating that the conclusion is derived from them. It is always our aim to derive the conclusion under the scope of the premises alone. But our official derivation system will allow appeal to certain *auxiliary* assumptions in addition to premises. Any such assumption comes with a scope line of its own—indicating the range over which *it* applies. Thus, for example, derivations might be structured as follows:

(G)

a.	$\mathcal{A}$	P(remise)
b.	$\mathcal{B}$	P(remise)
c.	$\mathcal{C}$	A(ssumption)
d.		
e.	$\mathcal{G}$	(goal)

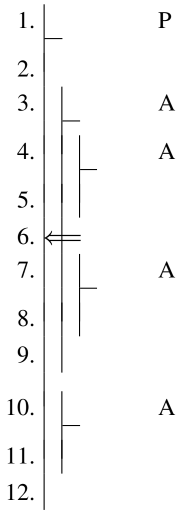
(H)

a.	$\mathcal{A}$	P(remise)
b.	$\mathcal{B}$	P(remise)
c.	$\mathcal{C}$	A(ssumption)
d.	$\mathcal{D}$	A(ssumption)
e.		
f.		
g.	$\mathcal{G}$	(goal)

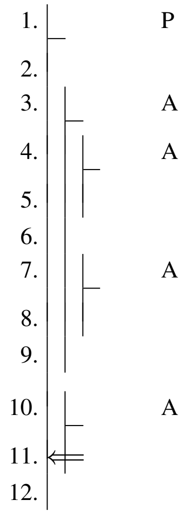
In each, there are premises $\mathcal{A}$ through $\mathcal{B}$ at the top and goal $\mathcal{G}$ at the bottom. As indicated by the main leftmost scope line, the premises apply throughout the derivations, and the goal is derived under them. In case (G), there is an additional assumption at (c). As indicated by its scope line, that assumption applies from (c)–(d). In (H), there are a pair of additional assumptions. As indicated by the associated scope lines, the first applies over (c)–(f), and the second over (d)–(e). We will say that an auxiliary assumption, together with the formulas that fall under its scope, is a *subderivation*. Thus (G) has a subderivation on (c)–(d). (H) has a pair of subderivations, one on (c)–(f), and another on (d)–(e). A derivation or subderivation may *include* various other subderivations. Any subderivation begins with an auxiliary assumption. In general we *cite* a subderivation by listing the line number on which it begins, then a dash, and the line number on which its scope line ends.

In contexts without auxiliary assumptions, we have been able freely to appeal to any formula already in our "hand." Where there are auxiliary assumptions, however, we may appeal only to *accessible* subderivations and formulas. A formula is *accessible* at a given stage when it is obtained under assumptions all of which continue to apply. But scope lines indicate the range over which assumptions apply. In practice then, for justification of a formula at line number i we can appeal only to formulas which appear immediately against scope lines extending as far as i—these are the formulas obtained under assumptions that continue to apply. Thus, for example, with the scope structure as in (I) below, in the justification of line (6),

(I)



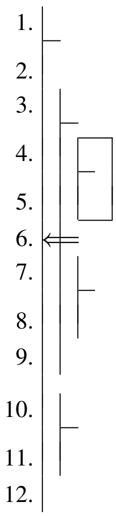
(J)



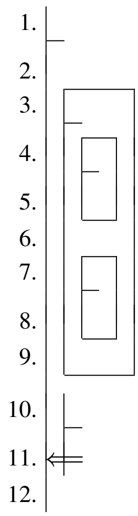
we could appeal only to formulas at (1), (2), and (3), for these are the only ones immediately against scope lines extending as far as (6). To see this, notice that scope lines extending as far as (6) are ones cut by the arrow at (6). Formulas at (4) and (5) are not against a line extending that far. Similarly, as indicated by the arrow in (J), for the justification of (11), we could appeal only to formulas at (1), (2), and (10). Formulas at other line numbers are not immediately against scope lines extending as far as (11). The accessible formulas are ones derived under assumptions all of which continue to apply.

It may be helpful to think of a completed subderivation as a sort of "box." So long as you are under the scope of an assumption, the box is open and you can "see" the formulas under its scope. However, once you exit from an assumption, the box is closed, and the formulas inside are no longer available.

(I′)



(J′)



Thus, again, at line (6) of (I′) the formulas at (4)–(5) are locked away so that the only accessible lines are (1)–(3). Similarly, at line (11) of (J′) all of (3)–(9) is unavailable.

Our aim is always to obtain the goal against the leftmost scope line—under the scope of the premises alone—and if the only formulas accessible for the goal's justification are also against the leftmost scope line, it may appear mysterious why we would ever introduce auxiliary assumptions and subderivations at all. What is the point of auxiliary assumptions, if formulas under their scope are inaccessible for justification of the formula we want? The answer is that though the formulas inside a box are unavailable *the box* may still be useful. Some of our rules will appeal to entire subderivations (to the boxes), rather than to the formulas in them. A subderivation is *accessible* at a given stage when *it* is obtained under assumptions all of which continue to apply. In practice, what this means is that for a formula at line i, we can appeal to a box (to a subderivation) only if *it* (its scope line) is against a line which extends down to i.

Thus at line (6) of (I′), we would not be able to appeal to the formulas on lines (4) and (5)—they are inside the closed box. However, we *would* be able to appeal to the *box* on lines (4)–(5), for *it* is against a scope line cut by the arrow. Similarly, at line (11) of (J′) we are not able to appeal to formulas on any of the lines (3)–(9), for they are inside the closed boxes. Similarly, we cannot appeal to the *boxes* on (4)–(5) or (7)–(8) for they are locked inside the larger box. However, we can appeal to the larger subderivation on (3)–(9) insofar as it is against a line cut by the arrow. Observe that one can appeal to a box only after it is closed—so, for example, at (11) of (J′) there is not (yet) a closed box at (10)–(11) and so no available subderivation to which one may appeal. When a box is closed, its assumption is *discharged*.

So we have an answer to our question about the point of subderivations for reaching a conclusion: In our example, the justification for the conclusion at line (12) might appeal to the formulas on lines (1) and (2) or to the subderivations on lines (3)–(9) and (10)–(11). Again line (12) does not have access to the *formulas* inside the subderivations from lines (3)–(9) and (10)–(11). So the subderivations are accessible even where the formulas inside them are not.

First rule of *NDs*. All this will become more concrete as we turn now to the rules of our official system *ND* and its initial fragment *NDs*. Let us set aside rules of the preliminary system *NP* and begin rules of *NDs* from scratch. We can reinforce the point about accessibility of *formulas* by introducing the first, and simplest, rule of *NDs*. If a formula $\mathcal{P}$ appears on an accessible line a of a derivation, we may repeat it by the rule *reiteration*, with justification a R.

$$\text{R} \qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} & \\ & & \\ & \mathcal{P} & \text{a R} \end{array}$$

It should be obvious why reiteration satisfies our basic condition on rules. If $\mathcal{P}$ is true, *of course* $\mathcal{P}$ is true. So this rule could never lead from a formula that is true to one that is not. Observe, though, that the line a must be *accessible*. Given scope lines as in (I) and leaving aside assumption lines (which are always justified 'A'), if the assumption at line (3) were a formula $\mathscr{P}$, we could conclude $\mathscr{P}$ with justification 3 R at lines (5), (6), (8), or (9). We could not obtain $\mathscr{P}$ with the same justification at (11) or (12) without violating the rule, because (3) is not accessible for justification of (11) or (12). You should be clear about why this is so.

*E6.3. Consider a derivation with the following structure:

```
1. |_         P
2. | |_       A
3. | |
4. | |_       A
5. | | |_     A
6. | | |
7. | |
8. |
```

For each of the lines (3), (6), (7), and (8) which lines are accessible? which subderivations (if any) are accessible? That is, complete the following table:

	accessible lines	accessible subderivations
line 3		
line 6		
line 7		
line 8		

*E6.4. Suppose in a derivation with structure as in E6.3 we have obtained a formula $\mathscr{A}$ on line (3). (i) On what lines could we conclude $\mathscr{A}$ by 3 R? Suppose there is

Definitions for Auxiliary Assumptions

SD An auxiliary assumption, together with the formulas that fall under its scope, is a *subderivation*.

FA A formula is *accessible* at a given stage when it is obtained under assumptions all of which continue to apply.

SA A subderivation is *accessible* at a given stage when it (as a whole) is obtained under assumptions all of which continue to apply.

In practice, what this means is that for justification of a formula at line i we can appeal to another formula only if it is immediately against a scope line extending as far as i.

And in practice, for justification of a formula at line i, we can appeal to a subderivation only if its whole *scope line* is itself immediately against a scope line extending as far as i.

a formula $\mathcal{B}$ on line (4). (ii) On what lines could we conclude $\mathcal{B}$ by 4 R? Hint: This is just a question about accessibility, asking where it is possible to use lines (3) and (4).

6.2 Sentential

We introduced the idea of a derivation by the preliminary system *NP*. We have introduced notions of accessibility. And, setting aside the rules of *NP*, we have seen the first rule R of *NDs*. We now turn to the rest of the rules of *NDs*, including rules for arbitrary sentential forms—for arbitrary forms involving $\sim$ and $\rightarrow$ (and so $\wedge$, $\vee$, and $\leftrightarrow$). Of course expressions of a quantificational language may have sentential forms, and if this is so the rules apply to them. For the most part, though, we simply focus on expressions of our sentential language $\mathcal{L}_s$. In a derivation, each formula is either a premise, an auxiliary assumption, or is justified by the rules. In addition to reiteration, *NDs* includes two rules for each of the five sentential operators—for a total of eleven rules. For each of the operators, there is an 'I' or *introduction* rule, and an 'E' or *exploitation* rule.[2] As we will see, this division helps structure the way we approach derivations. There are sections to introduce the rules (6.2.1–6.2.3), for discussion of strategy (6.2.4), and for an extended system *NDs+* (6.2.5).

6.2.1 $\rightarrow$ and $\wedge$

Let us start with the I- and E-rules for $\rightarrow$ and $\wedge$. We have already seen the exploitation rule for $\rightarrow$. It is R1 of system *NP*. If formulas $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ and appear on accessible lines a and b of a derivation, we may conclude $\mathcal{Q}$ with justification *a,b* $\rightarrow$E.

$$
\rightarrow\text{E} \qquad
\begin{array}{r|ll}
\text{a.} & \mathcal{P} \rightarrow \mathcal{Q} & \\
\text{b.} & \mathcal{P} & \\
 & & \\
 & \mathcal{Q} & \text{a,b} \rightarrow\text{E}
\end{array}
$$

Intuitively, if it is true that *if* $\mathcal{P}$ *then* $\mathcal{Q}$ and it is true that $\mathcal{P}$, then $\mathcal{Q}$ must be true as well. And on table (D) we saw that if both $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ are true, then $\mathcal{Q}$ is true. Notice that we *do not* somehow get the $\mathcal{P}$ from $\mathcal{P} \rightarrow \mathcal{Q}$. Rather, we exploit $\mathcal{P} \rightarrow \mathcal{Q}$ when, given that $\mathcal{P}$ also is true, we use $\mathcal{P}$ together with $\mathcal{P} \rightarrow \mathcal{Q}$ to conclude $\mathcal{Q}$. So this rule requires two input "cards." The $\mathcal{P} \rightarrow \mathcal{Q}$ card sits idle without a $\mathcal{P}$ to activate it. The order in which $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ appear does not matter so long as they are both accessible. However, you should cite them in the standard order—line for the conditional first, then the antecedent. As in the axiomatic system from Chapter 3, this rule is sometimes called *modus ponens*.

Here is an example. We show, $L, L \rightarrow (A \wedge K), (A \wedge K) \rightarrow (L \rightarrow P) \vdash_{NDs} P$.

[2] I- and E-rules are often called *introduction* and *elimination* rules. This can lead to confusion as E-rules do not necessarily eliminate anything.

(K)

$$\begin{array}{rl|ll}
1. & L & \text{P} \\
2. & L \to (A \wedge K) & \text{P} \\
3. & (A \wedge K) \to (L \to P) & \text{P} \\
\hline
4. & A \wedge K & 2,1 \to\text{E} \\
5. & L \to P & 3,4 \to\text{E} \\
6. & P & 5,1 \to\text{E}
\end{array}$$

$L \to (A \wedge K)$ and L and are of the form $\mathscr{P} \to \mathscr{Q}$ and $\mathscr{P}$ where L is the $\mathscr{P}$ and $A \wedge K$ is $\mathscr{Q}$. So we use them to conclude $A \wedge K$ by →E on (4). But then $(A \wedge K) \to (L \to P)$ and $A \wedge K$ are of the form $\mathscr{P} \to \mathscr{Q}$ and $\mathscr{P}$, so we use them to conclude $\mathscr{Q}$, in this case, $L \to P$, on line (5). Finally $L \to P$ and L are of the form $\mathscr{P} \to \mathscr{Q}$ and $\mathscr{P}$, and we use them to conclude P on (6).

Notice that,

(L)

$$\begin{array}{rl|lll}
1. & (A \to B) \wedge C & \text{P} & \\
2. & A & \text{P} & \\
\hline
3. & B & 1,2 \to\text{E} & \text{!Mistake!}
\end{array}$$

misapplies the rule. $(A \to B) \wedge C$ is not of the form $\mathscr{P} \to \mathscr{Q}$—the main operator being $\wedge$, so that the formula is of the form $\mathscr{P} \wedge \mathscr{Q}$. The rule →E applies just to formulas with main operator →. If we want to use $(A \to B) \wedge C$ with A to conclude B, we would first have to isolate $A \to B$ on a line of its own. We introduce a rule for this just below (and we might have done it in *NP*). But we do not yet have the required rule in *NDs*.

→I is our first rule that requires a subderivation. Once we understand this rule, the rest are mere variations on a theme. →I takes as its input an entire subderivation. Given an accessible subderivation which begins with assumption $\mathscr{P}$ on line a and ends with $\mathscr{Q}$ against the assumption's scope line at b, one may conclude $\mathscr{P} \to \mathscr{Q}$ with justification *a*-*b* →I.

→I

$$\begin{array}{r|l|ll}
\text{a.} & & \mathscr{P} & \text{A } (\mathscr{Q}, \to\text{I}) \\
\cline{3-3}
 & & & \\
\text{b.} & & \mathscr{Q} & \\
 & \mathscr{P} \to \mathscr{Q} & & \text{a-b} \to\text{I}
\end{array}
\qquad \text{or} \qquad
\begin{array}{r|l|ll}
\text{a.} & & \mathscr{P} & \text{A } (g, \to\text{I}) \\
\cline{3-3}
 & & & \\
\text{b.} & & \mathscr{Q} & \\
 & \mathscr{P} \to \mathscr{Q} & & \text{a-b} \to\text{I}
\end{array}$$

So $\mathscr{P} \to \mathscr{Q}$ is justified by a subderivation that begins with assumption $\mathscr{P}$ and ends with $\mathscr{Q}$. Note that the auxiliary assumption comes with a parenthetical *exit strategy:* In this case the exit strategy includes the *formula* $\mathscr{Q}$ with which the subderivation is to end, and an indication of the rule (→I) by which exit is to be made. We might write out the entire formula inside the parentheses as indicated on the left. In practice, however, this is tedious, and it is easier just to write the formula at the bottom of the scope line where we will need it in the end. Thus in the parentheses on the right 'g' is a simple *pointer* to the goal formula at the end of the scope line. Note that the pointer is empty unless there is a formula to which it points, and *the exit strategy therefore is not complete unless the goal formula is stated*. In this case, the strategy includes the pointer to the goal formula, along with the indication of the rule (→I) by which exit is to be made. Again, at the time we make the assumption, we write the $\mathcal{Q}$ down as part of the strategy for exiting the subderivation. But this does not mean the $\mathcal{Q}$ is justified! The $\mathcal{Q}$ is rather introduced as a new goal. Notice also that the justification *a-b* →I does not refer to the *formulas* on lines a and b. These are inaccessible. Rather, the justification appeals to the subderivation which begins on line a and ends on line b—where this subderivation is accessible even though the formulas in it are not. So there is a difference between the comma and the dash, as they appear in justifications.

For this rule, we assume the antecedent, reach the consequent, then discharge the assumption and conclude to the conditional by →I. Intuitively, if an assumption $\mathcal{P}$ leads to $\mathcal{Q}$ then we know that *if* $\mathcal{P}$ *then* $\mathcal{Q}$. On truth tables, if there is a sententially valid argument from some premises $\mathcal{A}_1 \dots \mathcal{A}_n$ and $\mathcal{P}$ to conclusion $\mathcal{Q}$, then there is no row where $\mathcal{A}_1 \dots \mathcal{A}_n$ are true and $\mathcal{P}$ is true but $\mathcal{Q}$ is false—but this is just to say that there is no row where $\mathcal{A}_1 \dots \mathcal{A}_n$ are true and $\mathcal{P} \to \mathcal{Q}$ is false; so $\mathcal{A}_1 \dots \mathcal{A}_n$ entail $\mathcal{P} \to \mathcal{Q}$.

For an example, suppose we are confronted with the following:

(M)

1.	$A \to B$	P
2.	$B \to C$	P
	$A \to C$	

In general, we use an introduction rule to *produce* some formula—typically one already given as a goal. →I generates $\mathcal{P} \to \mathcal{Q}$ given a subderivation that starts with the $\mathcal{P}$ and ends with the $\mathcal{Q}$. Thus to reach $A \to C$, we need a subderivation that starts with A and ends with C. So we set up to reach $A \to C$ with the assumption A and an exit strategy to produce $A \to C$ by →I. For this we set the consequent C as a subgoal.

1.	$A \to B$	P
2.	$B \to C$	P
3.	$\quad A$	A (g, →I)
	$\quad C$	
	$A \to C$	3-_ →I

Again, we have not yet reached C or $A \to C$. Rather, we have assumed A and set C as a subgoal, with the strategy of terminating our subderivation by an application of →I. This much is stated in the exit strategy. We are not in a position to fill in the entire justification for $A \to C$, but there is no harm filling in what we can, to remind us where we are going. As it happens, the new goal C is easy to get.

$$\begin{array}{lll}
1. & A \to B & \text{P} \\
2. & \underline{B \to C} & \text{P} \\
3. & \quad | \; \underline{A} & \text{A } (g, \to\text{I}) \\
4. & \quad | \; B & 1,3 \to\text{E} \\
5. & \quad | \; C & 2,4 \to\text{E} \\
 & A \to C & 3\text{-}_ \to\text{I}
\end{array}$$

Having reached C, and so completed the subderivation, we are in a position to execute our exit strategy and conclude $A \to C$ by $\to$I.

$$\begin{array}{lll}
1. & A \to B & \text{P} \\
2. & \underline{B \to C} & \text{P} \\
3. & \quad | \; \underline{A} & \text{A } (g, \to\text{I}) \\
4. & \quad | \; B & 1,3 \to\text{E} \\
5. & \quad | \; C & 2,4 \to\text{E} \\
6. & A \to C & 3\text{-}5 \to\text{I}
\end{array}$$

We appeal to the subderivation that starts with the assumption of the antecedent, and reaches the consequent. Notice that the $\to$I setup is driven, not by the premises, but by where we want to get. We will say something more systematic about strategy once we have introduced all the rules. But here is the fundamental idea: *think goal directedly*. We begin with $A \to C$ as a goal. Our idea for producing it leads to C as a new goal. And the new goal is relatively easy to obtain.

Here is another example, one that should illustrate the above point about strategy as well as the rule. Say we want to show $A \vdash_{NDs} B \to (C \to A)$.

(N)

$$\begin{array}{lll}
1. & \underline{A} & \text{P} \\
 & & \\
 & B \to (C \to A) &
\end{array}$$

Since the goal is of the form $\mathcal{P} \to \mathcal{Q}$, we set up to get it by $\to$I.

$$\begin{array}{lll}
1. & \underline{A} & \text{P} \\
2. & \quad | \; \underline{B} & \text{A } (g, \to\text{I}) \\
 & & \\
 & \quad | \; C \to A & \\
 & B \to (C \to A) & 2\text{-}_ \to\text{I}
\end{array}$$

We need a subderivation that starts with the antecedent and ends with the consequent. So we assume the antecedent, and set the consequent as a new goal. In this case, the new goal $C \to A$ has main operator $\to$, so we set up again to reach *it* by $\to$I.

$$\begin{array}{rll}
1. & A & \text{P} \\
2. & \quad B & \text{A } (g, \to\text{I}) \\
3. & \quad\quad C & \text{A } (g, \to\text{I}) \\
 & \quad\quad A & \\
 & \quad C \to A & 3\text{-}_ \to\text{I} \\
 & B \to (C \to A) & 2\text{-}_ \to\text{I}
\end{array}$$

The pointer g in an exit strategy points to the goal formula at the bottom of its scope line. Thus g for assumption B at (2) points to $C \to A$ at the bottom of its line, and g for assumption C at (3) points to A at the bottom of *its* line. Again, for the conditional, we assume the antecedent, and set the consequent as a new goal. And this last goal is particularly easy to reach. It follows immediately by reiteration from (1). Then it is a simple matter of executing the exit strategies with which our auxiliary assumptions were introduced.

$$\begin{array}{rll}
1. & A & \text{P} \\
2. & \quad B & \text{A } (g, \to\text{I}) \\
3. & \quad\quad C & \text{A } (g, \to\text{I}) \\
4. & \quad\quad A & 1 \text{ R} \\
5. & \quad C \to A & 3\text{-}4 \to\text{I} \\
6. & B \to (C \to A) & 2\text{-}5 \to\text{I}
\end{array}$$

The subderivation which begins on (3) and ends on (4) begins with the antecedent and ends with the consequent of $C \to A$. So we conclude $C \to A$ on (5) by 3-4 $\to$I. The subderivation which begins on (2) and ends at (5) begins with the antecedent and ends with the consequent of $B \to (C \to A)$. So we reach $B \to (C \to A)$ on (6) by 2-5 $\to$I. Notice again how our overall reasoning is driven by the goals, rather than the premises and assumptions. It is sometimes difficult to motivate strategy when derivations are short and relatively easy. But this sort of thinking will serve you well as problems get more difficult!

Given what we have done, the E- and I-rules for $\wedge$ are completely straightforward. First the exploitation rule: If $\mathcal{P} \wedge \mathcal{Q}$ appears on some accessible line a of a derivation, then you may move to the $\mathcal{P}$ or to the $\mathcal{Q}$ with justification a $\wedge$E.

$$\wedge\text{E} \qquad \begin{array}{rll}
\text{a.} & \mathcal{P} \wedge \mathcal{Q} & \\
 & \mathcal{P} & \text{a } \wedge\text{E}
\end{array} \qquad\qquad \begin{array}{rll}
\text{a.} & \mathcal{P} \wedge \mathcal{Q} & \\
 & \mathcal{Q} & \text{a } \wedge\text{E}
\end{array}$$

Either qualifies as an instance of the rule. The left-hand case was R3 from *NP*. Intuitively, $\wedge$E should be clear. If $\mathcal{P}$ *and* $\mathcal{Q}$ is true, then $\mathcal{P}$ is true. And if $\mathcal{P}$ *and* $\mathcal{Q}$ is true, then $\mathcal{Q}$ is true. We saw a table for the left-hand case in (D). The other is similar. The $\wedge$ introduction rule is equally straightforward. If $\mathcal{P}$ and $\mathcal{Q}$ appear on accessible lines a and b of a derivation, then you may move to $\mathcal{P} \wedge \mathcal{Q}$ with justification *a,b* $\wedge$I.

$$\wedge\text{I}\qquad \begin{array}{r|ll} \text{a.} & \mathscr{P} & \\ \text{b.} & \mathscr{Q} & \\ & & \\ & \mathscr{P} \wedge \mathscr{Q} & \text{a,b } \wedge\text{I} \end{array}$$

The order in which $\mathscr{P}$ and $\mathscr{Q}$ appear is irrelevant, though you should cite them in the specified order, line for the left conjunct first, and then for the right. If $\mathscr{P}$ is true and $\mathscr{Q}$ is true, then $\mathscr{P}$ *and* $\mathscr{Q}$ is true. Similarly, on a table, any line with both $\mathscr{P}$ and $\mathscr{Q}$ true has $\mathscr{P} \wedge \mathscr{Q}$ true.

Here is a simple example, demonstrating the *associativity* of conjunction.

$$\text{(O)}\qquad \begin{array}{r|ll} 1. & A \wedge (B \wedge C) & \text{P} \\ \hline 2. & A & 1\ \wedge\text{E} \\ 3. & B \wedge C & 1\ \wedge\text{E} \\ 4. & B & 3\ \wedge\text{E} \\ 5. & C & 3\ \wedge\text{E} \\ 6. & A \wedge B & 2,4\ \wedge\text{I} \\ 7. & (A \wedge B) \wedge C & 6,5\ \wedge\text{I} \end{array}$$

Notice that we could not get the B alone or the C alone without first isolating $B \wedge C$ on (3). As before, our rules apply just to the *main* operator. In effect, we take apart the premise with the E-rule, and put the conclusion together with the I-rule. Of course, as with $\rightarrow$I and $\rightarrow$E, rules for other operators do not always let us get to the parts and put them together in this simple and symmetric way.

A final example brings together all of the rules so far (except R).

$$\text{(P)}\qquad \begin{array}{r|l|ll} 1. & \multicolumn{2}{l}{A \rightarrow C} & \text{P} \\ \hline 2. & & A \wedge B & \text{A } (g, \rightarrow\text{I}) \\ \cline{3-3} 3. & & A & 2\ \wedge\text{E} \\ 4. & & C & 1,3 \rightarrow\text{E} \\ 5. & & B & 2\ \wedge\text{E} \\ 6. & & B \wedge C & 5,4\ \wedge\text{I} \\ 7. & \multicolumn{2}{l}{(A \wedge B) \rightarrow (B \wedge C)} & \text{2-6} \rightarrow\text{I} \end{array}$$

We set up to obtain the overall goal by $\rightarrow$I. This generates $B \wedge C$ as a subgoal. We get $B \wedge C$ by getting the B and the C.

Here is our guiding idea for strategy (which may now seem obvious): As you focus on a goal, to generate a formula with any main operator, consider producing it by the corresponding introduction rule. Thus if the main operator of a goal or subgoal is $\rightarrow$, consider producing the formula by $\rightarrow$I; if the main operator of a goal is $\wedge$, consider producing it by $\wedge$I. You make use of a formula with main operator $\rightarrow$ by $\rightarrow$E and of a formula with main operator $\wedge$ with $\wedge$E. This much should be sufficient for you to approach the following exercises. As you approach these and other derivations, you may find the *NDs* quick reference on page 225 helpful. As you work the derivations, it is good simply to leave plenty of space on the page for your derivation as you state goal formulas, and let there be blank lines if room remains.[3]

Words to the wise:

- A common mistake made by beginning students is to assimilate other rules to $\wedge$E and $\wedge$I—moving, say, from $\mathscr{P} \rightarrow \mathscr{Q}$ alone to $\mathscr{P}$ or $\mathscr{Q}$, or from $\mathscr{P}$ and $\mathscr{Q}$ to $\mathscr{P} \rightarrow \mathscr{Q}$. *Do not forget what you have learned! Do not make this mistake!* The $\wedge$ rules are particularly easy. But each operator has its own special character. Thus $\rightarrow$E requires two "cards" to play. And $\rightarrow$I takes a subderivation as input.

- Another common mistake is to assume a formula $\mathscr{P}$ merely because it would be nice to have access to $\mathscr{P}$. *Do not make this mistake!* An assumption always comes with an exit strategy, and is useful only for application of the exit rule. At this stage, then, the *only* reason to assume $\mathscr{P}$ is to produce a formula of the sort $\mathscr{P} \rightarrow \mathscr{Q}$ by $\rightarrow$I.

- Our little system *NP* introduced the idea of a derivation game. But we are introducing *ND* from scratch. At this stage, then, the only rules for derivations in *NDs* are R, $\rightarrow$I, $\rightarrow$E, $\wedge$I, and $\wedge$E.

E6.5. Complete the following derivations in *NDs* by filling in justifications for each line. Hint: It may be convenient to print or xerox the problems, and fill in your answers directly on the copy.

a.

$$
\begin{array}{r|l}
1. & (A \wedge B) \rightarrow C \\
2. & B \wedge A \\
\hline
3. & B \\
4. & A \\
5. & A \wedge B \\
6. & C
\end{array}
$$

b.

$$
\begin{array}{r|l}
1. & (R \rightarrow L) \wedge [(S \vee R) \rightarrow (T \leftrightarrow K)] \\
2. & (R \rightarrow L) \rightarrow (S \vee R) \\
\hline
3. & R \rightarrow L \\
4. & S \vee R \\
5. & (S \vee R) \rightarrow (T \leftrightarrow K) \\
6. & T \leftrightarrow K
\end{array}
$$

[3]Typing on a computer it is easy to push lines down if you need more room. It is not so easy with pencil and paper, and worse with pen. If you are not already using it, the Symbolic Logic APPlication (SLAPP) available from `https://tonyroyphilosophy.net/slapp/` is an electronic option including both checking and help. (See also Chapter 13, page 676 note 12.)

*c.

$$\begin{array}{rl}
1. & B \\
2. & (A \rightarrow B) \rightarrow (B \rightarrow (L \wedge S)) \\
3. & \quad A \\
4. & \quad B \\
5. & A \rightarrow B \\
6. & B \rightarrow (L \wedge S) \\
7. & L \wedge S \\
8. & S \\
9. & L \\
10. & S \wedge L
\end{array}$$

d.

$$\begin{array}{rl}
1. & A \wedge B \\
2. & \quad C \\
3. & \quad A \\
4. & \quad A \wedge C \\
5. & C \rightarrow (A \wedge C) \\
6. & \quad C \\
7. & \quad B \\
8. & \quad B \wedge C \\
9. & C \rightarrow (B \wedge C) \\
10. & [C \rightarrow (A \wedge C)] \wedge [C \rightarrow (B \wedge C)]
\end{array}$$

e.

$$\begin{array}{rl}
1. & (A \wedge S) \rightarrow C \\
2. & \quad A \\
3. & \quad\quad S \\
4. & \quad\quad A \wedge S \\
5. & \quad\quad C \\
6. & \quad S \rightarrow C \\
7. & A \rightarrow (S \rightarrow C)
\end{array}$$

E6.6. The following are not legitimate *NDs* derivations. In each case, explain why.

*a.

$$\begin{array}{rll}
1. & (A \wedge B) \wedge (C \rightarrow B) & \text{P} \\
2. & A & 1 \wedge\text{E}
\end{array}$$

b.

$$\begin{array}{rll}
1. & (A \wedge B) \wedge (C \rightarrow A) & \text{P} \\
2. & C & \text{P} \\
3. & A & 1,2 \rightarrow\text{E}
\end{array}$$

c.

$$\begin{array}{rll}
1. & (R \wedge S) \wedge (C \rightarrow A) & \text{P} \\
2. & C \rightarrow A & 1 \wedge\text{E} \\
3. & A & 2 \rightarrow\text{E}
\end{array}$$

d.

1.	$A \to B$	P
2.	$\quad A \wedge C$	A (g, $\to$I)
3.	$\quad A$	2 $\wedge$E
4.	B	1,3 $\to$E

e.

1.	$A \to B$	P
2.	$\quad A \wedge C$	A (g, $\to$I)
3.	$\quad A$	2 $\wedge$E
4.	$\quad B$	1,3 $\to$E
5.	$\quad C$	2 $\wedge$E
6.	$\quad B \wedge C$	4,5 $\wedge$I

Hint: This last problem (e) does not break any derivation rule. However, it still fails to derive $B \wedge C$ from the premise. Explain why.

E6.7. Provide derivations to show each of the following.

a. $A \wedge B \vdash_{NDs} B \wedge A$

*b. $A \wedge B,\ B \to C \vdash_{NDs} C$

c. $A \wedge (A \to (A \wedge B)) \vdash_{NDs} B$

d. $A \wedge B,\ B \to (C \wedge D) \vdash_{NDs} A \wedge D$

*e. $A \to (A \to B) \vdash_{NDs} A \to B$

f. $A,\ (A \wedge B) \to (C \wedge D) \vdash_{NDs} B \to C$

g. $C \to A,\ C \to (A \to B) \vdash_{NDs} C \to (A \wedge B)$

*h. $A \to B,\ B \to C \vdash_{NDs} (A \wedge K) \to C$

i. $A \to B \vdash_{NDs} (A \wedge C) \to (B \wedge C)$

j. $D \wedge E,\ (D \to F) \wedge (E \to G) \vdash_{NDs} F \wedge G$

k. $O \to B,\ B \to S,\ S \to L \vdash_{NDs} O \to L$

*l. $A \to B \vdash_{NDs} (C \to A) \to (C \to B)$

m. $A \to (B \to C) \vdash_{NDs} B \to (A \to C)$

n. $A \to (B \to C),\ D \to B \vdash_{NDs} A \to (D \to C)$

o. $A \to B \vdash_{NDs} A \to (C \to B)$

6.2.2 $\sim$ and $\vee$

Now let us consider the I- and E-rules for $\sim$ and $\vee$. The two rules for $\sim$ are quite similar to one another. Each appeals to a single subderivation. For $\sim$I, given an accessible subderivation which begins with assumption $\mathcal{P}$ on line a, and ends with a formula of the form $\mathcal{Q} \wedge \sim\mathcal{Q}$ against its scope line on line b, one may conclude $\sim\mathcal{P}$ by *a*-*b* $\sim$I. For $\sim$E, given an accessible subderivation which begins with assumption $\sim\mathcal{P}$ on line a, and ends with a formula of the form $\mathcal{Q} \wedge \sim\mathcal{Q}$ against its scope line on line b, one may conclude $\mathcal{P}$ by *a*-*b* $\sim$E.

$\sim$I

a.	$\quad\mid \mathcal{P}$	A (c, $\sim$I)
b.	$\quad\mid \mathcal{Q} \wedge \sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a-b $\sim$I

$\sim$E

a.	$\quad\mid \sim\mathcal{P}$	A (c, $\sim$E)
b.	$\quad\mid \mathcal{Q} \wedge \sim\mathcal{Q}$	
	$\mathcal{P}$	a-b $\sim$E

$\sim$I introduces an expression with main operator tilde, adding tilde to the assumption $\mathcal{P}$. $\sim$E exploits the assumption $\sim\mathcal{P}$, with a result that takes the tilde off. For these rules, the formula $\mathcal{Q}$ may be *any* formula, so long as $\sim\mathcal{Q}$ is *it* with a tilde in front. Because $\mathcal{Q}$ may be any formula, when we declare our exit strategy for the assumption, we might have no particular goal formula in mind. So, where g always points to a formula written at the bottom of a scope line, c is not a pointer to any particular formula. Rather, when we declare our exit strategy, we merely indicate our intent to obtain some contradiction, and then to exit by $\sim$I or $\sim$E.

Intuitively, if an assumption leads to a result that is false, the assumption is wrong. So if the assumption $\mathcal{P}$ leads to both $\mathcal{Q}$ and $\sim\mathcal{Q}$ and so to $\mathcal{Q} \wedge \sim\mathcal{Q}$, then we can discharge the assumption and conclude $\sim\mathcal{P}$; and if the assumption $\sim\mathcal{P}$ leads to $\mathcal{Q}$ and $\sim\mathcal{Q}$ and so $\mathcal{Q} \wedge \sim\mathcal{Q}$, then we discharge the assumption and conclude $\mathcal{P}$. On tables, there can be no row where both $\mathcal{Q}$ and $\sim\mathcal{Q}$ are true; so if every row where premises $\mathcal{A}_1 \ldots \mathcal{A}_n$ and $\mathcal{P}$ are true would have to make both $\mathcal{Q}$ and $\sim\mathcal{Q}$ true, there is no row where $\mathcal{A}_1 \ldots \mathcal{A}_n$ and $\mathcal{P}$ are true; so on a row where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are true $\sim\mathcal{P}$ is true. Similarly when the assumption is $\sim\mathcal{P}$, a row where premises $\mathcal{A}_1 \ldots \mathcal{A}_n$ are true has $\mathcal{P}$ true.

Here are some examples of these rules. Notice that, again, we introduce subderivations with the overall goal in mind.

(Q)

1.	$A \rightarrow B$	P
2.	$A \rightarrow \sim B$	P
3.	$\quad\mid A$	A (c, $\sim$I)
4.	$\quad\mid B$	1,3 $\rightarrow$E
5.	$\quad\mid \sim B$	2,3 $\rightarrow$E
6.	$\quad\mid B \wedge \sim B$	4,5 $\wedge$I
7.	$\sim A$	3-6 $\sim$I

We begin with the goal of obtaining $\sim A$. The natural way to obtain this is by $\sim$I. So we set up a subderivation with that in mind. Since the goal is $\sim A$, we begin with A

and go for a contradiction. In this case, the contradiction is easy to obtain by a couple applications of $\rightarrow$E and then $\wedge$I.

Here is another case that may be more interesting:

$$
\text{(R)}\qquad
\begin{array}{r|l|ll}
1. & \sim A & & \text{P}\\
2. & B \rightarrow A & & \text{P}\\
\hline
3. & & L \wedge B & \text{A } (c, \sim\text{I})\\
4. & & B & 3\ \wedge\text{E}\\
5. & & A & 2{,}4 \rightarrow\text{E}\\
6. & & A \wedge \sim A & 5{,}1\ \wedge\text{I}\\
7. & \sim(L \wedge B) & & 3\text{-}6 \sim\text{I}
\end{array}
$$

This time, the original goal is $\sim(L \wedge B)$. It is of the form $\sim\mathcal{P}$, so we set up to obtain it with a subderivation that begins with the $\mathcal{P}$, that is, $L \wedge B$. In this case, the contradiction is $A \wedge \sim A$. Once we have the contradiction, we simply apply our exit strategy.

A simplification. For any sentential or quantificational language $\mathcal{L}$ let $\perp$ (bottom) abbreviate some sentence of the form $\mathcal{Z} \wedge \sim\mathcal{Z}$—for $\mathcal{L}_s$ let $\perp$ just be $Z \wedge \sim Z$. Adopt a rule $\perp$I as on the left below,

$$
\perp\text{I}\qquad
\begin{array}{r|ll}
\text{a.} & \mathcal{Q} & \\
\text{b.} & \sim\mathcal{Q} & \\
 & & \\
 & \perp & \text{a,b } \perp\text{I}
\end{array}
\qquad\qquad
\text{(S)}\qquad
\begin{array}{r|l|ll}
1. & \mathcal{Q} & & \\
2. & \sim\mathcal{Q} & & \\
\hline
3. & & \sim\perp & \text{A } (c, \sim\text{E})\\
4. & & \mathcal{Q} \wedge \sim\mathcal{Q} & 1{,}2\ \wedge\text{I}\\
5. & \perp & & 3\text{-}4 \sim\text{E}
\end{array}
$$

Given $\mathcal{Q}$ and $\sim\mathcal{Q}$ on accessible lines, we move directly to $\perp$ by $\perp$I. This is an example of a *derived* rule. For given $\mathcal{Q}$ and $\sim\mathcal{Q}$, we can always derive $\perp$ as in (S) on the right. Thus we allow ourselves to shortcut the routine by introducing $\perp$I as a derived rule. We will see examples of additional derived rules in section 6.2.5. For now, the important thing is that since $\perp$ abbreviates $Z \wedge \sim Z$ we *operate* with $\perp$ as we might operate with $Z \wedge \sim Z$. Especially, given this abbreviation, our $\sim$I and $\sim$E rules appear in forms,

$$
\sim\text{I}\qquad
\begin{array}{r|l|ll}
\text{a.} & & \mathcal{P} & \text{A } (c, \sim\text{I})\\
 & & & \\
\text{b.} & & \perp & \\
 & \sim\mathcal{P} & & \text{a-b} \sim\text{I}
\end{array}
\qquad\qquad
\sim\text{E}\qquad
\begin{array}{r|l|ll}
\text{a.} & & \sim\mathcal{P} & \text{A } (c, \sim\text{E})\\
 & & & \\
\text{b.} & & \perp & \\
 & \mathcal{P} & & \text{a-b} \sim\text{E}
\end{array}
$$

Since $\perp$ is (abbreviates) $Z \wedge \sim Z$, the subderivations for $\sim$I and $\sim$E are appropriately concluded with $\perp$.[4] With $\perp$ as their last line, subderivations for $\sim$I and $\sim$E have a

[4] $\perp$ is often introduced as a primitive symbol. We have chosen not to extend the primitives, and so to treat it as an abbreviation. On the above account, then, one might derive $\perp$ from Z and $\sim Z$ by $\wedge$I; or use $\wedge$E to conclude Z or $\sim Z$ from $\perp$.

particular goal sentence very much like $\to$I. However, the $\mathcal{Q}$ and $\sim\mathcal{Q}$ required to obtain $\bot$ by $\bot$I are the same as would be required for $\mathcal{Q} \wedge \sim\mathcal{Q}$ on the original form of the rules. For this reason, we declare our exit strategy with a *c* rather than *g* any time the goal is $\bot$. At one level, this simplification is a mere notational convenience: having obtained $\mathcal{Q}$ and $\sim\mathcal{Q}$, we move to $\bot$, instead of writing out the complex conjunction $\mathcal{Q} \wedge \sim\mathcal{Q}$. However, there are contexts where it will be convenient to have a *particular* contradiction as goal. Thus this is the standard form in which we use these rules.

Here is an example of the rules in this form, this time for $\sim$E.

(T)

$$\begin{array}{rlll}
1. & \mid & \sim\sim A & \text{P} \\
2. & \mid & \mid\ \sim A & \text{A } (c, \sim\text{E}) \\
3. & \mid & \mid\ \bot & 2,1\ \bot\text{I} \\
4. & \mid & A & \text{2-3 } \sim\text{E}
\end{array}$$

It is no surprise that we can derive A from $\sim\sim A$. This is how to do it in *NDs*. Again, we begin from the goal. In this case the goal is A, and we can get it with a subderivation that starts with $\sim A$, by a $\sim$E exit strategy. In this case the $\mathcal{Q}$ and $\sim\mathcal{Q}$ for $\bot$I are $\sim A$ and $\sim\sim A$—that is $\sim A$ and $\sim A$ with a tilde in front of it. Though very often (at least in the beginning) an atomic and its negation will do for your contradiction, $\mathcal{Q}$ and $\sim\mathcal{Q}$ need not be simple. Observe that $\sim$E is a strange and powerful rule: Though an E-rule, effectively it can be used in pursuit of any goal whatsoever—to obtain formula $\mathcal{P}$ by $\sim$E, all one has to do is obtain a contradiction from the assumption of $\mathcal{P}$ with a tilde in front. As in this last example (T), $\sim$E is particularly useful when the goal is an atomic formula, and thus without a main operator, so that there is no straightforward way for regular introduction rules to apply. In this way, it plays the role of a sort of "backdoor" introduction rule.

The $\vee$I and $\vee$E rules apply methods we have already seen. For $\vee$I, given an accessible formula $\mathcal{P}$ on line *a*, one may move to either $\mathcal{P} \vee \mathcal{Q}$ or to $\mathcal{Q} \vee \mathcal{P}$ for any formula $\mathcal{Q}$, with justification *a* $\vee$I.

$\vee$I

$$\begin{array}{rlll}
\text{a.} & \mid & \mathcal{P} & \\
 & \mid & & \\
 & \mid & \mathcal{P} \vee \mathcal{Q} & \text{a } \vee\text{I}
\end{array}
\qquad\qquad
\begin{array}{rlll}
\text{a.} & \mid & \mathcal{P} & \\
 & \mid & & \\
 & \mid & \mathcal{Q} \vee \mathcal{P} & \text{a } \vee\text{I}
\end{array}$$

The left-hand case was R4 from *NP*. Table (D) exhibits the left-hand case. And the other side should be clear as well: Any row of a table where $\mathcal{P}$ is true has both $\mathcal{P} \vee \mathcal{Q}$ and $\mathcal{Q} \vee \mathcal{P}$ true.

Here is a simple example:

(U)

$$\begin{array}{rlll}
1. & \mid & P & \text{P} \\
2. & \mid & (P \vee Q) \to R & \text{P} \\
3. & \mid & P \vee Q & 1\ \vee\text{I} \\
4. & \mid & R & 2,3\ \to\text{E}
\end{array}$$

It is easy to get R once we have $P \vee Q$. And we build $P \vee Q$ directly from the P. Note that we could have done the derivation as well if (2) had been, say,

$(P \vee [K \wedge (L \leftrightarrow T)]) \rightarrow R$ and we used $\vee$I to add $[K \wedge (L \leftrightarrow T)]$ to the P all at once.

The inputs to $\vee$E are a formula of the form $\mathcal{P} \vee \mathcal{Q}$ and *two* subderivations. Given an accessible formula of the form $\mathcal{P} \vee \mathcal{Q}$ on line a, with an accessible subderivation beginning with assumption $\mathcal{P}$ on line b and ending with conclusion $\mathcal{C}$ against its scope line at c, and an accessible subderivation beginning with assumption $\mathcal{Q}$ on line d and ending with conclusion $\mathcal{C}$ against its scope line at e, one may conclude $\mathcal{C}$ with justification *a,b-c,d-e* $\vee$E.

$\vee$E	a.	$\mathcal{P} \vee \mathcal{Q}$	
	b.	$\quad \mathcal{P}$	A (g, a$\vee$E)
	c.	$\quad \mathcal{C}$	
	d.	$\quad \mathcal{Q}$	A (g, a$\vee$E)
	e.	$\quad \mathcal{C}$	
		$\mathcal{C}$	a,b-c,d-e $\vee$E

Given a disjunction $\mathcal{P} \vee \mathcal{Q}$, one subderivation begins with $\mathcal{P}$, and the other with $\mathcal{Q}$; both conclude with $\mathcal{C}$. This time our exit strategy includes markers for the new subgoals, along with a notation that we exit by appeal to the disjunction on line a and $\vee$E. Intuitively, if we know it is one or the other, and *both* lead to some conclusion, then the conclusion must be true. Here is an example a student gave me near graduation time: She and her mother were shopping for a graduation dress. They narrowed it down to dress A or dress B. Dress A was expensive, and if they bought it, her mother would be mad. But dress B was ugly and if they bought it the student would complain and her mother would be mad. Conclusion: her mother would be mad—and this without knowing which dress they were going to buy! On a truth table, if rows where $\mathcal{P}$ is true have $\mathcal{C}$ true, and rows where $\mathcal{Q}$ is true have $\mathcal{C}$ true, then any row with $\mathcal{P} \vee \mathcal{Q}$ true must have one of $\mathcal{P}$ or $\mathcal{Q}$ true and so $\mathcal{C}$ true as well.

Here are a couple of examples. The first is straightforward, and illustrates both the $\vee$I and $\vee$E rules.

(V)	1.	$A \vee B$	P
	2.	$A \rightarrow C$	P
	3.	$\quad A$	A (g, 1$\vee$E)
	4.	$\quad C$	2,3 $\rightarrow$E
	5.	$\quad B \vee C$	4 $\vee$I
	6.	$\quad B$	A (g, 1$\vee$E)
	7.	$\quad B \vee C$	6 $\vee$I
	8.	$B \vee C$	1,3-5,6-7 $\vee$E

We have the disjunction $A \vee B$ as premise, and original goal $B \vee C$. And we set up to obtain the goal by $\vee$E. For this, one subderivation starts with A and ends with $B \vee C$, and the other starts with B and ends with $B \vee C$. As it happens, these subderivations are easy to complete.

Very often, beginning students resist using $\vee$E—no doubt because it is relatively messy. *But this is a mistake—$\vee$E is your friend!* In fact, with this rule, we have a case where it pays to look at accessible formulas for general strategy. If you have an accessible line of the form $\mathscr{P} \vee \mathscr{Q}$, go for your goal, whatever it is, by $\vee$E. Here is why: As you go for the goal in the first subderivation, you have whatever sentences were accessible before, *plus* $\mathscr{P}$; and as you go for the goal in the second subderivation, you have whatever sentences were accessible before *plus* $\mathscr{Q}$. So you can only be better off in your quest to reach the goal. In many cases where an accessible formula has main operator $\vee$, there is no way to complete the derivation except by $\vee$E. The above example (V) is a case in point.

Here is a relatively messy example, which should help you be sure you understand the $\vee$ rules. It illustrates the *associativity* of disjunction.

(W)

1.	$A \vee (B \vee C)$	P
2.	$\quad A$	A (g, 1$\vee$E)
3.	$\quad A \vee B$	2 $\vee$I
4.	$\quad (A \vee B) \vee C$	3 $\vee$I
5.	$\quad B \vee C$	A (g, 1$\vee$E)
6.	$\quad\quad B$	A (g, 5$\vee$E)
7.	$\quad\quad A \vee B$	6 $\vee$I
8.	$\quad\quad (A \vee B) \vee C$	7 $\vee$I
9.	$\quad\quad C$	A (g, 5$\vee$E)
10.	$\quad\quad (A \vee B) \vee C$	9 $\vee$I
11.	$\quad (A \vee B) \vee C$	5,6-8,9-10 $\vee$E
12.	$(A \vee B) \vee C$	1,2-4,5-11 $\vee$E

The premise has main operator $\vee$. So we set up to obtain the goal by $\vee$E. This gives us subderivations starting with A and $B \vee C$, each with $(A \vee B) \vee C$ as goal. The first is easy to complete by a couple instances of $\vee$I. But the assumption of the second, $B \vee C$ has main operator $\vee$. So we set up to obtain *its* goal by $\vee$E. This gives us subderivations starting with B and C, each again having $(A \vee B) \vee C$ as goal. Again, these are easy to complete by application of $\vee$I. The final result follows by the planned applications of $\vee$E. If you have been able to follow this case, you are doing well!

E6.8. Complete the following derivations by filling in justifications for each line. Hint: Begin by identifying the exit strategy for auxiliary assumptions; then the rest will be straightforward.

a.

1. | $\sim B$
2. | $(\sim A \vee C) \to (B \wedge C)$
3. | | $\sim A$
4. | | $\sim A \vee C$
5. | | $B \wedge C$
6. | | B
7. | | $\bot$
8. | A

b.

1. | R
2. | $\sim(S \vee T)$
3. | | $R \to S$
4. | | S
5. | | $S \vee T$
6. | | $\bot$
7. | $\sim(R \to S)$

c.

1. | $(R \wedge S) \vee (K \wedge L)$
2. | | $R \wedge S$
3. | | R
4. | | S
5. | | $S \wedge R$
6. | | $(S \wedge R) \vee (L \wedge K)$
7. | | $K \wedge L$
8. | | K
9. | | L
10. | | $L \wedge K$
11. | | $(S \wedge R) \vee (L \wedge K)$
12. | $(S \wedge R) \vee (L \wedge K)$

d.

1. | $A \vee B$
2. | | A
3. | | | $A \to B$
4. | | | B
5. | | $(A \to B) \to B$
6. | | B
7. | | | $A \to B$
8. | | | B
9. | | $(A \to B) \to B$
10. | $(A \to B) \to B$

e.

1.	$\sim B$	
2.	$\sim A \rightarrow (A \vee B)$	
3.	$\quad \sim A$	
4.	$\quad A \vee B$	
5.	$\quad\quad A$	
6.	$\quad\quad A$	
7.	$\quad\quad B$	
8.	$\quad\quad\quad \sim A$	
9.	$\quad\quad\quad \bot$	
10.	$\quad\quad A$	
11.	$\quad A$	
12.	$\quad \bot$	
13.	A	

E6.9. The following are not legitimate *NDs* derivations. In each case, explain why.

a.

1.	$A \vee B$	P
2.	B	1 $\vee$E

b.

1.	$\sim A$	P
2.	$B \rightarrow A$	P
3.	$\quad B$	A (c, $\sim$I)
4.	$\quad A$	2,3 $\rightarrow$E
5.	$\sim B$	3-4 $\sim$I

*c.

1.	W	P
2.	$\quad R$	A (c, $\sim$I)
3.	$\quad\quad \sim W$	A (c, $\sim$E)
4.	$\quad\quad \bot$	1,3 $\bot$I
5.	$\quad W$	3-4 $\sim$E
6.	$\sim R$	2-5 $\sim$I

d.

1.	$A \vee B$	P
2.	$\quad A$	A (g, 1$\vee$E)
3.	$\quad A$	2 R
4.	$\quad B$	A (g, 1$\vee$E)
5.	$\quad A$	3 R
6.	A	1,2-3,4-5 $\vee$E

e.

1.	$\mid A \vee B$	P
2.	$\mid\ \mid A$	A (g, 1$\vee$E)
3.	$\mid\ \mid A$	2 R
4.	$\mid\ \mid A$	A (g, $\rightarrow$I)
5.	$\mid\ \mid\ \mid B$	A (g, 1$\vee$E)
6.	$\mid\ \mid\ \mid A$	4 R
7.	$\mid\ \mid A$	4 R
8.	$\mid A$	1,2-3,5-6 $\vee$E

E6.10. Produce derivations to show each of the following.

a. $\sim A \vdash_{NDs} \sim(A \wedge B)$

b. $A \vdash_{NDs} \sim\sim A$

*c. $\sim A \rightarrow B, \sim B \vdash_{NDs} A$

d. $A \rightarrow B \vdash_{NDs} \sim(A \wedge \sim B)$

e. $\sim A \rightarrow B,\ B \rightarrow A \vdash_{NDs} A$

f. $A \wedge B \vdash_{NDs} (R \leftrightarrow S) \vee B$

*g. $A \vee (A \wedge B) \vdash_{NDs} A$

h. $S, (B \vee C) \rightarrow \sim S \vdash_{NDs} \sim B$

i. $A \vee B,\ A \rightarrow B,\ B \rightarrow A \vdash_{NDs} A \wedge B$

j. $A \rightarrow B, (B \vee C) \rightarrow D,\ D \rightarrow \sim A \vdash_{NDs} \sim A$

k. $A \vee B \vdash_{NDs} B \vee A$

*l. $A \rightarrow \sim B \vdash_{NDs} B \rightarrow \sim A$

m. $(A \wedge B) \rightarrow \sim A \vdash_{NDs} A \rightarrow \sim B$

n. $A \vee \sim\sim B \vdash_{NDs} A \vee B$

o. $A \vee B, \sim B \vdash_{NDs} A$

6.2.3 $\leftrightarrow$

We complete our presentation of rules for *NDs* with the rules $\leftrightarrow$E and $\leftrightarrow$I. Given that $\mathcal{P} \leftrightarrow \mathcal{Q}$ abbreviates the same as $(\mathcal{P} \rightarrow \mathcal{Q}) \wedge (\mathcal{Q} \rightarrow \mathcal{P})$, it is not surprising that rules for $\leftrightarrow$ work like ones for arrow, but going two ways. For $\leftrightarrow$E, if formulas $\mathcal{P} \leftrightarrow \mathcal{Q}$ and $\mathcal{P}$ appear on accessible lines a and b of a derivation, we may conclude $\mathcal{Q}$ with justification *a,b* $\leftrightarrow$E; and similarly but in the other direction, if formulas $\mathcal{P} \leftrightarrow \mathcal{Q}$ and $\mathcal{Q}$ appear on accessible lines a and b of a derivation, we may conclude $\mathcal{P}$ with justification *a,b* $\leftrightarrow$E.

$$\leftrightarrow\text{E} \qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\ \text{b.} & \mathcal{P} & \\ & & \\ & \mathcal{Q} & \text{a,b} \leftrightarrow\text{E} \end{array} \qquad\qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\ \text{b.} & \mathcal{Q} & \\ & & \\ & \mathcal{P} & \text{a,b} \leftrightarrow\text{E} \end{array}$$

$\mathcal{P} \leftrightarrow \mathcal{Q}$ thus works like either $\mathcal{P} \rightarrow \mathcal{Q}$ or $\mathcal{Q} \rightarrow \mathcal{P}$. Intuitively given $\mathcal{P}$ if and *only if* $\mathcal{Q}$, then if $\mathcal{P}$ is true, $\mathcal{Q}$ is true. And given $\mathcal{P}$ *if* and only if $\mathcal{Q}$, then if $\mathcal{Q}$ is true $\mathcal{P}$ is true. On tables, if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true, then $\mathcal{P}$ and $\mathcal{Q}$ have the same truth value. So if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true and $\mathcal{P}$ is true, $\mathcal{Q}$ is true as well; and if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true and $\mathcal{Q}$ is true, $\mathcal{P}$ is true as well.

Given that $\mathcal{P} \leftrightarrow \mathcal{Q}$ can be exploited like $\mathcal{P} \rightarrow \mathcal{Q}$ or $\mathcal{Q} \rightarrow \mathcal{P}$, it is not surprising that introducing $\mathcal{P} \leftrightarrow \mathcal{Q}$ is like introducing both $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{Q} \rightarrow \mathcal{P}$. The input to $\leftrightarrow$I is *two* subderivations. Given an accessible subderivation beginning with assumption $\mathcal{P}$ on line a and ending with conclusion $\mathcal{Q}$ against its scope line on b, and an accessible subderivation beginning with assumption $\mathcal{Q}$ on line c and ending with conclusion $\mathcal{P}$ against its scope line on d, one may conclude $\mathcal{P} \leftrightarrow \mathcal{Q}$ with justification, *a-b,c-d* $\leftrightarrow$I.

$$\leftrightarrow\text{I} \qquad \begin{array}{r|l|ll} \text{a.} & & \mathcal{P} & \text{A } (g, \leftrightarrow\text{I}) \\ \cline{3-3} & & & \\ \text{b.} & & \mathcal{Q} & \\ & & & \\ \text{c.} & & \mathcal{Q} & \text{A } (g, \leftrightarrow\text{I}) \\ \cline{3-3} & & & \\ \text{d.} & & \mathcal{P} & \\ & \multicolumn{2}{l}{\mathcal{P} \leftrightarrow \mathcal{Q}} & \text{a-b,c-d} \leftrightarrow\text{I} \end{array}$$

Intuitively, if an assumption $\mathcal{P}$ leads to $\mathcal{Q}$ and the assumption $\mathcal{Q}$ leads to $\mathcal{P}$, then we know that $\mathcal{P}$ *only if* $\mathcal{Q}$, and $\mathcal{P}$ *if* $\mathcal{Q}$—which is to say that $\mathcal{P}$ if and only if $\mathcal{Q}$. On truth tables, if there is a sententially valid argument from premises $\mathcal{A}_1 \ldots \mathcal{A}_n$ and $\mathcal{P}$ to conclusion $\mathcal{Q}$, then there is no row where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are true and $\mathcal{P}$ is true and $\mathcal{Q}$ is false; and if there is a sententially valid argument from $\mathcal{A}_1 \ldots \mathcal{A}_n$ and $\mathcal{Q}$ to conclusion $\mathcal{P}$, then there is no row where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are true and $\mathcal{Q}$ is true and $\mathcal{P}$ is false; so on rows where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are true, it is not the case that one of $\mathcal{P}$ or $\mathcal{Q}$ is true and the other is false; so the biconditional $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true.

NDs **Quick Reference**

R (*reiteration*)

a.	$\mathcal{P}$	
	$\mathcal{P}$	a R

$\sim$I (*negation intro*)

a.	$\mid\ \mathcal{P}$	A (c, $\sim$I)
b.	$\mid\ \mathcal{Q} \wedge \sim\mathcal{Q}$ ($\bot$)	
	$\sim\mathcal{P}$	a-b $\sim$I

$\sim$E (*negation exploit*)

a.	$\mid\ \sim\mathcal{P}$	A (c, $\sim$E)
b.	$\mid\ \mathcal{Q} \wedge \sim\mathcal{Q}$ ($\bot$)	
	$\mathcal{P}$	a-b $\sim$E

$\wedge$I (*conjunction intro*)

a.	$\mathcal{P}$	
b.	$\mathcal{Q}$	
	$\mathcal{P} \wedge \mathcal{Q}$	a,b $\wedge$I

$\wedge$E (*conjunction exploit*)

a.	$\mathcal{P} \wedge \mathcal{Q}$	
	$\mathcal{P}$	a $\wedge$E

$\wedge$E (*conjunction exploit*)

a.	$\mathcal{P} \wedge \mathcal{Q}$	
	$\mathcal{Q}$	a $\wedge$E

$\vee$I (*disjunction intro*)

a.	$\mathcal{P}$	
	$\mathcal{P} \vee \mathcal{Q}$	a $\vee$I

$\vee$I (*disjunction intro*)

a.	$\mathcal{P}$	
	$\mathcal{Q} \vee \mathcal{P}$	a $\vee$I

$\vee$E (*disjunction exploit*)

a.	$\mathcal{P} \vee \mathcal{Q}$	
b.	$\mid\ \mathcal{P}$	A (g, a$\vee$E)
c.	$\mid\ \mathcal{C}$	
d.	$\mid\ \mathcal{Q}$	A (g, a$\vee$E)
e.	$\mid\ \mathcal{C}$	
	$\mathcal{C}$	a,b-c,d-e $\vee$E

$\to$I (*conditional intro*)

a.	$\mid\ \mathcal{P}$	A (g, $\to$I)
b.	$\mid\ \mathcal{Q}$	
	$\mathcal{P} \to \mathcal{Q}$	a-b $\to$I

$\to$E (*conditional exploit*)

a.	$\mathcal{P} \to \mathcal{Q}$	
b.	$\mathcal{P}$	
	$\mathcal{Q}$	a,b $\to$E

$\leftrightarrow$I (*biconditional intro*)

a.	$\mid\ \mathcal{P}$	A (g, $\leftrightarrow$I)
b.	$\mid\ \mathcal{Q}$	
c.	$\mid\ \mathcal{Q}$	A (g, $\leftrightarrow$I)
d.	$\mid\ \mathcal{P}$	
	$\mathcal{P} \leftrightarrow \mathcal{Q}$	a-b,c-d $\leftrightarrow$I

$\leftrightarrow$E (*biconditional exploit*)

a.	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b.	$\mathcal{P}$	
	$\mathcal{Q}$	a,b $\leftrightarrow$E

$\leftrightarrow$E (*biconditional exploit*)

a.	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b.	$\mathcal{Q}$	
	$\mathcal{P}$	a,b $\leftrightarrow$E

derived rule: **$\bot$I** (*bottom intro*)

a.	$\mathcal{Q}$	
b.	$\sim\mathcal{Q}$	
	$\bot$	a,b $\bot$I

Here are a couple of examples. The first is straightforward, and exercises both the $\leftrightarrow$I and $\leftrightarrow$E rules. We show, $A \leftrightarrow B$, $B \leftrightarrow C \vdash_{NDs} A \leftrightarrow C$.

(X)

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	A	A (g, $\leftrightarrow$I)
4.	B	1,3 $\leftrightarrow$E
5.	C	2,4 $\leftrightarrow$E
6.	C	A (g, $\leftrightarrow$I)
7.	B	2,6 $\leftrightarrow$E
8.	A	1,7 $\leftrightarrow$E
9.	$A \leftrightarrow C$	3-5,6-8 $\leftrightarrow$I

Our original goal is $A \leftrightarrow C$. So it is natural to set up subderivations to get it by $\leftrightarrow$I. Once we have done this, the subderivations are easily completed by applications of $\leftrightarrow$E.

Here is an interesting case that again exercises both rules. We show, $A \leftrightarrow (B \leftrightarrow C), C \vdash_{NDs} A \leftrightarrow B$.

(Y)

1.	$A \leftrightarrow (B \leftrightarrow C)$	P
2.	C	P
3.	A	A (g, $\leftrightarrow$I)
4.	$B \leftrightarrow C$	1,3 $\leftrightarrow$E
5.	B	4,2 $\leftrightarrow$E
6.	B	A (g, $\leftrightarrow$I)
7.	B	A (g, $\leftrightarrow$I)
8.	C	2 R
9.	C	A (g, $\leftrightarrow$I)
10.	B	6 R
11.	$B \leftrightarrow C$	7-8,9-10 $\leftrightarrow$I
12.	A	1,11 $\leftrightarrow$E
13.	$A \leftrightarrow B$	3-5,6-12 $\leftrightarrow$I

We begin by setting up the subderivations to get $A \leftrightarrow B$ by $\leftrightarrow$I. The first is easily completed with a couple applications of $\leftrightarrow$E. To reach the goal for the second by means of the premise (1) we need $B \leftrightarrow C$ as our second "card." So we set up to reach *that*. As it happens, the extra subderivations at (7)–(8) and (9)–(10) are easy to complete. Again, if you have followed so far, you are doing well. We will be in a better position to *create* such derivations after our discussion of strategy.

So much for the rules of *NDs*. Before we turn in the next section to strategy, let us note a couple of features of the rules that may so-far have gone without notice.

First, premises are not always necessary for *NDs* derivations. Thus, for example, $\vdash_{NDs} A \to A$.

(Z)

	$A \to A$	(goal)

1.	A	A (g, $\to$I)
2.	A	1 R
3.	$A \to A$	1-2 $\to$I

If there are no premises, do not panic! Begin in the usual way. In this case, the original goal is $A \to A$. So we set up to obtain it by $\to$I. And the subderivation is particularly simple. Notice that our derivation of $A \to A$ corresponds to the fact from truth tables that $\vDash_s A \to A$. And we *need* to be able to derive $A \to A$ from no premises if there is to be the right sort of correspondence between derivations in *NDs* and semantic validity—if we are to have $\Gamma \vDash_s \mathcal{P}$ iff $\Gamma \vdash_{NDs} \mathcal{P}$.

Second, observe again that every subderivation comes with an exit strategy. The exit strategy says whether you intend to complete the subderivation with a particular goal or by obtaining a contradiction, and then how the subderivation is to be used once complete. There are just five rules which appeal to a subderivation: $\to$I, $\sim$I, $\sim$E, $\vee$E, and $\leftrightarrow$I. You will complete the subderivation, and then use it by one of these rules. So these are the *only* rules which may appear in an exit strategy. If you do not understand this, then you need to go back and think about the rules until you do.

Finally, it is worth noting a strange sort of case, with application to rules that can take more than one input of the same type. Consider a simple demonstration that $A \vdash_{NDs} A \wedge A$. We might proceed as in (AA) on the left,

(AA)

1.	A	P
2.	A	1 R
3.	$A \wedge A$	1,2 $\wedge$I

(AB)

1.	A	P
2.	$A \wedge A$	1,1 $\wedge$I

We begin with A, reiterate so that A appears on different lines, and apply $\wedge$I. But we might have proceeded as in (AB) on the right. The rule requires an accessible line on which the left conjunct appears—which we have at (1)—and an accessible line on which the right conjunct appears *which we also have* on (1). So the rule takes an input for the left conjunct and an input for the right—they just happen to be the same thing. A similar point applies to rules $\vee$E and $\leftrightarrow$I which take more than one subderivation as input. Suppose we want to show $A \vee A \vdash_{NDs} A$.[5]

(AC)

1.	$A \vee A$	P
2.	A	A (g, 1$\vee$E)
3.	A	2 R
4.	A	A (g, 1$\vee$E)
5.	A	4 R
6.	A	1,2-3,4-5 $\vee$E

(AD)

1.	$A \vee A$	P
2.	A	A (g, 1$\vee$E)
3.	A	2 R
4.	A	1,2-3,2-3 $\vee$E

[5] I am reminded of a character in *Groundhog Day* (film, 1993) who repeatedly asks, "Am I right or am I right?" If he is right or he is right, it follows that he is right.

In (AC), we begin in the usual way to get the main goal by $\vee$E. This leads to the subderivations (2)–(3) and (4)–(5), the first moving from the left disjunct to the goal, and the second from the right disjunct to the goal. But the left and right disjuncts are the same. So we might have simplified as in (AD). $\vee$E still requires three inputs: First an accessible disjunction, which we find on (1); second an accessible subderivation which moves from the left disjunct to the goal, which we find on (2)–(3); third a subderivation which moves from the right disjunct to the goal—but we have this on (2)–(3). So the justification at (4) of (AD) appeals to the three relevant facts, by appeal to the same subderivation twice. Similarly one could imagine a quick-and-dirty demonstration that $\vdash_{NDs} A \leftrightarrow A$.

E6.11. Complete the following derivations by filling in justifications for each line.

a.

$$\begin{array}{rl}
1. & \mid \underline{A \leftrightarrow B} \\
2. & \mid \quad \mid \underline{A} \\
3. & \mid \quad \mid B \\
4. & \mid A \to B
\end{array}$$

b.

$$\begin{array}{rl}
1. & \mid A \leftrightarrow B \\
2. & \mid \underline{\sim B} \\
3. & \mid \quad \mid \underline{A} \\
4. & \mid \quad \mid B \\
5. & \mid \quad \mid \bot \\
6. & \mid \sim A
\end{array}$$

c.

$$\begin{array}{rl}
1. & \mid \quad \mid \underline{A \leftrightarrow \sim A} \\
2. & \mid \quad \mid \quad \mid \underline{A} \\
3. & \mid \quad \mid \quad \mid \sim A \\
4. & \mid \quad \mid \quad \mid \bot \\
5. & \mid \quad \mid \sim A \\
6. & \mid \quad \mid A \\
7. & \mid \quad \mid \bot \\
8. & \mid \sim(A \leftrightarrow \sim A)
\end{array}$$

d.
1. | | A
2. | | | $\sim A$
3. | | | A
4. | | $\sim A \rightarrow A$
5. | | $\sim A \rightarrow A$
6. | | | $\sim A$
7. | | | A
8. | | | $\bot$
9. | | A
10. | $A \leftrightarrow (\sim A \rightarrow A)$

e.
1. | $\sim A$
2. | $\sim B$
3. | | A
4. | | | $\sim B$
5. | | | $\bot$
6. | | B
7. | | B
8. | | | $\sim A$
9. | | | $\bot$
10. | | A
11. | $A \leftrightarrow B$

E6.12. The following are not legitimate *NDs* derivations. In each case, explain why.

a.
1. | A P
2. | B P
3. | $A \leftrightarrow B$ 1,2 $\leftrightarrow$I

b.
1. | $A \rightarrow B$ P
2. | B P
3. | A 1,2 $\rightarrow$E

*c.
1. | $A \leftrightarrow B$ P
2. | A 1 $\leftrightarrow$E

d.

1.	B	P
2.	$\mid A$	A (g, $\leftrightarrow$I)
3.	$\mid B$	1 R
4.	$\mid B$	A (g, $\leftrightarrow$I)
5.	$\mid A$	2 R
6.	$A \leftrightarrow B$	2-3,4-5 $\leftrightarrow$I

e.

1.	$\sim A$	P
2.	$\mid B$	A (g, $\rightarrow$I)
3.	$\mid \mid \sim A$	A (g, $\leftrightarrow$I)
4.	$\mid \mid B$	2 R
5.	$\mid B$	2 R
6.	$B \rightarrow B$	2-5 $\rightarrow$I
7.	$\mid B$	A (g, $\leftrightarrow$I)
8.	$\mid \sim A$	1 R
9.	$\sim A \leftrightarrow B$	3-4,7-8 $\leftrightarrow$I

E6.13. Produce derivations to show each of the following.

*a. $(A \wedge B) \leftrightarrow A \vdash_{NDs} A \rightarrow B$

b. $A \leftrightarrow (A \vee B) \vdash_{NDs} B \rightarrow A$

c. $A \leftrightarrow B, B \leftrightarrow C, C \leftrightarrow D, \sim A \vdash_{NDs} \sim D$

d. $A \leftrightarrow B \vdash_{NDs} (A \rightarrow B) \wedge (B \rightarrow A)$

*e. $A \leftrightarrow (B \wedge C), B \vdash_{NDs} A \leftrightarrow C$

f. $(A \rightarrow B) \wedge (B \rightarrow A) \vdash_{NDs} A \leftrightarrow B$

g. $A \rightarrow (B \leftrightarrow C) \vdash_{NDs} (A \wedge B) \leftrightarrow (A \wedge C)$

h. $A \leftrightarrow B, C \leftrightarrow D \vdash_{NDs} (A \wedge C) \leftrightarrow (B \wedge D)$

i. $\vdash_{NDs} A \leftrightarrow A$

j. $\vdash_{NDs} (A \wedge B) \leftrightarrow (B \wedge A)$

*k. $\vdash_{NDs} \sim\sim A \leftrightarrow A$

l. $\vdash_{NDs} (A \leftrightarrow B) \rightarrow (B \leftrightarrow A)$

m. $(A \wedge B) \leftrightarrow (A \wedge C) \vdash_{NDs} A \rightarrow (B \leftrightarrow C)$

n. $\sim A \rightarrow B, A \rightarrow \sim B \vdash_{NDs} \sim A \leftrightarrow B$

o. $A, B \vdash_{NDs} \sim A \leftrightarrow \sim B$

6.2.4 Strategy

It is natural to introduce derivation rules, as we have, with relatively simple cases. And you may or may not have been able to see from the start in some cases how derivations would go. But derivations are not always simple, and it is beyond human power always to see how they go. Perhaps this has already been an issue! However, as with chess or other games of strategy, it is possible to say a good deal about how to approach problems effectively. We have said quite a bit already. In this section, we pull together some of the themes and present the material more systematically.

In doing derivations there are two fundamentally different contexts. In the one case, you have some accessible lines, and want a definite goal sentence. In the other, there are some accessible lines, and you want a contradiction.

a.	$\mathcal{A}$		a.	$\mathcal{A}$	
b.	$\mathcal{B}$		b.	$\mathcal{B}$	
	$\mathcal{G}$	(goal sentence)		$\bot$	(contradiction)

The different contexts motivate separate *strategies for a goal* and *strategies for a contradiction*. In the first case, strategies for a goal help *reach* a known goal formula. But in the other case you want some $\mathcal{Q}$ and $\sim\mathcal{Q}$, where it may not be clear what this $\mathcal{Q}$ should be; thus strategies for a contradiction help *find* the formula you need. First, strategies for a goal.

Strategies for a Goal

For natural derivation systems, the overriding strategy is to *work goal directedly*. What you do at any stage is directed primarily, not by what you have, but by where you want to be. Suppose you are trying to show that $\Gamma \vdash_{NDs} \mathcal{P}$. You are given $\mathcal{P}$ as your goal. Perhaps it is tempting to begin by using E-rules to "see what you can get" from the members of Γ. There is nothing wrong with a bit of this in order to simplify your premises (like arranging the cards in your hand into some manageable order), but the main work of doing a derivation does not begin until you focus on the goal. This is not to say that your premises play no role in strategic thinking. Rather, it is to rule out doing things with them which are not purposefully directed at the end. In the ordinary case, applying the strategies for your goal dictates some new goal; applying strategies for this new goal dictates another; and so forth, until you come to a goal that is easily achieved.

The following *strategies for a goal* are arranged in rough priority order:

SG
1. If accessible lines contain explicit contradiction, use $\sim$E to reach goal.
2. Given an accessible formula with main operator $\vee$, use $\vee$E to reach goal.
3. If goal is "in" accessible lines (set goals and) attempt to exploit it out.

4. To reach goal with main operator $\star$, use $\star$I (careful with $\vee$).
5. Try $\sim$E (especially for atomics and sentences with $\vee$ as main operator).

If a high priority strategy applies, use it. If one does not apply, simply "fall through" to the next. The priority order is not necessarily a frequency order. The frequency will likely be something like SG4, SG3, SG5, SG2, SG1. But high priority strategies are such that you should adopt them if they are available—even though most often you will fall through to ones that are more frequently used. I take up the strategies in the priority order.

SG1. *If accessible lines contain explicit contradiction, use* $\sim$E *to reach goal*. For goal $\mathcal{B}$, with an explicit contradiction accessible, you can simply *assume* $\sim\mathcal{B}$, use your contradiction, and conclude $\mathcal{B}$.

given

a.	$\mathcal{A}$	
b.	$\sim\mathcal{A}$	
	$\mathcal{B}$	(goal)

use

a.	$\mathcal{A}$	
b.	$\sim\mathcal{A}$	
c.	$\quad\sim\mathcal{B}$	A (c, $\sim$E)
d.	$\quad\bot$	a,b $\bot$I
	$\mathcal{B}$	c-d $\sim$E

That is it! No matter what your goal is, given an accessible contradiction, you can reach that goal by $\sim$E. Since this strategy always delivers, you should jump on it whenever it is available. As an example, try to show, $A, \sim A \vdash_{NDs} (R \wedge S) \rightarrow T$. Your derivation need not involve $\rightarrow$I. (This section will be most valuable if you do work these examples, and so think through the steps.) Here it is in two stages:

(AE)

1.	A	P
2.	$\sim A$	P
3.	$\quad\sim[(R \vee S) \rightarrow T]$	A (c, $\sim$E)
	$(R \vee S) \rightarrow T$	3-_ $\sim$E

1.	A	P
2.	$\sim A$	P
3.	$\quad\sim[(R \vee S) \rightarrow T]$	A (c, $\sim$E)
4.	$\quad\bot$	1,2 $\bot$I
5.	$(R \vee S) \rightarrow T$	3-4 $\sim$E

As soon as we see the accessible contradiction, we assume the negation of our goal, with a plan to exit by $\sim$E. This is accomplished on the left. Then it is a simple matter of applying the contradiction, and going to the conclusion by $\sim$E.

For this strategy, it is not required that accessible lines "contain" a contradiction only when you already have $\mathcal{Q}$ and $\sim\mathcal{Q}$ for $\bot$I. However, the intent is that there should be some straightforward way to obtain them from accessible lines. If you can do this, then your derivation is over: assume the opposite, extract the contradiction, and apply $\sim$E to reach the goal. If there is no simple way to obtain a contradiction, fall through to the next strategy.

SG2. *Given an accessible formula with main operator* $\vee$*, use* $\vee$*E to reach goal*. As suggested above, you may prefer to avoid $\vee$E. But this is a mistake—$\vee$E is your friend! Suppose you have some accessible lines including a disjunction $\mathcal{A} \vee \mathcal{B}$ with goal $\mathcal{C}$. If you go for *that very goal* by $\vee$E, the result is a pair of subderivations with goal $\mathcal{C}$—where, in the one case, all those very same accessible lines *and* $\mathcal{A}$ are accessible, and in the other case, all those very same lines *and* $\mathcal{B}$ are accessible. So, in each subderivation, you can only be better off in your attempt to reach $\mathcal{C}$.

given

a.	$\mathcal{A} \vee \mathcal{B}$	
	$\mathcal{C}$	(goal)

use

a.	$\mathcal{A} \vee \mathcal{B}$	
b.	$\mid\ \mathcal{A}$	A (g, a$\vee$E)
	$\mid$	
c.	$\mid\ \mathcal{C}$	(goal)
d.	$\mid\ \mathcal{B}$	A (g, a$\vee$E)
	$\mid$	
e.	$\mid\ \mathcal{C}$	(goal)
	$\mathcal{C}$	a,b-c,d-e $\vee$E

As an example, try to show, $A \to B,\ A \vee (A \wedge B) \vdash_{NDs} A \wedge B$. Try showing it without $\vee$E! Here is the derivation in two stages:

(AF)

1.	$A \to B$	P
2.	$A \vee (A \wedge B)$	P
3.	$\mid\ A$	A (g, 2$\vee$E)
	$\mid$	
	$\mid\ A \wedge B$	
	$\mid\ A \wedge B$	A (g, 2$\vee$E)
	$\mid\ A \wedge B$	
	$A \wedge B$	2,3-_,_ $\vee$E

1.	$A \to B$	P
2.	$A \vee (A \wedge B)$	P
3.	$\mid\ A$	A (g, 2$\vee$E)
4.	$\mid\ B$	1,3 $\to$E
5.	$\mid\ A \wedge B$	3,4 $\wedge$I
6.	$\mid\ A \wedge B$	A (g, 2$\vee$E)
7.	$\mid\ A \wedge B$	6 R
8.	$A \wedge B$	2,3-5,6-7 $\vee$E

When we start, there is no accessible contradiction. So we fall through to SG2. Since a premise has main operator $\vee$, we set up to get the goal by $\vee$E. This leads to a pair of simple subderivations. Once we do this, we treat the disjunction as effectively "used up" so that SG2 does not apply to it again. Notice that there is almost nothing one *could* do except set up this way—and that once you do, it is easy!

SG3. *If goal is "in" accessible lines (set goals and) attempt to exploit it out*. In most derivations, you will work toward goals which are successively closer to what can be obtained directly from accessible lines. And you finally come to a goal which can be obtained directly. If it can be obtained directly, do so! In some cases, however, you will come to a stage where your goal exists in accessible lines but can be obtained only by means of some other result. In this case, you can set that other result as a *new* goal. A typical case is as follows:

given	a.	$\mathcal{A} \to \mathcal{B}$		*use*	a.	$\mathcal{A} \to \mathcal{B}$	
					b.	$\mathcal{A}$	(goal)
		$\mathcal{B}$	(goal)			$\mathcal{B}$	a,b $\to$E

The $\mathcal{B}$ exists in the premises. You cannot get it without the $\mathcal{A}$. So you set $\mathcal{A}$ as a new goal and use it to get the $\mathcal{B}$. This strategy applies whenever the complete goal exists in accessible lines, and can be obtained by reiteration, by an E-rule, or by an E-rule with some new goal. Observe that the strategy would not apply in case you have $A \to B$ and are going for A. Then the goal exists as part of a premise all right. But there is no obvious result such that obtaining it would give you a way to exploit $A \to B$ to get the A.

As an example, let us try to show $(A \to B) \wedge (B \to C)$, $A \leftrightarrow (L \leftrightarrow S)$, $(L \leftrightarrow S) \wedge H \vdash_{NDs} C$. Here is the derivation in four stages:

(AG)	1.	$(A \to B) \wedge (B \to C)$	P	1.	$(A \to B) \wedge (B \to C)$	P
	2.	$A \leftrightarrow (L \leftrightarrow S)$	P	2.	$A \leftrightarrow (L \leftrightarrow S)$	P
	3.	$(L \leftrightarrow S) \wedge H$	P	3.	$(L \leftrightarrow S) \wedge H$	P
	4.	$B \to C$	1 $\wedge$E	4.	$B \to C$	1 $\wedge$E
				5.	$A \to B$	1 $\wedge$E
					A	
		B			B	5,_ $\to$E
		C	4,_ $\to$E		C	4,_ $\to$E

The original goal C exists in the premises, as the consequent of the right conjunct of (1). It is easy to isolate the $B \to C$, but this leaves us with the B as a new goal to get the C. B also exists in the premises, as the consequent of the left conjunct of (1). Again, it is easy to isolate $A \to B$, but this leaves us with A as a new goal.

1.	$(A \to B) \wedge (B \to C)$	P	1.	$(A \to B) \wedge (B \to C)$	P
2.	$A \leftrightarrow (L \leftrightarrow S)$	P	2.	$A \leftrightarrow (L \leftrightarrow S)$	P
3.	$(L \leftrightarrow S) \wedge H$	P	3.	$(L \leftrightarrow S) \wedge H$	P
4.	$B \to C$	1 $\wedge$E	4.	$B \to C$	1 $\wedge$E
5.	$A \to B$	1 $\wedge$E	5.	$A \to B$	1 $\wedge$E
	$L \leftrightarrow S$		6.	$L \leftrightarrow S$	3 $\wedge$E
	A	2,_ $\leftrightarrow$E	7.	A	2,6 $\leftrightarrow$E
	B	5,_ $\to$E	8.	B	5,7 $\to$E
	C	4,_ $\to$E	9.	C	4,8 $\to$E

But A also exists in the premises, at the left side of (2); to get it, we set $L \leftrightarrow S$ as a goal. But $L \leftrightarrow S$ exists in the premises, and is easy to get by $\wedge$E. So we complete the derivation with the steps that motivated the subgoals in the first place. Observe the way we move from one goal to the next, until finally there is a stage where SG3 applies in its simplest form, so that $L \leftrightarrow S$ is obtained directly. Another example of this strategy is derivation (Y) above where we needed A to complete the second subderivation and so set $B \leftrightarrow C$ as goal.

SG4. *To reach goal with main operator* $\star$*, use* $\star$*I* (careful with $\vee$). This is the most frequently used strategy, the one most likely to structure your derivation as a whole. $\sim$E to the side, the basic structure of I-rules and E-rules in *NDs* gives you just one way to generate a formula with main operator $\star$, whatever that may be. In the ordinary case, then, you can *expect* to obtain a formula with main operator $\star$ by the corresponding I-rule. Thus, for a typical example,

given

	$\mathscr{A} \to \mathscr{B}$	(goal)

use

a.	$\mathscr{A}$	A (g, $\to$I)
b.	$\mathscr{B}$	(goal)
	$\mathscr{A} \to \mathscr{B}$	a-b $\to$I

And this is not the only context where SG4 applies. It makes sense to consider it for formulas with any main operator. Be cautious, however, for formulas with main operator $\vee$. There are cases where it is possible to prove a disjunction, but not to prove it by $\vee$I—as one might have conclusive reason to believe the butler *or* the maid did it, without conclusive reason to believe the butler did it, or conclusive reason to believe the maid did it (perhaps the butler and maid were the only ones with means and motive). You should consider the strategy for $\vee$. But it does not always work.

As an example, let us show $D \vdash_{NDs} A \to (B \to (C \to D))$. Here is the derivation in four stages:

(AH)

1.	D	P
2.	A	A (g, $\to$I)
	$B \to (C \to D)$	
	$A \to (B \to (C \to D))$	2-_ $\to$I

1.	D	P
2.	A	A (g, $\to$I)
3.	B	A (g, $\to$I)
	$C \to D$	
	$B \to (C \to D)$	3-_ $\to$I
	$A \to (B \to (C \to D))$	2-_ $\to$I

Initially, there is no contradiction or disjunction in the premises, and neither do we see the goal. So we fall through to strategy SG4 and, since the main operator of the goal is $\to$, set up to get it by $\to$I. This gives us $B \to (C \to D)$ as a new goal. Since this has main operator $\to$, and it remains that other strategies do not apply, we fall through to SG4, and set up to get it by $\to$I. This gives us $C \to D$ as a new goal.

1.	D	P
2.	$\quad A$	A (g, $\to$I)
3.	$\quad\quad B$	A (g, $\to$I)
4.	$\quad\quad\quad C$	A (g, $\to$I)
	$\quad\quad\quad D$	
	$\quad\quad C \to D$	4-_ $\to$I
	$\quad B \to (C \to D)$	3-_ $\to$I
	$A \to (B \to (C \to D))$	2-_ $\to$I

1.	D	P
2.	$\quad A$	A (g, $\to$I)
3.	$\quad\quad B$	A (g, $\to$I)
4.	$\quad\quad\quad C$	A g, $\to$I)
5.	$\quad\quad\quad D$	1 R
6.	$\quad\quad C \to D$	4-5 $\to$I
7.	$\quad B \to (C \to D)$	3-6 $\to$I
8.	$A \to (B \to (C \to D))$	2-7 $\to$I

As before, with $C \to D$ as the goal, there is no contradiction on accessible lines, no accessible formula has main operator $\vee$, and the goal does not itself appear on accessible lines. Since the main operator is $\to$, we set up again to get it by $\to$I. This gives us D as a new subgoal. But D does exist on an accessible line. Thus we are faced with a particularly simple instance of strategy SG3. To complete the derivation, we simply reiterate D from (1), and follow our exit strategies as planned.

SG5. *Try $\sim$E* (especially for atomics and sentences with $\vee$ as main operator). The previous strategy has no application to atomics, because they *have* no main operator, and we have suggested that it is problematic for disjunctions. This last strategy applies particularly in those cases. So it is applicable in cases where other strategies seem not to apply.

given

$\mathcal{A}$	(goal)

use

a.	$\quad \sim\mathcal{A}$	A (c, $\sim$E)
b.	$\quad \bot$	
	$\mathcal{A}$	a-b $\sim$E

It is possible to obtain *any* formula by $\sim$E, by assuming its negation and going for a contradiction. So this strategy is generally applicable. It cannot hurt: If you could have reached goal $\mathcal{A}$ anyway, you can still obtain $\mathcal{A}$ under the assumed $\sim\mathcal{A}$ and use the resultant contradiction to reach $\mathcal{A}$ outside of the subderivation. And it may help: As for $\vee$E, all the lines from before *plus* the new assumption are accessible; in many cases, the assumption puts you in a position to make progress you would not have been able to make before.

As a simple example of the strategy, try showing $\sim A \to B, \sim B \vdash_{NDs} A$. Here is the derivation in two stages:

(AI)

1.	$\sim A \to B$	P
2.	$\sim B$	P
3.	$\quad \sim A$	A (c, $\sim$E)
	$\quad \bot$	
	A	3-_ $\sim$E

1.	$\sim A \to B$	P
2.	$\sim B$	P
3.	$\quad \sim A$	A (c, $\sim$E)
4.	$\quad B$	1,3 $\to$E
5.	$\quad \bot$	4,2 $\bot$I
6.	A	3-5 $\sim$E

There is no contradiction in the premises, no formula has main operator $\vee$ and, though $\sim A$ is the antecedent of (1), there is no obvious way to exploit the premise to isolate the A. The goal A has no operators, so it has no main operator and strategy SG4 does not apply. So we fall through to strategy SG5, and set up to get the goal by $\sim$E. In this case, the subderivation is particularly easy to complete.

Sometimes the occasion between this strategy and SG1 can seem obscure (and, in the end, it may not be all that important to separate them). However, for the first strategy, accessible lines *by themselves* are sufficient for a contradiction and so motivate the assumption. In this example, from the premises we have $\sim B$, but cannot get the B and so do not have a contradiction from the premises alone. So SG1 does not apply. For SG5, in contrast to SG1, the contradiction becomes available only after you make the assumption.

Here is an extended example which combines a number of the strategies considered so far. We show that $B \vee A \vdash_{NDs} \sim A \rightarrow B$. You want especially to absorb the strategy-based *mode of thinking* as a way to approach exercises.

(AJ)

$$\begin{array}{rl@{\quad}l}
1. & \mid\ B \vee A & \text{P} \\
 & \mid\!\underline{\quad} & \\
 & \mid & \\
 & \mid\ \sim A \rightarrow B &
\end{array}$$

There is no contradiction in the premise; so strategy SG1 is inapplicable. Strategy SG2 tells us to go for the goal by $\vee$E. Another option is to fall through to SG4 and go for $\sim A \rightarrow B$ by $\rightarrow$I and then apply $\vee$E to get the B, but $\rightarrow$I has lower priority and let us follow the official procedure.

$$\begin{array}{rl@{\quad}l}
1. & \mid\ B \vee A & \text{P} \\
 & \mid\!\underline{\quad} & \\
2. & \mid\ \mid\ B & \text{A } (g, 1\vee\text{E}) \\
 & \mid\ \mid\!\underline{\quad} & \\
 & \mid\ \mid & \\
 & \mid\ \mid\ \sim A \rightarrow B & \\
 & \mid\ \mid\ A & \text{A } (g, 1\vee\text{E}) \\
 & \mid\ \mid\!\underline{\quad} & \\
 & \mid\ \mid & \\
 & \mid\ \mid\ \sim A \rightarrow B & \\
 & \mid\ \sim A \rightarrow B & 1,2\text{-}_,_\ \vee\text{E}
\end{array}$$

Given an accessible line with main operator $\vee$, use $\vee$E to reach goal.

Having set up for $\vee$E on line (1), we treat $B \vee A$ as effectively "used up" and so out of the picture. Concentrating, for the moment, on the first subderivation, there is no contradiction on accessible lines; neither is there another accessible disjunction; and the goal is not in accessible lines. So we fall through to SG4.

1.	$B \vee A$	P	
2.	$\quad B$	A (g, 1$\vee$E)	
3.	$\quad\quad \sim A$	A (g, $\to$I)	
	$\quad\quad B$		
	$\quad \sim A \to B$	3-_ $\to$I	To reach goal with main operator $\to$, use $\to$I.
	$\quad A$	A (g, 1$\vee$E)	
	$\quad \sim A \to B$		
	$\sim A \to B$	1,2-_,_ $\vee$E	

In this case, the subderivation is easy to complete. The new goal, B exists as such on an accessible line. So we are faced with a simple instance of SG3, and so can complete the subderivation.

1.	$B \vee A$	P	
2.	$\quad B$	A (g, 1$\vee$E)	
3.	$\quad\quad \sim A$	A (g, $\to$I)	
4.	$\quad\quad B$	2 R	The first subderivation is completed by reiterating B from line (2), and following the exit strategy.
5.	$\quad \sim A \to B$	3-4 $\to$I	
6.	$\quad A$	A (g, 1$\vee$E)	
	$\quad \sim A \to B$		
	$\sim A \to B$	1,2-5,6-_ $\vee$E	

For the second main subderivation lines (2)–(5) are inaccessible. Tick off in your head: there is no accessible contradiction; neither is there another accessible formula with main operator $\vee$; and the goal is not in accessible lines. So we fall through to strategy SG4.

1.	$B \vee A$	P	
2.	$\quad B$	A (g, 1$\vee$E)	
3.	$\quad\quad \sim A$	A (g, $\to$I)	
4.	$\quad\quad B$	2 R	
5.	$\quad \sim A \to B$	3-4 $\to$I	
6.	$\quad A$	A (g, 1$\vee$E)	To reach goal with main operator $\to$, use $\to$I.
7.	$\quad\quad \sim A$	A (g, $\to$I)	
	$\quad\quad B$		
	$\quad \sim A \to B$	7-_ $\to$I	
	$\sim A \to B$	1,2-5,6-_ $\vee$E	

But this time there *is* an accessible contradiction at (6) and (7). So SG1 applies, and we are in a position to complete the derivation as follows:

1.	$\mid B \vee A$	P
2.	$\mid\ \mid B$	A (g, 1$\vee$E)
3.	$\mid\ \mid\ \mid \sim A$	A (g, $\rightarrow$I)
4.	$\mid\ \mid\ \mid B$	2 R
5.	$\mid\ \mid \sim A \rightarrow B$	3-4 $\rightarrow$I
6.	$\mid\ \mid A$	A (g, 1$\vee$E)
7.	$\mid\ \mid\ \mid \sim A$	A (g, $\rightarrow$I)
8.	$\mid\ \mid\ \mid\ \mid \sim B$	A (c, $\sim$E)
9.	$\mid\ \mid\ \mid\ \mid \bot$	6,7 $\bot$I
10.	$\mid\ \mid\ \mid B$	8-9 $\sim$E
11.	$\mid\ \mid \sim A \rightarrow B$	7-10 $\rightarrow$I
12.	$\mid \sim A \rightarrow B$	1,2-5,6-11 $\vee$E

If accessible lines contain explicit contradiction, use $\sim$E to reach goal.

This derivation is fairly complicated! But we did not need to see how the whole thing would go from the start. Indeed, it is hard to see how one could do so. Rather it was enough to see, at each stage, what to do next. That is the beauty of our goal-oriented approach.

A brief remark before we turn to exercises: In going for a contradiction, as from SG4 or SG5, the new goal is not a definite formula—any contradiction is sufficient for the rule and for a derivation of $\bot$. But each of our strategies for a goal presupposes a known goal sentence. In going for a contradiction there is no definite goal formula—so this presupposition is not met, and strategies for a goal do not apply. This motivates the "strategies for a contradiction" of the next section. For now, I will say just this: If there is a contradiction to be had, and you can reduce formulas on accessible lines to atomics and negated atomics, the contradiction *will* appear at that level. So one way to go for a contradiction is simply by applying E-rules to accessible lines, to generate what atomics and negated atomics you can.

Proofs for the following theorems are left as exercises. You should not start them now, but wait for the assignment in E6.16. The first three may remind you of axioms from Chapter 3 and the fourth has an application in Part IV. The others foreshadow rules from the system *NDs+*, which we will see shortly.

T6.1. $\vdash_{NDs} \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

T6.2. $\vdash_{NDs} (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

*T6.3. $\vdash_{NDs} (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$

T6.4. $\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}), \mathcal{D} \rightarrow (\mathcal{C} \rightarrow \mathcal{E}), \mathcal{D} \rightarrow \mathcal{B} \vdash_{NDs} \mathcal{A} \rightarrow (\mathcal{D} \rightarrow \mathcal{E})$

T6.5. $\mathcal{A} \rightarrow \mathcal{B}, \sim\mathcal{B} \vdash_{NDs} \sim\mathcal{A}$

T6.6. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{NDs} \mathcal{A} \rightarrow \mathcal{C}$

T6.7. $\mathcal{A} \vee \mathcal{B}, \sim\mathcal{A} \vdash_{NDs} \mathcal{B}$

T6.8. $\mathcal{A} \vee \mathcal{B}, \sim\mathcal{B} \vdash_{NDs} \mathcal{A}$

T6.9. $\mathcal{A} \leftrightarrow \mathcal{B}, \sim\mathcal{A} \vdash_{NDs} \sim\mathcal{B}$

T6.10. $\mathcal{A} \leftrightarrow \mathcal{B}, \sim\mathcal{B} \vdash_{NDs} \sim\mathcal{A}$

T6.11. $\vdash_{NDs} (\mathcal{A} \wedge \mathcal{B}) \leftrightarrow (\mathcal{B} \wedge \mathcal{A})$

T6.12. $\vdash_{NDs} (\mathcal{A} \leftrightarrow \mathcal{B}) \leftrightarrow (\mathcal{B} \leftrightarrow \mathcal{A})$

*T6.13. $\vdash_{NDs} (\mathcal{A} \vee \mathcal{B}) \leftrightarrow (\mathcal{B} \vee \mathcal{A})$

T6.14. $\vdash_{NDs} (\mathcal{A} \rightarrow \mathcal{B}) \leftrightarrow (\sim\mathcal{B} \rightarrow \sim\mathcal{A})$

T6.15. $\vdash_{NDs} [\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})] \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C}]$

T6.16. $\vdash_{NDs} [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})] \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}]$

T6.17. $\vdash_{NDs} \mathcal{A} \leftrightarrow \sim\sim\mathcal{A}$

T6.18. $\vdash_{NDs} \mathcal{A} \leftrightarrow (\mathcal{A} \wedge \mathcal{A})$

T6.19. $\vdash_{NDs} \mathcal{A} \leftrightarrow (\mathcal{A} \vee \mathcal{A})$

T6.20. $\vdash_{NDs} [\mathcal{A} \vee (\mathcal{B} \vee \mathcal{C})] \leftrightarrow [(\mathcal{A} \vee \mathcal{B}) \vee \mathcal{C}]$

E6.14. For each of the following, (i) which goal strategy applies? and (ii) what is the next step? If the strategy calls for a new subgoal, show the subgoal; if it calls for a subderivation, set up the subderivation. In each case, *explain* your response. Hint: Each goal strategy applies once.

*a.
$$\begin{array}{rl|l} 1. & \sim A \vee B & \text{P} \\ 2. & A & \text{P} \\ \hline & & \\ & B & \end{array}$$

b.
$$\begin{array}{rl|l} 1. & J \wedge S & \text{P} \\ 2. & S \rightarrow K & \text{P} \\ \hline & & \\ & K & \end{array}$$

*c.
$$\begin{array}{rl|l} 1. & \sim A \leftrightarrow B & \text{P} \\ \hline & & \\ & B \leftrightarrow \sim A & \end{array}$$

d.
$$\begin{array}{rl|l} 1. & A \leftrightarrow \sim B & \text{P} \\ 2. & \sim A & \text{P} \\ \hline & & \\ & B & \end{array}$$

e.
$$\begin{array}{rl|l} 1. & A \wedge B & \text{P} \\ 2. & \sim A & \text{P} \\ \hline & & \\ & K \vee J & \end{array}$$

E6.15. Produce derivations to show each of the following. If you get stuck, you will find strategy hints in the *Answers to Selected Exercises*.

*a. $A \leftrightarrow (A \rightarrow B) \vdash_{NDs} A \rightarrow B$

*b. $(A \vee B) \rightarrow (B \leftrightarrow D),\ B \vdash_{NDs} B \wedge D$

*c. $\sim(A \wedge C),\ \sim(A \wedge C) \leftrightarrow B \vdash_{NDs} A \vee B$

*d. $A \wedge (C \wedge \sim B),\ (A \vee D) \rightarrow \sim E \vdash_{NDs} \sim E$

*e. $A \rightarrow B,\ B \rightarrow C \vdash_{NDs} A \rightarrow C$

*f. $(A \wedge B) \rightarrow (C \wedge D) \vdash_{NDs} [(A \wedge B) \rightarrow C] \wedge [(A \wedge B) \rightarrow D]$

*g. $A \rightarrow (B \rightarrow C),\ (A \wedge D) \rightarrow E,\ C \rightarrow D \vdash_{NDs} (A \wedge B) \rightarrow E$

*h. $(A \to B) \wedge (B \to C), [(D \vee E) \vee H] \to A, \sim(D \vee E) \wedge H \vdash_{NDs} C$

*i. $A \to (B \wedge C), \sim C \vdash_{NDs} \sim(A \wedge D)$

*j. $A \to (B \to C), D \to B \vdash_{NDs} A \to (D \to C)$

*k. $A \to (B \to C) \vdash_{NDs} \sim C \to \sim(A \wedge B)$

*l. $(A \wedge \sim B) \to \sim A \vdash_{NDs} A \to B$

*m. $\sim A \vdash_{NDs} A \to B$

*n. $\sim B \leftrightarrow A, C \to B, A \wedge C \vdash_{NDs} \sim K$

*o. $\sim A \leftrightarrow \sim B \vdash_{NDs} A \leftrightarrow B$

*p. $(A \vee B) \vee C, B \leftrightarrow C \vdash_{NDs} C \vee A$

*q. $\vdash_{NDs} A \to (A \vee B)$

*r. $\vdash_{NDs} A \to (B \to A)$

*s. $\vdash_{NDs} (A \leftrightarrow B) \to (A \to B)$

*t. $\vdash_{NDs} (A \wedge \sim A) \to (B \wedge \sim B)$

*u. $\vdash_{NDs} (A \to B) \to [(C \to A) \to (C \to B)]$

*v. $\vdash_{NDs} [(A \to B) \wedge \sim B] \to \sim A$

*w. $\vdash_{NDs} A \to [B \to (A \to B)]$

*x. $\vdash_{NDs} \sim A \to [(B \wedge A) \to C]$

*y. $\vdash_{NDs} (A \to B) \to [\sim B \to \sim(A \wedge D)]$

*E6.16. Produce derivations to demonstrate each of T6.1–T6.20. These are a mix—some repetitious, some challenging. But when we need the results later, we will be glad to have done them now. Hint: Do not worry if one or two get a bit longer than you are used to—they should!

Strategies for a Contradiction

We come now to our second set of strategies. Each of our strategies for a goal presupposes a known goal sentence—the strategies for a goal say how to go about reaching *this* goal or that. In going for a contradiction, however, the $\mathscr{Q}$ and $\sim\mathscr{Q}$ may not be known. Where the goal is unknown, our strategies for a goal do not apply. This motivates *strategies for a contradiction*. Again, the strategies are in rough priority order.

SC
1. Break accessible formulas down into atomics and negated atomics.
2. Given an available disjunction, go for $\bot$ by $\vee$E.
3. Set as goal the opposite of some negation (something that cannot itself be broken down); then apply strategies for a goal to reach it.
4. For some $\mathscr{P}$ such that both $\mathscr{P}$ and $\sim\mathscr{P}$ lead to contradiction: Assume $\mathscr{P}$ ($\sim\mathscr{P}$), obtain the first contradiction, and conclude $\sim\mathscr{P}$ ($\mathscr{P}$); then obtain the second contradiction—this is the one you want.

Again, the priority order is not the frequency order. The frequency is likely to be something like SC1, SC3, SC4, SC2. Also sometimes, but not always, SC3 and SC4 coincide: in deriving the opposite of some negation, you end up assuming a $\mathscr{P}$ such that $\mathscr{P}$ and $\sim\mathscr{P}$ lead to contradiction.

SC1. *Break accessible formulas down into atomics and negated atomics.* As we have already said, if there is a contradiction to be had, and you can break accessible formulas into atomics and negated atomics, the contradiction *will* appear at that level. Thus, for example,

(AK)

1.	$A \wedge B$	P
2.	$C \to \sim B$	P
3.	| C	A (c, $\sim$I)
	| $\bot$	
	$\sim C$	3-_ $\sim$I

1.	$A \wedge B$	P
2.	$C \to \sim B$	P
3.	| C	A (c, $\sim$I)
4.	| $\sim B$	2,3 $\to$E
5.	| A	1 $\wedge$E
6.	| B	1 $\wedge$E
7.	| $\bot$	6,4 $\bot$I
8.	$\sim C$	3-7 $\sim$I

Our strategy for the main goal is SG4 with an application of $\sim$I. Then the aim is to obtain a contradiction. And our first thought is to break accessible lines down to atomics and negated atomics. Perhaps this example is too simple. And you may wonder about the point of getting A at (5)—there *is* no need for A at (5). But this merely illustrates the point: If you can get to atomics and negated atomics ("randomly" as it were) the contradiction will appear in the end.

As another example, try showing $A \wedge (B \wedge \sim C), \sim F \rightarrow D, (A \wedge D) \rightarrow C \vdash_{NDs} F$. Here is the derivation completed in two stages:

(AL)

1.	$A \wedge (B \wedge \sim C)$	P
2.	$\sim F \rightarrow D$	P
3.	$(A \wedge D) \rightarrow C$	P
4.	$\quad \sim F$	A (c, $\sim$E)
	$\quad \perp$	
	F	4-_ $\sim$E

1.	$A \wedge (B \wedge \sim C)$	P
2.	$\sim F \rightarrow D$	P
3.	$(A \wedge D) \rightarrow C$	P
4.	$\quad \sim F$	A (c, $\sim$E)
5.	$\quad D$	2,4 $\rightarrow$E
6.	$\quad A$	1 $\wedge$E
7.	$\quad A \wedge D$	6,5 $\wedge$I
8.	$\quad C$	3,7 $\rightarrow$E
9.	$\quad B \wedge \sim C$	1 $\wedge$E
10.	$\quad \sim C$	9 $\wedge$E
11.	$\quad \perp$	8,10 $\perp$I
12.	F	4-11 $\sim$E

This time, our strategy for the goal falls through to SG5. After that, again, our goal is to obtain a contradiction—and our first thought is to break accessible formulas down to atomics and negated atomics. The assumption $\sim F$ gets us D with (2). We can get A from (1), and then C with the A and D together. Then $\sim C$ follows from (1) by a couple applications of $\wedge$E. You might proceed to get the atomics in a different order, but the basic idea of any such derivation is likely to be the same.

SC2. *Given an available disjunction, go for $\perp$ by $\vee$E.* In many cases, you will have applied $\vee$E by SG2 prior to setting up for $\sim$E or $\sim$I. Then the disjunction is "used up" and unavailable for this strategy. Sometimes, however, a disjunction remains or becomes available inside a subderivation for a tilde rule. In any such case, SC2 has high priority for the same reasons as SG2: You can only be *better off* in your attempt to reach a contradiction inside the subderivations for $\vee$E than before. So the strategy says to take the $\perp$ you need for $\sim$E or $\sim$I, and go for *it* by $\vee$E.

given

a.	$\quad \mathscr{P}$	A (c, $\sim$I)
b.	$\quad \mathscr{A} \vee \mathscr{B}$	
	$\quad \perp$	
	$\sim \mathscr{P}$	a-_ $\sim$I

use

a.	$\quad \mathscr{P}$	A (c, $\sim$I)
b.	$\quad \mathscr{A} \vee \mathscr{B}$	
c.	$\quad\quad \mathscr{A}$	A (c, b$\vee$E)
d.	$\quad\quad \perp$	(goal)
e.	$\quad\quad \mathscr{B}$	A (c, b$\vee$E)
f.	$\quad\quad \perp$	(goal)
g.	$\quad \perp$	b,c-d,e-f $\vee$E
	$\sim \mathscr{P}$	a-g $\sim$I

We go for $\perp$ in each of the subderivations for $\vee$E. Since the subderivations for $\vee$E have goal $\perp$, they have exit strategy c rather than g.

Here is an example. We show $\sim A \wedge \sim B \vdash_{NDs} \sim(A \vee B)$. The derivation is in four stages.

(AM)

1.	$\sim A \wedge \sim B$	P
2.	$\vert\ A \vee B$	A (c, $\sim$I)
	$\vert$	
	$\vert\ \bot$	
	$\sim(A \vee B)$	2-_ $\sim$I

1.	$\sim A \wedge \sim B$	P
2.	$\vert\ A \vee B$	A (c, $\sim$I)
3.	$\vert\ \vert\ A$	A (c, 2$\vee$E)
	$\vert\ \vert$	
	$\vert\ \vert\ \bot$	
	$\vert\ \vert\ B$	A (c, 2$\vee$E)
	$\vert\ \vert$	
	$\vert\ \vert\ \bot$	
	$\vert\ \bot$	2,3-_,_ $\vee$E
	$\sim(A \vee B)$	2-_ $\sim$I

In this case, our strategy for the goal is SG4. We might obtain $\sim A$ and $\sim B$ from (1), but after that there are no more atomics or negated atomics to be had. However the assumption line is itself a disjunction available for $\vee$E. So SC2 applies, and we set up with $\bot$ as the goal for $\vee$E.

1.	$\sim A \wedge \sim B$	P
2.	$\vert\ A \vee B$	A (c, $\sim$I)
3.	$\vert\ \vert\ A$	A (c, 2$\vee$E)
4.	$\vert\ \vert\ \sim A$	1 $\wedge$E
5.	$\vert\ \vert\ \bot$	3,4 $\bot$I
6.	$\vert\ \vert\ B$	A (c, 2$\vee$E)
	$\vert\ \vert$	
	$\vert\ \vert\ \bot$	
	$\vert\ \bot$	2,3-5,6-_ $\vee$E
	$\sim(A \vee B)$	2-_ $\sim$I

1.	$\sim A \wedge \sim B$	P
2.	$\vert\ A \vee B$	A (c, $\sim$I)
3.	$\vert\ \vert\ A$	A (c, 2$\vee$E)
4.	$\vert\ \vert\ \sim A$	1 $\wedge$E
5.	$\vert\ \vert\ \bot$	3,4 $\bot$I
6.	$\vert\ \vert\ B$	A (c, 2$\vee$E)
7.	$\vert\ \vert\ \sim B$	1 $\wedge$E
8.	$\vert\ \vert\ \bot$	6,7 $\bot$I
9.	$\vert\ \bot$	2,3-5,6-8 $\vee$E
10.	$\sim(A \vee B)$	2-9 $\sim$I

With $\bot$ as goal, strategies for a contradiction continue to apply. The first subderivation is easily completed from atomics and negated atomics. And the second is completed the same way. Observe that it is only because of our assumptions for $\vee$E that we are able to get the contradictions at all. And we expose another advantage of our standard use of $\bot$: While $\bot$ is a particular sentence, we obtained it by A and $\sim A$ in one subderivation and B and $\sim B$ in the other. $\vee$E would not apply to subderivations concluding with different contradictions $A \wedge \sim A$ and $B \wedge \sim B$. But once we have obtained $\bot$ in each, we are in a position to exit by $\vee$E in the usual way and so to apply $\sim$I.

SC3. *Set as goal the opposite of some negation (something that cannot itself be broken down); then apply strategies for a goal to reach it.* You will find yourself using this strategy often. In the ordinary case, if accessible formulas cannot be broken into atomics and negated atomics, it is because complex forms are “sealed off” by main operator $\sim$. The tilde blocks SC1 or SC2. But you can turn this lemon to lemonade: Taking the complex $\sim\mathcal{Q}$ as one half of a contradiction, set $\mathcal{Q}$ as goal. For some complex $\mathcal{Q}$,

given

a.	$\sim\mathcal{Q}$	
b.	$\mid\ \mathcal{A}$	A (c, $\sim$I)
	$\mid\ \bot$	
	$\sim\mathcal{A}$	b-_ $\sim$I

use

a.	$\sim\mathcal{Q}$	
b.	$\mid\ \mathcal{A}$	A (c, $\sim$I)
c.	$\mid\ \mathcal{Q}$	(goal)
d.	$\mid\ \bot$	c,a $\bot$I
	$\sim\mathcal{A}$	b-d $\sim$I

We are after a contradiction. Supposing that we cannot break $\sim\mathcal{Q}$ into its parts, our efforts to apply other strategies for a contradiction are frustrated. But SC3 offers an alternative: Set $\mathcal{Q}$ itself as a new goal and use this with $\sim\mathcal{Q}$ to reach $\bot$. Then strategies for the new goal take over. If we reach the new goal, we have the contradiction we need.

As an example, try showing $B, \sim(A \rightarrow B) \vdash_{NDs} \sim A$. Here is the derivation in four stages:

(AN)

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\mid\ A$	A (c, $\sim$I)
	$\mid\ \bot$	
	$\sim A$	3-_ $\sim$I

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\mid\ A$	A (c, $\sim$I)
	$\mid\ A \rightarrow B$	
	$\mid\ \bot$	_,2 $\bot$I
	$\sim A$	3-_ $\sim$I

Our strategy for the goal is SG4; for main operator $\sim$ we set up to get the goal by $\sim$I. So we need a contradiction. In this case, there is nothing to be done by way of obtaining atomics and negated atomics, and there is no disjunction. So we fall through to strategy SC3. $\sim(A \rightarrow B)$ on (2) has main operator $\sim$, so we set $A \rightarrow B$ as a new subgoal with the idea to use it for contradiction.

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\mid\ A$	A (c, $\sim$I)
4.	$\mid\ \mid\ A$	A (g, $\rightarrow$I)
	$\mid\ \mid\ B$	
	$\mid\ A \rightarrow B$	4-_ $\rightarrow$I
	$\mid\ \bot$	_,2 $\bot$I
	$\sim A$	3-_ $\sim$I

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\mid\ A$	A (c, $\sim$I)
4.	$\mid\ \mid\ A$	A (g, $\rightarrow$I)
5.	$\mid\ \mid\ B$	1 R
6.	$\mid\ A \rightarrow B$	4-5 $\rightarrow$I
7.	$\mid\ \bot$	6,2 $\bot$I
8.	$\sim A$	3-7 $\sim$I

Since $A \to B$ is a definite subgoal, we proceed with strategies for the goal in the usual way. The main operator is $\to$ so we set up to get it by $\to$I. The subderivation is particularly easy to complete. And we finish by executing the exit strategies as planned.

SC4. *For some $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction:* Assume $\mathcal{P}$ ($\sim\mathcal{P}$), obtain the first contradiction, and conclude $\sim\mathcal{P}$ ($\mathcal{P}$); then obtain the second contradiction—this is the one you want.

given

a.	$\mathcal{A}$	A (c, $\sim$I)
	$\perp$	
	$\sim\mathcal{A}$	a-_ $\sim$I

use

a.	$\mathcal{A}$	A (c, $\sim$I)
b.	$\mathcal{P}$	A (c, $\sim$I)
c.	$\perp$	
	$\sim\mathcal{P}$	b-c $\sim$I
d.	$\perp$	
	$\sim\mathcal{A}$	a-d $\sim$I

The essential point is that both $\mathcal{P}$ and $\sim\mathcal{P}$ somehow lead to contradiction. Given this, you can assume one of them and use the first contradiction to obtain the other; and once you have obtained this other formula, the desired contradiction results from it. The intuition behind this strategy is like that for the $\vee$E rule: $\mathcal{P}$ has to be one way or the other; if both ways lead to contradiction, contradiction follows. The strategy shows how to extract that contradiction—and is often a powerful way of making progress when none seems possible by other means.

Let us try to show $A \leftrightarrow B, B \leftrightarrow C, C \leftrightarrow \sim A \vdash_{NDs} K$. Here is the derivation in four stages:

(AO)

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
	$\perp$	
	K	4-_ $\sim$E

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
5.	A	A (c, $\sim$I)
	$\perp$	
	$\sim A$	5-_ $\sim$I
	$\perp$	
	K	4-_ $\sim$E

Our strategy for the goal falls through to SG5 (or we might see it as an obscure instance of SG1). We assume the negation of the goal, and go for a contradiction. In this case, there are no atomics or negated atomics to be had, there is no disjunction, and no formula is itself a negation such that we could build up to the opposite. So we fall through to SC4. This requires a formula such that both it and its negation lead to contradiction. Finding such a formula can be difficult! However, in this case, A does the job: Given A we can use $\leftrightarrow$E to reach $\sim A$ and so contradiction; and given $\sim A$ we can use $\leftrightarrow$E to reach A and so contradiction. So, following SC4, we assume one of them to get the other.

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
5.	A	A (c, $\sim$I)
6.	B	1,5 $\leftrightarrow$E
7.	C	2,6 $\leftrightarrow$E
8.	$\sim A$	3,7 $\leftrightarrow$E
9.	$\perp$	5,8 $\perp$I
10.	$\sim A$	5-9 $\sim$I
	$\perp$	
	K	4-_ $\sim$E

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
5.	A	A (c, $\sim$I)
6.	B	1,5 $\leftrightarrow$E
7.	C	2,6 $\leftrightarrow$E
8.	$\sim A$	3,7 $\leftrightarrow$E
9.	$\perp$	5,8 $\perp$I
10.	$\sim A$	5-9 $\sim$I
11.	C	3,10 $\leftrightarrow$E
12.	B	2,11 $\leftrightarrow$E
13.	A	1,12 $\leftrightarrow$E
14.	$\perp$	13,10 $\perp$I
15.	K	4-14 $\sim$E

The first contradiction appears easily at the level of atomics and negated atomics. This gives us $\sim A$. And with $\sim A$, the second contradiction also comes easily, at the level of atomics and negated atomics.

Though it can be useful, as we have said, this strategy is often difficult to see. And there is no obvious way to give a strategy for using the strategy! The best thing to say is that you should *look for it* when the other strategies seem to fail.

Let us consider an extended example which combines some of the strategies. We show that $\sim A \rightarrow B \vdash_{NDs} B \vee A$.

(AP)

1.	$\sim A \rightarrow B$	P
	$B \vee A$	

To start, there is a definite goal. We do not see a contradiction in the premises; there is no formula with main operator $\vee$ in the premises; and the goal does not appear in the premises. So we might try going for the goal by $\vee$I in application of SG4. This would require getting a B or an A. It is reasonable to go this way, but it turns out to be a dead end. (You should convince yourself that this is so.) Thus we fall through to SG5.

1.	$\sim A \to B$	P
2.	$\sim(B \vee A)$	A (c, $\sim$E)
	$\bot$	
	$B \vee A$	2-_ $\sim$E

Especially considering our goal has main operator $\vee$, set up to get the goal by $\sim$E.

Now we need a contradiction. For this, our first thought is to go for atomics and negated atomics. But there is nothing to be done. Similarly, there is no formula with main operator $\vee$. So we fall through to SC3 and continue as follows:

1.	$\sim A \to B$	P
2.	$\sim(B \vee A)$	A (c, $\sim$E)
	$B \vee A$	
	$\bot$	_,2 $\bot$I
	$B \vee A$	2-_ $\sim$E

Given a negation that cannot be broken down, set up to get the contradiction by building up to the opposite.

It might seem that we have made no progress, since our new goal is no different than the original! But there is progress insofar as we have an accessible formula not available before (more on this in a moment). At this stage, we *can* get the goal by $\vee$I. Either side will work, but it is easier to start with the A. So we set up for that.

1.	$\sim A \to B$	P
2.	$\sim(B \vee A)$	A (c, $\sim$E)
	A	
	$B \vee A$	_ $\vee$I
	$\bot$	_,2 $\bot$I
	$B \vee A$	2-_ $\sim$E

For a goal with main operator $\vee$, go for the goal by $\vee$I

Now the goal is atomic. Again, there is no contradiction or formula with main operator $\vee$ on accessible lines. The goal is not on accessible lines in any form we can hope to exploit. And the goal has no main operator. So, again, we fall through to SG5.

1.	$\sim A \to B$	P
2.	$\sim(B \vee A)$	A (c, $\sim$E)
3.	$\sim A$	A (c, $\sim$E)
	$\bot$	
	A	3-_ $\sim$E
	$B \vee A$	_ $\vee$I
	$\bot$	_,2 $\bot$I
	$B \vee A$	2-_ $\sim$E

Especially for atomics, go for the goal by $\sim$E

Again, to obtain the contradiction, our first thought is to get atomics and negated atomics. We can get B from lines (1) and (3) by $\to$E. But that is all. So we will not get a contradiction from atomics and negated atomics alone. There is no formula with main operator $\vee$. However, the possibility of getting a B suggests that we *can* build up to the opposite of line (2). That is, we complete the subderivation as follows, and follow our exit strategies to complete the whole.

1.	$\sim A \to B$	P
2.	$\mid\ \sim(B \vee A)$	A (c, $\sim$E)
3.	$\mid\ \mid\ \sim A$	A (c, $\sim$E)
4.	$\mid\ \mid\ B$	1,3 $\to$E
5.	$\mid\ \mid\ B \vee A$	4 $\vee$I
6.	$\mid\ \mid\ \bot$	5,2 $\bot$I
7.	$\mid\ A$	3-6 $\sim$E
8.	$\mid\ B \vee A$	7 $\vee$I
9.	$\mid\ \bot$	8,2 $\bot$I
10.	$B \vee A$	2-9 $\sim$E

Get the contradiction by building up to the opposite of an existing negation.

A couple of comments: First, observe that we build up to the opposite of $\sim(B \vee A)$ *twice*, coming at it from different directions. First we obtain the left side B and use $\vee$I to obtain the whole, then the right side A and use $\vee$I to obtain the whole. This "double use" is typical with negated disjunctions. Second, note that this derivation might be reconceived as an instance of SC4. $\sim A$ gets us B, and so $B \vee A$, which contradicts $\sim(B \vee A)$. But A gets us $B \vee A$ which again contradicts $\sim(B \vee A)$. So both A and $\sim A$ lead to contradiction; so we assume one ($\sim A$), and get the first contradiction; this gets us A, from which the second contradiction follows.

The general pattern of this derivation is typical for goal formulas with main operator $\vee$. For $\mathcal{P} \vee \mathcal{Q}$ we may not be able to prove either $\mathcal{P}$ or $\mathcal{Q}$ from scratch—so that the formula is not directly provable by $\vee$I. However, it may be *indirectly* provable. If it is provable at all, it *must* be that the negation of one side forces the other. So it must be possible to get the $\mathcal{P}$ or the $\mathcal{Q}$ under the *additional* assumption that the other is false. This makes possible an argument of the following form:

(AQ)

a.	$\mid\ \sim(\mathcal{P} \vee \mathcal{Q})$	A (c, $\sim$E)
b.	$\mid\ \mid\ \sim\mathcal{P}$	A (c, $\sim$E)
	$\mid\ \mid\ \vdots$	
c.	$\mid\ \mid\ \mathcal{Q}$	
d.	$\mid\ \mid\ \mathcal{P} \vee \mathcal{Q}$	c $\vee$I
e.	$\mid\ \mid\ \bot$	d,a $\bot$I
f.	$\mid\ \mathcal{P}$	b-e $\sim$E
g.	$\mid\ \mathcal{P} \vee \mathcal{Q}$	f $\vee$I
h.	$\mid\ \bot$	g,a $\bot$I
i.	$\mathcal{P} \vee \mathcal{Q}$	a-h $\sim$E

The "work" in this routine is getting from the negation of one side of the disjunction to the other. Thus if from the assumption $\sim\mathcal{P}$ it is possible to derive $\mathcal{Q}$, all the rest is automatic. We have just seen an extended example (AP) of this pattern. It may be seen as an application of SC3 or SC4 (or both). Where a disjunction may be provable but not provable by $\vee$I, it *will* work by this method. Observe that $\vee$I still plays an essential role—only not as the main strategy. In difficult cases when the goal is a disjunction, it is wise to think about whether you can get one side from the negation of the other. If you can, set up as above. (And reconsider this method when we get to a simplified version in the extended system *NDs+*.)

This example was fairly difficult! You may see some longer, but you will not see many harder. The strategies are not a cookbook for performing all derivations—doing derivations remains an art. But the strategies will give you a good start, and take you a long way through the exercises that follow. The theorems immediately below again foreshadow rules of *NDs+*.

T6.21. $\vdash_{NDs} \mathcal{A} \vee \sim\mathcal{A}$ *principle of excluded middle*

*T6.22. $\vdash_{NDs} \sim(\mathcal{A} \wedge \mathcal{B}) \leftrightarrow (\sim\mathcal{A} \vee \sim\mathcal{B})$

T6.23. $\vdash_{NDs} \sim(\mathcal{A} \vee \mathcal{B}) \leftrightarrow (\sim\mathcal{A} \wedge \sim\mathcal{B})$

T6.24. $\vdash_{NDs} (\sim\mathcal{A} \to \mathcal{B}) \leftrightarrow (\mathcal{A} \vee \mathcal{B})$

T6.25. $\vdash_{NDs} (\mathcal{A} \to \mathcal{B}) \leftrightarrow (\sim\mathcal{A} \vee \mathcal{B})$

T6.26. $\vdash_{NDs} [\mathcal{A} \wedge (\mathcal{B} \vee \mathcal{C})] \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \vee (\mathcal{A} \wedge \mathcal{C})]$

T6.27. $\vdash_{NDs} [\mathcal{A} \vee (\mathcal{B} \wedge \mathcal{C})] \leftrightarrow [(\mathcal{A} \vee \mathcal{B}) \wedge (\mathcal{A} \vee \mathcal{C})]$

T6.28. $\vdash_{NDs} (\mathcal{A} \leftrightarrow \mathcal{B}) \leftrightarrow [(\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})]$

T6.29. $\vdash_{NDs} (\mathcal{A} \leftrightarrow \mathcal{B}) \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \vee (\sim\mathcal{A} \wedge \sim\mathcal{B})]$

T6.30. $\vdash_{NDs} [\mathcal{A} \leftrightarrow (\mathcal{B} \leftrightarrow \mathcal{C})] \leftrightarrow [(\mathcal{A} \leftrightarrow \mathcal{B}) \leftrightarrow \mathcal{C}]$

E6.17. Each of the following begins with a simple application of $\sim$I or $\sim$E. Complete the derivations, and *explain* your use of strategies for a contradiction. Hint: Each of the strategies for a contradiction is used at least once.

*a.

1.	$A \wedge B$	P
2.	$\sim(A \wedge C)$	P
3.	$\quad C$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim C$	3-_ $\sim$I

b.

1.	$(\sim B \vee \sim A) \rightarrow D$	P
2.	$C \wedge \sim D$	P
3.	$\quad \sim B$	A (c, $\sim$E)
	$\quad \bot$	
	B	3-_ $\sim$E

c.

1.	$A \wedge B$	P
2.	$\quad \sim A \vee \sim B$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim(\sim A \vee \sim B)$	2-_ $\sim$I

d.

1.	$A \leftrightarrow \sim A$	P
2.	$\quad B$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim B$	2-_ $\sim$I

e.

1.	$\sim(A \rightarrow B)$	P
2.	$\quad \sim A$	A (c, $\sim$E)
	$\quad \bot$	
	A	2-_ $\sim$E

E6.18. Produce derivations to show each of the following.

*a. $A \rightarrow \sim(B \wedge C),\ B \rightarrow C \vdash_{NDs} A \rightarrow \sim B$

*b. $\vdash_{NDs} \sim(A \rightarrow A) \rightarrow A$

*c. $A \vee B \vdash_{NDs} \sim(\sim A \wedge \sim B)$

*d. $\sim(A \wedge B), \sim(A \wedge \sim B) \vdash_{NDs} \sim A$

*e. $\vdash_{NDs} A \vee \sim A$

*f. $\vdash_{NDs} A \vee (A \to B)$

*g. $A \vee \sim B, \sim A \vee \sim B \vdash_{NDs} \sim B$

*h. $A \leftrightarrow (\sim B \vee C), B \to C \vdash_{NDs} A$

*i. $A \leftrightarrow B \vdash_{NDs} (C \leftrightarrow A) \leftrightarrow (C \leftrightarrow B)$

*j. $A \leftrightarrow \sim(B \leftrightarrow \sim C), \sim(A \vee B) \vdash_{NDs} C$

*k. $[C \vee (A \vee B)] \wedge (C \to E), A \to D, D \to \sim A \vdash_{NDs} C \vee B$

*l. $\sim(A \to B), \sim(B \to C) \vdash_{NDs} \sim D$

*m. $C \to \sim A, \sim(B \wedge C) \vdash_{NDs} (A \vee B) \to \sim C$

*n. $\sim(A \leftrightarrow B) \vdash_{NDs} \sim A \leftrightarrow B$

*o. $A \leftrightarrow B, B \leftrightarrow \sim C \vdash_{NDs} \sim(A \leftrightarrow C)$

*p. $A \vee B, \sim B \vee C, \sim C \vdash_{NDs} A$

*q. $(\sim A \vee C) \vee D, D \to \sim B \vdash_{NDs} (A \wedge B) \to C$

*r. $A \vee D, \sim D \leftrightarrow (E \vee C), (C \wedge B) \vee [C \wedge (F \to C)] \vdash_{NDs} A$

*s. $(A \vee B) \vee (C \wedge D), (A \leftrightarrow E) \wedge (B \to F), G \leftrightarrow \sim(E \vee F), C \to B \vdash_{NDs} \sim G$

*t. $(A \vee B) \wedge \sim C, \sim C \to (D \wedge \sim A), B \to (A \vee E) \vdash_{NDs} E \vee F$

*E6.19. Produce derivations to demonstrate each of T6.21–T6.29. Note that demonstration of T6.30 (from left to right) is left for E6.20e.

E6.20. Produce derivations to show each of the following. These are particularly challenging. If you can get them, you are doing very well!

a. $(A \vee B) \to (A \vee C) \vdash_{NDs} A \vee (B \to C)$

b. $A \to (B \vee C) \vdash_{NDs} (A \to B) \vee (A \to C)$

c. $(A \leftrightarrow B) \leftrightarrow (C \leftrightarrow D) \vdash_{NDs} (A \leftrightarrow C) \to (B \to D)$

d. $\sim(A \leftrightarrow B), \sim(B \leftrightarrow C), \sim(C \leftrightarrow A) \vdash_{NDs} \sim K$

e. $A \leftrightarrow (B \leftrightarrow C) \vdash_{NDs} (A \leftrightarrow B) \leftrightarrow C$

6.2.5 The System *NDs+*

We turn now to some derived rules that will be useful for streamlining derivations. *NDs+* includes all the rules of *NDs*, with some additional inference rules and new *replacement* rules. It is not possible to derive anything in *NDs+* that cannot already be derived in *NDs*. Thus the new rules do not add extra derivation power. They are rather "shortcuts" for things that can already be done in *NDs*. This is particularly obvious in the case of the inference rules.

We have already seen $\bot$I as a first example of a derived rule. As described on page 217 it is possible to derive $\bot$ from any $\mathcal{Q}$ and $\sim\mathcal{Q}$. It is possible also to introduce a companion $\bot$E as below and justified by the derivation on the right.

$\bot$E

a.	$\bot$	
	$\mathcal{P}$	a $\bot$E

(AR)

1.	$\bot$	P
2.	$\quad\sim\mathcal{P}$	A (c, $\sim$E)
3.	$\quad\bot$	1 R
4.	$\mathcal{P}$	2-3 $\sim$E

From a contradiction, one can derive anything.[6] Again, the justification for this rule is that it does not let you do anything that you could not already do in *NDs*. In contexts where SG1 applies, this rule shortcuts a step, and cleans out a distracting subderivation.

For other new rules, suppose in an *NDs* derivation we have $\mathcal{P} \rightarrow \mathcal{Q}$ and $\sim\mathcal{Q}$ and want to reach $\sim\mathcal{P}$. No doubt, we would proceed as follows:

(AS)

1.	$\mathcal{P} \rightarrow \mathcal{Q}$	P
2.	$\sim\mathcal{Q}$	P
3.	$\quad\mathcal{P}$	A (c, $\sim$I)
4.	$\quad\mathcal{Q}$	1,3 $\rightarrow$E
5.	$\quad\bot$	4,2 $\bot$I
6.	$\sim\mathcal{P}$	3-5 $\sim$I

We assume $\mathcal{P}$, get the contradiction, and conclude by $\sim$I. Perhaps you have done this so many times that you can do it in your sleep. In *NDs+* you are given a way to shortcut the routine, and go directly from an accessible $\mathcal{P} \rightarrow \mathcal{Q}$ on a, and an accessible $\sim\mathcal{Q}$ on b to $\sim\mathcal{P}$ with justification *a,b* MT (*modus tollens*).

MT

a	$\mathcal{P} \rightarrow \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a,b MT

Again, the justification for this is that the rule does not let you do anything that you could not already do in *NDs*. So if the rules of *NDs* preserve truth, this rule preserves truth. And, as a matter of fact, we already demonstrated that $\mathcal{P} \rightarrow \mathcal{Q}, \sim\mathcal{Q} \vdash_{NDs} \sim\mathcal{P}$ in T6.5.

---

[6] This rule is sometimes known as *ex falso quodlibet*, which translates, "from falsehood anything (follows)."

NB

a	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b	$\sim\mathcal{P}$	
	$\sim\mathcal{Q}$	a,b NB

a	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a,b NB

NB (*negated biconditional*) lets you move from a biconditional and the negation of one side, to the negation of the other. It is like MT, but with the arrow going both ways. The parts are justified in T6.9 and T6.10.

DS

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{P}$	
	$\mathcal{Q}$	a,b DS

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\mathcal{P}$	a,b DS

DS (*disjunctive syllogism*) lets you move from a disjunction and the negation of one side, to the other side of the disjunction. The two parts are justified by T6.7 and T6.8.

HS

a	$\mathcal{O} \rightarrow \mathcal{P}$	
b	$\mathcal{P} \rightarrow \mathcal{Q}$	
	$\mathcal{O} \rightarrow \mathcal{Q}$	a,b HS

HS (*hypothetical syllogism*) is a principle of transitivity by which you may string a pair of conditionals together into one. It is justified by T6.6.

Each of these rules should be clear and easy to use. Here is an example that puts most of the new rules together into one derivation:

(AT)

1.	$A \leftrightarrow B$	P
2.	$\sim B$	P
3.	$A \vee (C \rightarrow D)$	P
4.	$D \rightarrow B$	P
5.	$\sim A$	1,2 NB
6.	$C \rightarrow D$	3,5 DS
7.	$C \rightarrow B$	6,4 HS
8.	$\sim C$	7,2 MT

1.	$A \leftrightarrow B$	P
2.	$\sim B$	P
3.	$A \vee (C \rightarrow D)$	P
4.	$D \rightarrow B$	P
5.	$\quad A$	A (g, 3$\vee$E)
6.	$\quad\quad C$	A (c, $\sim$I)
7.	$\quad\quad B$	1,5 $\leftrightarrow$E
8.	$\quad\quad \bot$	7,2 $\bot$I
9.	$\quad \sim C$	6-8 $\sim$I
10.	$\quad C \rightarrow D$	A (g, 3$\vee$E)
11.	$\quad\quad C$	A (c, $\sim$I)
12.	$\quad\quad D$	10,11 $\rightarrow$E
13.	$\quad\quad B$	4,12 $\rightarrow$E
14.	$\quad\quad \bot$	13,2 $\bot$I
15.	$\quad \sim C$	11-14 $\sim$I
16.	$\sim C$	3,5-9,10-15 $\vee$E

We can do it by our normal methods purely with the rules of *NDs* as on the right. But it is easier with the shortcuts from *NDs+* as on the left. It may take you some time to "see" applications of the new rules when you are doing derivations, but the simplification makes it worth getting used to them.

The replacement rules of *NDs+* are different from ones we have seen before in two respects. First, replacement rules go in two directions. Consider the following simple rule:

DN $\qquad \mathscr{P} \lhd\rhd \sim\sim\mathscr{P}$

According to DN (*double negation*), given $\mathscr{P}$ on an accessible line a, you may move to $\sim\sim\mathscr{P}$ with justification a DN; and given $\sim\sim\mathscr{P}$ on an accessible line a, you may move to $\mathscr{P}$ with justification a DN. This two-way rule is justified by T6.17, in which we showed $\vdash_{NDs} \mathscr{P} \leftrightarrow \sim\sim\mathscr{P}$. Given $\mathscr{P}$ we could use the routine from one half of the derivation to reach $\sim\sim\mathscr{P}$, and given $\sim\sim\mathscr{P}$ we could use the routine from the other half of the derivation to reach $\mathscr{P}$.

But further, we can use replacement rules to replace a subformula that is just a proper part of another formula. Thus, for example, in the following list, we could move in one step by DN from the formula on the left to any of the ones on the right, and from any of the ones on the right to the one on the left.

(AU) $\qquad A \wedge (B \to C)$

$$\begin{array}{l} \sim\sim[A \wedge (B \to C)] \\ \sim\sim A \wedge (B \to C) \\ A \wedge \sim\sim(B \to C) \\ A \wedge (\sim\sim B \to C) \\ A \wedge (B \to \sim\sim C) \end{array}$$

The first application is of the sort we have seen before, in which the whole formula is replaced. In the second, the replacement is between the subformulas A and $\sim\sim A$. In the third, between the subformulas $(B \to C)$ and $\sim\sim(B \to C)$. The fourth switches B and $\sim\sim B$, and the last C and $\sim\sim C$. Thus the DN rule allows the substitution of any subformula $\mathscr{P}$ with one of the form $\sim\sim\mathscr{P}$, and vice versa.

The application of replacement rules to subformulas is not so easily justified as their application to whole formulas. A complete justification that *NDs+* does not let you go beyond what can be derived in *NDs* will have to wait for Part III. Roughly, though, the idea is this: Given a complex formula, we can take it apart, do the replacement, and then put it back together. Here is a very simple example from above:

(AV)

1.	$A \wedge (B \to C)$	P
2.	$A \wedge \sim\sim(B \to C)$	1 DN

1.	$A \wedge (B \to C)$	P
2.	A	1 $\wedge$E
3.	$B \to C$	1 $\wedge$E
4.	$\quad \sim(B \to C)$	A (c, $\sim$I)
5.	$\quad \bot$	3,4 $\bot$I
6.	$\sim\sim(B \to C)$	4-5 $\sim$I
7.	$A \wedge \sim\sim(B \to C)$	2,6 $\wedge$I

On the left, we make the move from $A \wedge (B \to C)$ to $A \wedge \sim\sim(B \to C)$ in one step by DN. On the right, using ordinary inference rules, we begin by taking off the

A. Then we convert $B \to C$ to $\sim\sim(B \to C)$, and put it back together with the A. Though we will not be able to show that this sort of thing is generally possible until Part III, for now I will continue to say that replacement rules are "justified" by the corresponding biconditionals. As it happens, for replacement rules, the biconditionals play a crucial role in the demonstration that $\Gamma \vdash_{NDs} \mathcal{P}$ iff $\Gamma \vdash_{NDs+} \mathcal{P}$.

The rest of the replacement rules work the same way.

Com
$$\begin{array}{l} \mathcal{P} \wedge \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \wedge \mathcal{P} \\ \mathcal{P} \vee \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \vee \mathcal{P} \\ \mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \leftrightarrow \mathcal{P} \end{array}$$

Com (*commutation*) lets you reverse the order of formulas in a conjunction, disjunction, or biconditional. By Com you could go from, say, $A \wedge (B \vee C)$ to $(B \vee C) \wedge A$, switching the order around $\wedge$, or from $A \wedge (B \vee C)$ to $A \wedge (C \vee B)$, switching the order around $\vee$. You should be clear about why this is so. The different forms are justified by T6.11, T6.13, and T6.12.

Assoc
$$\begin{array}{l} \mathcal{O} \wedge (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \wedge \mathcal{Q} \\ \mathcal{O} \vee (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \vee \mathcal{P}) \vee \mathcal{Q} \\ \mathcal{O} \leftrightarrow (\mathcal{P} \leftrightarrow \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \leftrightarrow \mathcal{P}) \leftrightarrow \mathcal{Q} \end{array}$$

Assoc (*association*) lets you shift parentheses for conjunctions, disjunctions, and biconditionals. The different forms are justified by T6.16, T6.20, and T6.30.

Idem
$$\begin{array}{l} \mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \wedge \mathcal{P} \\ \mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{P} \end{array}$$

Idem (*idempotence)* exposes the equivalence between $\mathcal{P}$ and $\mathcal{P} \wedge \mathcal{P}$, and between $\mathcal{P}$ and $\mathcal{P} \vee \mathcal{P}$. The two forms are justified by T6.18 and T6.19.

Impl
$$\begin{array}{l} \mathcal{P} \to \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \mathcal{Q} \\ \sim\mathcal{P} \to \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{Q} \end{array}$$

Impl (*implication*) lets you move between a conditional and a corresponding disjunction. Thus, for example, by the first form of Impl you could move from $A \to (\sim B \vee C)$ to $\sim A \vee (\sim B \vee C)$, using the rule from left to right, or to $A \to (B \to C)$, using the rule from right to left. As we will see, this rule can be particularly useful. The two forms are justified by T6.24 and T6.25.

Trans
$$\mathcal{P} \to \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{Q} \to \sim\mathcal{P}$$

Trans (*transposition*) lets you reverse the antecedent and consequent around a conditional—subject to the addition or removal of negations. From left to right, this rule should remind you of MT, as Trans plus $\to$E has the same effect as one application of MT. Trans is justified by T6.14.

DeM
$$\begin{array}{l} \sim(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \sim\mathcal{Q} \\ \sim(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \wedge \sim\mathcal{Q} \end{array}$$

DeM (*DeMorgan*) should remind you of equivalences we learned in Chapter 5, for *not both* (the first form) and *neither nor* (the second form). This rule also can be very useful. The two forms are justified by T6.22 and T6.23.

Exp	$\mathcal{O} \to (\mathcal{P} \to \mathcal{Q}) \;\triangleleft\!\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \to \mathcal{Q}$

Exp (*exportation*) is another equivalence that may have arisen in translation. It is justified by T6.15.

Equiv	$\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\!\triangleright\; (\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P})$
	$\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\!\triangleright\; (\mathcal{P} \wedge \mathcal{Q}) \vee (\sim\mathcal{P} \wedge \sim\mathcal{Q})$

Equiv (*equivalence*) converts between a biconditional and the corresponding pair of conditionals, or converts between a biconditional and a corresponding pair of conjunctions. The two forms are justified by T6.28 and T6.29.

Dist	$\mathcal{O} \wedge (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\!\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \vee (\mathcal{O} \wedge \mathcal{Q})$
	$\mathcal{O} \vee (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\!\triangleright\; (\mathcal{O} \vee \mathcal{P}) \wedge (\mathcal{O} \vee \mathcal{Q})$

Dist (*distribution*) works something like the mathematical principle for multiplying across a sum. In each case, moving from left to right, the operator from outside attaches to each of the parts inside the parenthesis, and the operator from inside becomes the main operator. The two forms are justified by T6.26 and T6.27.

Thus end the rules of *NDs+*. They are a lot to absorb at once. But you do not need to absorb all the rules at once. Again, the rules do not let you do anything you could not already do in *NDs*. For the most part, you should proceed as if you were in *NDs*. If an *NDs+* shortcut occurs to you, use it. You will gradually become familiar with more and more of the special *NDs+* rules. Perhaps, though, we can make a few observations about strategy that will get you started. First, again, do not get too distracted by the extra rules! You should continue with the overall goal-directed approach from *NDs*. There are, however, a few contexts where special rules from *NDs+* can make a substantive difference. I comment on three.

First, as we have seen, in *NDs* formulas with $\vee$ can be problematic. $\vee$E is awkward to apply, and $\vee$I does not always work. In simple cases, DS can get you out of $\vee$E. But this is not always so, and you will want to keep $\vee$E among your standard strategies. More importantly, Impl can convert between awkward goal formulas with main operator $\vee$ and more manageable ones with main operator $\to$. Although a disjunction may be derivable, but not by $\vee$I, if a conditional is derivable, it *is* derivable by $\to$I. Thus to reach a goal with main operator $\vee$, consider going for the corresponding $\to$, and converting with Impl.

given

	$\mathcal{A} \vee \mathcal{B}$	(goal)

use

a.	$\sim\mathcal{A}$	A (g, $\to$I)
b.	$\mathcal{B}$	(goal)
c.	$\sim\mathcal{A} \to \mathcal{B}$	a-b $\to$I
	$\mathcal{A} \vee \mathcal{B}$	c Impl

And the other form of Impl may be helpful for a goal of the sort $\sim\mathcal{A} \vee \mathcal{B}$. Here is a quick example:

NDs+ Quick Reference

NDs+ includes all the rules of *NDs* and,

Inference Rules:

⊥I (*bottom intro*)

$$\begin{array}{l|ll} \text{a.} & \mathcal{Q} & \\ \text{b.} & \sim\mathcal{Q} & \\ & & \\ & \bot & \text{a,b } \bot\text{I} \end{array}$$

⊥E (*bottom exploit*)

$$\begin{array}{l|ll} \text{a.} & \bot & \\ & & \\ & \mathcal{P} & \text{a } \bot\text{E} \end{array}$$

MT (*Modus Tollens*)

$$\begin{array}{l|ll} \text{a} & \mathcal{P} \to \mathcal{Q} & \\ \text{b} & \sim\mathcal{Q} & \\ & & \\ & \sim\mathcal{P} & \text{a,b MT} \end{array}$$

NB (*Negated Biconditional*)

$$\begin{array}{l|ll} \text{a} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\ \text{b} & \sim\mathcal{P} & \\ & & \\ & \sim\mathcal{Q} & \text{a,b NB} \end{array}$$

NB (*Negated Biconditional*)

$$\begin{array}{l|ll} \text{a} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\ \text{b} & \sim\mathcal{Q} & \\ & & \\ & \sim\mathcal{P} & \text{a,b NB} \end{array}$$

DS (*Disjunctive Syllogism*)

$$\begin{array}{l|ll} \text{a} & \mathcal{P} \vee \mathcal{Q} & \\ \text{b} & \sim\mathcal{P} & \\ & & \\ & \mathcal{Q} & \text{a,b DS} \end{array}$$

DS (*Disjunctive Syllogism*)

$$\begin{array}{l|ll} \text{a} & \mathcal{P} \vee \mathcal{Q} & \\ \text{b} & \sim\mathcal{Q} & \\ & & \\ & \mathcal{P} & \text{a,b DS} \end{array}$$

HS (*Hypothetical Syllogism*)

$$\begin{array}{l|ll} \text{a} & \mathcal{O} \to \mathcal{P} & \\ \text{b} & \mathcal{P} \to \mathcal{Q} & \\ & & \\ & \mathcal{O} \to \mathcal{Q} & \text{a,b HS} \end{array}$$

Replacement Rules:

DN	$\mathcal{P} \lhd\rhd \sim\sim\mathcal{P}$	Idem	$\mathcal{P} \lhd\rhd \mathcal{P} \wedge \mathcal{P}$ $\mathcal{P} \lhd\rhd \mathcal{P} \vee \mathcal{P}$
Assoc	$\mathcal{O} \wedge (\mathcal{P} \wedge \mathcal{Q}) \lhd\rhd (\mathcal{O} \wedge \mathcal{P}) \wedge \mathcal{Q}$ $\mathcal{O} \vee (\mathcal{P} \vee \mathcal{Q}) \lhd\rhd (\mathcal{O} \vee \mathcal{P}) \vee \mathcal{Q}$ $\mathcal{O} \leftrightarrow (\mathcal{P} \leftrightarrow \mathcal{Q}) \lhd\rhd (\mathcal{O} \leftrightarrow \mathcal{P}) \leftrightarrow \mathcal{Q}$	Com	$\mathcal{P} \wedge \mathcal{Q} \lhd\rhd \mathcal{Q} \wedge \mathcal{P}$ $\mathcal{P} \vee \mathcal{Q} \lhd\rhd \mathcal{Q} \vee \mathcal{P}$ $\mathcal{P} \leftrightarrow \mathcal{Q} \lhd\rhd \mathcal{Q} \leftrightarrow \mathcal{P}$
Exp	$\mathcal{O} \to (\mathcal{P} \to \mathcal{Q}) \lhd\rhd (\mathcal{O} \wedge \mathcal{P}) \to \mathcal{Q}$	Trans	$\mathcal{P} \to \mathcal{Q} \lhd\rhd \sim\mathcal{Q} \to \sim\mathcal{P}$
DeM	$\sim(\mathcal{P} \wedge \mathcal{Q}) \lhd\rhd \sim\mathcal{P} \vee \sim\mathcal{Q}$ $\sim(\mathcal{P} \vee \mathcal{Q}) \lhd\rhd \sim\mathcal{P} \wedge \sim\mathcal{Q}$	Impl	$\mathcal{P} \to \mathcal{Q} \lhd\rhd \sim\mathcal{P} \vee \mathcal{Q}$ $\sim\mathcal{P} \to \mathcal{Q} \lhd\rhd \mathcal{P} \vee \mathcal{Q}$
Dist	$\mathcal{O} \wedge (\mathcal{P} \vee \mathcal{Q}) \lhd\rhd (\mathcal{O} \wedge \mathcal{P}) \vee (\mathcal{O} \wedge \mathcal{Q})$ $\mathcal{O} \vee (\mathcal{P} \wedge \mathcal{Q}) \lhd\rhd (\mathcal{O} \vee \mathcal{P}) \wedge (\mathcal{O} \vee \mathcal{Q})$		
Equiv	$\mathcal{P} \leftrightarrow \mathcal{Q} \lhd\rhd (\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P})$ $\mathcal{P} \leftrightarrow \mathcal{Q} \lhd\rhd (\mathcal{P} \wedge \mathcal{Q}) \vee (\sim\mathcal{P} \wedge \sim\mathcal{Q})$		

(AW)

1.	$\quad \sim A$	A (g, $\rightarrow$I)
2.	$\quad \sim A$	1 R
3.	$\sim A \rightarrow \sim A$	1-2 $\rightarrow$I
4.	$A \vee \sim A$	3 Impl

1.	$\quad \sim(A \vee \sim A)$	A (c, $\sim$E)
2.	$\quad\quad A$	A (c, $\sim$I)
3.	$\quad\quad A \vee \sim A$	2 $\vee$I
4.	$\quad\quad \perp$	3,1 $\perp$I
5.	$\quad \sim A$	2-4 $\sim$I
6.	$\quad A \vee \sim A$	5 $\vee$I
7.	$\quad \perp$	6,1 $\perp$I
8.	$A \vee \sim A$	1-7 $\sim$E

The derivation on the left using Impl is completely trivial, requiring just a derivation of $\sim A \rightarrow \sim A$. But the derivation on the right is not. It falls through to SG5, and then requires a challenging application of SC3 or SC4. This proposed strategy replaces or simplifies the pattern (AQ) for disjunctions described on page 250. Observe that the *work*—getting from the negation of one side of a disjunction to the other—is exactly the same. It is only that we use the derived rule to simplify away the distracting and messy setup.

Second, among the most useless formulas for exploitation in *NDs* are ones with main operator $\sim$. But the combination of DeM, Impl, Equiv, and DN let you "push" negations into arbitrary formulas. Thus you can convert formulas with main operator $\sim$ into a more useful form. To see how these rules can be manipulated, consider the following sequence:

(AX)

1.	$\sim(A \rightarrow B)$	P
2.	$\sim(\sim A \vee B)$	1 Impl
3.	$\sim\sim A \wedge \sim B$	2 DeM
4.	$A \wedge \sim B$	3 DN

We begin with the negation as main operator, and end with a negation only against an atomic. This sort of thing is often very useful. For example, in going for a contradiction, you have the option of "breaking down" a formula with main operator $\sim$ rather than automatically building up to its opposite, according to SC3.

Finally, observe that derivations which can be conducted entirely be replacement rules are "reversible." Thus, for a simple case,

(AY)

1.	$\quad \sim(A \wedge \sim B)$	A (g, $\leftrightarrow$I)
2.	$\quad \sim A \vee \sim\sim B$	1 DeM
3.	$\quad \sim A \vee B$	2 DN
4.	$\quad A \rightarrow B$	3 Impl
5.	$\quad A \rightarrow B$	A (g, $\leftrightarrow$I)
6.	$\quad \sim A \vee B$	5 Impl
7.	$\quad \sim A \vee \sim\sim B$	6 DN
8.	$\quad \sim(A \wedge \sim B)$	7 DeM
9.	$\sim(A \wedge \sim B) \leftrightarrow (A \rightarrow B)$	1-4,5-8 $\leftrightarrow$I

We set up for $\leftrightarrow$I in the usual way. Then the subderivations work by precisely the same steps, DeM, DN, Impl, but in the reverse order. This is not surprising since replacement rules work in both directions. Notice that reversal does *not* generally work where regular inference rules are involved.

The rules of *NDs+* are not a "magic bullet" to make all difficult derivations go away! Rather, with the derived rules, we set aside a certain sort of difficulty that should no longer worry us, so that we are in a position to take on new challenges without becoming overwhelmed by details.

E6.21. Produce derivations to show each of the following.

*a. $(H \wedge G) \rightarrow (L \vee K), G \wedge H \vdash_{NDs+} K \vee L$

*b. $\vdash_{NDs+} [(A \wedge B) \rightarrow (B \wedge A)] \wedge [\sim(A \wedge B) \rightarrow \sim(B \wedge A)]$

*c. $[(K \wedge J) \vee I] \vee \sim Y, Y \wedge [(I \vee K) \rightarrow F] \vdash_{NDs+} F \vee N$

*d. $\sim L \vee (\sim Z \vee \sim U), (U \wedge G) \vee H, Z \vdash_{NDs+} L \rightarrow H$

*e. $F \rightarrow (\sim G \vee H), F \rightarrow G, \sim(H \vee I) \vdash_{NDs+} F \rightarrow J$

*f. $F \rightarrow (G \rightarrow H), \sim I \rightarrow (F \vee H), F \rightarrow G \vdash_{NDs+} I \vee H$

g. $G \rightarrow (H \wedge \sim K), H \leftrightarrow (L \wedge I), \sim I \vee K \vdash_{NDs+} \sim G$

h. $\sim(Z \vee \sim X) \vee (\sim X \rightarrow \sim Y), X \rightarrow Z, Z \rightarrow Y \vdash_{NDs+} X \leftrightarrow Y$

i. $\vdash_{NDs+} [A \vee (B \vee C)] \leftrightarrow [C \vee (B \vee A)]$

j. $\vdash_{NDs+} [A \rightarrow (B \leftrightarrow C)] \leftrightarrow (A \rightarrow [(\sim B \vee C) \wedge (\sim C \vee B)])$

k. $\vdash_{NDs+} (A \vee [B \rightarrow (A \rightarrow B)]) \leftrightarrow (A \vee [(\sim A \vee \sim B) \vee B])$

l. $\vdash_{NDs+} [\sim A \rightarrow (\sim B \rightarrow C)] \rightarrow [(A \vee B) \vee (\sim\sim B \vee C)]$

m. $\vdash_{NDs+} (\sim A \leftrightarrow \sim A) \leftrightarrow [\sim(\sim A \rightarrow A) \leftrightarrow (A \rightarrow \sim A)]$

n. $\vdash_{NDs+} (A \rightarrow B) \vee (B \rightarrow C)$

o. $\vdash_{NDs+} [(A \rightarrow B) \rightarrow A] \rightarrow A$

E6.22. For each of the following, produce a good translation including interpretation function. Then use a derivation to show that the argument is valid in *NDs+*. The first two are suggested from the history of philosophy; the last is our familiar case from page 2.

a. We have knowledge about numbers.
If Platonism is true, then numbers are not in spacetime.
Either numbers are in spacetime, or we do not interact with them.
We have knowledge about numbers only if we interact with them.

Platonism is not true.

b. There is evil.
If god is good, then there is no evil unless god has morally sufficient reasons for allowing it.
If god is both omnipotent and omniscient, then god does not have morally sufficient reasons for allowing evil.

God is not good, omnipotent, and omniscient.

c. If Bob goes to the fair, then so do Daniel and Edward. Albert goes to the fair only if Bob or Carol go. If Daniel goes, then Edward goes only if Fred goes. But not both Fred and Albert go. So Albert goes to the fair only if Carol goes too.

d. If I think dogs fly, then I am insane or they have really big ears. But if dogs do not have really big ears, then I am not insane. So either I do not think dogs fly, or they have really big ears.

e. If the maid did it, then it was done with a revolver only if it was done in the parlor. But if the butler is innocent, then the maid did it unless it was done in the parlor. The maid did it only if it was done with a revolver, while the butler is guilty if it did happen in the parlor. So the butler is guilty.

E6.23. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Derivations as games, and the condition on rules.

b. Accessibility, and auxiliary assumptions.

c. The rules $\vee$I and $\vee$E.

d. The strategies for a goal.

e. The strategies for a contradiction.

6.3 Quantificational

Our full system *ND* includes all the rules for *NDs*, along with new I- and E-rules for quantifiers and equality—so it includes reiteration, with I- and E-rules for $\sim$, $\to$, $\leftrightarrow$, $\wedge$, $\vee$, and then I- and E-rules for $\forall$, $\exists$, and $=$. Thus *ND* completes the basic structure of I- and E-rules. We leave aside derived rules from *NDs+* (except $\bot$I) until they are included again with *ND+*. After some quick introductory remarks, there are sections for the quantifier rules (6.3.1, 6.3.2), for discussion of strategy (6.3.3), then for the equality rules (6.3.4), and for the extended system *ND+* (6.3.5).

First, we do not sacrifice any of the *NDs* rules we have so far. All these rules apply to formulas of quantificational languages as well as to formulas of sentential ones. Thus, for example, $Fx \to \forall xFx$ and Fx are of the form $\mathscr{P} \to \mathscr{Q}$ and $\mathscr{P}$. So we might move from them to $\forall xFx$ by $\to$E as before. And similarly for other rules. Here is a short example:

(AZ)

1.	$\forall xFx \wedge \exists x\forall y(Hx \vee Zy)$	P
2.	$\quad Kx$	A (g, $\to$I)
3.	$\quad \forall xFx$	1 $\wedge$E
4.	$Kx \to \forall xFx$	2-3 $\to$I

The goal is of the form $\mathscr{P} \to \mathscr{Q}$; so we set up to get it in the usual way. And the subderivation is particularly simple. Notice that formulas of the sort $\forall x(Kx \to Fx)$ and Kx are *not* of the form $\mathscr{P} \to \mathscr{Q}$ and $\mathscr{P}$. The main operator of $\forall x(Kx \to Fx)$ is $\forall x$, not $\to$. So $\to$E does not apply. That is why we need new rules for the quantificational operators.

For our quantificational rules, we need a couple of notions already introduced in Chapter 3. Again, for any formula $\mathcal{A}$, variable x, and term t, say $\mathcal{A}^{x}_{t}$ is $\mathcal{A}$ with all the free instances of x replaced by t. And t is *free for x in $\mathcal{A}$* iff all the variables in the replacing instances of t remain free after substitution in $\mathcal{A}^{x}_{t}$. Thus, for example,

(BA) $\quad (\forall xRxy \vee Px)^{x}_{y}$ is $\forall xRxy \vee Py$

There are three instances of x in $\forall xRxy \vee Px$, but only the last is free; so y is substituted only for that instance. Since the substituted y is free in the resultant expression, y is free for x in $\forall xRxy \vee Px$. Similarly,

(BB) $\quad [\forall x(x = y) \vee Ryx]^{y}_{f^1x}$ is $\forall x(x = f^1x) \vee Rf^1xx$

Both instances of y in $\forall x(x = y) \vee Ryx$ are free; so our substitution replaces both. But the x in the first instance of f^1x is bound upon substitution; so f^1x is not free for y in $\forall x(x = y) \vee Ryx$. In contrast, f^1z goes into the same places but is free for y in $\forall x(x = y) \vee Ryx$.

Some quick applications: If x is not free in $\mathcal{A}$, then replacing every free instance of x in $\mathcal{A}$ with some term results in no change; so if x is not free in $\mathcal{A}$, then $\mathcal{A}^{x}_{t}$ is

$\mathcal{A}$. Similarly, $\mathcal{A}^x_x$ is just $\mathcal{A}$ itself. Further, any variable x is sure to be free for itself in a formula $\mathcal{A}$—if every *free* instance of variable x is "replaced" with x, then the replacing instances are sure to be free. Similarly variable-free terms (like constants) are sure to be free for a variable x in a formula $\mathcal{A}$; if a term has no variables, no variable in the replacing term is bound upon substitution. And if $\mathcal{A}$ is quantifier-free then any t is free for variable x in $\mathcal{A}$; if $\mathcal{A}$ has no quantifiers, then no variable in t can be bound upon substitution.

With these concepts, we are ready to turn to our rules. We begin with the easier ones, and work from there.

6.3.1 ∀E and ∃I

∀E and ∃I are straightforward. For the former, for any variable x, given an accessible formula $\forall x\mathcal{P}$ on line a, if term t is free for x in $\mathcal{P}$, one may move to $\mathcal{P}^x_t$ with justification, a ∀E.

∀E	a. $\forall x\mathcal{P}$		provided t is free for x in $\mathcal{P}$
	$\mathcal{P}^x_t$	a ∀E	

∀E removes a quantifier and substitutes a term t for resulting free instances of x, so long as t is free in the resulting formula. We sometimes say that variable x is *instantiated* by term t. Thus, for example, $\forall x\exists yLxy$ is of the form $\forall x\mathcal{P}$, where $\mathcal{P}$ is $\exists yLxy$. So by ∀E we can move from $\forall x\exists yLxy$ to $\exists yLay$, removing the quantifier and substituting a for x. And similarly, since the complex terms f^1a and g^2zb are free for x in $\exists yLxy$, ∀E legitimates moving from $\forall x\exists yLxy$ to $\exists yLf^1ay$ or $\exists yLg^2zby$. What we cannot do is move from $\forall x\exists yLxy$ to $\exists yLyy$ or $\exists yLf^1yy$. These violate the constraint insofar as a variable of the substituted term is bound by a quantifier in the resulting formula.

Intuitively, the motivation for this rule is clear: If $\mathcal{P}$ is satisfied for *every* assignment to variable x, then it is sure to be satisfied for the thing assigned to t, whatever that thing may be. Thus, for example, if everyone loves someone, $\forall x\exists yLxy$, it is sure to be the case that Al, and Al's father love someone—that $\exists yLay$ and $\exists yLf^1ay$. But from everyone loves someone, it does not follow that anyone loves themselves, that $\exists yLyy$, or that anyone is loved by their father $\exists yLf^1yy$. Though we know Al and Al's father loves someone, we do not know who that someone might be. We therefore require that the replacing term be independent of quantifiers in the rest of the formula.

Here are some examples. Notice that we continue to apply bottom-up goal-oriented thinking.

(BC)

1.	$\forall x \forall y Hxy$	P
2.	$Hcf^2ab \rightarrow \forall z Kz$	P
3.	$\forall y Hcy$	1 $\forall$E
4.	Hcf^2ab	3 $\forall$E
5.	$\forall z Kz$	2,4 $\rightarrow$E
6.	Kb	5 $\forall$E

Our original goal is Kb. We could get this by $\forall$E if we had $\forall z Kz$. So we set that as a subgoal. This leads to Hcf^2ab as another subgoal. And we get this from (1) by two applications of $\forall$E. The constant c is free for x in $\forall y Hxy$ so we move from $\forall x \forall y Hxy$ to $\forall y Hcy$ by $\forall$E. And the complex term f^2ab is free for y in Hcy, so we move from $\forall y Hcy$ to Hcf^2ab by $\forall$E. And similarly, we get Kb from $\forall z Kz$ by $\forall$E.

Here is another example, also illustrating strategic thinking:

(BD)

1.	$\forall x Bx$	P
2.	$\forall x(Cx \rightarrow \sim Bx)$	P
3.	$\quad Ca$	A (c, $\sim$I)
4.	$\quad Ca \rightarrow \sim Ba$	2 $\forall$E
5.	$\quad \sim Ba$	4,3 $\rightarrow$E
6.	$\quad Ba$	1 $\forall$E
7.	$\quad \bot$	6,5 $\bot$I
8.	$\sim Ca$	3-7 $\sim$I

Our original goal is $\sim Ca$; so we set up to get it by $\sim$I. And our contradiction appears at the level of atomics and negated atomics. The constant a is free for x in $Cx \rightarrow \sim Bx$. So we move from $\forall x(Cx \rightarrow \sim Bx)$ to $Ca \rightarrow \sim Ba$ by $\forall$E. And similarly, we move from $\forall x Bx$ to Ba by $\forall$E. Notice that we could use $\forall$E to instantiate the universal quantifiers to *any* terms. We pick the constant a because it does us some good in the context of our assumption Ca—itself driven by the goal $\sim Ca$. And it is typical to "swoop" in with universal quantifiers to put variables on terms that matter in a given context.

$\exists$I is also straightforward. For variable x, given an accessible formula $\mathcal{P}^x_t$ on line a, where term t is free for x in formula $\mathcal{P}$, one may move to $\exists x\mathcal{P}$, with justification, a $\exists$I.

$\exists$I

a.	$\mathcal{P}^x_t$		
			provided t is free for x in $\mathcal{P}$
	$\exists x\mathcal{P}$	a $\exists$I	

So for example one might move from Fa to $\exists x Fx$. Note that the statement of this rule is somewhat in reverse from the way one expects it to be: Supposing that t is free for x in $\mathcal{P}$, when one removes the quantifier from the *result* and replaces every free instance of x with t one ends up with the *start*. A consequence is that one starting formula might legitimately lead to different results by $\exists$I. Thus if $\mathcal{P}$ is any of Fxx, Fxa, or Fax, then $\mathcal{P}^x_a$ is Faa. So $\exists$I allows a move from Faa to any of $\exists x Fxx$,

$\exists xFax$, or $\exists xFxa$. In doing a derivation, there is a sense in which we replace one or more instances of a in Faa with x, and add the quantifier to get the result. But then notice that not every instance of the term need be replaced. Officially the rule is stated the other way: Removing the quantifier from the result and replacing free instances of the variable yields the initial formula. Be clear about this in your mind. The requirement that t be free for x in $\mathcal{P}$ prevents moving from $\forall yLyy$ or $\forall yLf^1yy$ to $\exists x\forall yLxy$. The term from which we generalize must be free in the sense that it has no bound variable!

Again, the motivation for this rule is clear. If $\mathcal{P}$ is satisfied for the individual assigned to t, it is sure to be satisfied for *some* individual. Thus, for example, if Al or Al's father love everyone, $\forall yLay$ or $\forall yLf^1ay$, it is sure to be the case that someone loves everyone $\exists x\forall yLxy$. But from the premise that everyone loves themselves $\forall yLyy$, or that everyone is loved by their father $\forall yLf^1yy$ it does not follow that someone loves everyone. Again, the constraint on the rule requires that the term on which we generalize be independent of quantifiers in the rest of the formula.

Here are a couple of examples. The first is relatively simple. The second illustrates the "duality" between $\forall$E and $\exists$I.

(BE)

1.	Ha	P
2.	$\exists yHy \to \forall xJx$	P
3.	$\exists yHy$	1 $\exists$I
4.	$\forall xJx$	2,3 $\to$E
5.	Ja	4 $\forall$E
6.	$Ha \wedge Ja$	1,5 $\wedge$I
7.	$\exists x(Hx \wedge Jx)$	6 $\exists$I

$Ha \wedge Ja$ is $(Hx \wedge Jx)^x_a$ so we can get $\exists x(Hx \wedge Jx)$ from $Ha \wedge Ja$ by $\exists$I. Ha is already a premise, so we set Ja as a subgoal. Ja comes by $\forall$E from $\forall xJx$, and to get this we set $\exists yHy$ as another subgoal. And $\exists yHy$ follows directly by $\exists$I from Ha. Observe that, for now, the natural way to produce a formula with main operator $\exists$ is by $\exists$I. You should fold this into your strategic thinking.

For the second example, recall from translations that $\sim\forall x\sim\mathcal{P}$ is equivalent to $\exists x\mathcal{P}$, and $\sim\exists x\sim\mathcal{P}$ is equivalent to $\forall x\mathcal{P}$. Given this, it turns out that we can use the universal rule with an effect something like $\exists$I, and the existential rule with an effect like $\forall$E. The following pair of derivations illustrate this point:

(BF)

1.	Pa	P
2.	$\quad \forall x\sim Px$	A (c, $\sim$I)
3.	$\quad \sim Pa$	2 $\forall$E
4.	$\quad \bot$	1,3 $\bot$I
5.	$\sim\forall x\sim Px$	2-4 $\sim$I

(BG)

1.	$\sim\exists x\sim Px$	P
2.	$\quad \sim Pa$	A (c, $\sim$E)
3.	$\quad \exists x\sim Px$	2 $\exists$I
4.	$\quad \bot$	3,1 $\bot$I
5.	Pa	2-4 $\sim$E

By $\exists$I we could move from Pa to $\exists xPx$ in one step. In (BF) we use the universal rule to move from the same premise to the equivalent $\sim\forall x\sim Px$. Indeed, $\exists xPx$ abbreviates this very expression. Similarly, by $\forall$E we could move from $\forall xPx$ to Pa

in one step. In (BG), we move to the same result from the equivalent $\sim\exists x\sim Px$ by the existential rule. Thus there is a sense in which, in the presence of rules for negation, the work done by one of these quantifier rules is very similar to, or can substitute for, the work done by the other.

E6.24. Complete the following derivations by filling in justifications for each line. Then for each application of $\forall$E or $\exists$I, explain how the "free for" constraint is met.

a.
1. $\forall x(Ax \to Bxf^1x)$
2. $\forall xAx$
3. Af^1c
4. $Af^1c \to Bf^1cf^1f^1c$
5. $Bf^1cf^1f^1c$

*b.
1. Gaa
2. $\exists yGay$
3. $\exists x\exists yGxy$

c.
1. $\forall x(Rx \wedge Jx)$
2. $Rk \wedge Jk$
3. Rk
4. Jk
5. $Jk \wedge Rk$
6. $\exists y(Jy \wedge Ry)$

d.
1. $\exists x(Rx \wedge Gx) \to \forall yFy$
2. $\forall zGz$
3. Ra
4. Ga
5. $Ra \wedge Ga$
6. $\exists x(Rx \wedge Gx)$
7. $\forall yFy$
8. Fg^2ax

e.
1. $\sim\exists zFg^1z$
2. | $\forall xFx$
3. | Fg^1k
4. | $\exists zFg^1z$
5. | $\bot$
6. $\sim\forall xFx$

E6.25. The following are not legitimate *ND* derivations. In each case, explain why.

a.
1. $\forall x Fx \leftrightarrow Gx$ P
2. $Fj \leftrightarrow Gj$ 1 $\forall$E

*b.
1. $\forall x \exists y Gxy$ P
2. $\exists y Gyy$ 1 $\forall$E

c.
1. $\forall y(Fay \to Gy)$ P
2. $Fay \to Gf^1b$ 1 $\forall$E

d.
1. $\forall y Gf^2xyy$ P
2. $\exists x \forall y Gxy$ 1 $\exists$I

e.
1. Gj P
2. $\exists x Gf^1x$ 1 $\exists$I

E6.26. Provide derivations to show each of the following.

*a. $\forall x Fx \vdash_{ND} Fa \wedge Fb$

*b. $\forall x \forall y Fxy \vdash_{ND} Fab \wedge Fba$

c. $\forall x(Gf^1x \to \forall y Ayx), Gf^1b \vdash_{ND} Af^1cb$

d. $\forall x \forall y(Hxy \to Dyx), \sim Dab \vdash_{ND} \sim Hba$

e. $\vdash_{ND} [\forall x \forall y Fxy \wedge \forall x(Fxx \to A)] \to A$

f. $Fa, Ga \vdash_{ND} \exists x(Fx \wedge Gx)$

*g. $Gaf^1z \vdash_{ND} \exists x \exists y Gxy$

h. $\vdash_{ND} (Fa \vee Fb) \to \exists x Fx$

i. $Gaa \vdash_{ND} \exists x \exists y(Kxx \to Gxy)$

j. $\forall x Fx, Ga \vdash_{ND} \exists y(Fy \wedge Gy)$

*k. $\forall x(Fx \to Gx), \exists y Gy \to Ka \vdash_{ND} Fa \to \exists x Kx$

l. $\forall x \forall y Hxy \vdash_{ND} \exists y \exists x Hyx$

m. $\forall x(\sim Bx \to Kx), \sim Kf^1x \vdash_{ND} Bf^1x$

n. $\forall x \forall y(Fxy \to \sim Fyx) \vdash_{ND} \exists z \sim Fzz$

o. $\forall x(Fx \to Gx), Fa \vdash_{ND} \exists x(\sim Gx \to Hx)$

6.3.2 ∀I and ∃E

In parallel with ∀E and ∃I, rules for ∀I and ∃E are a linked pair. ∀I is as follows: For variables v and x, given an accessible formula $\mathcal{P}^x_v$ at line a—where v is free for x in $\mathcal{P}$, v is not free in any undischarged assumption, and v is not free in $\forall x\mathcal{P}$—one may move to $\forall x\mathcal{P}$ with justification a ∀I.

∀I	a. $\mathcal{P}^x_v$		provided (i) v is free for x in $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathcal{P}$
	$\forall x\mathcal{P}$	a ∀I	

The form of this rule is like ∃I with t a variable: Instead of going from $\mathcal{P}^x_t$ to the existential quantification $\exists x\mathcal{P}$, we move from $\mathcal{P}^x_v$ to the universally quantified $\forall x\mathcal{P}$. The underlying difference is in the special constraints.

First, constraints (i) and (iii) are automatically met when v is x. For x is sure to be free for x in $\mathcal{P}$; and x is not free in $\forall x\mathcal{P}$. And when v is other than x, constraints (i) and (iii) together require that x and v appear free in just the same places of $\mathcal{P}$ and $\mathcal{P}^x_v$. If v is free for x in $\mathcal{P}$, then v is free in $\mathcal{P}^x_v$ everywhere x is free in $\mathcal{P}$. If v is not free in $\forall x\mathcal{P}$, then v is free in $\mathcal{P}^x_v$ only where x is free in $\mathcal{P}$—put the other way around, if $\mathcal{P}^x_v$ has free instances of v in addition to ones that replace instances of x, then $\mathcal{P}$ itself has some free instances of v, so that those instances remain free in $\forall x\mathcal{P}$ and the third condition fails. This two-way requirement is not present for ∃I. Thus, for an example, $Avyv$ and $Axyx$ have x and v free in just the same places; by ∃I one could move from $Avyv$ to $\exists xAxyv$, $\exists xAvyx$, or $\exists xAxyx$; but only a move to $\forall xAxyx$ satisfies constraints (i) and (iii) of the universal rule.

In addition, v cannot be free in an auxiliary assumption still in effect when ∀I is applied. Recall that a formula is true when it is satisfied on every variable assignment. As it turns out (and we shall see in detail in Part II), the truth of a formula with a free variable is therefore equivalent to the truth of its universal quantification. But this is not so under the scope of an assumption in which the variable is free. Under the scope of an assumption with a free variable, we effectively *constrain* the range of assignments under consideration to ones where the assumption is satisfied. Thus under any such assumption, the move to a universal quantification is not justified. However outside the scope of an assumption in which v is free, assignments are unconstrained and the move from $\mathcal{P}^x_v$ to $\forall x\mathcal{P}$ is justified. Again, observe that no such constraint is required for ∃I, which depends on satisfaction for just a single individual, so that any assignment and term will do.

Once you get your mind around them, these constraints are not difficult. Somehow, though, managing them is a common source of frustration for beginning students. However, there is a simple way to be sure that the constraints are met. Suppose you have been following the strategies, along the lines from before, and come to a goal of the sort $\forall x\mathcal{P}$. It is natural to expect to get this by ∀I from $\mathcal{P}^x_v$. You will be sure to satisfy the constraints if you set $\mathcal{P}^x_v$ as a subgoal, *where v does not appear elsewhere in the derivation*. If v does not otherwise appear in the derivation, (i) there cannot be any v-quantifier in $\mathcal{P}$, so v is sure to be free for x in $\mathcal{P}$. If v does not otherwise appear in the derivation, (ii) v cannot appear in any assumption, and so be free in an undischarged assumption. And if v does not otherwise appear in the derivation, (iii) it cannot appear at all in $\forall x\mathcal{P}$, and so cannot be free in $\forall x\mathcal{P}$. It is not always *necessary* to use a new variable in order to satisfy the constraints, and sometimes it is possible to simplify derivations by clever variable selection. However, we shall make it our standard procedure to do so.

Here are some examples. The first is very simple, but illustrates the basic idea underlying the rule.

(BH)

1.	$\forall x(Hx \wedge Mx)$	P
	Hj	
	$\forall yHy$	_ $\forall$I

1.	$\forall x(Hx \wedge Mx)$	P
2.	$Hj \wedge Mj$	1 $\forall$E
3.	Hj	2 $\wedge$E
4.	$\forall yHy$	3 $\forall$I

The goal is $\forall yHy$. So, picking a variable new to the derivation, we set up to get this by $\forall$I from Hj. This goal is easy to obtain from the premise by $\forall$E and $\wedge$E. If every x is such that both Hx and Mx, it is not surprising that every y is such that Hy. The general content from the quantifier is converted to the form with free variables, manipulated by ordinary rules, and converted back to quantified form. This is typical.

Another example has free variables in an auxiliary assumption.

(BI)

1.	$\forall x(Ex \to Sx)$	P
2.	$\forall z(Sz \to Kz)$	P
	$Ej \to Kj$	
	$\forall x(Ex \to Kx)$	_ $\forall$I

1.	$\forall x(Ex \to Sx)$	P
2.	$\forall z(Sz \to Kz)$	P
3.	$\quad Ej$	A (g, $\to$I)
4.	$\quad Ej \to Sj$	1 $\forall$E
5.	$\quad Sj$	4,3 $\to$E
6.	$\quad Sj \to Kj$	2 $\forall$E
7.	$\quad Kj$	6,5 $\to$E
8.	$Ej \to Kj$	3-7 $\to$I
9.	$\forall x(Ex \to Kx)$	8 $\forall$I

Given the goal $\forall x(Ex \to Kx)$, we immediately set up to get it by $\forall$I from $Ej \to Kj$. At this stage, j does not appear elsewhere in the derivation and we can therefore be sure that the constraints will be met when it comes time to apply $\forall$I. The derivation is completed by the usual strategies. Observe that j appears in an auxiliary assumption at (3). This is no problem insofar as the assumption is discharged by the time $\forall$I is applied. Inside the subderivation, however, we would not be able to conclude, say, $\forall xSx$ from (5) or $\forall xKx$ from (7), since at that stage the variable j is free in the undischarged assumption. But, of course, given the strategies there should be no temptation whatsoever to do so. For when we set up for $\forall$I, we set up to do it in a way that is sure to satisfy the constraints.

A last example introduces multiple quantifiers and, again, emphasizes the importance of following the strategies. Insofar as the conclusion merely exchanges variables with the premise, it is no surprise that there is a way for it to be done.

(BJ)

$$\begin{array}{r|ll}
1. & \forall x(Gx \to \forall yFyx) & \text{P} \\
\hline
 & & \\
 & Gj \to \forall xFxj & \\
 & \forall y(Gy \to \forall xFxy) & _\ \forall\text{I}
\end{array}
\qquad
\begin{array}{r|l|ll}
1. & \multicolumn{2}{l}{\forall x(Gx \to \forall yFyx)} & \text{P} \\
\hline
2. & & Gj & \text{A } (g, \to\text{I}) \\
\cline{3-3}
 & & & \\
 & & Fkj & \\
 & & \forall xFxj & _\ \forall\text{I} \\
 & \multicolumn{2}{l}{Gj \to \forall xFxj} & 2\text{-}_\ \to\text{I} \\
 & \multicolumn{2}{l}{\forall y(Gy \to \forall xFxy)} & _\ \forall\text{I}
\end{array}$$

First, we set up to get $\forall y(Gy \to \forall xFxy)$ from $Gj \to \forall xFxj$. The variable j does not appear in the derivation, so we expect that the constraints on ∀I will be satisfied. But our new goal is a conditional, so we set up to go for it by →I in the usual way. This leads to $\forall xFxj$ as a goal, and we set up to get it from Fkj, where k does not otherwise appear in the derivation. Observe that we have at this stage an undischarged assumption in which j appears free. However, our plan is to generalize on k. Since k is new at this stage, we are fine. Of course, this assumes that we are following the strategies so that our new variable automatically avoids variables free in assumptions under which this instance of ∀I falls. This goal is easily obtained and the derivation completed as follows:

$$\begin{array}{r|l|ll}
1. & \multicolumn{2}{l}{\forall x(Gx \to \forall yFyx)} & \text{P} \\
\hline
2. & & Gj & \text{A } (g, \to\text{I}) \\
\cline{3-3}
3. & & Gj \to \forall yFyj & 1\ \forall\text{E} \\
4. & & \forall yFyj & 3,2\ \to\text{E} \\
5. & & Fkj & 4\ \forall\text{E} \\
6. & & \forall xFxj & 5\ \forall\text{I} \\
7. & \multicolumn{2}{l}{Gj \to \forall xFxj} & 2\text{-}6\ \to\text{I} \\
8. & \multicolumn{2}{l}{\forall y(Gy \to \forall xFxy)} & 7\ \forall\text{I}
\end{array}$$

When we apply ∀I the first time, we replace k with x and add the x-quantifier. When we apply ∀I the second time, we replace each instance of j with y and add the y-quantifier. This is just how we planned for the rules to work.

∃E appeals to both a formula and a subderivation. For variables v and x, given an accessible formula $\exists x\mathcal{P}$ at a, and an accessible subderivation beginning with $\mathcal{P}^x_v$ at b and ending with $\mathcal{Q}$ against its scope line at c—where v is free for x in $\mathcal{P}$, v is free in no undischarged assumption, and v is not free in $\exists x\mathcal{P}$ or in $\mathcal{Q}$—one may move to $\mathcal{Q}$, with justification *a,b-c* ∃E.

∃E

$$\begin{array}{r|l|ll}
\text{a.} & \multicolumn{2}{l}{\exists x\mathcal{P}} & \\
\text{b.} & & \mathcal{P}^x_v & \text{A } (g, \text{a}\exists\text{E}) \\
\cline{3-3}
 & & & \\
\text{c.} & & \mathcal{Q} & \\
 & \multicolumn{2}{l}{\mathcal{Q}} & \text{a,b-c } \exists\text{E}
\end{array}$$

provided (i) v is free for x in $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\exists x\mathcal{P}$ or in $\mathcal{Q}$

Notice that the assumption comes with an exit strategy as usual. We can think of this rule on analogy with $\vee$E. A universally quantified expression is something like a big conjunction: if $\forall x\mathcal{P}$, then this element of U is $\mathcal{P}$ *and* that element of U is $\mathcal{P}$ *and*.... And an existentially quantified expression is something like a big disjunction: if $\exists x\mathcal{P}$, then this element of U is $\mathcal{P}$ *or* that element of U is $\mathcal{P}$ *or*.... As though it were a massive $\vee$E, then, we have that something is $\mathcal{P}$, and need to show that $\mathcal{Q}$ follows no matter which thing it happens to be. The constraints guarantee that our reasoning works for any individual to which the assumption applies. Given this, we are in a position to conclude that $\mathcal{Q}$.

Again, if you are following the strategies, a simple way to guarantee that the constraints are met is to use a variable new to the derivation for the assumption. Suppose you are going for goal $\mathcal{Q}$. In parallel with $\vee$, when presented with an accessible formula with main operator $\exists$, it is wise to go for the entire goal by $\exists$E.

(BK)

a.	$\exists x\mathcal{P}$	
	$\mathcal{Q}$	(goal)

a.	$\exists x\mathcal{P}$	
b.	$\quad \mathcal{P}^{x}_{v}$	A (g, a$\exists$E)
c.	$\quad \mathcal{Q}$	(goal)
	$\mathcal{Q}$	a,b-c $\exists$E

Observe that v is free in assumption (b); this is no problem for the requirement (ii) that v is not free in an undischarged auxiliary assumption, insofar as $\exists$E is applied only after the assumption is discharged. And if v does not otherwise appear in the derivation, then (i) there is no v-quantifier in $\mathcal{P}$ and v is sure to be free for x in $\mathcal{P}$. If v does not otherwise appear in the derivation (ii) v does not appear in any other assumption and so is not free in any undischarged auxiliary assumption. And if v does not otherwise appear in the derivation (iii) v does not appear in either $\exists x\mathcal{P}$ or in $\mathcal{Q}$ and so is not free in $\exists x\mathcal{P}$ or in $\mathcal{Q}$. Thus we adopt the same simple expedient to guarantee that the constraints are met. Of course, this presupposes we are following the strategies enough so that other assumptions are in place when we make the assumption for $\exists$E, and that we are clear about the exit strategy, so that we know what $\mathcal{Q}$ will be. The variable is new relative to this much setup.

Here are some examples. The first is particularly simple, and should seem intuitively right. Notice again that given an accessible formula with main operator $\exists$, we go directly for the goal by $\exists$E.

(BL)

1.	$\exists x(Fx \wedge Gx)$	P
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)
	$\quad \exists xFx$	
	$\exists xFx$	1,2-_ $\exists$E

1.	$\exists x(Fx \wedge Gx)$	P
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)
3.	$\quad Fj$	2 $\wedge$E
4.	$\quad \exists xFx$	3 $\exists$I
5.	$\exists xFx$	1,2-4 $\exists$E

Given an accessible formula with main operator $\exists$, we go for the goal by $\exists$E. This gives us a subderivation with the same goal, and our assumption with the new variable.

As it turns out, this goal is easy to obtain, with instances of $\wedge$E and $\exists$I. We could not do $\forall$I to introduce $\forall x Fx$ under the scope of the assumption with j free. But $\exists$I is not so constrained. So we complete the derivation as above. If some x is such that both Fx and Gx then of course some x is such that Fx. Again, we are able to take the quantifier off, manipulate the expressions with free variables, and put the quantifier back on.

Observe that the following is a mistake. It violates the third constraint that the variable $\mathcal{v}$ to which we instantiate the existential is not free in the formula $\mathcal{Q}$ that results from $\exists$E.

(BM)

1.	$\exists x(Fx \wedge Gx)$	P	
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)	
3.	$\quad Fj$	2 $\wedge$E	
4.	Fj	1,2-3 $\exists$E	!Mistake!
5.	$\exists x Fx$	4 $\exists$I	

If you are following the strategies, there should be no temptation to do this. In the above example (BL), we go for the *goal* $\exists x Fx$ by $\exists$E. At that stage, the variable of the assumption j is new to the derivation and so does not appear in the goal. So all is well. This case (BM) does not introduce a variable that is new relative to the goal of the subderivation, and so runs into trouble.

Very often, a goal from $\exists$E is existentially quantified—for introducing an existential quantifier may be a way to bind the variable from the assumption, so that it is not free in the goal. In fact, we do not have to think much about this, insofar as we explicitly introduce the assumption by a variable not in the goal. However, it is not always the case that the goal for $\exists$E is existentially quantified. Here is a simple case of that sort:

(BN)

1.	$\exists x Fx$	P
2.	$\forall z(\exists y Fy \to Gz)$	P
3.	$\quad Fj$	A (g, 1$\exists$E)
	$\quad \forall x Gx$	
	$\forall x Gx$	1,3-_ $\exists$E

1.	$\exists x Fx$	P
2.	$\forall z(\exists y Fy \to Gz)$	P
3.	$\quad Fj$	A (g, 1$\exists$E)
4.	$\quad \exists y Fy \to Gk$	2 $\forall$E
5.	$\quad \exists y Fy$	3 $\exists$I
6.	$\quad Gk$	4,5 $\to$E
7.	$\quad \forall x Gx$	6 $\forall$I
8.	$\forall x Gx$	1,3-7 $\exists$E

Again, given an existential premise, we set up to reach the goal by $\exists$E, where the variable in the assumption is new. In this case, the goal is universally quantified, and illustrates the point that any formula may be the goal for $\exists$E. In this case, we reach the goal in the usual way. To reach $\forall x Gx$ set Gk as goal; at this stage, k is new to the derivation, and so not free in any undischarged assumption. So there is no problem about $\forall$I. Then it is a simple matter of exploiting accessible lines for the result.

Here is an example with multiple quantifiers. It is another case which makes sense insofar as the premise and conclusion merely exchange variables.

(BO)

1.	$\exists x(Fx \wedge \exists yGxy)$	P
2.	$\quad Fj \wedge \exists yGjy$	A (g, 1$\exists$E)
	$\quad \exists y(Fy \wedge \exists xGyx)$	
	$\exists y(Fy \wedge \exists xGyx)$	1,2-_ $\exists$E

1.	$\exists x(Fx \wedge \exists yGxy)$	P
2.	$\quad Fj \wedge \exists yGjy$	A (g, 1$\exists$E)
3.	$\quad \exists yGjy$	2 $\wedge$E
4.	$\quad\quad Gjk$	A (g, 3$\exists$E)
	$\quad\quad \exists y(Fy \wedge \exists xGyx)$	
	$\quad \exists y(Fy \wedge \exists xGyx)$	3,4-_ $\exists$E
	$\exists y(Fy \wedge \exists xGyx)$	1,2-_ $\exists$E

The premise is an existential, so we go for the goal by $\exists$E. This gives us the first subderivation, with the same goal and new variable j substituted for x. But just a bit of simplification gives us another existential on line (3). Thus, following the standard strategies, we set up to go for the goal again by $\exists$E. At this stage, j is no longer new, so we set up another subderivation with new variable k substituted for y. Now the derivation is reasonably straightforward.

1.	$\exists x(Fx \wedge \exists yGxy)$	P
2.	$\quad Fj \wedge \exists yGjy$	A (g, 1$\exists$E)
3.	$\quad \exists yGjy$	2 $\wedge$E
4.	$\quad\quad Gjk$	A (g, 3$\exists$E)
5.	$\quad\quad \exists xGjx$	4 $\exists$I
6.	$\quad\quad Fj$	2 $\wedge$E
7.	$\quad\quad Fj \wedge \exists xGjx$	6,5 $\wedge$I
8.	$\quad\quad \exists y(Fy \wedge \exists xGyx)$	7 $\exists$I
9.	$\quad \exists y(Fy \wedge \exists xGyx)$	3,4-8 $\exists$E
10.	$\exists y(Fy \wedge \exists xGyx)$	1,2-9 $\exists$E

$\exists$I applies in the scope of the subderivations. And we put Fj and $\exists xGjx$ together so that the outer quantifier goes on properly, with y in the right slots.

Finally, observe that $\forall$I and $\exists$I also constitute a dual to one another. The derivations to show this are relatively difficult to create. But to not worry about that. It is enough to understand the steps. For the parallel to $\forall$I, suppose the constraints are met for a derivation of $\forall xPx$ from Pj. And for the parallel to $\exists$E, suppose it is possible to derive Q by $\exists$E from $\exists xPx$; so from application of that rule, in a subderivation, we can get Q from Pj.

(BP)

1.	Pj	P
2.	$\mid\ \exists x{\sim}Px$	A (c, ~I)
3.	$\mid\ \mid\ {\sim}Pj$	A (c, 2$\exists$E)
4.	$\mid\ \mid\ \bot$	1,3 $\bot$I
5.	$\mid\ \bot$	2,3-4 $\exists$E
6.	${\sim}\exists x{\sim}Px$	2-5 ~I

(BQ)

1.	${\sim}\forall x{\sim}Px$	P
2.	$\mid\ {\sim}Q$	A (c, ~E)
3.	$\mid\ \mid\ Pj$	A (c, ~I)
	$\mid\ \mid\ \vdots$	
4.	$\mid\ \mid\ Q$	(somehow)
5.	$\mid\ \mid\ \bot$	4,2 $\bot$I
6.	$\mid\ {\sim}Pj$	3-5 ~I
7.	$\mid\ \forall x{\sim}Px$	6 $\forall$I
8.	$\mid\ \bot$	7,1 $\bot$I
9.	Q	2-8 ~E

Where Pj is a premise, it would be possible to derive $\forall xPx$ in one step by $\forall$I. But in (BP) from the same start we derive the equivalent ${\sim}\exists x{\sim}Px$ by the existential rule. Because conditions for the universal rule apply, j is not free in any undischarged assumption, j is free for x in ${\sim}Px$, and j is not free in $\exists x{\sim}Px$; in addition, it matters that $\bot$ abbreviates a *sentence* and so includes no free instance of j. So the constraints on $\exists$E are satisfied. (The variable j of the assumption at (3) is not *new*—still, constraints are met insofar as j appears only in the premise.) Similarly, if it is possible to derive Q by $\exists$E from $\exists xPx$, we would set up a subderivation starting with Pj, derive Q and use $\exists$E to exit with the Q. In (BQ) we begin with the equivalent ${\sim}\forall x{\sim}Px$ and, supposing it is possible in a subderivation to derive Q from Pj, use the universal rule to derive Q. Again, because conditions for the existential rule apply, j is free for x in ${\sim}Px$, j is not free in $\forall x{\sim}Px$, and j is not free in ${\sim}Q$ or other undischarged assumptions. So the constraints on $\forall$I are satisfied. Thus, again, there is a sense in which in the presence of rules for negation, the work done by one of these quantifier rules is very similar to, or can substitute for, the work done by the other.

E6.27. Complete the following derivations by filling in justifications for each line. Then for each application of $\forall$I or $\exists$E show that the constraints are met by running through each of the three requirements.

a.

1.	$\forall x(Hx \rightarrow Rx)$	
2.	$\forall yHy$	
3.	$Hj \rightarrow Rj$	
4.	Hj	
5.	Rj	
6.	$\forall zRz$	

*b.

$$\begin{array}{rl}
1. & \forall y(Fy \to Gy)\\
2. & \exists zFz\\
3. & \mid\ Fj\\
4. & \mid\ Fj \to Gj\\
5. & \mid\ Gj\\
6. & \mid\ \exists xGx\\
7. & \exists xGx
\end{array}$$

c.

$$\begin{array}{rl}
1. & \exists x\forall y\forall zHxyz\\
2. & \mid\ \forall y\forall zHjyz\\
3. & \mid\ \forall zHjf^1kz\\
4. & \mid\ Hjf^1kf^1k\\
5. & \mid\ \exists xHxf^1kf^1k\\
6. & \mid\ \forall y\exists xHxf^1yf^1y\\
7. & \forall y\exists xHxf^1yf^1y
\end{array}$$

d.

$$\begin{array}{rl}
1. & \forall y\forall x(Fx \to By)\\
2. & \mid\ \exists xFx\\
3. & \mid\ \mid\ Fj\\
4. & \mid\ \mid\ \forall x(Fx \to Bk)\\
5. & \mid\ \mid\ Fj \to Bk\\
6. & \mid\ \mid\ Bk\\
7. & \mid\ Bk\\
8. & \exists xFx \to Bk\\
9. & \forall y(\exists xFx \to By)
\end{array}$$

e.

$$\begin{array}{rl}
1. & \exists x(Fx \to \forall yGy)\\
2. & \mid\ Fj \to \forall yGy\\
3. & \mid\ \mid\ Fj\\
4. & \mid\ \mid\ \forall yGy\\
5. & \mid\ \mid\ Gk\\
6. & \mid\ Fj \to Gk\\
7. & \mid\ \forall y(Fj \to Gy)\\
8. & \mid\ \exists x\forall y(Fx \to Gy)\\
9. & \exists x\forall y(Fx \to Gy)
\end{array}$$

E6.28. The following are not legitimate *ND* derivations. In each case, explain why.

*a.

$$\begin{array}{rll}
1. & Gjy \to Fjy & \text{P}\\
2. & \forall z(Gzy \to Fjy) & 1\ \forall\text{I}
\end{array}$$

b.
$$\begin{array}{rll}
1. & \exists x\forall yByx & \text{P}\\
2. & \quad\vert\ \forall yByy & \text{A }(g,\ 1\exists\text{E})\\
3. & \quad\vert\ Baa & 2\ \forall\text{E}\\
4. & Baa & 1{,}2\text{-}3\ \exists\text{E}
\end{array}$$

c.
$$\begin{array}{rll}
1. & \exists xByx & \text{P}\\
2. & \quad\vert\ Byy & \text{A }(g,\ 1\exists\text{E})\\
3. & \quad\vert\ \exists yByy & 2\ \exists\text{I}\\
4. & \exists yByy & 1{,}2\text{-}3\ \exists\text{E}
\end{array}$$

d.
$$\begin{array}{rll}
1. & \forall x\exists yLxy & \text{P}\\
2. & \exists yLjy & 1\ \forall\text{E}\\
3. & \quad\vert\ Ljk & \text{A }(g,\ 2\exists\text{E})\\
4. & \quad\vert\ \forall xLxk & 3\ \forall\text{I}\\
5. & \quad\vert\ \exists y\forall xLxy & 4\ \exists\text{I}\\
6. & \exists y\forall xLxy & 2{,}3\text{-}5\ \exists\text{E}
\end{array}$$

e.
$$\begin{array}{rll}
1. & \forall x(Hx \rightarrow Gx) & \text{P}\\
2. & \exists xHx & \text{P}\\
3. & \quad\vert\ Hj & \text{A }(g,\ 2\exists\text{E})\\
4. & \quad\vert\ Hj \rightarrow Gj & 1\ \forall\text{E}\\
5. & \quad\vert\ Gj & 4{,}3\ \rightarrow\text{E}\\
6. & Gj & 2{,}3\text{-}5\ \exists\text{E}\\
7. & \forall xGx & 6\ \forall\text{I}
\end{array}$$

E6.29. Provide derivations to show each of the following.

*a. $\forall xKxx \vdash_{ND} \forall zKzz$

b. $\exists xKxx \vdash_{ND} \exists zKzz$

*c. $\forall x{\sim}Kx,\ \forall x({\sim}Kx \rightarrow {\sim}Sx) \vdash_{ND} \forall x(Hx \vee {\sim}Sx)$

d. $\vdash_{ND} \forall xHf^1x \rightarrow \forall xHf^1g^1x$

e. $\forall x\forall y(Gy \rightarrow Fx) \vdash_{ND} \forall x(\forall yGy \rightarrow Fx)$

*f. $\exists yByyy \vdash_{ND} \exists x\exists y\exists zBxyz$

g. $\forall x[(Hx \wedge {\sim}Kx) \rightarrow Ix],\ \exists y(Hy \wedge Gy),\ \forall x(Gx \wedge {\sim}Kx) \vdash_{ND} \exists y(Iy \wedge Gy)$

h. $\forall x(Ax \rightarrow Bx) \vdash_{ND} \exists zAz \rightarrow \exists zBz$

i. $\exists x{\sim}(Cx \vee {\sim}Rx) \vdash_{ND} \exists x{\sim}Cx$

j. $\exists x(Nx \vee Lxx),\ \forall x{\sim}Nx \vdash_{ND} \exists yLyy$

*k. $\forall x \forall y(Fx \to Gy) \vdash_{ND} \forall x(Fx \to \forall y Gy)$

l. $\forall x(Fx \to \forall y Gy) \vdash_{ND} \forall x \forall y(Fx \to Gy)$

m. $\exists x(Mx \wedge \sim Kx), \exists y(\sim Oy \wedge Wy) \vdash_{ND} \exists x \exists y(\sim Kx \wedge \sim Oy)$

n. $\forall x(Fx \to \exists y Gxy) \vdash_{ND} \forall x[Fx \to \exists y(Gxy \vee \sim Hxy)]$

o. $\exists x(Jxa \wedge Cb), \exists x(Sx \wedge Hxx), \forall x[(Cb \wedge Sx) \to \sim Ax] \vdash_{ND} \exists z(\sim Az \wedge Hzz)$

6.3.3 Strategy

Our strategies remain very much as before. They are modified only to accommodate the parallels between $\wedge$ and $\forall$, and between $\vee$ and $\exists$. I restate the strategies in their modified form, and give some examples of each. As before, we begin with strategies for reaching a determinate goal.

SG
1. If accessible lines contain explicit contradiction, use $\sim$E to reach goal.
2. Given an accessible formula with main operator $\exists$ or $\vee$, use $\exists$E or $\vee$E to reach goal (watch "screened" variables).
3. If goal is "in" accessible lines (set goals and) attempt to exploit it out.
4. To reach goal with main operator $\star$, use $\star$I (careful with $\vee$ and $\exists$).
5. Try $\sim$E (especially for atomics and formulas with $\vee$ or $\exists$ as main operator).

And we have strategies for reaching a contradiction.

SC
1. Break accessible formulas down into atomics and negated atomics.
2. Given an available existential or disjunction, go for $\perp$ by $\exists$E or $\vee$E (watch "screened" variables).
3. Set as goal the opposite of some negation (something that cannot itself be broken down); then apply strategies for a goal to reach it.
4. For some $\mathscr{P}$ such that both $\mathscr{P}$ and $\sim\mathscr{P}$ lead to contradiction: Assume $\mathscr{P}$ ($\sim\mathscr{P}$), obtain the first contradiction, and conclude $\sim\mathscr{P}$ ($\mathscr{P}$); then obtain the second contradiction—this is the one you want.

As before, these are listed in priority order, though the frequency order may be different. If a high priority strategy does not apply, simply fall through to one that does. In each case, you may want to refer back to the corresponding section in the sentential case for further discussion and examples.

SG1. *If accessible lines contain explicit contradiction, use* ~*E to reach goal*. The statement is unchanged from before. If accessible lines contain an explicit contradiction, we can assume the negation of our goal, bring the contradiction under the assumption, and conclude to the original goal. Since this always works, we want to jump on it whenever it is available. The only thing to add for the quantificational case is that accessible lines might "contain" a contradiction that is just a short step away buried in quantified expressions. Thus, for example,

(BR)

1.	$\forall x Fx$	P
2.	$\forall y \sim Fy$	P
	Gz	

1.	$\forall x Fx$	P
2.	$\forall y \sim Fy$	P
3.	$\quad \sim Gz$	A (c, ~E)
4.	$\quad Fx$	1 ∀E
5.	$\quad \sim Fx$	2 ∀E
6.	$\quad \bot$	4,5 ⊥I
7.	Gz	3-6 ~E

Though $\forall x Fx$ and $\forall y \sim Fy$ are not themselves an explicit contradiction, they lead by ∀E directly to expressions that are. Given the analogy between $\wedge$ and $\forall$, it is as if we had both $Fa \wedge \ldots \wedge Fb$ and $\sim Fa \wedge \ldots \wedge \sim Fb$ in the premises. In this case, we would not hesitate to go for the goal by ~E. And similarly here.

SG2. *Given an accessible formula with main operator* ∃ *or* ∨*, use* ∃*E or* ∨*E to reach goal* (watch "screened" variables). What is new for this strategy is the existential quantifier. Motivation is the same as before: With goal $\mathcal{Q}$, and an accessible line with main operator ∃, go for the goal by ∃E. Then you have all the same accessible formulas as before, with the addition of the assumption. So you will (typically) be better off in your attempt to reach $\mathcal{Q}$. We have already emphasized this strategy in introducing the rules. Here is an example:

(BS)

1.	$\exists x Fx$	P
2.	$\exists y Gy$	P
3.	$\exists z Fz \rightarrow \forall y Fy$	P
4.	$\quad Fj$	A (g, 1∃E)
5.	$\quad\quad Gk$	A (g, 2∃E)
	$\quad\quad \exists x(Fx \wedge Gx)$	
	$\quad \exists x(Fx \wedge Gx)$	2,5-_ ∃E
	$\exists x(Fx \wedge Gx)$	1,4-_ ∃E

1.	$\exists x Fx$	P
2.	$\exists y Gy$	P
3.	$\exists z Fz \rightarrow \forall y Fy$	P
4.	$\quad Fj$	A (g, 1∃E)
5.	$\quad\quad Gk$	A (g, 2∃E)
6.	$\quad\quad \exists z Fz$	4 ∃I
7.	$\quad\quad \forall y Fy$	3,6 →E
8.	$\quad\quad Fk$	7 ∀E
9.	$\quad\quad Fk \wedge Gk$	8,5 ∧I
10.	$\quad\quad \exists x(Fx \wedge Gx)$	9 ∃I
11.	$\quad \exists x(Fx \wedge Gx)$	2,5-10 ∃E
12.	$\exists x(Fx \wedge Gx)$	1,4-11 ∃E

The premise at (3) has main operator → and so is not existentially quantified. But the first two premises have main operator ∃. So we set up to reach the goal with two applications of ∃E. It does not matter which we do first as, either way, we end up with the same accessible formulas to reach the goal at the innermost subderivation. Once we have the subderivations set up, the rest is straightforward.

Given what we have said, it might appear mysterious how one could be anything but better off going directly for a goal by ∃E or ∨E. But consider the derivations below:

(BT)

1.	$\forall x \exists y Fxy$	P
2.	$\forall x \forall y (Fxy \rightarrow Gxy)$	P
3.	$\exists y Fjy$	1 ∀E
4.	$\quad Fjk$	A (g, 3∃E)
5.	$\quad \forall y (Fjy \rightarrow Gjy)$	2 ∀E
6.	$\quad Fjk \rightarrow Gjk$	5 ∀E
7.	$\quad Gjk$	6,4 →E
8.	$\quad \exists y Gjy$	7 ∃I
9.	$\quad \forall x \exists y Gxy$	!Mistake!
10.	$\forall x \exists y Gxy$	3,4-9 ∃E

(BU)

1.	$\forall x \exists y Fxy$	P
2.	$\forall x \forall y (Fxy \rightarrow Gxy)$	P
3.	$\exists y Fjy$	1 ∀E
4.	$\quad Fjk$	A (g, 3∃E)
5.	$\quad \forall y (Fjy \rightarrow Gjy)$	2 ∀E
6.	$\quad Fjk \rightarrow Gjk$	5 ∀E
7.	$\quad Gjk$	6,4 →E
8.	$\quad \exists y Gjy$	7 ∃I
9.	$\exists y Gjy$	3,4-8 ∃E
10.	$\forall x \exists y Gxy$	9 ∀I

In derivation (BT), we isolate the existential on line (3) and go for the goal, $\forall x \exists y Gxy$ by ∃E. But something is in fact lost when we set up for the subderivation—the variable j, that was not in any undischarged assumption and therefore available for ∀I, gets "screened off" by the assumption and so lost for universal generalization. So at step (9), we are blocked from using (8) and ∀I to reach the goal. The problem is solved in (BU) by letting variable j pass into the subderivation and back out, where it is available again for ∀I. We pass over our second strategy for a goal until we have a new goal in which j is free. This way there is no call to generalize on j under the scope of the assumption. The restriction on ∃E blocks a goal in which k is free, but there is no problem about j.

SG3. *If goal is "in" accessible lines (set goals and) attempt to exploit it out*. The statement of this strategy is the same as before. The only thing to add is that we should consider the instances of a universally quantified expression as already "in" the expression (as if it were a big conjunction). Thus, for example,

(BV)

1.	$Ga \rightarrow \forall x Fx$	P
2.	$\forall x Gx$	P
	$\forall x Fx$	
	Fa	_ ∀E

1.	$Ga \rightarrow \forall x Fx$	P
2.	$\forall x Gx$	P
3.	Ga	2 ∀E
4.	$\forall x Fx$	1,3 →E
5.	Fa	4 ∀E

The original goal Fa is "in" the consequent of (1), $\forall x Fx$. So we set $\forall x Fx$ as a subgoal. This leads to Ga as another subgoal, and we find this "in" the premise at (2).

Here is a more complicated case. When extracting a goal that involves multiple quantifiers and terms it can sometimes help to pencil a "map" for how quantifiers are to be applied.

(BW)

$$
\begin{array}{r|ll}
1. & \forall x \forall y W \overset{a}{x} \overset{b}{y} & \text{P} \\
2. & \forall x \forall y \forall z (W \overset{a}{x} \overset{b}{y} \overset{b}{z} \rightarrow R \overset{b}{z} \overset{a}{x}) & \text{P} \\
\hline
 & & \\
 & Rba &
\end{array}
\qquad
\begin{array}{r|ll}
1. & \forall x \forall y Wxby & \text{P} \\
2. & \forall x \forall y \forall z (Wxyz \rightarrow Rzx) & \text{P} \\
\hline
3. & \forall y Waby & 1\ \forall\text{E} \\
4. & Wabb & 3\ \forall\text{E} \\
5. & \forall y \forall z (Wayz \rightarrow Rza) & 2\ \forall\text{E} \\
6. & \forall z (Wabz \rightarrow Rza) & 5\ \forall\text{E} \\
7. & Wabb \rightarrow Rba & 6\ \forall\text{E} \\
8. & Rba & 7,4 \rightarrow\text{E}
\end{array}
$$

Working back from the goal, we want Rba from the consequent of (2); this tells us how to instantiate z and x in (2); then in order to connect with (1) we instantiate y to b. From this x and y in (1) go to a and b. Then the plan is easily executed.

SG4. *To reach goal with main operator ⋆, use* ⋆I (careful with ∨ and ∃). As before, this is your "bread-and-butter" strategy. You will come to it over and over. Of new applications, the most automatic is for ∀. For a simple case,

(BX)

$$
\begin{array}{r|ll}
1. & \forall x Gx & \text{P} \\
2. & \forall y Fy & \text{P} \\
\hline
 & & \\
 & Fj \wedge Gj & \\
 & \forall z (Fz \wedge Gz) & \_\ \forall\text{I}
\end{array}
\qquad
\begin{array}{r|ll}
1. & \forall x Gx & \text{P} \\
2. & \forall y Fy & \text{P} \\
\hline
3. & Gj & 1\ \forall\text{E} \\
4. & Fj & 2\ \forall\text{E} \\
5. & Fj \wedge Gj & 4,3\ \wedge\text{I} \\
6. & \forall z (Fz \wedge Gz) & 5\ \forall\text{I}
\end{array}
$$

Given a goal with main operator ∀, we immediately set up to get it by ∀I. This leads to $Fj \wedge Gj$ with the new variable j as a subgoal. After that, completing the derivation is easy. Observe that this strategy does not always work for formulas with main operators ∨ and ∃.

SG5. *Try ∼E* (especially for atomics and formulas with ∨ or ∃ as main operator). Recall that atomics now include more than just sentence letters. Thus this strategy applies to goals of the sort Fab or Gz. And, just as one might have good reason to accept that $\mathcal{P}$ or $\mathcal{Q}$ without having good reason to accept that $\mathcal{P}$, or that $\mathcal{Q}$, so one might have reason to accept that $\exists x \mathcal{P}$ without having reason to accept that any particular individual is $\mathcal{P}$—as one might be quite confident that *someone* did it, without evidence sufficient to convict any particular individual. Thus there are contexts where it is possible to derive $\exists x \mathcal{P}$ but not possible to reach it directly by ∃I. SG5 has special application in those contexts. Thus consider the following example:

(BY)

1.	$\sim\forall xAx$	P
2.	$\mid\ \sim\exists x\sim Ax$	A (c, $\sim$E)
	$\mid$	
	$\mid\ \bot$	
	$\exists x\sim Ax$	2-_ $\sim$E

1.	$\sim\forall xAx$	P
2.	$\mid\ \sim\exists x\sim Ax$	A (c, $\sim$E)
3.	$\mid\ \mid\ \sim Aj$	A (c, $\sim$E)
4.	$\mid\ \mid\ \exists x\sim Ax$	3 $\exists$I
5.	$\mid\ \mid\ \bot$	4,2 $\bot$I
6.	$\mid\ Aj$	3-5 $\sim$E
7.	$\mid\ \forall xAx$	6 $\forall$I
8.	$\mid\ \bot$	7,1 $\bot$I
9.	$\exists x\sim Ax$	2-8 $\sim$E

Our initial goal is $\exists x\sim Ax$. There is no contradiction; there is no disjunction or existential; we do not see the goal in the premise; and attempts to reach the goal by $\exists$I are doomed to fail. So we fall through to SG5, and set up to reach the goal by $\sim$E. As it happens, the contradiction is not easy to get! We can think of the derivation as involving applications of either SC3 or SC4. We take up this sort of case below. For now, the important point is just the setup on the left.

Where strategies for a goal apply in the context of some determinate goal, strategies for a contradiction apply when the goal is just some contradiction—and any contradiction will do. Again, there is nothing fundamentally changed from the sentential case, though we can illustrate some special quantificational applications.

SC1. *Break accessible formulas down into atomics and negated atomics*. This works just as before. The only point to emphasize for the quantificational case is one we made for SG1 above, that relevant atomics may be "contained" in quantified expressions. So going for atomics and negated atomics may include "shaking" quantified expressions to see what falls out. Here is a simple example:

(BZ)

1.	$\sim Fa$	P
2.	$\mid\ \forall x(Fx \wedge Gx)$	A (c, $\sim$I)
	$\mid$	
	$\mid\ \bot$	
	$\sim\forall x(Fx \wedge Gx)$	2-_ $\sim$I

1.	$\sim Fa$	P
2.	$\mid\ \forall x(Fx \wedge Gx)$	A (c, $\sim$I)
3.	$\mid\ Fa \wedge Ga$	2 $\forall$E
4.	$\mid\ Fa$	3 $\wedge$E
5.	$\mid\ \bot$	4,1 $\bot$I
6.	$\sim\forall x(Fx \wedge Gx)$	2-5 $\sim$I

Our strategy for the goal is SG4. For an expression with main operator $\sim$, we go for the goal by $\sim$I. We already have $\sim Fa$ toward a contradiction at the level of atomics and negated atomics. And Fa comes from the universally quantified expression by $\forall$E.

SC2. *Given an available existential or disjunction, go for $\bot$ by $\exists$E or $\vee$E* (watch "screened" variables). As before, in many cases you will have applied $\exists$E or $\vee$E by SG2 prior to setting up for $\sim$E or $\sim$I. Then the existential or disjunction is "used up" and unavailable for this strategy. However it may be that an existential or disjunction becomes or remains available inside a subderivation for a tilde rule. In any such case, this strategy has high priority for the same reasons as before: In your attempt to reach a contradiction, you have all the same accessible formulas as before, with the addition of the assumption. So you will (typically) be better off in your attempt to reach a contradiction. Here is an example:

(CA)

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists x Ax$	A (c, $\sim$I)
	$\quad \bot$	
	${\sim}\exists x Ax$	2-_ $\sim$I

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists x Ax$	A (c, $\sim$I)
3.	$\quad\quad Aj$	A (c, 2$\exists$E)
	$\quad\quad \bot$	
	$\quad \bot$	2,3-_ $\exists$E
	${\sim}\exists x Ax$	2-_ $\sim$I

We set up to reach the main goal by $\sim$I. This gives us an existentially quantified expression at (2), where the goal is a contradiction. SC2 tells us to go for $\bot$ by $\exists$E. Observe that, because the goal is $\bot$, the exit strategy is c rather than g. But by application of SC1, this subderivation is easy.

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists x Ax$	A (c, $\sim$I)
3.	$\quad\quad Aj$	A (c, 2$\exists$E)
4.	$\quad\quad {\sim}Aj$	1 $\forall$E
5.	$\quad\quad \bot$	3,4 $\bot$I
6.	$\quad \bot$	2,3-5 $\exists$E
7.	${\sim}\exists x Ax$	2-6 $\sim$I

The contradiction results with Aj on line (3) and ${\sim}Aj$ "contained" on line (1). But as occurs with the parallel goal-directed strategy, the contradiction would not even have been possible without the assumption Aj for $\exists$E.

As can occur with applications of SG2, it is wise to be careful about applications of this strategy when assumptions for $\exists$E or $\vee$E "screen off" variables that would otherwise be available for $\forall$I. Here is an example to illustrate the point:

(CB)

1.	$\sim\forall x\exists yGxy$	P
2.	$\forall x\forall y(Fxy \to Gxy)$	P
3.	$\forall x\exists yFxy$	A (c, $\sim$I)
4.	$\exists yFjy$	3 $\forall$E
5.	Fjk	A (c, 4$\exists$E)
6.	$\forall y(Fjy \to Gjy)$	2 $\forall$E
7.	$Fjk \to Gjk$	6 $\forall$E
8.	Gjk	7,5 $\to$E
9.	$\exists yGjy$	8 $\exists$I
10.	$\forall x\exists yGxy$	!Mistake!
11.	$\bot$	10,1 $\bot$I
12.	$\bot$	4,5-11 $\exists$E
13.	$\sim\forall x\exists yFxy$	3-12 $\sim$I

(CC)

1.	$\sim\forall x\exists yGxy$	P
2.	$\forall x\forall y(Fxy \to Gxy)$	P
3.	$\forall x\exists yFxy$	A (c, $\sim$I)
4.	$\exists yFjy$	3 $\forall$E
5.	Fjk	A (g, 4$\exists$E)
6.	$\forall y(Fjy \to Gjy)$	2 $\forall$E
7.	$Fjk \to Gjk$	6 $\forall$E
8.	Gjk	7,5 $\to$E
9.	$\exists yGjy$	8 $\exists$I
10.	$\exists yGjy$	4,5-9 $\exists$E
11.	$\forall x\exists yGxy$	10 $\forall$I
12.	$\bot$	11,1 $\bot$I
13.	$\sim\forall x\exists yFxy$	3-12 $\sim$I

In derivation (CB), we isolate the existential on line (4) and set up to go for contradiction by $\exists$E. But something is in fact lost when we set up for the subderivation—the variable j, that was not in any undischarged assumption and therefore available for $\forall$I, gets "screened off" by the assumption and so lost for universal generalization. So at step (10), we are blocked from using (9) and $\forall$I to reach the goal. Again, the problem is solved in (CC) by letting variable j pass into the subderivation and back out, where it is available for $\forall$I. As before, we pass over the second strategy for a contradiction until we have a new goal in which j is free. And we apply $\exists$E for it.

SC3. *Set as goal the opposite of some negation (something that cannot itself be broken down); then apply strategies for a goal to reach it.* In principle, this strategy is unchanged from before, though of course there are new applications for quantified expressions. Here is a quick example:

(CD)

1.	$\sim\exists xAx$	P
2.	Aj	A (c, $\sim$I)
	$\bot$	
	$\sim Aj$	2-_ $\sim$I
	$\forall x\sim Ax$	_ $\forall$I

1.	$\sim\exists xAx$	P
2.	Aj	A (c, $\sim$I)
3.	$\exists xAx$	2 $\exists$I
4.	$\bot$	3,1 $\bot$I
5.	$\sim Aj$	2-4 $\sim$I
6.	$\forall x\sim Ax$	5 $\forall$I

Our strategy for the goal is SG4. We plan on reaching $\forall x\sim Ax$ by $\forall$I. So we set $\sim Aj$ as a subgoal. Again the strategy for the goal is SG4, and we set up to get $\sim Aj$ by $\sim$I. Other than the assumption itself, there are no atomics and negated atomics to be had. There is no available existential or disjunction. But the premise is a negated expression. So we set $\exists xAx$ as a goal. And this is easy, as it comes in one step by $\exists$I. (CC) above is another example of this. Needing a contradiction, we build up to the opposite of the formula on line (1).

SC4. *For some $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction:* Assume $\mathcal{P}$ ($\sim\mathcal{P}$), obtain the first contradiction, and conclude $\sim\mathcal{P}$ ($\mathcal{P}$); then obtain the second contradiction—this is the one you want. As in the sentential case, this strategy often coincides with SC3—in building up to the opposite of something that cannot be broken down, one assumes a $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ result in contradiction. Corresponding to the pattern with $\vee$, this often happens when some accessible expression is a negated existential. Here is a challenging example:

(CE)

1.	$\forall x(\sim Ax \rightarrow Kx)$	P
2.	$\sim\forall yKy$	P
3.	$\quad \sim\exists wAw$	A (c, $\sim$E)
	$\quad \bot$	
	$\exists wAw$	3-_ $\sim$E

1.	$\forall x(\sim Ax \rightarrow Kx)$	P
2.	$\sim\forall yKy$	P
3.	$\quad \sim\exists wAw$	A (c, $\sim$E)
4.	$\quad\quad Aj$	A (c, $\sim$I)
5.	$\quad\quad \exists wAw$	4 $\exists$I
6.	$\quad\quad \bot$	5,3 $\bot$I
7.	$\quad \sim Aj$	4-6 $\sim$I
8.	$\quad \sim Aj \rightarrow Kj$	1 $\forall$E
9.	$\quad Kj$	8,7 $\rightarrow$E
10.	$\quad \forall yKy$	9 $\forall$I
11.	$\quad \bot$	10,2 $\bot$I
12.	$\exists wAw$	3-11 $\sim$E

Once we decide that we cannot get the goal directly by $\exists$I, the strategy for a goal falls through to SG5. And, as it turns out, both Aj and $\sim Aj$ lead to contradiction. So we assume one and get the contradiction; this gives us the other which leads to contradiction as well. The decision to assume Aj may seem obscure! But it is a common pattern: Given $\sim\exists x\mathcal{P}$, assume an instance $\mathcal{P}^x_v$ for some variable v, or at least something that will yield $\mathcal{P}^x_v$. Then $\exists$I gives you $\exists x\mathcal{P}$, and so the first contradiction. So you conclude $\sim\mathcal{P}^x_v$—and this *outside* the scope of the assumption, where $\forall$I and the like might apply for v. In effect, you come with an instance of the existential "underneath" its negation, this leads to contradiction and so to a negation of the instance—which has some chance to give you what you want. For another example of this pattern, see (BY) above.

Notice that such cases can also be understood as driven by applications of SC3. In (CE), we set the opposite of the formula on (2) as goal. This leads to Kj and then $\sim Aj$ as subgoals. To reach $\sim Aj$, we assume Aj, and get this by building to the opposite of $\sim\exists wAw$. And similarly in (BY).

Again, these strategies are not a cookbook for performing all derivations—doing derivations remains an art. But the strategies will give you a good start, and take you a long way through the exercises that follow, including derivation of the theorems immediately below.

For derivation of the following theorems, as a matter of notation, let $\mathcal{Q}(x)$, $\mathcal{Q}(x, y)$ and such indicate that $\mathcal{Q}$ may have instances of the indicated variables free—and, in context, $\mathcal{Q}$ without the parenthetical notation that the variables are not free. Then $\mathcal{Q}(t)$

is $\mathcal{Q}(x)^x_t$. This will let you "track" substituted terms in the usual way. So for T6.31 you show $\forall x\mathcal{P}(x) \to \mathcal{P}(t)$, and for T6.34a, $\forall x(\mathcal{P} \wedge \mathcal{Q}(x)) \leftrightarrow (\mathcal{P} \wedge \forall x\mathcal{Q}(x))$.

Observe that, unless explicitly stated, we cannot be sure that an arbitrary t is *free* for x in $\mathcal{Q}(x)$. However, as always, x is free for (free instances of) x in $\mathcal{Q}$. (And, more generally, t must be free for x in $\mathcal{Q}$ if it has no variable beyond x.) It turns out that this is sufficient for demonstration of the following theorems, which you will be able to work without variable exchange (when you get to them in E6.33).

*T6.31. $\vdash_{ND} \forall x\mathcal{P} \to \mathcal{P}^x_t$ where term t is free for variable x in formula $\mathcal{P}$

*T6.32. $\vdash_{ND} \forall x\forall y\mathcal{P} \leftrightarrow \forall y\forall x\mathcal{P}$

T6.33. $\vdash_{ND} \exists x\exists y\mathcal{P} \leftrightarrow \exists y\exists x\mathcal{P}$

T6.34. Where x is not free in $\mathcal{P}$,

*(a) $\vdash_{ND} \forall x(\mathcal{P} \wedge \mathcal{Q}) \leftrightarrow (\mathcal{P} \wedge \forall x\mathcal{Q})$

(b) $\vdash_{ND} \exists x(\mathcal{P} \wedge \mathcal{Q}) \leftrightarrow (\mathcal{P} \wedge \exists x\mathcal{Q})$

(c) $\vdash_{ND} \forall x(\mathcal{Q} \wedge \mathcal{P}) \leftrightarrow (\forall x\mathcal{Q} \wedge \mathcal{P})$

(d) $\vdash_{ND} \exists x(\mathcal{Q} \wedge \mathcal{P}) \leftrightarrow (\exists x\mathcal{Q} \wedge \mathcal{P})$

*(e) $\vdash_{ND} \forall x(\mathcal{P} \vee \mathcal{Q}) \leftrightarrow (\mathcal{P} \vee \forall x\mathcal{Q})$

(f) $\vdash_{ND} \exists x(\mathcal{P} \vee \mathcal{Q}) \leftrightarrow (\mathcal{P} \vee \exists x\mathcal{Q})$

(g) $\vdash_{ND} \forall x(\mathcal{Q} \vee \mathcal{P}) \leftrightarrow (\forall x\mathcal{Q} \vee \mathcal{P})$

(h) $\vdash_{ND} \exists x(\mathcal{Q} \vee \mathcal{P}) \leftrightarrow (\exists x\mathcal{Q} \vee \mathcal{P})$

(i) $\vdash_{ND} \forall x(\mathcal{P} \to \mathcal{Q}) \leftrightarrow (\mathcal{P} \to \forall x\mathcal{Q})$

*(j) $\vdash_{ND} \exists x(\mathcal{P} \to \mathcal{Q}) \leftrightarrow (\mathcal{P} \to \exists x\mathcal{Q})$

(k) $\vdash_{ND} \forall x(\mathcal{Q} \to \mathcal{P}) \leftrightarrow (\exists x\mathcal{Q} \to \mathcal{P})$

(l) $\vdash_{ND} \exists x(\mathcal{Q} \to \mathcal{P}) \leftrightarrow (\forall x\mathcal{Q} \to \mathcal{P})$

T6.35. $\vdash_{ND} \exists x(\mathcal{P} \vee \mathcal{Q}) \leftrightarrow (\exists x\mathcal{P} \vee \exists x\mathcal{Q})$

T6.36. $\vdash_{ND} \forall x(\mathcal{P} \wedge \mathcal{Q}) \leftrightarrow (\forall x\mathcal{P} \wedge \forall x\mathcal{Q})$

T6.37. $\vdash_{ND} \sim\forall x\mathcal{P} \leftrightarrow \exists x{\sim}\mathcal{P}$

T6.38. $\vdash_{ND} \sim\exists x\mathcal{P} \leftrightarrow \forall x{\sim}\mathcal{P}$

E6.30. For each of the following, (i) which strategies for a goal apply? and (ii) what are the next two steps? If the strategies call for a new subgoal, show the subgoal; if they call for a subderivation, set up the subderivation. In each case *explain* your response. Hint: Each of the strategies for a goal is used at least once.

*a.

1.	$\exists x\exists y(Fxy \wedge Gyx)$	P
	$\exists x\exists yFyx$	

b.

1.	$\forall y[(Hy \wedge Fy) \rightarrow Gy]$	P
2.	$\forall zFz \wedge \sim\forall xKxb$	P
	$\forall x(Hx \rightarrow Gx)$	

c.

1.	$\forall x\forall y(Gy \rightarrow Rxy)$	P
2.	$\forall x(Hx \rightarrow Gx)$	P
3.	Hb	P
	Rab	

d.

1.	$\forall x\forall y(Rxy \rightarrow \sim Ryx)$	P
2.	Raa	P
	$\exists z\exists ySyz$	

e.

1.	$\sim\forall x(Fx \vee A)$	P
	$\exists x{\sim}Fx$	

E6.31. Each of the following sets up an application of ~I or ~E for SG4 or SG5. Complete the derivations, and *explain* your use of strategies for a contradiction. Hint: Each of the strategies for a contradiction is used at least once.

*a.

1.	$\sim\exists x(Fx \wedge Gx)$	P
2.	Fj	A (g, $\rightarrow$I)
3.	Gj	A (c, $\sim$I)
	$\perp$	
	$\sim Gj$	3-_ $\sim$I
	$Fj \rightarrow \sim Gj$	2-_ $\rightarrow$I
	$\forall x(Fx \rightarrow \sim Gx)$	_ $\forall$I

b.

1.	$\forall x(Fx \rightarrow \forall y \sim Fy)$	P
2.	$\exists x Fx$	A (c, $\sim$I)
	$\perp$	
	$\sim\exists x Fx$	2-_ $\sim$I

c.

1.	$\forall x(Fx \rightarrow \forall y Rxy)$	P
2.	$\sim Rab$	P
3.	Fa	A (c, $\sim$I)
	$\perp$	
	$\sim Fa$	3-_ $\sim$I

d.

1.	$\sim\forall x Fx$	P
2.	$\sim\exists x(\sim Fx \vee A)$	A (c, $\sim$E)
	$\perp$	
	$\exists x(\sim Fx \vee A)$	2-_ $\sim$E

e.

1.	$\exists x(Ax \leftrightarrow \sim Ax)$	A (c, $\sim$I)
	$\perp$	
	$\sim\exists x(Ax \leftrightarrow \sim Ax)$	1-_ $\sim$I

E6.32. Produce derivations to show each of the following. Though no full answers are provided, strategy hints are available for the first problems. If you get the last few on your own, you are doing very well!

*a. $\forall x(\sim Bx \rightarrow \sim Wx), \exists x Wx \vdash_{ND} \exists x Bx$

*b. $\forall x \forall y \forall z Gxyz \vdash_{ND} \forall x \forall y \forall z(Hxyz \rightarrow Gzyx)$

*c. $\forall x[Ax \rightarrow \forall y(\sim Dxy \leftrightarrow Bf^1 f^1 y)], \forall x(Ax \wedge \sim Bx) \vdash_{ND} \forall x Df^1 x f^1 x$

*d. $\forall x(Hx \to \forall yRxyb), \forall x\forall z(Razx \to Sxzz) \vdash_{ND} Ha \to \exists xSxcc$

*e. $\sim\forall x(Fx \wedge Abx) \leftrightarrow \sim\forall xKx, \forall y[\exists x\sim(Fx \wedge Abx) \wedge Ryy] \vdash_{ND} \sim\forall xKx$

*f. $\forall x\forall y(Dxy \to Cxy), \forall x\exists yDxy, \forall x\forall y(Cyx \to Dxy) \vdash_{ND} \exists x\exists y(Cxy \wedge Cyx)$

*g. $\forall x\forall y[(Ry \vee Dx) \to \sim Ky], \forall x\exists y(Ax \to \sim Ky), \exists x(Ax \vee Rx) \vdash_{ND} \exists x\sim Kx$

*h. $\forall y(My \to Ay), \exists x\exists y[(Bx \wedge Mx) \wedge (Ry \wedge Syx)], \exists xAx \to \forall y\forall z(Syz \to Ay)$
$\vdash_{ND} \exists x(Rx \wedge Ax)$

*i. $\forall x\forall y[(Hby \wedge Hxb) \to Hxy], \forall z(Bz \to Hbz), \exists x(Bx \wedge Hxb)$
$\vdash_{ND} \exists z[Bz \wedge \forall y(By \to Hzy)]$

*j. $\forall x\exists yRxy, \forall x\forall y(Rxy \to Ryx) \vdash_{ND} \forall x\exists y(Rxy \wedge Ryx)$

*k. $\forall x((Fx \wedge \sim Kx) \to \exists y[(Fy \wedge Hyx) \wedge \sim Ky]),$
$\forall x[(Fx \wedge \forall y[(Fy \wedge Hyx) \to Ky]) \to Kx] \to Ma \vdash_{ND} Ma$

*l. $\forall x\forall y[(Gx \wedge Gy) \to (Hxy \to Hyx)], \forall x\forall y\forall z([(Gx \wedge Gy) \wedge Gz] \to$
$[(Hxy \wedge Hyz) \to Hxz]) \vdash_{ND} \forall w([Gw \wedge \exists z(Gz \wedge Hwz)] \to Hww)$

*m. $\forall x\forall y[(Ax \wedge By) \to Cxy], \exists y[Ey \wedge \forall w(Hw \to Cyw)], \forall x\forall y\forall z[(Cxy \wedge$
$Cyz) \to Cxz], \forall w(Ew \to Bw) \vdash_{ND} \forall z\forall w[(Az \wedge Hw) \to Czw]$

*n. $\forall x\exists y\forall z(Axyz \vee Bzyx), \sim\exists x\exists y\exists zBzyx \vdash_{ND} \forall x\exists y\forall zAxyz$

*o. $A \to \exists xFx \vdash_{ND} \exists x(A \to Fx)$

*p. $\forall xFx \to A \vdash_{ND} \exists x(Fx \to A)$

q. $\forall x(Fx \to Gx), \forall x\forall y(Rxy \to Syx), \forall x\forall y(Sxy \to Syx)$
$\vdash_{ND} \forall x[\exists y(Fx \wedge Rxy) \to \exists y(Gx \wedge Sxy)]$

r. $\exists y\forall xRxy, \forall x(Fx \to \exists ySyx), \forall x\forall y(Rxy \to \sim Sxy) \vdash_{ND} \exists x\sim Fx$

s. $\exists x\forall y[(Fx \vee Gy) \to \forall z(Hxy \to Hyz)], \exists z\forall x\sim Hxz \vdash_{ND} \exists y\forall x(Fy \to \sim Hyx)$

t. $\forall x\forall y[\exists zHyz \to Hxy] \vdash_{ND} \exists x\exists yHxy \to \forall x\forall yHxy$

u. $\exists x(Fx \wedge \forall y[(Gy \wedge Hy) \to \sim Sxy]), \forall x\forall y([(Fx \wedge Gy) \wedge Jy] \to \sim Sxy),$
$\forall x\forall y([(Fx \wedge Gy) \wedge Rxy] \to Sxy), \exists x(Gx \wedge (Jx \vee Hx))$
$\vdash_{ND} \exists x\exists y((Fx \wedge Gy) \wedge \sim Rxy)$

v. $\exists x\forall y[\exists z(Fzy \to \exists wFyw) \to Fxy] \vdash_{ND} \exists xFxx$

w. $\vdash_{ND} \exists x\forall y(Fx \to Fy)$

x. $\vdash_{ND} \exists x(\exists yFy \to Fx)$

y. $\vdash_{ND} \forall x\exists y\forall z[\exists wTxyw \to \exists wTxzw]$

*E6.33. Produce derivations to demonstrate each of the results from T6.31–T6.33, T6.34a,b,e, and T6.35–T6.38. For the first five, for each application of a quantifier rule explain how its restrictions are met. Challenge: finish the results of T6.34.

6.3.4 =I and =E

We complete the system *ND* with I- and E-rules for equality. Strictly, $=$ is not an operator; it is a two-place relation symbol. However, because its interpretation is standardized across all interpretations, it is possible to introduce rules for its behavior.

The =I rule is particularly simple. At any stage in a derivation, for any term t, one may write down $t = t$ with justification =I.

=I

$$\begin{array}{l|ll} & t = t & =\text{I} \end{array}$$

Strictly, without any inputs, this is an *axiom schema* of the sort we encountered in Chapter 3—a form whose instances may be asserted at any stage in a derivation. Motivation should be clear. Since for any m in the universe U, $\langle \mathsf{m}, \mathsf{m} \rangle$ is in the interpretation of $=$, $t = t$ is sure to be satisfied, no matter what the assignment to t might be. Thus, in $\mathscr{L}_q$, $a = a$, $x = x$, and $f^2az = f^2az$ are formulas that might be justified by =I.

=E is more interesting and, in practice, more useful. Say an arbitrary term is *free* in a formula iff every variable in it is free. And say $\mathscr{P}^t/_{\mathscr{s}}$ is $\mathscr{P}$ where some, but not necessarily all, free instances of term t are replaced by term $\mathscr{s}$. Then, given an accessible formula $\mathscr{P}$ on line a and the atomic formula $t = \mathscr{s}$ or $\mathscr{s} = t$ on accessible line b, one may move to $\mathscr{P}^t/_{\mathscr{s}}$ where $\mathscr{s}$ is free for all the replaced instances of t in $\mathscr{P}$, with justification *a,b* =E.

=E

$$\begin{array}{r|ll} \text{a.} & \mathscr{P} & \\ \text{b.} & t = \mathscr{s} & \\ & & \\ & \mathscr{P}^t/_{\mathscr{s}} & \text{a,b =E} \end{array} \qquad \begin{array}{r|ll} \text{a.} & \mathscr{P} & \\ \text{b.} & \mathscr{s} = t & \\ & & \\ & \mathscr{P}^t/_{\mathscr{s}} & \text{a,b =E} \end{array}$$

provided that term $\mathscr{s}$ is free for all the replaced instances of term t in formula $\mathscr{P}$

If the assignment to some terms is the same, this rule lets us replace free instances of the one term by the other in any formula. Again, the motivation should be clear. On trees, the only thing that matters about a term is the thing to which it refers. So if $\mathscr{P}$ with term t is satisfied, and the assignment to t is the same as the assignment to $\mathscr{s}$, then $\mathscr{P}$ with $\mathscr{s}$ in place of t should be satisfied as well. When a term is not free, it is not the assignment to the term that is doing the work, but rather the way it is bound. So we restrict ourselves to contexts where it is just the assignment that matters!

Because we need not replace all free instances of one term with the other, this rule has some special applications that are worth noticing. Consider the formulas $Raba$ and $a = b$. The following lists all the formulas that could be derived from them in one step by =E.

(CF)

$$
\begin{array}{rl|ll}
1. & Raba & \text{P} \\
2. & a = b & \text{P} \\
\hline
3. & Rbba & 1,2 =\text{E} \\
4. & Rabb & 1,2 =\text{E} \\
5. & Rbbb & 1,2 =\text{E} \\
6. & Raaa & 1,2 =\text{E} \\
7. & a = a & 2,2 =\text{E} \\
8. & b = b & 2,2 =\text{E}
\end{array}
$$

(3) and (4) replace one instance of a with b. (5) replaces both instances of a with b. (6) replaces the instance of b with a. We could reach, say, $Raab$, but this would require another step—which we could take from any of (4), (5), or (6). You should be clear about why this is so. (7) and (8) are different. We have a formula $a = b$, and an equality $a = b$. In (7) we use the equality to replace one instance of b in the formula with a. In (8) we use the equality to replace one instance of a in the formula with b. Of course (7) and (8) might equally have been derived by =I. Notice also that =E is not restricted to atomic formulas, or to simple terms. Thus, for example,

(CG)

$$
\begin{array}{rl|ll}
1. & \forall y(Rag^1x \wedge Kf^2f^2azy) & \text{P} \\
2. & g^1x = f^2az & \text{P} \\
\hline
3. & \forall y(Raf^2az \wedge Kf^2f^2azy) & 1,2 =\text{E} \\
4. & \forall y(Rag^1x \wedge Kf^2g^1xy) & 1,2 =\text{E}
\end{array}
$$

ND Quick Reference

ND includes all the rules of *NDs* and,

$\forall$E (*universal exploit*)

$$
\begin{array}{l|ll}
\text{a.} & \forall x\mathcal{P} & \\
& \mathcal{P}^x_t & \text{a } \forall\text{E}
\end{array}
$$

$\exists$I (*existential intro*)

$$
\begin{array}{l|ll}
\text{a.} & \mathcal{P}^x_t & \\
& \exists x\mathcal{P} & \text{a } \exists\text{I}
\end{array}
$$

provided t is free for x in $\mathcal{P}$

$\forall$I (*universal intro*)

$$
\begin{array}{l|ll}
\text{a.} & \mathcal{P}^x_v & \\
& \forall x\mathcal{P} & \text{a } \forall\text{I}
\end{array}
$$

$\exists$E (*existential exploit*)

$$
\begin{array}{l|ll}
\text{a.} & \exists x\mathcal{P} & \\
\text{b.} & \quad \mathcal{P}^x_v & \text{A } (g, \text{a}\exists\text{E}) \\
\text{c.} & \quad \mathcal{Q} & \\
& \mathcal{Q} & \text{a,b-c } \exists\text{E}
\end{array}
$$

provided (i) v is free for x in $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathcal{P}/\exists x\mathcal{P}$ or in $\mathcal{Q}$

=I (*equality intro*)

$$
\begin{array}{|ll}
t = t & =\text{I}
\end{array}
$$

=E (*equality exploit*)

$$
\begin{array}{l|l}
\text{a} & \mathcal{P} \\
\text{b} & t = s \\
& \mathcal{P}^t/_s
\end{array}
\qquad
\begin{array}{|ll}
\mathcal{P} & \\
s = t & \\
\mathcal{P}^t/_s & \text{a,b } =\text{E}
\end{array}
$$

provided that term s is free for all the replaced instances of term t in formula $\mathcal{P}$

lists steps that are legitimate applications of =E to (1) and (2). If the second premise were $g^1x = f^2ay$, however, we could not use it with (1) to reach say, $\forall y(Raf^2ay \wedge Kf^2f^2azy)$, since f^2ay is not free for g^1x in $\forall y(Rag^1x \wedge Kf^2f^2azy)$. And of course, we could not replace either y or f^2f^2azy in $\forall y(Rag^1x \wedge Kf^2f^2azy)$ since they are not free.

There is not much new to say about strategy, except that you should include =E among the stock of rules you use to identify what is "contained" in accessible lines. It may be that a goal is contained in accessible lines, when terms only need to be switched by some equality. Thus, for goal Fa, with Fb explicitly available, it might be worth setting $a = b$ as a subgoal, with the intent of using the equality to switch the terms.

Rather than dwell on strategy as such, let us proceed directly to a few substantive applications. First, you should find derivation of the following theorems straightforward. Thus, for example, T6.39 and T6.42 take just one step (and none require more than five lines). The first three may remind you of axioms from Chapter 3. The others represent important features of equality.

T6.39. $\vdash_{ND}\ x = x$

*T6.40. $\vdash_{ND}\ (x_i = y) \rightarrow (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$

T6.41. $\vdash_{ND}\ (x_i = y) \rightarrow (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \rightarrow \mathcal{R}^n x_1 \ldots y \ldots x_n)$

T6.42. $\vdash_{ND}\ t = t$ *reflexivity of equality*

T6.43. $\vdash_{ND}\ (t = s) \rightarrow (s = t)$ *symmetry of equality*

T6.44. $\vdash_{ND}\ (r = s) \rightarrow [(s = t) \rightarrow (r = t)]$ *transitivity of equality*

Note that with symmetry, given $s = t$ it follows that $t = s$. So that reasoning goes both ways and $\vdash_{ND}\ t = s \leftrightarrow s = t$.

Here is reasoning of a frequently-encountered type. Suppose we want to show that the following is valid in *ND:*

	$\exists x[(Dx \wedge \forall y(Dy \rightarrow x = y)) \wedge Bx]$	The dog is barking
(CH)	$\exists x(Dx \wedge Cx)$	Some dog is chasing a cat
	$\exists x[Dx \wedge (Bx \wedge Cx)]$	Some dog is barking and chasing a cat

Using the methods of Chapter 5, this might translate something like the argument on the right. We set out to do the derivation in the usual way.

1.	$\exists x[(Dx \wedge \forall y(Dy \rightarrow x = y)) \wedge Bx]$	P
2.	$\exists x(Dx \wedge Cx)$	P
3.	$(Dj \wedge \forall y(Dy \rightarrow j = y)) \wedge Bj$	A (g, 1$\exists$E)
4.	$Dk \wedge Ck$	A (g, 2$\exists$E)
	$Dj \wedge (Bj \wedge Cj)$	
	$\exists x[Dx \wedge (Bx \wedge Cx)]$	_ $\exists$I
	$\exists x[Dx \wedge (Bx \wedge Cx)]$	2,4-_ $\exists$E
	$\exists x[Dx \wedge (Bx \wedge Cx)]$	1,3-_ $\exists$E

Given two existentials in the premises, we set up to get the goal by two applications of $\exists$E. And if we had $Dj \wedge (Bj \wedge Cj)$ we could reach the goal by $\exists$I. Dj and Bj are easy to get from (3). But we do not have Cj. What we have is rather Ck. The existentials in the assumptions are instantiated to different (new) variables—and they *must* be so instantiated if we are to meet the constraints on $\exists$E. From $\exists x\mathcal{P}$ and $\exists x\mathcal{Q}$ it does not follow that any one thing is both $\mathcal{P}$ and $\mathcal{Q}$. In this case, however, we are given that there is just one dog. And we can use this to force an equivalence between j and k. Then we get the result by $=$E.

1.	$\exists x[(Dx \wedge \forall y(Dy \rightarrow x = y)) \wedge Bx]$	P
2.	$\exists x(Dx \wedge Cx)$	P
3.	$(Dj \wedge \forall y(Dy \rightarrow j = y)) \wedge Bj$	A (g, 1$\exists$E)
4.	$Dk \wedge Ck$	A (g, 2$\exists$E)
5.	Bj	3 $\wedge$E
6.	$Dj \wedge \forall y(Dy \rightarrow j = y)$	3 $\wedge$E
7.	Dj	6 $\wedge$E
8.	$\forall y(Dy \rightarrow j = y)$	6 $\wedge$E
9.	$Dk \rightarrow j = k$	8 $\forall$E
10.	Dk	4 $\wedge$E
11.	$j = k$	9,10 $\rightarrow$E
12.	Ck	4 $\wedge$E
13.	Cj	12,11 $=$E
14.	$Bj \wedge Cj$	5,13 $\wedge$I
15.	$Dj \wedge (Bj \wedge Cj)$	7,14 $\wedge$I
16.	$\exists x[Dx \wedge (Bx \wedge Cx)]$	15 $\exists$I
17.	$\exists x[Dx \wedge (Bx \wedge Cx)]$	2,4-16 $\exists$E
18.	$\exists x[Dx \wedge (Bx \wedge Cx)]$	1,3-17 $\exists$E

Though there are a few steps, the work to get it done is simple. This is a very common pattern: Arbitrary individuals are introduced as if they were distinct. But uniqueness clauses let us establish an identity between them. Given this, facts about the one transfer to the other by $=$E.

*E6.34. Produce derivations to show T6.39–T6.44. Hint: It may help to begin with concrete versions of the theorems and then move to the general case. Thus, for example, for T6.40, show that $\vdash_{ND} (y = j) \to (g^3xyz = g^3xjz)$. Then you will be able to show the general case.

E6.35. Produce derivations to show each of the following.

*a. $\vdash_{ND} \forall x\exists y(x = y)$

b. $\vdash_{ND} \forall x\exists y(f^1x = y)$

c. $\vdash_{ND} \forall x\forall y[(Fx \wedge \sim Fy) \to \sim(x = y)]$

d. $\forall x(Rxa \to x = c), \forall x(Rxb \to x = d), \exists x(Rxa \wedge Rxb) \vdash_{ND} c = d$

e. $\vdash_{ND} \forall x[\sim(f^1x = x) \to \forall y((f^1x = y) \to \sim(x = y))]$

f. $\vdash_{ND} \forall x\forall y[(f^1x = y \wedge f^1y = x) \to f^1f^1x = x]$

*g. $\exists x\exists yHxy, \forall y\forall z(Dyz \leftrightarrow Hzy), \forall x\forall y(\sim Hxy \vee x = y)$
$\vdash_{ND} \exists x(Hxx \wedge Dxx)$

h. $\forall x\forall y[(Rxy \wedge Ryx) \to x = y], \forall x\forall y(Rxy \to Ryx)$
$\vdash_{ND} \forall x[\exists y(Rxy \vee Ryx) \to Rxx]$

i. $\exists x\forall y(x = y \leftrightarrow Fy), \forall x(Gx \to Fx) \vdash_{ND} \forall x\forall y[(Gx \wedge Gy) \to x = y]$

j. $\forall x[Fx \to \exists y(Gyx \wedge \sim Gxy)], \forall x\forall y[(Fx \wedge Fy) \to x = y]$
$\vdash_{ND} \forall x(Fx \to \exists y\sim Fy)$

6.3.5 The System *ND+*

We conclude this section with some final derived rules. Again, it is not possible to derive anything with the extra rules that cannot already be derived in *ND*. Thus the new rules do not add extra derivation power. They are rather "shortcuts" for things that can already be done in *ND*. The full system *ND+* includes all the rules of *ND*, all the derived rules of *NDs+*, and some additional derived rules.

First, Sym (*symmetry*) reverses the order of terms in an equality.

Sym $\qquad t = s \lhd\rhd s = t$

This form is justified by T6.43 which, although it is not a biconditional, is symmetrical (!) so that given one equality we can reverse the terms to obtain the other.

Next some quantifier rules. First, QS (*quantifier switch*) switches the order of a pair of universal quantifiers, or of a pair of existential quantifiers.

QS $\qquad \forall x\forall y\mathcal{P} \lhd\rhd \forall y\forall x\mathcal{P} \qquad\qquad \exists x\exists y\mathcal{P} \lhd\rhd \exists y\exists x\mathcal{P}$

These forms are justified by T6.32 and T6.33. Notice that switching applies only where quantifiers are the same.

Then QD (*quantifier distribution*) distributes the universal quantifier over $\wedge$, and the existential over $\vee$.

QD $\quad \forall x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \forall x\mathcal{P} \wedge \forall x\mathcal{Q} \qquad \exists x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \exists x\mathcal{P} \vee \exists x\mathcal{Q}$

These forms are justified by T6.35 and T6.36. Observe that distribution does not work for $\forall x$ over $\vee$, or $\exists x$ over $\wedge$.

Next, QP (*quantifier placement*) collectes a series of principles like ones we saw in Chapter 5. Where x is not free in $\mathcal{P}$,

QP

$$\begin{array}{ll}
\forall x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \wedge \forall x\mathcal{Q} & \exists x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \wedge \exists x\mathcal{Q} \\
\forall x(\mathcal{Q} \wedge \mathcal{P}) \;\triangleleft\triangleright\; \forall x\mathcal{Q} \wedge \mathcal{P} & \exists x(\mathcal{Q} \wedge \mathcal{P}) \;\triangleleft\triangleright\; \exists x\mathcal{Q} \wedge \mathcal{P} \\
\\
\forall x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \vee \forall x\mathcal{Q} & \exists x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \vee \exists x\mathcal{Q} \\
\forall x(\mathcal{Q} \vee \mathcal{P}) \;\triangleleft\triangleright\; \forall x\mathcal{Q} \vee \mathcal{P} & \exists x(\mathcal{Q} \vee \mathcal{P}) \;\triangleleft\triangleright\; \exists x\mathcal{Q} \vee \mathcal{P} \\
\\
\forall x(\mathcal{P} \to \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \to \forall x\mathcal{Q} & \exists x(\mathcal{P} \to \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \to \exists x\mathcal{Q} \\
\forall x(\mathcal{Q} \to \mathcal{P}) \;\triangleleft\triangleright\; \exists x\mathcal{Q} \to \mathcal{P} & \exists x(\mathcal{Q} \to \mathcal{P}) \;\triangleleft\triangleright\; \forall x\mathcal{Q} \to \mathcal{P}
\end{array}$$

Notice the quantifier flip in the bottom line. These principles are justified by the results of T6.34.

In practice, QS, QD, and QP do not apply all that frequently—still it is good to recognize when expressions are equivalent but for the order and placement of quantifiers. Much more common is a very useful replacement rule:

QN

$$\begin{array}{l}
\sim\forall x\mathcal{P} \;\triangleleft\triangleright\; \exists x{\sim}\mathcal{P} \\
\sim\exists x\mathcal{P} \;\triangleleft\triangleright\; \forall x{\sim}\mathcal{P}
\end{array}$$

QN (*quantifier negation*) is another principle we encountered in Chapter 5. It lets you push or pull a negation across a quantifier, with a corresponding flip from one quantifier to the other. The forms are justified by T6.37 and T6.38.

Again, with DeM, Impl, and Equiv, QN lets you "push" a main operator $\sim$ to the inside of a formula. This can be especially useful. So, for example, given a negated universal on some accessible line, you can go directly to the (high priority) strategies SG2 or SC2: Push the negation through, get the existential, and go for the goal by $\exists$E as usual. Here is an example:

(CI)

$$\begin{array}{rll}
1. & \mid \sim\forall x(Fx \to Gx) & \text{P} \\
2. & \mid \exists x{\sim}(Fx \to Gx) & \text{1 QN} \\
3. & \mid\;\mid \sim(Fj \to Gj) & \text{A } (g, 2\exists\text{E}) \\
4. & \mid\;\mid \sim(\sim Fj \vee Gj) & \text{3 Impl} \\
5. & \mid\;\mid \sim\sim Fj \wedge \sim Gj & \text{4 DeM} \\
6. & \mid\;\mid \sim Gj & 5\ \wedge\text{E} \\
7. & \mid\;\mid \exists x{\sim}Gx & 6\ \exists\text{I} \\
8. & \mid \exists x{\sim}Gx & \text{2,3-7 }\exists\text{E}
\end{array}$$

$$\begin{array}{rll}
1. & \mid \sim\forall x(Fx \to Gx) & \text{P} \\
2. & \mid\;\mid \sim\exists x{\sim}Gx & \text{A } (c, \sim\text{E}) \\
3. & \mid\;\mid\;\mid Fj & \text{A } (g, \to\text{I}) \\
4. & \mid\;\mid\;\mid\;\mid \sim Gj & \text{A } (c, \sim\text{E}) \\
5. & \mid\;\mid\;\mid\;\mid \exists x{\sim}Gx & 4\ \exists\text{I} \\
6. & \mid\;\mid\;\mid\;\mid \bot & 5,2\ \bot\text{I} \\
7. & \mid\;\mid\;\mid Gj & \text{4-6 }\sim\text{E} \\
8. & \mid\;\mid Fj \to Gj & \text{3-7 }\to\text{I} \\
9. & \mid\;\mid \forall x(Fx \to Gx) & 8\ \forall\text{I} \\
10. & \mid\;\mid \bot & 9,1\ \bot\text{I} \\
11. & \mid \exists x{\sim}Gx & \text{2-10 }\sim\text{E}
\end{array}$$

The derivation on the left is much to be preferred over the one on the right, where we are caught up in a difficult case of SG5 and then SC3. But, after QN, the derivation on the left is straightforward—and would be relatively straightforward even if we missed the uses of Impl and DeM.

The rest of the rules for *ND+* apply to a species of *restricted* quantifier. In Chapter 5 we emphasized that the universal quantifier typically applies to expressions with main operator $\rightarrow$ and the existential to ones with $\wedge$. We can streamline operations on these expressions as follows. Take,

RQ $\quad (\forall x : \mathcal{B})\mathcal{P}$ abbreviates $\forall x(\mathcal{B} \rightarrow \mathcal{P})$

$\quad\quad (\exists x : \mathcal{B})\mathcal{P}$ abbreviates $\exists x(\mathcal{B} \wedge \mathcal{P})$

Read: 'for all x such that $\mathcal{B}$, $\mathcal{P}$' and 'for some x such that $\mathcal{B}$, $\mathcal{P}$'. In these expressions $\mathcal{B}$ restricts the range of things to which the quantifier applies. Important instances, encountered in the next section and especially in Part IV, are the *bounded* quantifiers as, $(\forall x : x < t)\mathcal{P}$ and $(\exists x : x < t)\mathcal{P}$ where x does not appear in t. These are usually compressed to $(\forall x < t)\mathcal{P}$ and $(\exists x < t)\mathcal{P}$. In these cases, $\mathcal{B}$ is $x < t$. For such expressions, we have natural I- and E-rules along with a replacement rule.

First the I- and E-rules for bounded quantifiers (∀I), (∀E), (∃I), (∃E), streamline what you can do with the unabbreviated forms.

(∀E)

a. $(\forall x : \mathcal{B})\mathcal{P}$
b. $\mathcal{B}^x_t$

$\mathcal{P}^x_t$

(∃I)

a. $\mathcal{P}^x_t$
b. $\mathcal{B}^x_t$

$(\exists x : \mathcal{B})\mathcal{P}$

provided t is free for x in $\mathcal{B}$ and $\mathcal{P}$

(∀I)

a. $\mathcal{B}^x_v$

$\mathcal{P}^x_v$

$(\forall x : \mathcal{B})\mathcal{P}$

(∃E)

a. $(\exists x : \mathcal{B})\mathcal{P}$
b. $\mathcal{P}^x_v$
c. $\mathcal{B}^x_v$

$\mathcal{Q}$

$\mathcal{Q}$

provided (i) v is free for x in $\mathcal{B}$ and $\mathcal{P}$, (ii) v is not free in any undischarged assumption, and (iii) v is not free in the quantified expression or $\mathcal{Q}$

For convenience, the assumption for (∃E) occupies two lines. Formal demonstration that these are derived rules in *ND* is left to Chapter 9. However, each is intuitive: In (∀E), the unabbreviated premises are $\forall x(\mathcal{B} \rightarrow \mathcal{P})$ and $\mathcal{B}^x_t$; then ∀E and →E give $\mathcal{P}^x_t$. In (∃I), the premises with ∧I and ∃I yield the unabbreviated conclusion $\exists x(\mathcal{B} \wedge \mathcal{P})$. For (∀I), given the subderivation, →I and ∀I yield the unabbreviated form. And for (∃E) the unabbreviated premise with the subderivation and ∃E yield $\mathcal{Q}$ (treating $\mathcal{P}^x_v$ and $\mathcal{B}^x_v$ as $(\mathcal{B} \wedge \mathcal{P})^x_v$).

Here is the replacement rule:

RQN $\quad \sim(\forall x : \mathcal{B})\mathcal{P} \;\lhd\rhd\; (\exists x : \mathcal{B})\sim\mathcal{P}$

$\quad\quad \sim(\exists x : \mathcal{B})\mathcal{P} \;\lhd\rhd\; (\forall x : \mathcal{B})\sim\mathcal{P}$

RQN (*restricted quantifier negation*) works by analogy with QN. Its demonstration requires a new theorem.

T6.45. The following are theorems of *ND*.

*(a) $\vdash_{ND} \sim(\forall x : \mathcal{B})\mathcal{P} \leftrightarrow (\exists x : \mathcal{B})\sim\mathcal{P}$

(b) $\vdash_{ND} \sim(\exists x : \mathcal{B})\mathcal{P} \leftrightarrow (\forall x : \mathcal{B})\sim\mathcal{P}$

Demonstration of this result is left to E6.37.

E6.36. Produce derivations to show each of the following. Work the last two with all the rules of *ND+* and then again but without quantifier placement rules. Hint: The latter are quite challenging!

a. $\sim(\exists x : \sim Rx)Sxx, Saa \vdash_{ND+} Ra$

b. $\forall x(\sim Axf^1x \vee \exists yBg^1y) \vdash_{ND+} \exists xAf^1xf^1f^1x \to \exists yBg^1y$

c. $(\forall x : \sim Cxb \vee Hx)Lxx, \exists y\sim Lyy \vdash_{ND+} \exists xCxb$

d. $\forall xFx, \forall zHz \vdash_{ND+} \sim\exists y(\sim Fy \vee \sim Hy)$

e. $\sim\exists x\forall y(Pxy \wedge \sim Qxy) \vdash_{ND+} \forall x\exists y(Pxy \to Qxy)$

f. $\exists y[(\forall xFx \to Ay) \vee (Ay \to \exists xFx)] \vdash_{ND+} \exists x\exists y[(Fx \to Ay) \vee (Ay \to Fx)]$

*g. $\sim(\exists x : Fx)Gx \vee \exists x\sim Gx, \forall yGy \vdash_{ND+} (\forall z : Fz)\sim Gz$

*h. $\forall x\forall y\exists zAf^1xyz, \forall x\forall y\forall z[Axyz \to \sim(Cxyz \vee Bzyx)]$
$\vdash_{ND+} \forall x\forall y\sim\forall zBzg^1yf^1g^1x$

i. $\sim\exists y(Ty \vee \exists x\sim Hxy) \vdash_{ND+} \forall x\forall yHxy \wedge \forall x\sim Tx$

j. $\exists x(Fx \to \exists y\sim Fy) \vdash_{ND+} \sim\forall xFx$

k. $\vdash_{ND+} (\forall x : Ax)Bx \vee \exists xAx$

l. $\forall x\forall y(Fx \leftrightarrow Gy) \vdash_{ND+} (\exists xFx \to \forall yGy) \wedge (\exists yGy \to \forall xFx)$

m. $\exists x(Fx \leftrightarrow Gx), (\forall x : Gx)(Hx \to Jx) \vdash_{ND+} \exists xJx \vee [\forall xFx \to (\exists x : Gx)\sim Hx]$

n. $(\exists x : \sim Bxa)\forall y(Cy \to \sim Gxy), \forall z[\sim\forall y(Wy \to Gzy) \to Bza]$
$\vdash_{ND+} (\forall x : Cx)\sim Wx$

*o. $\exists xFx \to \sim\forall yGy, \forall x(Kx \to \exists yJy), \exists y\sim Gy \to \exists xKx$
$\vdash_{ND+} \sim\exists xFx \vee \exists yJy$

p. $\exists zQz \to (\forall w : Lww)\sim Hw, \exists xBx \to (\forall y : Ay)Hy$
$\vdash_{ND+} (\exists w : Qw)Bw \to (\forall y : Lyy)\sim Ay$

q. $\sim\forall x(\sim Px \vee \sim Hx) \to \forall x[Cx \wedge (\forall y : Ly)Axy], (\exists x : Hx)(\forall y : Ly)Axy \to \forall x(Rx \wedge \forall yBxy) \vdash_{ND+} \sim\forall x\forall yBxy \to \forall x(\sim Px \vee \sim Hx)$

r. $\vdash_{ND+} (\exists xAx \to \exists xBx) \to \exists x(Ax \to Bx)$

ND+ Quick Reference

ND+ includes all the rules of *ND*, all the derived rules of *NDs+*, and,

Inference Rules:

($\forall$E) *restricted univ exploit*

$$\begin{array}{l|ll}
\text{a.} & (\forall x : \mathcal{B})\mathcal{P} & \\
\text{b.} & \mathcal{B}^x_t & \\
 & & \\
 & \mathcal{P}^x_t & \text{a,b } (\forall\text{E})
\end{array}$$

($\exists$I) *restricted exist intro*

$$\begin{array}{l|ll}
\text{a.} & \mathcal{P}^x_t & \\
\text{b.} & \mathcal{B}^x_t & \\
 & & \\
 & (\exists x : \mathcal{B})\mathcal{P} & \text{a,b } (\exists\text{I})
\end{array}$$

provided t is free for x in $\mathcal{B}$ and $\mathcal{P}$

($\forall$I) *restricted univ intro*

$$\begin{array}{l|l|ll}
\text{a.} & & \mathcal{B}^x_v & \text{A } (g, (\forall\text{I})) \\
\cline{3-3}
 & & & \\
\text{b.} & & \mathcal{P}^x_v & \\
 & \multicolumn{2}{l}{(\forall x : \mathcal{B})\mathcal{P}} & \text{a-b } (\forall\text{I})
\end{array}$$

($\exists$E) *restricted exist exploit*

$$\begin{array}{l|l|ll}
\text{a.} & \multicolumn{2}{l}{(\exists x : \mathcal{B})\mathcal{P}} & \\
\text{b.} & & \mathcal{P}^x_v & \text{A } (g, \text{a}(\exists\text{E})) \\
\text{c.} & & \mathcal{B}^x_v & \\
\cline{3-3}
 & & & \\
\text{d.} & & \mathcal{Q} & \\
 & \multicolumn{2}{l}{\mathcal{Q}} & \text{a,b-d } (\exists\text{E})
\end{array}$$

provided (i) v is free for x in $\mathcal{B}$ and $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $(\forall x : \mathcal{B})\mathcal{P}$ or in $(\exists x : \mathcal{B})\mathcal{P}$ or in $\mathcal{Q}$

Replacement Rules:

Sym	$s = t \;\triangleleft\triangleright\; t = s$	
QS	$\forall x \forall y \mathcal{P} \;\triangleleft\triangleright\; \forall y \forall x \mathcal{P}$	$\exists x \exists y \mathcal{P} \;\triangleleft\triangleright\; \exists y \exists x \mathcal{P}$
QD	$\forall x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \forall x \mathcal{P} \wedge \forall x \mathcal{Q}$	$\exists x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \exists x \mathcal{P} \vee \exists x \mathcal{Q}$
QP*	$\forall x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \wedge \forall x \mathcal{Q}$	$\exists x(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \wedge \exists x \mathcal{Q}$
	$\forall x(\mathcal{Q} \wedge \mathcal{P}) \;\triangleleft\triangleright\; \forall x \mathcal{Q} \wedge \mathcal{P}$	$\exists x(\mathcal{Q} \wedge \mathcal{P}) \;\triangleleft\triangleright\; \exists x \mathcal{Q} \wedge \mathcal{P}$
	$\forall x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \vee \forall x \mathcal{Q}$	$\exists x(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \vee \exists x \mathcal{Q}$
	$\forall x(\mathcal{Q} \vee \mathcal{P}) \;\triangleleft\triangleright\; \forall x \mathcal{Q} \vee \mathcal{P}$	$\exists x(\mathcal{Q} \vee \mathcal{P}) \;\triangleleft\triangleright\; \exists x \mathcal{Q} \vee \mathcal{P}$
	$\forall x(\mathcal{P} \to \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \to \forall x \mathcal{Q}$	$\exists x(\mathcal{P} \to \mathcal{Q}) \;\triangleleft\triangleright\; \mathcal{P} \to \exists x \mathcal{Q}$
	$\forall x(\mathcal{Q} \to \mathcal{P}) \;\triangleleft\triangleright\; \exists x \mathcal{Q} \to \mathcal{P}$	$\exists x(\mathcal{Q} \to \mathcal{P}) \;\triangleleft\triangleright\; \forall x \mathcal{Q} \to \mathcal{P}$
QN	$\sim\forall x \mathcal{P} \;\triangleleft\triangleright\; \exists x \sim\mathcal{P}$	
	$\sim\exists x \mathcal{P} \;\triangleleft\triangleright\; \forall x \sim\mathcal{P}$	
RQN	$\sim(\forall x : \mathcal{B})\mathcal{P} \;\triangleleft\triangleright\; (\exists x : \mathcal{B})\sim\mathcal{P}$	
	$\sim(\exists x : \mathcal{B})\mathcal{P} \;\triangleleft\triangleright\; (\forall x : \mathcal{B})\sim\mathcal{P}$	

*The replacement rule QP requires x not free in $\mathcal{P}$.

s. $\forall x \exists y(Ax \vee By) \vdash_{ND+} \exists y \forall x(Ax \vee By)$

t. $\forall x Fx \leftrightarrow \sim\exists x \exists y Rxy \vdash_{ND+} \exists x \forall y \forall z(Fx \to \sim Ryz)$

*E6.37. (i) Using rules of *ND*, prove unabbreviated versions for both parts of T6.45. (ii) Using I- and E-rules for the restricted quantifiers show the same, but without unabbreviation. Hint: using notation as for E6.33, apply quantifier rules without variable exchange.

E6.38. For each of the following, produce a translation into $\mathcal{L}_q$, including interpretation function and formal sentences, and show that the resulting arguments are valid in *ND+*.

a. If a first person is taller than a second, then the second is not taller than the first. So nobody is taller than themselves. (An asymmetric relation is irreflexive.)

b. A barber shaves all and only people who do not shave themselves. So there are no barbers.

c. Alice is taller than every other woman. If a first person is taller than a second, then the second is not taller than the first. So only Alice is taller than every other woman.

d. There is at most one dog, and at least one flea. Each flea has a dog for a host, and any dog hosts at most one flea. So there is exactly one flea.

e. Something is divine just in case nothing is conceived to be greater than it. Some (conceivable) object is divine. If something is divine but not real, then something is divine but conceived to be real. If one thing is divine and conceived to be real, and another is divine but not real, then the first is conceived to be greater than the second. So something is both divine and real.

   Hint: Let quantifiers range over *objects of conception* and so set $\mathsf{U} = \{\mathsf{o} \mid \mathsf{o}$ is conceivable$\}$. This, of course, is a version of Anselm's Ontological Argument according to which god is 'a being than which none greater can be conceived'. This version is simplified from Robinson, "A New Formalization of Anselm's Ontological Argument." For a good introductory discussion and alternate account, see Plantinga, *God, Freedom, and Evil*.

6.4 Applications: Q and PA

A very important application, one with which we will be extensively concerned later in the text, is to arithmetic. We encountered Peano Arithmetic in Chapter 3. We now consider a pair of theories, Robinson Arithmetic (Q) and then Peano Arithmetic (PA) once again.

For this, $\mathcal{L}_{NT}$ is like $\mathcal{L}_{NT}^{<}$ from section 2.3.5 but without $<$. As described in the language of arithmetic reference on the next page, there is the constant symbol $\emptyset$, the one-place function symbol S, two-place function symbols $+$, and $\times$, and the relation symbol $=$. We will find it convenient to let the variables be any of $a \dots z$ with or without positive integer subscripts. Let $s \leq t$ abbreviate $\exists u(u + s = t)$, and $s < t$ abbreviate $\exists u(Su + s = t)$ where u is some variable that does not appear in s or t. We also encounter restricted (bounded) quantifiers in the forms $(\forall x \leq t)\mathcal{P}$, $(\exists x \leq t)\mathcal{P}$, $(\forall x < t)\mathcal{P}$, and $(\exists x < t)\mathcal{P}$ where x does not occur in t (so t is independent of that which it bounds).

In derivations, we allow movement between these abbreviations and their unabbreviated forms with justification 'abv'. For the bounded quantifiers derived introduction and exploitation rules appear in the forms,

(∀E)

a. $(\forall x < t)\mathcal{P}$
b. $s < t$

$\mathcal{P}^{x}_{s}$

(∃I)

a. $\mathcal{P}^{x}_{s}$
b. $s < t$

$(\exists x < t)\mathcal{P}$

provided s is free for x in $\mathcal{P}$

(∀I)

a. | $v < t$

| $\mathcal{P}^{x}_{v}$

$(\forall x < t)\mathcal{P}$

(∃E)

a. $(\exists x < t)\mathcal{P}$
b. | $\mathcal{P}^{x}_{v}$
c. | $v < t$

| $\mathcal{Q}$

$\mathcal{Q}$

provided v is free for x in $\mathcal{P}$, not free in any undischarged assumption, and not free in the quantified expression or $\mathcal{Q}$

And similarly with '$\leq$' uniformly substituted for '$<$'. Insofar as any term is free for x in the inequalities $x \leq t$ and $x < t$, constraints are simplified somewhat relative to the formulation of section 6.3.5.[7]

$\mathcal{L}_{NT}$ has a standard interpretation N just like $\bar{\mathsf{N}}$ for $\mathcal{L}_{NT}^{<}$, but without the assignment to $<$. So the universe is the set $\mathbb{N}$ of natural numbers, $\emptyset$ is assigned zero, S the successor function, $+$ the addition function, $\times$ the multiplication function, and $=$ the equality relation. Officially, derivations are perfectly well defined apart from this or any other interpretation. All the same, the standard interpretation *motivates* axioms and results of Robinson and Peano arithmetic to follow.

6.4.1 Robinson Arithmetic, Q

Robinson arithmetic is a minimal theory of arithmetic just strong enough to support Gödel's incompleteness theorem from Part IV. We will say that a formula $\mathcal{P}$ is an

[7]Actually, not every term s is free for x in $\exists u(u + x = t)$ and $\exists u(Su + x = t)$; however for any s, by exchange of the bound variable, these expressions are equivalent to ones that have s free for x. In a given context, it is simplest to suppose u is some one variable (maybe z_{75}) not appearing in other terms.

$\mathcal{L}_{NT}$ Quick Reference

Vocabulary:

variables: $a \dots z$ with or without positive integer subscripts

constant: $\emptyset$

one-place function symbol: S

two-place function symbols: $+$, $\times$

relation symbol: $=$

Abbreviations:

where u does not appear in s or t,

$s \le t$ abbreviates $\exists u(u + s = t)$

$s < t$ abbreviates $\exists u(Su + s = t)$

and where x does not appear in t,

$(\forall x \le t)\mathcal{P}$ abbreviates $(\forall x : x \le t)\mathcal{P}$ which is $\forall x(x \le t \to \mathcal{P})$

$(\forall x < t)\mathcal{P}$ abbreviates $(\forall x : x < t)\mathcal{P}$ which is $\forall x(x < t \to \mathcal{P})$

$(\exists x \le t)\mathcal{P}$ abbreviates $(\exists x : x \le t)\mathcal{P}$ which is $\exists x(x \le t \wedge \mathcal{P})$

$(\exists x < t)\mathcal{P}$ abbreviates $(\exists x : x < t)\mathcal{P}$ which is $\exists x(x < t \wedge \mathcal{P})$

From section 6.3.5 (and page 300), the restricted quantifiers have derived introduction and exploitation rules ($\forall$E), ($\forall$I), ($\exists$E), ($\exists$I), and a restricted quantifier negation RQN. In derivations, abv moves between abbreviated and unabbreviated forms.

$\mathcal{L}_{NT}$ has a standard interpretation N with U the set $\mathbb{N}$ of natural numbers and,

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$\mathsf{N}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\mathsf{N}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

On this interpretation we may obtain derived semantic conditions for the inequalities and bounded quantifiers (see T12.3 and T12.4).

ND+ theorem of Robinson Arithmetic just in case $\mathcal{P}$ follows in *ND+* given as premises the following axioms for Robinson Arithmetic:[8]

Q 1. $\sim(Sx = \emptyset)$

2. $(Sx = Sy) \to (x = y)$

3. $(x + \emptyset) = x$

4. $(x + Sy) = S(x + y)$

[8] After R. Robinson, "An Essentially Undecidable Axiom System."

5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $\sim(x = \emptyset) \rightarrow \exists y(x = Sy)$

In the ordinary case we suppress mention of Q1–Q7 as premises, and simply write $\mathsf{Q} \vdash_{ND+} \mathcal{P}$ to indicate that $\mathcal{P}$ is an *ND+* theorem of Robinson arithmetic—that there is an *ND+* derivation of $\mathcal{P}$ which may include appeal to any of Q1–Q7.

The axioms set up a basic version of arithmetic on the natural numbers. On the standard interpretation N, $\emptyset$ is not the successor of any natural number (Q1); if the successor of x is the same as the successor of y, then x is y (Q2); x plus $\emptyset$ is equal to x (Q3); x plus one more than y is equal to one more than x plus y (Q4); x times $\emptyset$ is equal to $\emptyset$ (Q5); x times one more than y is equal to x times y plus x (Q6); and any number other than $\emptyset$ is a successor (Q7).

If $\mathcal{P}$ is derived directly from some of Q1–Q7 then it is an *ND+* theorem of Robinson Arithmetic. But if the members of a set Γ are *ND+* theorems of Robinson Arithmetic, and $\Gamma \vdash_{ND+} \mathcal{P}$, then $\mathcal{P}$ is an *ND+* theorem of Robinson Arithmetic as well—for any derivation of $\mathcal{P}$ from some theorems might be extended into one which derives the theorems, and then goes on from there to obtain $\mathcal{P}$. In the ordinary case, then, we *build* to increasingly complex results: Having once demonstrated a theorem by a derivation, we feel free simply to *cite* it as a premise in the next derivation. So the collection of formulas we count as premises increases from one derivation to the next.

Though the application to arithmetic is interesting, there is in principle *nothing different* about derivations for Q from ones we have done before: We are moving from premises to a goal by rules. As we make progress, however, there will be an increasing number of premises available, and it may be relatively challenging to recognize *which* premises are relevant to a given goal. As you work through problems, you may find the Robinson and Peano reference on page 313 helpful.

Let us start with some simple generalizations of Q1–Q7. As they are stated, Q1–Q7 are particular formulas involving *variables*. But they permit derivation of corresponding principles for arbitrary terms $\mathcal{s}$ and $\mathcal{t}$.

T6.46. $\mathsf{Q} \vdash_{ND+} \sim(S\mathcal{t} = \emptyset)$

1.	$\sim(Sx = \emptyset)$	Q1
2.	$\forall u \sim(Su = \emptyset)$	1 $\forall$I
3.	$\sim(S\mathcal{t} = \emptyset)$	2 $\forall$E

Observe that there are no undischarged assumptions, so x is not free in an undischarged assumption; and since $\sim(Su = \emptyset)$ has no quantifiers, term $\mathcal{t}$ must be free for u in $\sim(Su = \emptyset)$. So there is no problem about the restrictions on $\forall$I and $\forall$E. And since $\mathcal{t}$ is any term, substituting $\emptyset$, $(S\emptyset + y)$, and the like for $\mathcal{t}$, we have that $\sim(S\emptyset = \emptyset)$, $\sim(S(S\emptyset + y) = \emptyset)$, and the like are all instances of T6.46. For the next result, let u be a variable not in $\mathcal{t}$.

T6.47. Q $\vdash_{ND+} (S\mathscr{t} = S\mathscr{s}) \to (\mathscr{t} = \mathscr{s})$

1.	$Sx = Sy \to x = y$	Q2
2.	$\forall u[Sx = Su \to x = u]$	1 $\forall$I
3.	$\forall v \forall u[Sv = Su \to v = u]$	2 $\forall$I
4.	$\forall u[S\mathscr{t} = Su \to \mathscr{t} = u]$	3 $\forall$E
5.	$S\mathscr{t} = S\mathscr{s} \to \mathscr{t} = \mathscr{s}$	4 $\forall$E

Since u is not a variable in $\mathscr{t}$, (4) meets the constraint on $\forall$E. Q1 – Q7 are stated in terms of the particular variables x and y. We cannot be sure that y is not a variable in $\mathscr{t}$. However, $\mathscr{t}$ has at most finitely many variables. So we can be sure that there is *some* variable not in $\mathscr{t}$. And the derivation goes through once the quantifier is applied to it.

*T6.48. Q $\vdash_{ND+} (\mathscr{t} + \emptyset) = \mathscr{t}$

T6.49. Q $\vdash_{ND+} (\mathscr{t} + S\mathscr{s}) = S(\mathscr{t} + \mathscr{s})$

T6.50. Q $\vdash_{ND+} (\mathscr{t} \times \emptyset) = \emptyset$

T6.51. Q $\vdash_{ND+} (\mathscr{t} \times S\mathscr{s}) = ((\mathscr{t} \times \mathscr{s}) + \mathscr{t})$

T6.52. Q $\vdash_{ND+} \sim(\mathscr{t} = \emptyset) \to \exists w(\mathscr{t} = Sw)$ where variable w does not appear in $\mathscr{t}$

Given these results, we are ready for some that are more interesting. Let us show that $1 + 1 = 2$. That is, that Q $\vdash_{ND+} S\emptyset + S\emptyset = SS\emptyset$.

(CJ)

1.	$(S\emptyset + \emptyset) = S\emptyset$	T6.48
2.	$(S\emptyset + S\emptyset) = S(S\emptyset + \emptyset)$	T6.49
3.	$(S\emptyset + S\emptyset) = SS\emptyset$	2,1 =E

The first premise is an instance of T6.48 with $S\emptyset$ for $\mathscr{t}$. (2) is an instance of T6.49 that has $S\emptyset$ for $\mathscr{t}$ and $\emptyset$ for $\mathscr{s}$. Given the premises, this derivation is simple. With $(S\emptyset + \emptyset) = S\emptyset$ from (1), we can substitute $S\emptyset$ for $S\emptyset + \emptyset$ in the right side of (2) by =E. This is just what we do. Be sure you understand each step. In the same way, and more generally,

T6.53. Q $\vdash_{ND+} \mathscr{t} + S\emptyset = S\mathscr{t}$

Hint: You can do this by the same basic steps as above.

Observe the way Q3 and Q4 work together: Q3 (T6.48) gives the sum of any term t with $\emptyset$; and given the sum of t with any s, Q4 (T6.49) gives the sum of t and one more than s. So we can calculate the sum of t and zero from T6.48, and then with T6.49 get the sum of it and one, then it and two, and so forth. In this way, we calculate arbitrary sums. So, for example, Q $\vdash_{ND+}$ $SS\emptyset + SSS\emptyset = SSSSS\emptyset$. We start with T6.48 and T6.49.

(CK)

1.	$(SS\emptyset + \emptyset) = SS\emptyset$	T6.48
2.	$(SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset)$	T6.49
3.	$(SS\emptyset + S\emptyset) = SSS\emptyset$	2,1 =E

We use (1) to put the known value of $SS\emptyset + \emptyset$ into the right side of (2). Or we might simply have asserted (3) by T6.53. But now the value of $SS\emptyset + S\emptyset$ is known, and we can use T6.49 again.

(CL)

1.	$(SS\emptyset + \emptyset) = SS\emptyset$	T6.48
2.	$(SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset)$	T6.49
3.	$(SS\emptyset + S\emptyset) = SSS\emptyset$	2,1 =E
4.	$(SS\emptyset + SS\emptyset) = S(SS\emptyset + S\emptyset)$	T6.49
5.	$(SS\emptyset + SS\emptyset) = SSSS\emptyset$	4,3 =E

This time, we use (3) to put the known value of $SS\emptyset + S\emptyset$ into the right side of (4). And we can use T6.49 again to get the final result. Since we are in *ND+*, we sort the premises to the top to get,

(CM)

1.	$(SS\emptyset + \emptyset) = SS\emptyset$	T6.48
2.	$(SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset)$	T6.49
3.	$(SS\emptyset + SS\emptyset) = S(SS\emptyset + S\emptyset)$	T6.49
4.	$(SS\emptyset + SSS\emptyset) = S(SS\emptyset + SS\emptyset)$	T6.49
5.	$(SS\emptyset + S\emptyset) = SSS\emptyset$	2,1 =E
6.	$(SS\emptyset + SS\emptyset) = SSSS\emptyset$	3,5 =E
7.	$(SS\emptyset + SSS\emptyset) = SSSSS\emptyset$	4,6 =E

Again, $SS\emptyset + \emptyset$ is given from T6.48; we use multiple applications of T6.49 to increase the second term to $SSS\emptyset$ for the final result.

And similarly for multiplication: Q5 (T6.50) gives the product of any term t with $\emptyset$; and given the product of t with any s, Q6 (T6.51) gives the product of t and one more than s. So we can calculate the product t and zero from T6.50, and then with T6.51 get the product of it and one, it and two, and so forth. Thus, for example, Q $\vdash_{ND+}$ $S\emptyset \times SS\emptyset = SS\emptyset$.

(CN)

1.	$S\emptyset \times \emptyset = \emptyset$	T6.50
2.	$S\emptyset \times S\emptyset = (S\emptyset \times \emptyset) + S\emptyset$	T6.51
3.	$\emptyset + S\emptyset = S\emptyset$	T6.53
4.	$S\emptyset \times SS\emptyset = (S\emptyset \times S\emptyset) + S\emptyset$	T6.51
5.	$S\emptyset + S\emptyset = SS\emptyset$	T6.53
6.	$S\emptyset \times S\emptyset = \emptyset + S\emptyset$	2,1 =E
7.	$S\emptyset \times S\emptyset = S\emptyset$	6,3 =E
8.	$S\emptyset \times SS\emptyset = S\emptyset + S\emptyset$	4,7 =E
9.	$S\emptyset \times SS\emptyset = SS\emptyset$	8,5 =E

The basic pattern of working from one case to the next is as for addition. A difference is that the multiplications depend on additions—which require derivation of their own (in this case, T6.53).

So far, we have focused on variable-free terms built up from $\emptyset$. But nothing stops application to expressions in a more general form.

(CO)

1.	$(j + Sk) = S(j + k)$	T6.49
2.	$\quad \exists y(j + y = S\emptyset)$	A (g, $\rightarrow$I)
3.	$\quad\quad j + k = S\emptyset$	A (g, 2$\exists$E)
4.	$\quad\quad j + Sk = SS\emptyset$	1,3 =E
5.	$\quad\quad \exists y(j + y = SS\emptyset)$	4 $\exists$I
6.	$\quad \exists y(j + y = SS\emptyset)$	2,3-5 $\exists$E
7.	$\exists y(j + y = S\emptyset) \rightarrow \exists y(j + y = SS\emptyset)$	2-6 $\rightarrow$I
8.	$\forall x[\exists y(x + y = S\emptyset) \rightarrow \exists y(x + y = SS\emptyset)]$	7 $\forall$I

The basic setup for $\forall$I, $\rightarrow$I, and $\exists$E is by now routine. The real work is where we use (1) and (3) to obtain $j + Sk = SS\emptyset$. Here are a couple of theorems that will be of interest later:

T6.54. Q $\vdash_{ND+} t \leq \emptyset \rightarrow t = \emptyset$

Hints: Be sure you are clear about what is being asked for; at some stage, you will need abv to unpack the abbreviation. Do not forget that you can appeal to T6.46 and T6.52.

T6.55. Q $\vdash_{ND+} \sim(t < \emptyset)$

Hint: This comes to an application of SC4. Under assumptions for $\sim$I and then (after abv) $\exists$E, assume $\sim(t = \emptyset)$ to obtain a first contradiction; you will be able to obtain contradiction from $t = \emptyset$ as well.

Robinson Arithmetic is interesting. Its axioms are sufficient to prove arbitrary facts about particular numbers. Its language and derivation system are just strong enough to support Gödel's incompleteness result, on which it is not possible for a "nicely specified" consistent theory including a sufficient amount of arithmetic to have as consequences $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$ (Part IV). But we do not need Gödel's result to see that Robinson Arithmetic is (negation) incomplete: It turns out that many true generalizations are not provable in Robinson Arithmetic. So, for example, neither $\forall x \forall y[(x \times y) = (y \times x)]$, nor its negation is provable.[9] So Robinson Arithmetic is a particularly weak theory.

*E6.39. Produce derivations to show T6.48–T6.53. For any problem, you may appeal to results before.

*E6.40. Produce derivations to show each of the following. Along with theorems from the text, for any exercise you may appeal to ones before.

*a. Q $\vdash_{ND+}$ $(t + SS\emptyset) = SSt$
Hint: Do not forget that you can appeal to T6.53.

*b. Q $\vdash_{ND+}$ $(SS\emptyset \times SS\emptyset) = SSSS\emptyset$

c. Q $\vdash_{ND+}$ $(t + SSS\emptyset) = SSSt$

d. Q $\vdash_{ND+}$ $(SSS\emptyset \times SS\emptyset) = SSSSSS\emptyset$

e. Q $\vdash_{ND+}$ $(SSS\emptyset \times SS\emptyset) = (SS\emptyset \times SSS\emptyset)$

*f. Q $\vdash_{ND+}$ $\sim\exists x(x + SS\emptyset = S\emptyset)$
Hint: Do not forget that you can appeal to T6.46 and T6.47.

*g. Q $\vdash_{ND+}$ $\forall x[(x = \emptyset \lor x = S\emptyset) \to x \leq S\emptyset]$
Hint: You will need to unpack the abbreviation using abv.

h. Q $\vdash_{ND+}$ $\forall x[(x = \emptyset \lor x = S\emptyset) \to x < SS\emptyset]$

i. Q $\vdash_{ND+}$ $(\forall x \leq S\emptyset)(x = \emptyset \lor x = S\emptyset)$
Hint: You can use ($\forall$I) and T6.52, T6.49, T6.47 and T6.54.

j. Q $\vdash_{ND+}$ $(\forall x \leq S\emptyset)(x \leq SS\emptyset)$

E6.41. Produce derivations to show T6.54 and T6.55.

[9] A semantic demonstration of this negative result is left as an exercise for Chapter 7. But we already understand the basic idea from Chapter 4: To show that a conclusion does not follow, produce an interpretation on which the axioms are true but the conclusion is not.

6.4.2 Peano Arithmetic

Though Robinson Arithmetic leaves even standard results like commutation for multiplication unproven, it is possible to strengthen the axioms to obtain such results. Thus such standard generalizations are provable in Peano Arithmetic.[10] This is the system we encountered in Chapter 3, but now with *ND+*. So when $\mathcal{P}$ is derived from the axioms it is an *ND+ theorem of Peano Arithmetic*. For this, let PA1–PA6 be the same as Q1–Q6. Replace Q7 as follows: For any formula $\mathcal{P}$,

PA7 $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})] \rightarrow \forall x\mathcal{P}$

is an axiom. If a formula $\mathcal{P}$ applies to $\emptyset$, and for any x if $\mathcal{P}$ applies to x then it also applies to Sx, then $\mathcal{P}$ applies to every x. This form represents the *principle of mathematical induction*. While all the axioms of Q (and so PA1–PA6) are particular formulas, PA7 is an *axiom schema* insofar as indefinitely many formulas might be of that form. We will have much more to say about the principle of mathematical induction in Part II. For now, it is enough merely to *recognize* its instances. Thus, for example, if $\mathcal{P}$ is $\sim(x = Sx)$, then $\mathcal{P}^x_\emptyset$ is $\sim(\emptyset = S\emptyset)$, and $\mathcal{P}^x_{Sx}$ is $\sim(Sx = SSx)$. So,

$$[\sim(\emptyset = S\emptyset) \wedge \forall x(\sim(x = Sx) \rightarrow \sim(Sx = SSx))] \rightarrow \forall x\sim(x = Sx)$$

is an instance of the schema. You should see why this is so.

It will be convenient to have the principle of mathematical induction in a rule form. Given $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$ on accessible lines *a* and *b*, one may move to $\forall x\mathcal{P}$ with justification *a,b* IN.

IN

a.	$\mathcal{P}^x_\emptyset$	
b.	$\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$	
	$\forall x\mathcal{P}$	a,b IN

1.	$\mathcal{P}^x_\emptyset$	P
2.	$\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$	P
3.	$[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})] \rightarrow \forall x\mathcal{P}$	PA7
4.	$\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$	1,2 $\wedge$I
5.	$\forall x\mathcal{P}$	3,4 $\rightarrow$E

The rule is justified from PA7 by reasoning as on the right. That is, given $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$ on accessible lines, one can always conjoin them, then with an instance of PA7 as a premise reach $\forall x\mathcal{P}$ by $\rightarrow$E. The use of IN merely saves a couple steps, and avoids some relatively long formulas we would have to deal with using PA7 alone. Thus, from our previous example, where $\mathcal{P}$ is $\sim(x = Sx)$, we would need $\sim(\emptyset = S\emptyset)$ and $\forall x[\sim(x = Sx) \rightarrow \sim(Sx = SSx)]$ to move to $\forall x\sim(x = Sx)$ by IN. You should see that this is no different from before.

[10] After the work of R. Dedekind and G. Peano. For historical discussion, see Wang, "The Axiomatization of Arithmetic."

Since PA1–PA6 are the same as Q1–Q6, theorems of Q derived from just Q1–Q6 remain theorems of PA. Further, PA has a theorem like Q7. That is, with the aid of PA7, we shall be able to show,

$$\text{PA} \vdash_{ND+} \sim(t = \emptyset) \to \exists w(t = Sw) \qquad \text{where } w \text{ is not a variable in } t$$

Since it is to follow from PA1–PA7, the proof must, of course, not depend on Q7. But this is the same as T6.52, and has Q7 as an instance. Given this, any *ND+* theorem of Q is automatically an *ND+* theorem of PA—for we can derive this result, and use it as it would have been used in a derivation for Q. We thus freely use any theorem from Q in the derivations that follow.

With these axioms in hand, including the principle of mathematical induction, we set out to show some general principles of commutativity, associativity, and distribution for addition and multiplication. But we build gradually to them. For a first application of IN, let $\mathcal{P}$ be $(\emptyset + x) = x$; then $\mathcal{P}^x_\emptyset$ is $(\emptyset + \emptyset) = \emptyset$ and $\mathcal{P}^x_{Sx}$ is $(\emptyset + Sx) = Sx$.

T6.56. PA $\vdash_{ND+} (\emptyset + t) = t$

1.	$(\emptyset + \emptyset) = \emptyset$	T6.48
2.	$(\emptyset + Sj) = S(\emptyset + j)$	T6.49
3.	$\quad (\emptyset + j) = j$	A (g, →I)
4.	$\quad (\emptyset + Sj) = Sj$	2,3 =E
5.	$[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$	3-4 →I
6.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	5 ∀I
7.	$\forall x[(\emptyset + x) = x]$	1,6 IN
8.	$(\emptyset + t) = t$	7 ∀E

The key to this derivation, and others like it, is bringing IN into play. That we want to do this is sufficient to drive us to the following as setup:

a.	$(\emptyset + \emptyset) = \emptyset$	(goal)
b.	$\quad (\emptyset + j) = j$	A (g, →I)
c.	$\quad (\emptyset + Sj) = Sj$	(goal)
d.	$[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$	b-c →I
e.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	d ∀I
f.	$\forall x[(\emptyset + x) = x]$	a,e IN
	$(\emptyset + t) = t$	f ∀E

Our aim is to get the goal by ∀E from $\forall x[(\emptyset + x) = x]$. And we will get this by IN. So we need the inputs to IN: $\mathcal{P}^x_\emptyset$, that is, $(\emptyset + \emptyset) = \emptyset$, and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$, that is, $\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$. As is often the case, $\mathcal{P}^x_\emptyset$, here $(\emptyset + \emptyset) = \emptyset$, is easy to get. It is natural to obtain the latter by ∀I from $[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$, and to go for this by →I. Thus the work of the derivation is reaching goals (a) and (c). But that is not hard: (a) is an immediate instance of T6.48; and (c) follows from the equality on (b) with an instance of T6.49. We are in a better position to *think* about which (axioms or) theorems we need as premises once we have gone through this standard setup for IN. We will see this pattern many times.

T6.57. PA $\vdash_{ND+} (St + \emptyset) = S(t + \emptyset)$

1.	$(St + \emptyset) = St$	T6.48
2.	$(t + \emptyset) = t$	T6.48
3.	$(St + \emptyset) = S(t + \emptyset)$	1,2 =E

This simple derivation results by using the equality on (2) to justify a substitution for t in (1). This result forms the "zero case" for the one that follows.

T6.58. PA $\vdash_{ND+} (St + \mathscr{s}) = S(t + \mathscr{s})$

1.	$(St + \emptyset) = S(t + \emptyset)$	T6.57
2.	$(t + Sj) = S(t + j)$	T6.49
3.	$(St + Sj) = S(St + j)$	T6.49
4.	$\quad (St + j) = S(t + j)$	A (g, $\to$I)
5.	$\quad (St + Sj) = SS(t + j)$	3,4 =E
6.	$\quad (St + Sj) = S(t + Sj)$	5,2 =E
7.	$[(St + j) = S(t + j)] \to [(St + Sj) = S(t + Sj)]$	4-6 $\to$I
8.	$\forall x([(St + x) = S(t + x)] \to [(St + Sx) = S(t + Sx)])$	7 $\forall$I
9.	$\forall x[(St + x) = S(t + x)]$	1,8 IN
10.	$(St + \mathscr{s}) = S(t + \mathscr{s})$	9 $\forall$E

Again, the idea is to bring IN into play. Here $\mathscr{P}$ is $(St + x) = S(t + x)$. Given that we have the zero-case on line (1), with standard setup the derivation reduces to obtaining the formula on (6) given the assumption on (4). Line (6) is like (3) except for the right-hand side. So it is a matter of applying the equalities on (4) and (2) to reach the goal. You should study this derivation, to be sure that you follow the applications of =E—for we encounter such uses over and over.

T6.59. PA $\vdash_{ND+}$ $t + s = s + t$ *commutativity of addition*

1.	$t + \emptyset = t$	T6.48
2.	$\emptyset + t = t$	T6.56
3.	$t + Sj = S(t + j)$	T6.49
4.	$Sj + t = S(j + t)$	T6.58
5.	$t + \emptyset = \emptyset + t$	1,2 =E
6.	$\quad t + j = j + t$	A (g, $\rightarrow$I)
7.	$\quad t + Sj = S(j + t)$	3,6 =E
8.	$\quad t + Sj = Sj + t$	7,4 =E
9.	$[t + j = j + t] \rightarrow [t + Sj = Sj + t]$	6-8 $\rightarrow$I
10.	$\forall x([t + x = x + t] \rightarrow [t + Sx = Sx + t])$	9 $\forall$I
11.	$\forall x[t + x = x + t]$	5,10 IN
12.	$t + s = s + t$	11 $\forall$E

Again the derivation is by IN where $\mathcal{P}$ is $t + x = x + t$. We achieve the zero case on (5) from (1) and (2). So the derivation reduces to getting (8) given the assumption on (6). The left-hand side of (8) is like (3). So it is a matter of applying the equalities on (6) and then (4) to reach the goal. Once you have the basic setup, you are positioned to organize in your mind which equalities you have, and which are required to reach the goal.

T6.59 is an interesting result! No doubt, you have heard from your mother's knee that $t + s = s + t$. But it is a sweeping claim with application to *all* numbers. Surely you have not been able to test every case. But here we have a derivation of the result from the Peano axioms. And similarly for results that follow. Now that you have this result, recognize that you can use instances of it to switch around terms in additions—just as you would have done automatically for addition in elementary school.

*T6.60. PA $\vdash_{ND+}$ $(r + s) + \emptyset = r + (s + \emptyset)$

Hint: Begin with $(r + s) + \emptyset = r + s$ as an instance of T6.48. The derivation is then a simple matter of using T6.48 again to replace s in the right-hand side with $s + \emptyset$.

*T6.61. PA $\vdash_{ND+}$ $(r + s) + t = r + (s + t)$ *associativity of addition*

Hint: For an application of IN let $\mathcal{P}$ be $(r + s) + x = r + (s + x)$. You already have the zero case from T6.60. Inside the subderivation for $\rightarrow$I, use the assumption together with some instances of T6.49 to reach the goal.

Again, once you have this result, be aware that you can use its instances for association as you would have done long ago. It is good to *think* about what the different theorems give you, so that you can make sense of what to use where!

T6.62. PA $\vdash_{ND+} t \times S\emptyset = t$

Hint: This does not require IN. It is a rather a simple result which you can do in just a few lines.

T6.63. PA $\vdash_{ND+} \emptyset \times t = \emptyset$

Hint: For an application of IN, let $\mathscr{P}$ be $\emptyset \times x = \emptyset$. The derivation is easy enough with an application of T6.50 for the zero case, and instances of T6.51 and T6.48 for the main result.

T6.64. PA $\vdash_{ND+} St \times \emptyset = (t \times \emptyset) + \emptyset$

Hint: This does not require IN. It follows rather by some simple applications of T6.48 and T6.50.

T6.65. PA $\vdash_{ND+} St \times s = (t \times s) + s$

Hint: For this longish derivation, plan to reach the goal through IN where $\mathscr{P}$ is $St \times x = (t \times x) + x$. You will be able to use your assumption for $\rightarrow$I with an instance of T6.51 to show $St \times Sj = ((t \times j) + j) + St$. Then you should be able to manipulate the right-hand side into the result you want. You will need several theorems as premises.

T6.66. PA $\vdash_{ND+} t \times s = s \times t$ *commutativity for multiplication*

Hint: Plan on reaching the goal by IN where $\mathscr{P}$ is $t \times x = x \times t$. Apart from theorems for the zero case, you will need an instance of T6.51 and an instance of T6.65.

T6.67. PA $\vdash_{ND+} r \times (s + \emptyset) = (r \times s) + (r \times \emptyset)$

Hint: You will not need IN for this.

T6.68. PA $\vdash_{ND+} r \times (s + t) = (r \times s) + (r \times t)$ *distributivity*

Hint: Plan on reaching the goal by IN where $\mathscr{P}$ is $r \times (s + x) = (r \times s) + (r \times x)$. Under the assumption $r \times (s + j) = (r \times s) + (r \times j)$, perhaps the simplest thing is to start with $r \times (s + Sj) = r \times (s + Sj)$ by =I. Then the left side is what you want, and you can work on the right. Working on the right-hand side, $(s + Sj) = S(s + j)$ by T6.49. And $r \times S(s + j) = (r \times (s + j)) + r$ by T6.51. With this, you will be able to apply the assumption. Then further simplification should get you to your goal.

T6.69. PA $\vdash_{ND+} (s + t) \times r = (s \times r) + (t \times r)$ *distributivity*

Hint: You will not need IN for this. Rather, it is enough to use T6.68 with a few applications of T6.66.

T6.70. PA $\vdash_{ND+} (q + r) \times (s + t) = ((q \times s) + (q \times t)) + ((r \times s) + (r \times t))$

Hint: This is another application of distributivity. You may have encountered this result under the acronym 'FOIL' (first/outer/inner/last) in elementary algebra.

T6.71. PA $\vdash_{ND+} (s \times t) \times \emptyset = s \times (t \times \emptyset)$

Hint: This is easy without an application of IN.

T6.72. PA $\vdash_{ND+} (s \times t) \times r = s \times (t \times r)$ *associativity of multiplication*

Hint: Go after the goal by IN where $\mathcal{P}$ is $(s \times t) \times x = s \times (t \times x)$. You should be able to use the assumption $(s \times t) \times j = s \times (t \times j)$ with T6.51 to show that $(s \times t) \times Sj = (s \times (t \times j)) + (s \times t)$; then you can reduce the right hand side to what you want.

T6.73. PA $\vdash_{ND+} r + t = s + t \rightarrow r = s$ *cancellation law for addition*

Hint: Go for the goal by IN where $\mathcal{P}$ is $r + x = s + x \rightarrow r = s$.

T6.74. PA $\vdash_{ND+} \forall y[t \neq \emptyset \rightarrow (y \times t = \emptyset \times t \rightarrow y = \emptyset)]$

Hint: This does not require IN.

T6.75. PA $\vdash_{ND+} t \neq \emptyset \rightarrow (r \times t = s \times t \rightarrow r = s)$ *cancellation law for multiplication*

Hint: For this challenging derivation go for the goal by IN on x where $\mathcal{P}$ is $\forall y[t \neq \emptyset \rightarrow (y \times t = x \times t \rightarrow y = x)]$. You have T6.74 for the zero-case. Observe that we adopt the "slash" notation to indicate negated equality.

After you have completed the exercises, if you are looking for more to do, you might take a look at the additional results from T13.11 on page 649 of Chapter 13—which you now have the background to work.

Peano Arithmetic is sufficient for many "ordinary" results we could not obtain in Q alone. However, insofar as PA includes the language and results of Q, it too is sufficient for Gödel's incompleteness theorem. So PA is not (negation) complete, and it is not possible for a nicely specified consistent theory including PA to be such that it proves either $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$. But such results must wait for later.

Robinson and Peano Arithmetic (*ND+*)

Q/PA 1. $\sim(Sx = \emptyset)$

2. $(Sx = Sy) \to (x = y)$

3. $(x + \emptyset) = x$

4. $(x + Sy) = S(x + y)$

5. $(x \times \emptyset) = \emptyset$

6. $(x \times Sy) = [(x \times y) + x]$

Q7 $\sim(x = \emptyset) \to \exists y(x = Sy)$

PA7 $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})] \to \forall x\mathcal{P}$

$$\text{IN}\quad \begin{array}{r|ll} \text{a.} & \mathcal{P}^x_\emptyset & \\ \text{b.} & \forall x(\mathcal{P} \to \mathcal{P}^x_{Sx}) & \\ & & \\ & \forall x\mathcal{P} & \text{a,b IN} \end{array}$$

Derived from PA7

T6.46 Q $\vdash_{ND+}$ $\sim(St = \emptyset)$

T6.47 Q $\vdash_{ND+}$ $(St = Ss) \to (t = s)$

T6.48 Q $\vdash_{ND+}$ $(t + \emptyset) = t$

T6.49 Q $\vdash_{ND+}$ $(t + Ss) = S(t + s)$

T6.50 Q $\vdash_{ND+}$ $(t \times \emptyset) = \emptyset$

T6.51 Q $\vdash_{ND+}$ $(t \times Ss) = ((t \times s) + t)$

T6.52 Q $\vdash_{ND+}$ $\sim(t = \emptyset) \to \exists w(t = Sw)$ *where variable w does not appear in t*

T6.53 Q $\vdash_{ND+}$ $t + S\emptyset = St$

T6.54 Q $\vdash_{ND+}$ $t \leq \emptyset \to t = \emptyset$

T6.55 Q $\vdash_{ND+}$ $\sim(t < \emptyset)$

T6.56 PA $\vdash_{ND+}$ $(\emptyset + t) = t$

T6.57 PA $\vdash_{ND+}$ $(St + \emptyset) = S(t + \emptyset)$

T6.58 PA $\vdash_{ND+}$ $(St + s) = S(t + s)$

T6.59 PA $\vdash_{ND+}$ $t + s = s + t$ *commutativity of addition*

T6.60 PA $\vdash_{ND+}$ $(r + s) + \emptyset = r + (s + \emptyset)$

T6.61 PA $\vdash_{ND+}$ $(r + s) + t = r + (s + t)$ *associativity of addition*

T6.62 PA $\vdash_{ND+}$ $t \times S\emptyset = t$

T6.63 PA $\vdash_{ND+}$ $\emptyset \times t = \emptyset$

T6.64 PA $\vdash_{ND+}$ $St \times \emptyset = (t \times \emptyset) + \emptyset$

T6.65 PA $\vdash_{ND+}$ $St \times s = (t \times s) + s$

T6.66 PA $\vdash_{ND+}$ $t \times s = s \times t$ *commutativity for multiplication*

T6.67 PA $\vdash_{ND+}$ $r \times (s + \emptyset) = (r \times s) + (r \times \emptyset)$

T6.68 PA $\vdash_{ND+}$ $r \times (s + t) = (r \times s) + (r \times t)$ *distributivity*

T6.69 PA $\vdash_{ND+}$ $(s + t) \times r = (s \times r) + (t \times r)$ *distributivity*

T6.70 PA $\vdash_{ND+}$ $(q + r) \times (s + t) = ((q \times s) + (q \times t)) + ((r \times s) + (r \times t))$

T6.71 PA $\vdash_{ND+}$ $(s \times t) \times \emptyset = s \times (t \times \emptyset)$

T6.72 PA $\vdash_{ND+}$ $(s \times t) \times r = s \times (t \times r)$ *associativity of multiplication*

T6.73 PA $\vdash_{ND+}$ $r + t = s + t \to r = s$ *cancellation law for addition*

T6.74 PA $\vdash_{ND+}$ $\forall y[t \neq \emptyset \to (y \times t = \emptyset \times t \to y = \emptyset)]$

T6.75 PA $\vdash_{ND+}$ $t \neq \emptyset \to (r \times t = s \times t \to r = s)$ *cancellation law for multiplication*

In addition by abv we allow movement between abbreviated and unabbreviated forms for inequalities and bounded quantifiers.

*E6.42. Produce derivations to show T6.60–T6.75.

E6.43. Produce a derivation to show that PA proves the result of T6.52 and so that any *ND+* theorem of Q is an *ND+* theorem of PA. Hint: For an application of IN let $\mathscr{P}$ be $x \neq \emptyset \rightarrow \exists w(x = Sw)$.

E6.44. Produce derivations to show that T13.11a–T13.11af are theorems in *ND+*. The final few are left for an exercise in Chapter 13. In the textbook $\overline{1}$ abbreviates $S\emptyset$ (compare page 391), and some parentheses are dropped (SLAPP does not do this). Though *ND* requires premises to be listed at the top, this is not necessary for theorems — and you may find it convenient to cite theorems where they are used.

E6.45. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The rules ∀I and ∃E, including especially restrictions on the rules.

b. The axioms of Q and PA and the way theorems derive from them.

c. The relation between the rules of *ND* and the rules of *ND+*.

Part II

Transition: Reasoning About Logic

Introductory

We have expended a great deal of energy learning to do logic. What we have learned constitutes the complete classical predicate calculus with equality. This is a system of tremendous power including for reasoning in foundations of arithmetic.

But our work itself raises questions. In Chapter 4 we used truth trees and tables for an account of the conditions under which sentential formulas are true and arguments are valid. In the quantificational case, though, we were not able to use our graphical methods for a general account of truth and validity—there were simply too many branches, and too many interpretations, for a general account by means of trees. Thus there is an open question about whether and how quantificational validity can be shown.

And once we have introduced our notions of validity, many interesting questions can be asked about how they work: Are the arguments that are valid in *AD* the same as the ones that are valid in *ND*? Are the arguments that are valid in *ND* the same as the ones that are quantificationally valid? Are the theorems of Q the same as the theorems of PA? Are theorems of PA the same as the truths on N the standard interpretation for number theory? Is it possible for a computing device to identify the theorems of the different logical systems?

It is one thing to ask such questions, and perhaps amazing that there are demonstrable answers. We will come to that. However, in this short section we do not attempt answers. Rather, we put ourselves in a position to think about answers by introducing methods for thinking about logic. Thus this part looks both backward and forward: By our methods we plug the hole left from Chapter 4—in Chapter 7 we accomplish what could not be done with the tables and trees of Chapter 4, and are able to demonstrate quantificational validity. At the same time, we lay a foundation to ask and answer core questions about logic.

Chapter 7 begins with our basic method of reasoning from definitions. Chapter 8 introduces mathematical induction. These methods are important not only for results, but for their own sakes, as part of the broader "toolkit" that comes with mathematical logic.

Chapter 7

Direct Semantic Reasoning

It is the task of this chapter to think about reasoning directly from definitions. Frequently students who already reason quite skillfully with definitions flounder when asked to do so explicitly, in the style of this chapter.[1] Thus I propose to begin in a restricted context—one with which we are already familiar, using a fairly rigid framework as a guide. Perhaps you first learned to ride a bicycle with training wheels, but eventually learned to ride without them, and so to go faster, and to places other than the wheels would let you go. Similarly, in the end, we will want to apply our methods beyond the restricted context in which we begin, working outside the initial framework. But the framework should give us a good start. In this chapter, then, I introduce the framework in the context of reasoning for specifically *semantic* notions, and against the background of semantic reasoning we have already done.

In Chapter 4 we used truth trees and tables for an account of the conditions under which sentential formulas are true and arguments are valid. In the quantificational case though, we were not able to use our graphical methods for a general account of truth and validity—there were simply too many branches, and too many interpretations, for a general account by means of trees. For a complete account, we will need to reason more directly from the definitions. But the tables and trees *do* exhibit the semantic definitions. So we can build on what we have already done with them. Our goal will be to move past the tables and trees, and learn to function without them. After some introductory remarks in section 7.1, we start with the sentential case (section 7.2), and move to the quantificational (section 7.3).

[1]The ability to reason clearly and directly with definitions is important not only here, but also beyond. From Dennett's (often humorous) *Philosopher's Lexicon*, compare the verb *to chisholm*—after Roderick Chisholm, who was a master of the technique—where one proposes a definition; considers a counterexample; modifies to account for the example; considers another counterexample; modifies again; and so forth. As, "He started with definition (d.8) and kept chisholming away at it until he ended up with (d.$8^{\prime\prime\prime\prime\prime\prime\prime\prime}$)." Such reasoning is impossible to understand apart from explicit attention to consequences of definitions of the sort we have in mind.

7.1 Introductory

I begin with some considerations about what we are trying to accomplish, and how it is related to what we have done. At this stage, do not worry so much about details as about the overall nature of the project. With this in mind, consider the following row of a truth table, meant to show that $B \to C \nvDash_s \sim B$:

(A)

B	C	B	$\to$	C	/	$\sim$	B
T	T	T	***T***	T		***F***	T

Since there is an interpretation on which the premise is true and the conclusion is not, the argument is not sententially valid. Now, what justifies setting $B \to C$ to T and $\sim B$ to F? One might respond, "the truth tables." But the truth tables T($\to$) and T($\sim$) themselves derive from definition ST. And similarly the conclusion that the argument is not sententially valid derives from SV.

ST($\sim$) $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] = \mathsf{F}$; otherwise $\mathsf{I}[\sim\mathscr{P}] = \mathsf{F}$.

ST($\to$) $\mathsf{I}[(\mathscr{P} \to \mathscr{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] = \mathsf{F}$ or $\mathsf{I}[\mathscr{Q}] = \mathsf{T}$ (or both); otherwise $\mathsf{I}[(\mathscr{P} \to \mathscr{Q})] = \mathsf{F}$.

SV $\Gamma \vDash_s \mathscr{P}$ iff there is no sentential interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathscr{P}] = \mathsf{F}$.

In this case, $\mathsf{I}[C] = \mathsf{T}$; from this, reasoning as by $\vee$I, $\mathsf{I}[B] = \mathsf{F}$ or $\mathsf{I}[C] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[B \to C] = \mathsf{T}$. Similarly, $\mathsf{I}[B] = \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{F}$. And since we have produced an I such that $\mathsf{I}[B \to C] = \mathsf{T}$ but $\mathsf{I}[\sim B] = \mathsf{F}$, by SV, $B \to C \nvDash_s \sim B$. Up to now, we have used tables to express these conditions. But we *might* have reasoned directly:

(B) Consider any interpretation I such that $\mathsf{I}[B] = \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$. Since $\mathsf{I}[C] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$ or $\mathsf{I}[C] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[B \to C] = \mathsf{T}$. But since $\mathsf{I}[B] = \mathsf{T}$, by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{F}$. So there is a sentential interpretation I such that $\mathsf{I}[B \to C] = \mathsf{T}$ but $\mathsf{I}[\sim B] = \mathsf{F}$; so by SV, $B \to C \nvDash_s \sim B$.

Presumably, all this is "contained" in the one line of the truth table, when we use it to conclude that the argument is not sententially valid. Our aim is to "expose" reasoning in this way.

Similarly, consider the following table, meant to show that $\sim\sim A \vDash_s \sim A \to A$.

(C)

A	$\sim$	$\sim$	A	/	$\sim$	A	$\to$	A
T	***T***	F	T		F	T	***T***	T
F	***F***	T	F		T	F	***F***	F

Since there is no row where the premise is true and the conclusion is false, the argument is sententially valid. Again, ST($\sim$) and ST($\to$) justify the way you build the table. And SV lets you conclude that the argument is sententially valid. Thus the table represents reasoning as follows:

(D) For any sentential interpretation either (i) $\mathsf{I}[A] = \mathsf{T}$ or (ii) $\mathsf{I}[A] = \mathsf{F}$. Suppose (i); then $\mathsf{I}[A] = \mathsf{T}$; so $\mathsf{I}[\sim A] = \mathsf{F}$ or $\mathsf{I}[A] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$; from this either $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$; so it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. Suppose (ii); then $\mathsf{I}[A] = \mathsf{F}$; so by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{T}$; so by ST($\sim$) again, $\mathsf{I}[\sim\sim A] = \mathsf{F}$; so either $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$; so it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. From these together, no interpretation makes it the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. So by SV, $\sim\sim A \vDash_s \sim A \rightarrow A$.

Thus we might recapitulate reasoning in the table. Perhaps we typically "whip through" tables without explicitly considering all the definitions involved. But the definitions *are* involved when we complete the table.

In fact, (D) does not recapitulate the entire table (C). Thus at (i), for the conditional we do not establish $\mathsf{I}[\sim A] = \mathsf{F}$—it is enough that $\mathsf{I}[A] = \mathsf{T}$ so that $\mathsf{I}[\sim A] = \mathsf{F}$ or $\mathsf{I}[A] = \mathsf{T}$ and by ST($\rightarrow$), $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$. Similarly at (i) there is no need to make the point that $\mathsf{I}[\sim\sim A] = \mathsf{T}$. What matters is that $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$, so that $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \rightarrow A] = \mathsf{T}$, and it is therefore not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. Such "shortcuts" may reflect what you have already done when you realize that, say, a true conclusion eliminates the need to think about the premises on some row of a table. Even so, the idea of reasoning in this way corresponding to a 4, 8, or more (!) row table remains painful.

But there is a way out. Recall what happens when you apply the Chapter 4 "shortcut" table method to valid arguments. You start with the assumption that the premises are true and the conclusion is not. If the argument is valid, you reach some conflict so that it is not, in fact, possible to complete the row. Then, as we said on page 105, you know "in your heart" that the argument is valid. Let us turn this into an official argument form.

(E) Suppose $\sim\sim A \nvDash_s \sim A \rightarrow A$; then by SV, there is an I such that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. From the former, by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{F}$. But from the latter, by ST($\rightarrow$), $\mathsf{I}[\sim A] = \mathsf{T}$ and $\mathsf{I}[A] = \mathsf{F}$. So $\mathsf{I}[\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A] = \mathsf{F}$. This is impossible; reject the assumption: $\sim\sim A \vDash_s \sim A \rightarrow A$.

This is better. The assumption that the argument is invalid leads to the conclusion that for some I, $\mathsf{I}[\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A] = \mathsf{F}$; but this is impossible and we reject the assumption. The pattern is like $\sim$E in *ND*. This approach is particularly important insofar as we do not reason individually about each of the possible interpretations. This is nice in the sentential case, when there are too many to reason about conveniently. And in the quantificational case, we will not be *able* to argue individually about each of the possible interpretations. So we need to avoid talking about interpretations one by one.

Thus we arrive at two strategies: To show that an argument is invalid, we produce an interpretation, and show by the definitions that it makes the premises true and the conclusion not. That is what we did in (B) above. To show that an argument is valid, we assume the opposite, and show by the definitions that the assumption leads to contradiction. Again, that is what we did just above, at (E).

Before we get to the details, let us consider an important point about what we are trying to do: Our *reasoning* takes place in the metalanguage, based on the definitions stated in the metalanguage—where object-level expressions are *uninterpreted* apart from their definitions. To see this, ask yourself whether a sentence $\mathcal{P}$ conflicts with $\mathcal{P} \uparrow \mathcal{P}$. "Well," you might respond, "I have never encountered this symbol '$\uparrow$' before, so I am not in a position to say." But that is the point: whether $\mathcal{P}$ conflicts with $\mathcal{P} \uparrow \mathcal{P}$ depends entirely on a definition for *up arrow* '$\uparrow$'. As it happens, this symbol is typically read "not both" as given by what might be a further clause of ST.[2]

ST($\uparrow$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \uparrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$ (or both); otherwise $\mathsf{I}[(\mathcal{P} \uparrow \mathcal{Q})] = \mathsf{F}$.

The resultant table is,

T($\uparrow$)

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \uparrow \mathcal{Q}$
T	T	***F***
T	F	***T***
F	T	***T***
F	F	***T***

$\mathcal{P} \uparrow \mathcal{Q}$ is false when $\mathcal{P}$ and $\mathcal{Q}$ are both T, and otherwise true. Given this, $\mathcal{P}$ does conflict with $\mathcal{P} \uparrow \mathcal{P}$. Suppose $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P} \uparrow \mathcal{P}] = \mathsf{T}$; from the latter, by ST($\uparrow$), $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; either way, $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; but this is impossible given our assumption that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. In fact, $\mathcal{P} \uparrow \mathcal{P}$ has the same table as $\sim\mathcal{P}$, and $\mathcal{P} \uparrow (\mathcal{Q} \uparrow \mathcal{Q})$ the same as $\mathcal{P} \rightarrow \mathcal{Q}$.

(F)

$\mathcal{P}$	$\mathcal{P} \uparrow \mathcal{P}$
T	***F***
F	***T***

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \uparrow$	$(\mathcal{Q} \uparrow \mathcal{Q})$
T	T	***T***	F
T	F	***F***	T
F	T	***T***	F
F	F	***T***	T

From this, we *might* have treated $\sim$ and $\rightarrow$, and so $\wedge$, $\vee$, and $\leftrightarrow$, all as abbreviations for expressions whose only operator is $\uparrow$. At best, however, this leaves official expressions incredibly difficult to read. Here is the point that matters: Operators have their significance entirely from the definitions. In this chapter, we make metalinguistic claims *about* object expressions, where these can only be based on the definitions. $\mathcal{P}$ and $\mathcal{P} \uparrow \mathcal{P}$ do not themselves conflict, apart from the definition which makes $\mathcal{P}$ with $\mathcal{P} \uparrow \mathcal{P}$ have the consequence that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P}] = \mathsf{F}$. And similarly operators with which we are more familiar gain their significance from the definition. At every stage, it is the *definitions* which justify conclusions.

7.2 Sentential

With this much said, it remains possible to become confused about details while working with the definitions. It is one thing to be able to follow such reasoning—as I

[2]An alternative symbol is a simple vertical line, '|'. Then it is (the Sheffer) *stroke*.

hope you have been able to do—and another to produce it. The idea now is to make use of something at which we are already good, doing derivations, to further structure and guide the way we proceed. The result will be a sort of derivation system for reasoning with metalinguistic expressions. We build up this system in stages.

7.2.1 Truth

Let us begin with some notation. Where the script characters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \ldots$ represent object expressions in the usual way, let the Fraktur characters $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}, \ldots$ represent *metalinguistic* expressions ('$\mathfrak{A}$' is the Fraktur 'A'). Thus $\mathfrak{A}$ might represent an expression of the sort $\mathsf{I}[B] = \mathsf{T}$. Then $\Rightarrow$ and $\Leftrightarrow$ are the metalinguistic conditional and biconditional respectively; $\neg$, $\vartriangle$, $\triangledown$, and $\perp$ are metalinguistic negation, conjunction, disjunction, and contradiction. In practice, negation is indicated by the slash ($\nvDash$) as well.

Now consider the following restatement of definition ST. Each clause is given in both a positive and a negative form. For any sentences $\mathcal{P}$ and $\mathcal{Q}$ and interpretation I,

ST ($\sim$)	$\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$	$\mathsf{I}[\sim\mathcal{P}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T}$
($\rightarrow$)	$\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$	$\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

Given the new symbols, and that the definitions make a sentence F exactly when it is not true, this is a simple restatement of ST. As we develop our metalinguistic derivation system, we will treat the metalinguistic biconditionals both as (replacement) rules and as axioms. Thus, for example, by the first form of ST($\sim$) it will be legitimate to move directly from $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ to $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$, moving from left to right across the arrow; and similarly but in the other direction from $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ to $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$. Alternatively, it will be appropriate to assert by ST($\sim$) the entire biconditional, that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$. For now, we will mostly use the biconditionals, in the first form, as rules.

To manipulate the definitions, we require some rules. These are like ones you have seen before, only pitched at the metalinguistic level.

com	$(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow (\mathfrak{B} \triangledown \mathfrak{A})$		$(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow (\mathfrak{B} \vartriangle \mathfrak{A})$	
idm	$\mathfrak{A} \Leftrightarrow (\mathfrak{A} \triangledown \mathfrak{A})$		$\mathfrak{A} \Leftrightarrow (\mathfrak{A} \vartriangle \mathfrak{A})$	
dem	$\neg(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow (\neg\mathfrak{A} \triangledown \neg\mathfrak{B})$		$\neg(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	
cnj	$\dfrac{\mathfrak{A}, \mathfrak{B}}{\mathfrak{A} \vartriangle \mathfrak{B}}$	$\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{A}}$	$\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{B}}$	
dsj	$\dfrac{\mathfrak{A}}{\mathfrak{A} \triangledown \mathfrak{B}}$	$\dfrac{\mathfrak{B}}{\mathfrak{A} \triangledown \mathfrak{B}}$	$\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{A}}{\mathfrak{B}}$	$\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{B}}{\mathfrak{A}}$
neg	$\mathfrak{A} \Leftrightarrow \neg\neg\mathfrak{A}$	$\begin{array}{l} \mid\ \mathfrak{A} \\ \mid\ \perp \\ \neg\mathfrak{A} \end{array}$	$\begin{array}{l} \mid\ \neg\mathfrak{A} \\ \mid\ \perp \\ \mathfrak{A} \end{array}$	bot $\dfrac{\mathfrak{A}, \neg\mathfrak{A}}{\perp}$

Each of these should remind you of rules from *ND* or *ND+*. In practice, we will allow generalized versions of cnj that let us move directly from $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_n$ to $\mathfrak{A}_1 \vartriangle \mathfrak{A}_2 \vartriangle \ldots \vartriangle \mathfrak{A}_n$. Similarly, we will allow applications of dsj and dem that skip officially required applications of neg. Thus, for example, instead of going by dem from $\neg(\mathfrak{A} \vartriangle \neg\mathfrak{B})$, to $\neg\mathfrak{A} \triangledown \neg\neg\mathfrak{B}$ and then by neg to $\neg\mathfrak{A} \triangledown \mathfrak{B}$, we might move by dem directly from $\neg(\mathfrak{A} \vartriangle \neg\mathfrak{B})$, to $\neg\mathfrak{A} \triangledown \mathfrak{B}$. We will also allow a version of dsj with a pair of subderivations (as for $\vee$E in *ND*). All this should become more clear as we proceed.

With definition ST and these rules, we can begin to reason about consequences of the definition. Suppose we want to show that an interpretation with $\mathsf{I}[A] = \mathsf{I}[B] = \mathsf{T}$ is such that $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$.

(G)

1. $\mathsf{I}[A] = \mathsf{T}$	prem	
2. $\mathsf{I}[B] = \mathsf{T}$	prem	
3. $\mathsf{I}[\sim B] \neq \mathsf{T}$	2 ST($\sim$)	
4. $\mathsf{I}[A] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] \neq \mathsf{T}$	1,3 cnj	
5. $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$	4 ST($\to$)	
6. $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$	5 ST($\sim$)	

We are given that $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[B] = \mathsf{T}$. From the latter, by ST($\sim$), $\mathsf{I}[\sim B] \neq \mathsf{T}$; so $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[\sim B] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$.

The reasoning on the left is a metalinguistic *derivation* in the sense that every step is either a premise or results by a definition or rule. You should be able to follow each step. And these derivations can be worked "bottom-up" in the usual way: From the main operator, we expect to obtain $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$ by ST($\sim$); for this we need $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$; again by the main operator, we expect to get this by ST($\to$) and so set $\mathsf{I}[A] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] \neq \mathsf{T}$ as goal; this requires both conjuncts; but the first is given, and the second results from $\mathsf{I}[B] = \mathsf{T}$ by ST($\sim$).

On the right, we simply "tell the story" of the derivation—mirroring it step for step. This latter style is the one we want to develop. As we shall see, it gives us power to go beyond where the derivations will take us. But the derivations serve a purpose. If we can do them, we can *use* them to construct reasoning of the sort we want. Each stage on one side corresponds to one on the other. So the derivations can guide us as we construct our reasoning, and constrain the moves we make. Note: First, on the right, we replace line references with language ("from the latter") meant to serve the same purpose. Second, the metalinguistic symbols, $\Rightarrow$, $\Leftrightarrow$, $\neg$, $\vartriangle$, $\triangledown$, $\perp\!\!\!\perp$, are replaced with ordinary language on the right side. Finally, on the right, though we cite every *definition* when we use it, we do not cite the additional *rules* (in this case cnj). To the extent that you can, it is good to have one line depend on the one before or in the immediate neighborhood, so as to minimize the need for extended references in the written version. And in general, as much as possible, you should strive to put the reader (and yourself at a later time) in a position to follow your reasoning—supposing just a basic familiarity with the definitions.

Consider now another example. Suppose we want to show that an interpretation with $\mathsf{I}[B] \neq \mathsf{T}$ is such that $\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$.

(H)

1.	$\mathsf{I}[B] \neq \mathsf{T}$	prem
2.	$\mathsf{I}[\sim B] = \mathsf{T}$	1 ST($\sim$)
3.	$\mathsf{I}[A] \neq \mathsf{T} \bigtriangledown \mathsf{I}[\sim B] = \mathsf{T}$	2 dsj
4.	$\mathsf{I}[A \to \sim B] = \mathsf{T}$	3 ST($\to$)
5.	$\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$	4 ST($\sim$)

We are given that $\mathsf{I}[B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{T}$; so $\mathsf{I}[A] \neq \mathsf{T}$ or $\mathsf{I}[\sim B] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[A \to \sim B] = \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$.

Observe how ST($\to$) requires $\mathsf{I}[\mathcal{A}] \neq \mathsf{T} \bigtriangledown \mathsf{I}[\sim\mathcal{B}] = \mathsf{T}$ to obtain $\mathsf{I}[\mathcal{A} \to \sim\mathcal{B}] = \mathsf{T}$. Thus we obtain the disjunctive (3) in order to get (4). In contrast, on (5) of (G), ST($\to$) takes the conjunctive $\mathsf{I}[A] = \mathsf{T} \bigtriangleup \mathsf{I}[\sim B] \neq \mathsf{T}$ for $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$. Keep these cases separate in your mind: from the left-hand side of ST($\to$), a disjunction for a true conditional; and from the right-hand side, a conjunction for a conditional that is not true.

Here is another derivation of the same result, this time beginning with assumption of the opposite (with justification, 'assp') and breaking down to the parts, for an application of neg.

(I)

1.	$\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$	assp
2.	$\mathsf{I}[A \to \sim B] \neq \mathsf{T}$	1 ST($\sim$)
3.	$\mathsf{I}[A] = \mathsf{T} \bigtriangleup \mathsf{I}[\sim B] \neq \mathsf{T}$	2 ST($\to$)
4.	$\mathsf{I}[\sim B] \neq \mathsf{T}$	3 cnj
5.	$\mathsf{I}[B] = \mathsf{T}$	4 ST($\sim$)
6.	$\mathsf{I}[B] \neq \mathsf{T}$	prem
7.	$\perp$	5,6 bot
8.	$\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$	1-7 neg

Suppose $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$; then from ST($\sim$), $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[\sim B] \neq \mathsf{T}$; so $\mathsf{I}[\sim B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[B] = \mathsf{T}$. But we are given that $\mathsf{I}[B] \neq \mathsf{T}$. This is impossible; reject the assumption: $\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$.

Notice again that the conditional which is not true yields a conjunction. This version takes a couple more lines. But it works as well and provides a useful illustration of the (neg) rule. As usual, reasoning on the one side mirrors that on the other. So we can use the metalinguistic derivation as a guide for the reasoning on the right. Again, we leave out the special metalinguistic symbols. And again we cite all instances of definitions, but not the additional rules.

These derivations are structurally much simpler than ones you have seen before from *AD* and *ND*. The challenge is accommodating new notation with the different mix of rules. As you work these and other problems, you may find the sentential metalinguistic reference on page 333 helpful.

> *Some perspective:* Our reasoning takes place in the metalanguage. Special symbols, $\bigtriangleup$, $\bigtriangledown$, and such just *are* the metalinguistic 'and', 'or', and the like. Thus our work is in the usual language we use to state definitions. This language comes with its own interpretation. Taken this way, the metalinguistic derivations themselves constitute metalinguistic reasonings. It is true that metalinguistic rules are given in terms of form. We thus impose formal *constraints* on our reasoning. But we have not introduced a new language whose symbols require interpretation (as for $\mathcal{L}_q$), and do not *justify* inferences by form (as for *ND*). So we have not *formalized* the metalanguage. Rather we have adopted the formal constraints in order to guide and structure our reasoning.

E7.1. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] \neq \mathsf{T}$, and $\mathsf{I}[C] = \mathsf{T}$. For each of the following, produce a metalinguistic derivation, and then informal reasoning to demonstrate either that it is or is not true on I. Hint: You may find a quick row of the truth table helpful to let you see which you want to show. Also, (e) is much easier than it looks.

*a. $\sim B \to C$

*b. $\sim B \to \sim C$

c. $\sim[(A \to \sim B) \to \sim C]$

d. $\sim[A \to (B \to \sim C)]$

e. $\sim A \to [((A \to B) \to C) \to \sim(\sim C \to B)]$

7.2.2 Validity

So far we have been able to reason about ST and truth. Let us now extend results to validity. For this, we need to augment our metalinguistic derivation system. Let 'S' be a metalinguistic existential quantifier—it asserts the existence of some *object*. For now, 'S' will appear only in contexts asserting the existence of *interpretations*. Thus, for example, $S\mathsf{I}(\mathsf{I}[\mathscr{P}] = \mathsf{T})$ says there is an interpretation I such that $\mathsf{I}[\mathscr{P}] = \mathsf{T}$, and $\neg S\mathsf{I}(\mathsf{I}[\mathscr{P}] = \mathsf{T})$ says it is not the case that there is an interpretation I such that $\mathsf{I}[\mathscr{P}] = \mathsf{T}$. Given this, we can state SV as follows, again in positive and negative forms:

SV $\quad \neg S\mathsf{I}(\mathsf{I}[\mathscr{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathscr{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] \neq \mathsf{T}) \Leftrightarrow \mathscr{P}_1 \ldots \mathscr{P}_n \vDash_s \mathscr{Q}$

$\quad\quad S\mathsf{I}(\mathsf{I}[\mathscr{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathscr{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] \neq \mathsf{T}) \Leftrightarrow \mathscr{P}_1 \ldots \mathscr{P}_n \nvDash_s \mathscr{Q}$

These should look familiar. An argument is valid when it is *not* the case that there is some interpretation that makes the premises true and the conclusion not. An argument is invalid if there is some interpretation that makes the premises true and the conclusion not.

Again, we need rules to manipulate the new operator. In general, whenever a metalinguistic *term* $\mathfrak{t}$ first appears outside the scope of a metalinguistic quantifier, it is labeled *arbitrary* or *particular*. For the sentential case, terms will be of the sort I, J, ... for *interpretations*, and mostly labeled 'particular' when they first appear apart from the quantifier S. Say $\mathfrak{A}[\mathfrak{t}]$ is some metalinguistic expression in which term $\mathfrak{t}$ appears, and $\mathfrak{A}[\mathfrak{u}]$ is like $\mathfrak{A}[\mathfrak{t}]$ but with free instances of $\mathfrak{t}$ replaced by $\mathfrak{u}$. Perhaps $\mathfrak{A}[\mathfrak{t}]$ is $\mathsf{I}[A] = \mathsf{T}$ and $\mathfrak{A}[\mathfrak{u}]$ is $\mathsf{J}[A] = \mathsf{T}$. Then,

exs $\quad \dfrac{\mathfrak{A}[\mathfrak{u}]}{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary or particular $\qquad \dfrac{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ particular and new

As an instance of the left-hand "introduction" rule, we might move from $\mathsf{J}[A] = \mathsf{T}$, for a J labeled either arbitrary or particular, to $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$. If interpretation J is such that $\mathsf{J}[A] = \mathsf{T}$, then there is *some* interpretation I such that $\mathsf{I}[A] = \mathsf{T}$. For the other "exploitation" rule, we may move from $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$ to the result that $\mathsf{J}[A] = \mathsf{T}$ so long as J is identified as *particular* and is new to the derivation, in the sense required for $\exists$E in Chapter 6. In particular, it must be that the term does not so-far appear outside the scope of a metalinguistic quantifier, and does not appear free in current goal expressions. Given that some I is such that $\mathsf{I}[A] = \mathsf{T}$, we set up J as a particular interpretation for which it is so.[3]

In addition, it will be helpful to allow a rule which lets us make assertions by *inspection* about already given interpretations—and we will limit justifications by (ins) just to assertions about interpretations (and, later, variable assignments). Thus, for example, in the context of an interpretation I on which $\mathsf{I}[A] = \mathsf{T}$, we might allow,

n. $\mathsf{I}[A] = \mathsf{T}$ ins (I particular)

as a line of one of our derivations. In this case, I is a *name* of the interpretation, and listed as particular on first use.

Now suppose we want to show that $(B \to \sim D), \sim B \not\models_s D$. Recall that our strategy for showing that an argument is invalid is to *produce* an interpretation, and show that it makes the premises true and the conclusion not. So consider an interpretation J such that $\mathsf{J}[B] \neq \mathsf{T}$ and $\mathsf{J}[D] \neq \mathsf{T}$. (A quick row of the truth table might help to identify this as the interpretation we want to consider.)

(J)

1.	$\mathsf{J}[B] \neq \mathsf{T}$	ins (J particular)
2.	$\mathsf{J}[B] \neq \mathsf{T} \triangledown \mathsf{J}[\sim D] = \mathsf{T}$	1 dsj
3.	$\mathsf{J}[B \to \sim D] = \mathsf{T}$	2 ST($\to$)
4.	$\mathsf{J}[\sim B] = \mathsf{T}$	1 ST($\sim$)
5.	$\mathsf{J}[D] \neq \mathsf{T}$	ins
6.	$\mathsf{J}[B \to \sim D] = \mathsf{T} \vartriangle \mathsf{J}[\sim B] = \mathsf{T} \vartriangle \mathsf{J}[D] \neq \mathsf{T}$	3,4,5 cnj
7.	$S\mathsf{I}(\mathsf{I}[B \to \sim D] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] = \mathsf{T} \vartriangle \mathsf{I}[D] \neq \mathsf{T})$	6 exs
8.	$B \to \sim D, \sim B \not\models_s D$	7 SV

(1) and (5) are by inspection of the interpretation J, where an individual name is always labeled "particular" when it first appears. At (6) we have a conclusion about interpretation J, and at (7) we generalize to the existential, for an application of SV at (8). Here is the corresponding informal reasoning:

> $\mathsf{J}[B] \neq \mathsf{T}$; so either $\mathsf{J}[B] \neq \mathsf{T}$ or $\mathsf{J}[\sim D] = \mathsf{T}$; so by ST($\to$), $\mathsf{J}[B \to \sim D] = \mathsf{T}$. But since $\mathsf{J}[B] \neq \mathsf{T}$, by ST($\sim$), $\mathsf{J}[\sim B] = \mathsf{T}$. And $\mathsf{J}[D] \neq \mathsf{T}$. So $\mathsf{J}[B \to \sim D] = \mathsf{T}$ and $\mathsf{J}[\sim B] = \mathsf{T}$ but $\mathsf{J}[D] \neq \mathsf{T}$. So there is an interpretation I such that $\mathsf{I}[B \to \sim D] = \mathsf{T}$ and $\mathsf{I}[\sim B] = \mathsf{T}$ but $\mathsf{I}[D] \neq \mathsf{T}$. So by SV, $B \to \sim D, \sim B \not\models_s D$.

[3]Insofar as I is bound in $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$, term I may itself be new in the sense that it does not so-far appear outside the scope of a quantifier. Thus we may be justified in moving from $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$ to $\mathsf{I}[A] = \mathsf{T}$, with I particular. However, as a matter of style, we will typically switch terms upon application of the exs rule.

It should be clear that this reasoning reflects that of the derivation. We show the argument is invalid by showing that there exists an interpretation on which the premises are true and the conclusion is not.

Say we want to show that $\sim(A \rightarrow B) \vDash_s A$. To show that an argument is valid, our idea has been to assume otherwise and show that the assumption leads to contradiction. So we might reason as follows:

(K)

1.	$\sim(A \rightarrow B) \nvDash_s A$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\sim(A \rightarrow B)] = \mathsf{T} \mathbin{\triangle} \mathsf{I}[A] \neq \mathsf{T})$	1 SV
3.	$\mathsf{J}[\sim(A \rightarrow B)] = \mathsf{T} \mathbin{\triangle} \mathsf{J}[A] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\sim(A \rightarrow B)] = \mathsf{T}$	3 cnj
5.	$\mathsf{J}[A \rightarrow B] \neq \mathsf{T}$	4 ST($\sim$)
6.	$\mathsf{J}[A] = \mathsf{T} \mathbin{\triangle} \mathsf{J}[B] \neq \mathsf{T}$	5 ST($\rightarrow$)
7.	$\mathsf{J}[A] = \mathsf{T}$	6 cnj
8.	$\mathsf{J}[A] \neq \mathsf{T}$	3 cnj
9.	$\perp$	7,8 bot
10.	$\sim(A \rightarrow B) \vDash_s A$	1-9 neg

Suppose $\sim(A \rightarrow B) \nvDash_s A$; then by SV there is some I such that $\mathsf{I}[\sim(A \rightarrow B)] = \mathsf{T}$ and $\mathsf{I}[A] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\sim(A \rightarrow B)] = \mathsf{T}$ and $\mathsf{J}[A] \neq \mathsf{T}$. From the former, by ST($\sim$), $\mathsf{J}[A \rightarrow B] \neq \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$. So both $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[A] \neq \mathsf{T}$. This is impossible; reject the assumption: $\sim(A \rightarrow B) \vDash_s A$.

At (2) we have the result that there is some interpretation on which the premise is true and the conclusion is not. At (3), we set up to reason about a particular J for which this is so. J does not so-far appear in the derivation, and does not appear in the goal at (9). So we instantiate to it. This puts us in a position to reason by ST. The pattern is typical. Given that the assumption leads to contradiction, we are justified in rejecting the assumption, and thus conclude that the argument is valid. It is important that we are able to show an argument is valid without reasoning individually about every possible interpretation of the basic sentences!

Notice that we can also reason generally about *forms*. Here is a case of that sort:

T7.4s. $\vDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$

1.	$\nvDash_s (\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})] \neq \mathsf{T})$	1 SV
3.	$\mathsf{J}[(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\sim\mathcal{Q} \to \sim\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[(\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q}] \neq \mathsf{T}$	3 ST($\to$)
5.	$\mathsf{J}[(\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q}] \neq \mathsf{T}$	4 cnj
6.	$\mathsf{J}[\sim\mathcal{Q} \to \mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	5 ST($\to$)
7.	$\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	6 cnj
8.	$\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$	7 ST($\sim$)
9.	$\mathsf{J}[\sim\mathcal{Q} \to \mathcal{P}] = \mathsf{T}$	6 cnj
10.	$\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T} \triangledown \mathsf{J}[\mathcal{P}] = \mathsf{T}$	9 ST($\to$)
11.	$\mathsf{J}[\mathcal{P}] = \mathsf{T}$	10,8 dsj
12.	$\mathsf{J}[\sim\mathcal{Q} \to \sim\mathcal{P}] = \mathsf{T}$	4 cnj
13.	$\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T} \triangledown \mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$	12 ST($\to$)
14.	$\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$	13,8 dsj
15.	$\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$	14 ST($\sim$)
16.	$\perp$	11,15 bot
17.	$\vDash_s (\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$	1-16 neg

> Suppose $\nvDash_s (\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$; then by SV there is some I such that $\mathsf{I}[(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{J}[\sim\mathcal{Q} \to \sim\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[(\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q}] \neq \mathsf{T}$; from the latter, by ST($\to$), $\mathsf{J}[\sim\mathcal{Q} \to \mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$; from the second of these, by ST($\sim$), $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$. Since $\mathsf{J}[\sim\mathcal{Q} \to \mathcal{P}] = \mathsf{T}$, by ST($\to$), $\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T}$ or $\mathsf{J}[\mathcal{P}] = \mathsf{T}$; but $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$, so $\mathsf{J}[\mathcal{P}] = \mathsf{T}$. Since $\mathsf{J}[\sim\mathcal{Q} \to \sim\mathcal{P}] = \mathsf{T}$, by ST($\to$), $\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T}$ or $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$; but $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$, so $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s (\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$.

Observe that the steps represented by (11) and (14) are not by cnj but by the dsj rule with $\mathfrak{A} \triangledown \mathfrak{B}$ and $\neg\mathfrak{A}$ for the result that $\mathfrak{B}$.[4] Observe also that contradictions are obtained at the *metalinguistic* level. Thus $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ at (11) does not contradict $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$ at (14). Of course, it is a short step to the result that $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$ which do contradict. As a general point of strategy, it is much easier to manage a conditional that is not true than a conditional that is true—for a conditional that is not true yields a conjunctive result, and one that is true a disjunctive result. Thus we begin above at (5) and (6) with the conditional that is not true, and *use* the results to set up applications of dsj. This is typical. Similarly we can show,

T7.1s. $\mathcal{P}, \mathcal{P} \to \mathcal{Q} \vDash_s \mathcal{Q}$

T7.2s. $\vDash_s \mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

T7.3s. $\vDash_s (\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

---

[4] Or, rather, we have $\neg\mathfrak{A} \triangledown \mathfrak{B}$ and $\mathfrak{A}$—and thus skip application of neg to obtain the proper $\neg\neg\mathfrak{A}$ for this application of dsj.

T7.1s–T7.4s should remind you of the axioms and rule of the sentential system *ADs* from Chapter 3. These results (or, rather, analogues for the quantificational case) play an important role for things to come.

Again to show that an argument is invalid, produce an interpretation; then use it for a demonstration that there exists an interpretation that makes premises true and the conclusion not. To show that an argument is valid, suppose otherwise; then demonstrate that your assumption leads to contradiction. The derivations then provide the pattern for your informal reasoning.

E7.2. Produce a metalinguistic derivation, and then informal reasoning to demonstrate each of the following. To show invalidity, you will have to *produce* an interpretation to which your argument refers.

*a. $A \rightarrow B, \sim A \nvDash_s \sim B$

*b. $A \rightarrow B, \sim B \vDash_s \sim A$

c. $A \rightarrow B, B \rightarrow C, C \rightarrow D \vDash_s A \rightarrow D$

d. $A \rightarrow B, B \rightarrow \sim A \vDash_s \sim A$

e. $A \rightarrow B, \sim A \rightarrow \sim B \nvDash_s \sim(A \rightarrow \sim B)$

f. $(\sim A \rightarrow B) \rightarrow A \vDash_s \sim A \rightarrow \sim B$

g. $\sim A \rightarrow \sim B, B \vDash_s \sim(B \rightarrow \sim A)$

h. $A \rightarrow B, \sim B \rightarrow A \nvDash_s A \rightarrow \sim B$

i. $\nvDash_s [(A \rightarrow B) \rightarrow (A \rightarrow C)] \rightarrow [(A \rightarrow B) \rightarrow C]$

j. $\vDash_s (A \rightarrow B) \rightarrow [(B \rightarrow \sim C) \rightarrow (C \rightarrow \sim A)]$

E7.3. Provide demonstrations for T7.1s–T7.3s in the informal style. Hint: You may or may not find metalinguistic derivations helpful as a guide.

7.2.3 Derived Clauses

Finally, for this section on sentential forms, we expand the range of our results by introducing derived clauses to definition ST. For this, we require some rules for $\Rightarrow$ and $\Leftrightarrow$.

cnd	$\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$	$\begin{array}{l} \left\lvert \begin{array}{l} \mathfrak{A} \\ \hline \mathfrak{B} \end{array} \right. \\ \mathfrak{A} \Rightarrow \mathfrak{B} \end{array}$	$\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{C}}{\mathfrak{A} \Rightarrow \mathfrak{C}}$	
bcnd	$\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$	$\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B}}{\mathfrak{A}}$	$\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{A}}{\mathfrak{A} \Leftrightarrow \mathfrak{B}}$	$\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B} \Leftrightarrow \mathfrak{C}}{\mathfrak{A} \Leftrightarrow \mathfrak{C}}$

We will also allow versions of cnd and bcnd which move from, say, $\mathfrak{A} \Rightarrow \mathfrak{B}$ and $\neg\mathfrak{B}$ to $\neg\mathfrak{A}$, and from $\mathfrak{A} \Leftrightarrow \mathfrak{B}$ and $\neg\mathfrak{A}$, to $\neg\mathfrak{B}$ (like MT and NB from *ND+*). And we will allow generalized versions of these rules moving directly from, say, $\mathfrak{A} \Rightarrow \mathfrak{B}$, $\mathfrak{B} \Rightarrow \mathfrak{C}$, and $\mathfrak{C} \Rightarrow \mathfrak{D}$ to $\mathfrak{A} \Rightarrow \mathfrak{D}$; and similarly, from $\mathfrak{A} \Leftrightarrow \mathfrak{B}$, $\mathfrak{B} \Leftrightarrow \mathfrak{C}$, and $\mathfrak{C} \Leftrightarrow \mathfrak{D}$ to $\mathfrak{A} \Leftrightarrow \mathfrak{D}$. In this last case, the natural informal description is, $\mathfrak{A}$ iff $\mathfrak{B}$; $\mathfrak{B}$ iff $\mathfrak{C}$; $\mathfrak{C}$ iff $\mathfrak{D}$; so $\mathfrak{A}$ iff $\mathfrak{D}$. In real cases, however, repetition of terms can be awkward and get in the way of reading. In practice, then, the pattern collapses to, $\mathfrak{A}$ iff $\mathfrak{B}$; iff $\mathfrak{C}$; iff $\mathfrak{D}$; so $\mathfrak{A}$ iff $\mathfrak{D}$—where this is understood as in the official version.

Also, when demonstrating that $\mathfrak{A} \Rightarrow \mathfrak{B}$, in many cases, it is helpful to get $\mathfrak{B}$ by neg; officially, the pattern is as on the left,

$$
\begin{array}{l}
\left|\begin{array}{l}
\underline{\mathfrak{A}} \\
\left|\begin{array}{l} \underline{\neg\mathfrak{B}} \\ \bot \end{array}\right. \\
\mathfrak{B}
\end{array}\right. \\
\mathfrak{A} \Rightarrow \mathfrak{B}
\end{array}
$$

But the result is automatic once we derive a contradiction from $\mathfrak{A}$ and $\neg\mathfrak{B}$; so, in practice, this pattern collapses into:

$$
\begin{array}{l}
\left|\begin{array}{l}
\underline{\mathfrak{A} \vartriangle \neg\mathfrak{B}} \\
\bot
\end{array}\right. \\
\mathfrak{A} \Rightarrow \mathfrak{B}
\end{array}
$$

So to demonstrate a conditional, it is enough to derive a contradiction from the antecedent and negation of the consequent. Let us also include among our metalinguistic definitions, abb as a metalinguistic counterpart to abv (as for example on page 300). This is to be understood as justifying biconditionals $\mathfrak{A}[\mathscr{P}'] \Leftrightarrow \mathfrak{A}[\mathscr{P}]$ where $\mathscr{P}'$ abbreviates $\mathscr{P}$. So, for example, by abb $\mathsf{I}[\mathscr{P}'] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T}$. Such a biconditional can be used as either an axiom or a rule.

We are now in a position to produce derived clauses for ST. We have already seen derived tables from Chapter 4. Now we demonstrate the conditions.

$$
\begin{array}{lll}
\text{ST}' & (\wedge) & \mathsf{I}[\mathscr{P} \wedge \mathscr{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] = \mathsf{T} \\
 & & \mathsf{I}[\mathscr{P} \wedge \mathscr{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathscr{Q}] \neq \mathsf{T} \\
 & (\vee) & \mathsf{I}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T} \triangledown \mathsf{I}[\mathscr{Q}] = \mathsf{T} \\
 & & \mathsf{I}[\mathscr{P} \vee \mathscr{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] \neq \mathsf{T} \\
 & (\leftrightarrow) & \mathsf{I}[\mathscr{P} \leftrightarrow \mathscr{Q}] = \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] = \mathsf{T}) \triangledown (\mathsf{I}[\mathscr{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] \neq \mathsf{T}) \\
 & & \mathsf{I}[\mathscr{P} \leftrightarrow \mathscr{Q}] \neq \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] \neq \mathsf{T}) \triangledown (\mathsf{I}[\mathscr{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] = \mathsf{T})
\end{array}
$$

Again, you should recognize the derived clauses based on what you already know from truth tables.

First, consider the positive form for ST$'(\wedge)$. We reason about the arbitrary interpretation. The demonstration begins by abb, and strings together biconditionals to reach the final result.

$$
\text{(L)}\quad
\begin{array}{rll}
1. & \mathsf{I}[\mathscr{P} \wedge \mathscr{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\sim(\mathscr{P} \rightarrow \sim\mathscr{Q})] = \mathsf{T} & \text{abb (I arbitrary)} \\
2. & \mathsf{I}[\sim(\mathscr{P} \rightarrow \sim\mathscr{Q})] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P} \rightarrow \sim\mathscr{Q}] \neq \mathsf{T} & \text{ST}(\sim) \\
3. & \mathsf{I}[\mathscr{P} \rightarrow \sim\mathscr{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathscr{Q}] \neq \mathsf{T} & \text{ST}(\rightarrow) \\
4. & \mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathscr{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] = \mathsf{T} & \text{ST}(\sim) \\
5. & \mathsf{I}[\mathscr{P} \wedge \mathscr{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathscr{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathscr{Q}] = \mathsf{T} & \text{1,2,3,4 bcnd}
\end{array}
$$

This time the interpretation is arbitrary insofar as the reasoning applies to any interpretation whatsoever. This derivation puts together a string of biconditionals of the form $\mathfrak{A} \Leftrightarrow \mathfrak{B}$, $\mathfrak{B} \Leftrightarrow \mathfrak{C}$, $\mathfrak{C} \Leftrightarrow \mathfrak{D}$, $\mathfrak{D} \Leftrightarrow \mathfrak{E}$; the conclusion follows by bcnd. Notice that we use the abbreviation and first two definitions as axioms, to state the biconditionals. Technically, (4) results from an implicit $\mathfrak{A} \Leftrightarrow \mathfrak{A}$—that is, $\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$—followed by ST($\sim$) as a replacement rule, substituting $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ for $\mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$ on the right-hand side. In the "collapsed" biconditional form, the result is as follows:

> By abb, $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{P} \rightarrow \sim\mathcal{Q}] \neq \mathsf{T}$; by ST($\rightarrow$), iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$. So $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$.

In this abbreviated form, each stage implies the next from start to finish. But similarly, each stage implies the one before from finish to start. So one might think of it as demonstrating conditionals in both directions all at once for eventual application of bcnd. Because we have just shown a biconditional, it follows immediately that $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ just in case the right hand side fails—just in case one of $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$. However, we can also make the point directly.

> By abb, $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})] \neq \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{P} \rightarrow \sim\mathcal{Q}] = \mathsf{T}$; by ST($\rightarrow$), iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{Q}] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$. So $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$.

Reasoning for ST$'$($\vee$) is similar. For ST$'$($\leftrightarrow$) it will be helpful to introduce, as a derived rule, a sort of distribution principle.

dst $\quad [(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Leftrightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$

To show this, our basic idea is to obtain the conditional going in both directions, and then apply bcnd. The argument from left to right is given in box (N) on the following page. The conditional is demonstrated in the "collapsed" form, where we assume the antecedent with the negation of the consequent and go for a contradiction. Note the little subderivation at (11)–(14); we have accumulated disjunctions at (3), (4), (8), and (10), but do not have any of the "sides"; to make headway, we assume the negation of one side; this feeds into dsj and neg (the idea is related to SC4). Demonstration of the conditional in the other direction is left as an exercise. Given dst, you should be able to demonstrate ST($\leftrightarrow$), also in the collapsed biconditional style. You will begin by observing by abb that $\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\sim((\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \rightarrow \mathcal{P}))] = \mathsf{T}$; by ST($\sim$) iff. . . . The negative side is relatively straightforward, and does not require dst.

Having established the derived clauses for ST$'$, we can use them directly in our reasoning. Thus, for example, let us show that $B \vee (A \wedge \sim C), (C \rightarrow A) \leftrightarrow B \not\vDash_s \sim(A \wedge C)$. For this, consider an interpretation J such that $\mathsf{J}[A] = \mathsf{J}[B] = \mathsf{J}[C] = \mathsf{T}$.

(M)

1.	$\mathsf{J}[B] = \mathsf{T}$	ins (J particular)
2.	$\mathsf{J}[B] = \mathsf{T} \triangledown \mathsf{J}[A \wedge \sim C] = \mathsf{T}$	1 dsj
3.	$\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$	2 ST′(∨)
4.	$\mathsf{J}[A] = \mathsf{T}$	ins
5.	$\mathsf{J}[C] \neq \mathsf{T} \triangledown \mathsf{J}[A] = \mathsf{T}$	4 dsj
6.	$\mathsf{J}[C \to A] = \mathsf{T}$	5 ST(→)
7.	$\mathsf{J}[C \to A] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}$	10,1 cnj
8.	$(\mathsf{J}[C \to A] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}) \triangledown (\mathsf{J}[C \to A] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T})$	7 dsj
9.	$\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$	8 ST′(↔)
10.	$\mathsf{J}[C] = \mathsf{T}$	ins
11.	$\mathsf{J}[A] = \mathsf{T} \vartriangle \mathsf{J}[C] = \mathsf{T}$	4,10 cnj
12.	$\mathsf{J}[A \wedge C] = \mathsf{T}$	11 ST′(∧)
13.	$\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$	12 ST(∼)
14.	$\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T} \vartriangle \mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$	3,9,13 cnj
15.	$\mathcal{S}\mathsf{I}[\mathsf{I}[B \vee (A \wedge \sim C)] = \mathsf{T} \vartriangle \mathsf{I}[(C \to A) \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[\sim(A \wedge C)] \neq \mathsf{T}]$	14 exs
16.	$B \vee (A \wedge \sim C), (C \to A) \leftrightarrow B \not\vDash_s \sim(A \wedge C)$	15 SV

Since $\mathsf{J}[B] = \mathsf{T}$, either $\mathsf{J}[B] = \mathsf{T}$ or $\mathsf{J}[A \wedge \sim C] = \mathsf{T}$; so by ST′(∨), $\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$. Since $\mathsf{J}[A] = \mathsf{T}$, either $\mathsf{J}[C] \neq \mathsf{T}$ or $\mathsf{J}[A] = \mathsf{T}$; so by ST(→), $\mathsf{J}[C \to A] = \mathsf{T}$; so both $\mathsf{J}[C \to A] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$; so either both $\mathsf{J}[C \to A] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$ or both $\mathsf{J}[C \to A] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; so by ST′(↔), $\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$. Since $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[C] = \mathsf{T}$, by ST′(∧), $\mathsf{J}[A \wedge C] = \mathsf{T}$; so by ST(∼), $\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$. So $\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$ and $\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$ but $\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$; so there exists an interpretation I such that $\mathsf{I}[B \vee (A \wedge \sim C)] = \mathsf{T}$ and $\mathsf{I}[(C \to A) \leftrightarrow B] = \mathsf{T}$ but $\mathsf{I}[\sim(A \wedge C)] \neq \mathsf{T}$; so by SV, $B \vee (A \wedge \sim C), (C \to A) \leftrightarrow B \not\vDash_s \sim(A \wedge C)$.

(N)

1.	$[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \vartriangle \neg[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	assp
2.	$(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})$	1 cnj
3.	$\neg\mathfrak{A} \triangledown \mathfrak{B}$	2 cnj
4.	$\neg\mathfrak{B} \triangledown \mathfrak{A}$	2 cnj
5.	$\neg[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	1 cnj
6.	$\neg(\mathfrak{A} \vartriangle \mathfrak{B}) \vartriangle \neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	5 dem
7.	$\neg(\mathfrak{A} \vartriangle \mathfrak{B})$	6 cnj
8.	$\neg\mathfrak{A} \triangledown \neg\mathfrak{B}$	7 dem
9.	$\neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	6 cnj
10.	$\mathfrak{A} \triangledown \mathfrak{B}$	9 dem
11.	$\quad \mathfrak{A}$	assp
12.	$\quad \mathfrak{B}$	3,11 dsj
13.	$\quad \neg\mathfrak{B}$	8,11 dsj
14.	$\quad \perp$	12,13 bot
15.	$\neg\mathfrak{A}$	11-14 neg
16.	$\neg\mathfrak{B}$	4,15 dsj
17.	$\mathfrak{B}$	10,15 dsj
18.	$\perp$	17,16 bot
19.	$[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Rightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	1-18 cnd

Observe the use of dsj at (8) to feed into ST$'$($\leftrightarrow$) at (9). This is no different than we have done before, only with the relatively complex expressions.

Similarly we can show that $A \rightarrow (B \vee C)$, $C \leftrightarrow B$, $\sim C \models_s \sim A$. As usual, our strategy is to assume otherwise, and go for contradiction.

(O)

1.	$A \rightarrow (B \vee C), C \leftrightarrow B, \sim C \not\models_s \sim A$	assp
2.	$S\mathsf{I}(\mathsf{I}[A \rightarrow (B \vee C)] = \mathsf{T} \vartriangle \mathsf{I}[C \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[\sim C] = \mathsf{T} \vartriangle \mathsf{I}[\sim A] \neq \mathsf{T})$	1 SV
3.	$\mathsf{J}[A \rightarrow (B \vee C)] = \mathsf{T} \vartriangle \mathsf{J}[C \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[\sim C] = \mathsf{T} \vartriangle \mathsf{J}[\sim A] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\sim C] = \mathsf{T}$	3 cnj
5.	$\mathsf{J}[C] \neq \mathsf{T}$	4 ST($\sim$)
6.	$\mathsf{J}[C] \neq \mathsf{T} \triangledown \mathsf{J}[B] \neq \mathsf{T}$	5 dsj
7.	$\neg(\mathsf{J}[C] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T})$	6 dem
8.	$\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$	3 cnj
9.	$(\mathsf{J}[C] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}) \triangledown (\mathsf{J}[C] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T})$	8 ST$'$($\leftrightarrow$)
10.	$\mathsf{J}[C] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T}$	9,7 dsj
11.	$\mathsf{J}[\sim A] \neq \mathsf{T}$	3 cnj
12.	$\mathsf{J}[A] = \mathsf{T}$	11 ST($\sim$)
13.	$\mathsf{J}[A \rightarrow (B \vee C)] = \mathsf{T}$	3 cnj
14.	$\mathsf{J}[A] \neq \mathsf{T} \triangledown \mathsf{J}[B \vee C] = \mathsf{T}$	13 ST($\rightarrow$)
15.	$\mathsf{J}[B \vee C] = \mathsf{T}$	14,12 dsj
16.	$\mathsf{J}[B] = \mathsf{T} \triangledown \mathsf{J}[C] = \mathsf{T}$	15 ST$'$($\vee$)
17.	$\mathsf{J}[B] \neq \mathsf{T}$	10 cnj
18.	$\mathsf{J}[C] = \mathsf{T}$	16,17 dsj
19.	$\perp$	18,5 bot
20.	$A \rightarrow (B \vee C), C \leftrightarrow B, \sim C \models_s \sim A$	1-20 neg

Suppose $A \rightarrow (B \vee C)$, $C \leftrightarrow B$, $\sim C \not\models_s \sim A$; then by SV there is some I such that $\mathsf{I}[A \rightarrow (B \vee C)] = \mathsf{T}$ and $\mathsf{I}[C \leftrightarrow B] = \mathsf{T}$ and $\mathsf{I}[\sim C] = \mathsf{T}$ but $\mathsf{I}[\sim A] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[A \rightarrow (B \vee C)] = \mathsf{T}$ and $\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$ and $\mathsf{J}[\sim C] = \mathsf{T}$ but $\mathsf{J}[\sim A] \neq \mathsf{T}$. Since $\mathsf{J}[\sim C] = \mathsf{T}$, by ST($\sim$), $\mathsf{J}[C] \neq \mathsf{T}$; so either $\mathsf{J}[C] \neq \mathsf{T}$ or $\mathsf{J}[B] \neq \mathsf{T}$; so it is not the case that both $\mathsf{J}[C] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$. But $\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$; so by ST$'$($\leftrightarrow$), both $\mathsf{J}[C] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$, or both $\mathsf{J}[C] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; but not the former, so $\mathsf{J}[C] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$. $\mathsf{J}[\sim A] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[A] = \mathsf{T}$. But $\mathsf{J}[A \rightarrow (B \vee C)] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B \vee C] = \mathsf{T}$; but $\mathsf{J}[A] = \mathsf{T}$; so $\mathsf{J}[B \vee C] = \mathsf{T}$; so by ST$'$($\vee$), $\mathsf{J}[B] = \mathsf{T}$ or $\mathsf{J}[C] = \mathsf{T}$; but $\mathsf{J}[B] \neq \mathsf{T}$; so $\mathsf{J}[C] = \mathsf{T}$; but $\mathsf{J}[C] \neq \mathsf{T}$. This is impossible; reject the assumption: $A \rightarrow (B \vee C)$, $C \leftrightarrow B$, $\sim C \models_s \sim A$.

Note the move on lines (5)–(7) where we use dsj with dem to convert $\mathsf{J}[C] \neq \mathsf{T}$ into a negation useful at (10).

Though the metalinguistic derivations are useful to discipline the way we reason, in the end, you may find the written versions to be both quicker and easier to follow. As you work the exercises, try to free yourself from the derivations to work the informal versions independently—though you should continue to use derivations as a check for your work.

Metalinguistic Quick Reference (sentential)

DEFINITIONS:

ST $(\sim)$ $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ $\qquad$ $\mathsf{I}[\sim\mathcal{P}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T}$

$(\to)$ $\mathsf{I}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$ $\qquad$ $\mathsf{I}[\mathcal{P} \to \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

ST$'$ $(\wedge)$ $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

$\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

$(\vee)$ $\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

$\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

$(\leftrightarrow)$ $\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T})$

$\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T})$

SV $\neg\mathcal{S}\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash_s \mathcal{Q}$

$\mathcal{S}\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash_s \mathcal{Q}$

abb *Abbreviation* allows $\mathfrak{A}[\mathcal{P}'] \Leftrightarrow \mathfrak{A}[\mathcal{P}]$ where $\mathcal{P}'$ abbreviates $\mathcal{P}$.

RULES:

com $(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow (\mathfrak{B} \triangledown \mathfrak{A})$ $\qquad$ $(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow (\mathfrak{B} \vartriangle \mathfrak{A})$

idm $\mathfrak{A} \Leftrightarrow (\mathfrak{A} \triangledown \mathfrak{A})$ $\qquad$ $\mathfrak{A} \Leftrightarrow (\mathfrak{A} \vartriangle \mathfrak{A})$

dem $\neg(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow (\neg\mathfrak{A} \triangledown \neg\mathfrak{B})$ $\qquad$ $\neg(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$

cnj $\dfrac{\mathfrak{A}, \mathfrak{B}}{\mathfrak{A} \vartriangle \mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{A}}$ $\qquad$ $\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{B}}$

dsj $\dfrac{\mathfrak{A}}{\mathfrak{A} \triangledown \mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{B}}{\mathfrak{A} \triangledown \mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{A}}{\mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{B}}{\mathfrak{A}}$

neg $\mathfrak{A} \Leftrightarrow \neg\neg\mathfrak{A}$ $\qquad$ $\begin{array}{l} \begin{array}{|l} \mathfrak{A} \\ \hline \bot \end{array} \\ \neg\mathfrak{A} \end{array}$ $\qquad$ $\begin{array}{l} \begin{array}{|l} \neg\mathfrak{A} \\ \hline \bot \end{array} \\ \mathfrak{A} \end{array}$ $\qquad$ bot $\dfrac{\mathfrak{A}, \neg\mathfrak{A}}{\bot}$

exs $\dfrac{\mathfrak{A}[\mathfrak{u}]}{\mathcal{S}\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary or particular $\qquad$ $\dfrac{\mathcal{S}\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ particular and new

cnd $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$ $\qquad$ $\begin{array}{l} \begin{array}{|l} \mathfrak{A} \\ \hline \mathfrak{B} \end{array} \\ \mathfrak{A} \Rightarrow \mathfrak{B} \end{array}$ $\qquad$ $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{C}}{\mathfrak{A} \Rightarrow \mathfrak{C}}$ $\qquad$ $\begin{array}{l} \begin{array}{|l} \mathfrak{A} \vartriangle \neg\mathfrak{B} \\ \hline \bot \end{array} \\ \mathfrak{A} \Rightarrow \mathfrak{B} \end{array}$

bcnd $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B}}{\mathfrak{A}}$ $\qquad$ $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{A}}{\mathfrak{A} \Leftrightarrow \mathfrak{B}}$ $\qquad$ $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B} \Leftrightarrow \mathfrak{C}}{\mathfrak{A} \Leftrightarrow \mathfrak{C}}$

dst $[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Leftrightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$

ins *Inspection* allows assertions about interpretations and variable assignments.

E7.4. Produce informal reasoning to demonstrate each of the following.

a. $A \to (B \wedge C), \sim C \vDash_s \sim A$

*b. $\sim(A \leftrightarrow B), \sim A, \sim B \vDash_s C \wedge \sim C$

*c. $\sim(\sim A \wedge \sim B) \nvDash_s A \wedge B$

d. $\sim A \leftrightarrow \sim B \vDash_s B \to A$

e. $A \wedge (B \to C) \nvDash_s (A \wedge C) \vee (A \wedge B)$

f. $[(C \vee D) \wedge B] \to A, D \vDash_s B \to A$

g. $\nvDash_s A \vee ((C \to \sim B) \wedge \sim A)$

h. $D \to (A \to B), \sim A \to \sim D, C \wedge D \vDash_s B$

i. $(\sim A \vee B) \to (C \wedge D), \sim(\sim A \vee B) \nvDash_s \sim(C \wedge D)$

j. $A \wedge (B \vee C), (\sim C \vee D) \wedge (D \to \sim D) \vDash_s A \wedge B$

*E7.5. Complete the demonstration of derived clauses of ST′ by completing the demonstration for dst from right to left, and providing informal reasoning for both the positive and negative parts of ST′(∨) and ST′(↔).

E7.6. Extend definition ST as follows:

$(\uparrow)$ $\mathsf{I}[\mathcal{P} \uparrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$ $\qquad$ $\mathsf{I}[\mathcal{P} \uparrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

(compare page 320). Produce informal reasoning to show each of the following. Again, you may or may not find metalinguistic derivations helpful—but your reasoning should be no less clean than that guided by the rules.

*a. $\mathsf{I}[\mathcal{P} \uparrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{T}$

b. $\mathsf{I}[\mathcal{P} \uparrow \mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$

*c. $\mathsf{I}[\mathcal{P} \uparrow (\mathcal{Q} \uparrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T}$

d. $\mathsf{I}[(\mathcal{P} \uparrow \mathcal{P}) \uparrow (\mathcal{Q} \uparrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{T}$

e. $\mathsf{I}[(\mathcal{P} \uparrow \mathcal{Q}) \uparrow (\mathcal{P} \uparrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$

7.3 Quantificational

So far, we might have obtained sentential results for validity and invalidity by truth tables. But our method positions us to make progress for the quantificational case compared to what we were able to do before. Again we will depend on and gradually expand our metalinguistic derivation system as a guide.

7.3.1 Satisfaction

Given what we have done, it is easy to state definition SF for satisfaction at least as it applies to sentence letters, $\sim$, and $\rightarrow$. In this quantificational case, as described in Chapter 4, we are reasoning about *satisfaction*, and satisfaction depends not just on interpretations, but on interpretations with variable assignments. For $\mathcal{S}$ an arbitrary sentence letter and $\mathcal{P}$ and $\mathcal{Q}$ any formulas, where $\mathsf{I_d}$ is an interpretation I with variable assignment d,

SF (s) $\mathsf{I_d}[\mathcal{S}] = \mathsf{S} \Leftrightarrow \mathsf{I}[\mathcal{S}] = \mathsf{T}$ $\quad\quad$ $\mathsf{I_d}[\mathcal{S}] \neq \mathsf{S} \Leftrightarrow \mathsf{I}[\mathcal{S}] \neq \mathsf{T}$

($\sim$) $\mathsf{I_d}[\sim\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S}$ $\quad\quad$ $\mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S}$

($\rightarrow$) $\mathsf{I_d}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ $\quad\quad$ $\mathsf{I_d}[\mathcal{P} \rightarrow \mathcal{Q}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

Again, you should recognize this as a simple restatement from SF on page 118. Rules for manipulating the definitions remain as before. Already, then, we can produce derived clauses for $\vee$, $\wedge$, and $\leftrightarrow$.

SF′ ($\vee$) $\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\wedge$) $\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\leftrightarrow$) $\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S})$

$\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S})$

All these are like ones from before. For the first,

(P)

1. $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S}$ — abb (I, d arbitrary)
2. $\mathsf{I_d}[\sim\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ — SF($\rightarrow$)
3. $\mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ — SF($\sim$)
4. $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ — 1,2,3 bcnd

By abb, $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S}$; by SF($\rightarrow$), iff $\mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$. So $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$.

The reasoning is as before except that our condition for satisfaction depends on an interpretation with variable assignment rather than an interpretation alone.

Of course, given these definitions, we can use them in our reasoning. As a simple example, let us demonstrate that if $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

(Q)

1.	$\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \mathbin{\vartriangle} \mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$	assp (I, d arbitrary)
2.	$\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$	1 cnj
3.	$\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \mathbin{\triangledown} \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$	2 SF′(∨)
4.	$\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$	1 cnj
5.	$\mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$	4 SF(∼)
6.	$\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$	3,5 dsj
7.	$(\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \mathbin{\vartriangle} \mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}) \Rightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S}$	1-6 cnd

Suppose $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$; from the former, by SF′(∨), $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$; but $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$; so by SF(∼), $\mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$; so $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. So if $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

Again, basic reasoning is as in the sentential case, except that definitions are for satisfaction, and we carry along reference to variable assignments.

Observe that given $\mathsf{I}[A] = \mathsf{T}$ for a sentence letter A, to show that $\mathsf{I_d}[A \vee B] = \mathsf{S}$, we reason,

(R)

1.	$\mathsf{I}[A] = \mathsf{T}$	ins (I particular)
2.	$\mathsf{I_d}[A] = \mathsf{S}$	1 SF(s) (d arbitrary)
3.	$\mathsf{I_d}[A] = \mathsf{S} \mathbin{\triangledown} \mathsf{I_d}[B] = \mathsf{S}$	2 dsj
4.	$\mathsf{I_d}[A \vee B] = \mathsf{S}$	3 SF′(∨)

moving by SF(s) from the premise that the letter is true, to the result that it is satisfied, so that we are in a position to apply other clauses of the definition for satisfaction. SF(∼) and (→), and so SF′(∨), (∧), (↔), apply to *satisfaction* not truth! So we have to bridge from truth to satisfaction before those clauses can apply.

This much should be straightforward, but let us pause to demonstrate derived clauses for satisfaction, and reinforce familiarity with the quantificational definition SF. As you work these and other problems, you may find the quantificational metalinguistic reference on page 352 helpful.

E7.7. Produce metalinguistic derivations and then informal reasoning to complete demonstrations for the positive parts of SF′(∧) and SF′(↔). Hint: You have been through the reasoning before!

*E7.8. Consider an I such that $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] \neq \mathsf{T}$, and $\mathsf{I}[C] = \mathsf{T}$ and arbitrary d. For each of the expressions in E7.1, produce the metalinguistic derivation and then informal reasoning to demonstrate either that it is or is not *satisfied* on $\mathsf{I_d}$.

7.3.2 Truth and Validity

In the quantificational case, there is a distinction between satisfaction and truth. We have been working with the definition for satisfaction. But validity is defined in terms of truth. So to reason about validity, we need a bridge from satisfaction to truth that applies beyond the case of sentence letters. For this, let 'A' be a metalinguistic universal quantifier. So, for example, $A\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S})$ says that any variable assignment d is such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. Then we have,

TI $\quad \mathsf{I}[\mathcal{P}] = \mathsf{T} \Leftrightarrow A\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S}) \qquad \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \Leftrightarrow S\mathsf{d}(\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S})$

This restates the definition from section 4.2.4. $\mathcal{P}$ is true on I iff it is satisfied for any variable assignment d. $\mathcal{P}$ is not true on I iff it is not satisfied for some variable assignment d. Then definition QV is like SV.

QV $\quad \neg S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash \mathcal{Q}$

$\quad S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash \mathcal{Q}$

An argument is quantificationally valid just in case there is no interpretation on which the premises are true and the conclusion is not. Of course, we are now talking about quantificational interpretations as from section 4.2.

To manipulate the metalinguistic universal quantifier A, we will need some new rules. In Chapter 6, we used $\forall$E to instantiate to *any* term—variable, constant, or otherwise. But $\forall$I was restricted—the idea being to generalize only on variables for truly *arbitrary* individuals. Corresponding restrictions are enforced here by the way terms are introduced. We generalize *from* variables for arbitrary individuals, but may instantiate *to* variables or terms of any kind. The universal rules are,

unv $\quad \dfrac{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ of any type $\qquad \dfrac{\mathfrak{A}[\mathfrak{u}]}{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary and new

If some $\mathfrak{A}$ is true for any $\mathfrak{t}$, then it is true for individual $\mathfrak{u}$. Thus we might move from the generalization, $A\mathsf{d}(\mathsf{I_d}[A] = \mathsf{S})$ to the particular claim $\mathsf{I_h}[A] = \mathsf{S}$ for assignment h. For the right-hand "introduction" rule, we require that $\mathfrak{u}$ be arbitrary and new in the sense required for $\forall$I in Chapter 6. In particular, if $\mathfrak{u}$ is new to a derivation for goal $A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]$, $\mathfrak{u}$ will not appear free in any undischarged assumption when the universal rule is applied (typically, our derivations will be so simple that this will not be an issue). If we can show, say, $\mathsf{I_h}[A] = \mathsf{S}$ for arbitrary assignment h, then it is appropriate to move to the conclusion $A\mathsf{d}(\mathsf{I_d}[A] = \mathsf{S})$. We will also accept a metalinguistic quantifier negation, as in *ND+*.

qn $\quad \neg A\mathfrak{t}\mathfrak{A} \Leftrightarrow S\mathfrak{t}\neg\mathfrak{A} \qquad \neg S\mathfrak{t}\mathfrak{A} \Leftrightarrow A\mathfrak{t}\neg\mathfrak{A}$

With these definitions and rules, we are ready to reason about validity—at least for sentential forms. Suppose we want to show, T7.1. $\mathcal{P}, \mathcal{P} \to \mathcal{Q} \vDash \mathcal{Q}$

1.	$\mathcal{P}, \mathcal{P} \to \mathcal{Q} \nvDash \mathcal{Q}$	assp
2.	$\mathit{S}\mathsf{I}(\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	3 cnj
5.	$\mathit{S}\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{Q}] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] \neq \mathsf{S}$	5 exs (h particular)
7.	$\mathsf{J}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T}$	3 cnj
8.	$\mathit{A}\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{P} \to \mathcal{Q}] = \mathsf{S})$	7 TI
9.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P} \to \mathcal{Q}] = \mathsf{S}$	8 unv
10.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{J}_{\mathsf{h}}[\mathcal{Q}] = \mathsf{S}$	9 SF($\to$)
11.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S}$	10,6 dsj
12.	$\mathsf{J}[\mathcal{P}] = \mathsf{T}$	3 cnj
13.	$\mathit{A}\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{P}] = \mathsf{S})$	12 TI
14.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] = \mathsf{S}$	13 unv
15.	$\perp$	14,11 bot
16.	$\mathcal{P}, \mathcal{P} \to \mathcal{Q} \vDash \mathcal{Q}$	1-15 neg

As usual, we begin with the assumption that the theorem is not valid, and apply the definition of validity for the result that the premises are true and the conclusion not. The goal is a contradiction. What is interesting are the applications of TI to bridge between truth and satisfaction. Again, SF applies to satisfaction, not truth. We begin by working on the conclusion. Since the conclusion is not true, by TI with exs we introduce a new and particular variable assignment h on which the conclusion is not satisfied. Then, because the premises are true, by TI with unv the premises are satisfied on that very same assignment h. Then we use SF in the usual way. All this is like the strategy from *ND* by which we jump on existentials: If we had started with the premises, the requirement from exs that we instantiate $\mathit{S}\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{Q}] \neq \mathsf{S})$ to a *new* term would have forced a *different* variable assignment. But by beginning with the conclusion and coming with the universals from the premises after, we bring results into contact for contradiction.

> Suppose $\mathcal{P}$, $\mathcal{P} \to \mathcal{Q} \nvDash \mathcal{Q}$. Then by QV, there is some I such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T}$ but $\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J}_{\mathsf{d}}[\mathcal{Q}] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] \neq \mathsf{S}$. But since $\mathsf{J}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T}$, by TI, for any d, $\mathsf{J}_{\mathsf{d}}[\mathcal{P} \to \mathcal{Q}] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\mathcal{P} \to \mathcal{Q}] = \mathsf{S}$; so by SF($\to$), $\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S}$ or $\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] = \mathsf{S}$; but $\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] \neq \mathsf{S}$, so $\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S}$. But since $\mathsf{J}[\mathcal{P}] = \mathsf{T}$, by TI, for any d, $\mathsf{J}_{\mathsf{d}}[\mathcal{P}] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\mathcal{P}] = \mathsf{S}$. This is impossible; reject the assumption: $\mathcal{P}$, $\mathcal{P} \to \mathcal{Q} \vDash \mathcal{Q}$.

Similarly we can show,

T7.2. $\vDash \mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

T7.3. $\vDash (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

T7.4. $\vDash (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow [(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}]$

T7.5. There is no interpretation I and formula $\mathcal{P}$ such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$.

Hint: Your goal is to show $\neg S\mathsf{I}(\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{P}] = \mathsf{T})$. You can get this by neg.

In each case, the pattern is the same: Bridge assumptions about truth to definition SF by TI with exs and unv. Reasoning with SF is as before. Given the requirement that the metalinguistic existential quantifier always be instantiated to a *new* term it makes sense first to instantiate that which is not true, and so comes out as a metalinguistic existential, and then come with universals on "top" of terms already introduced. This is what we did above, and is like your derivation strategy in *ND*.

*E7.9. Produce metalinguistic derivations and informal reasoning to show that a,b,d,f,h from E7.4 are quantificationally valid.

E7.10. Provide demonstrations for T7.2, T7.3, T7.4, and T7.5 in the informal style. Hint: You may or may not decide that metalinguistic derivations would be helpful.

7.3.3 Terms and Atomics

So far, we have addressed only validity for sentential forms, and have not even seen the (r) and (∀) clauses for SF. We will get the quantifier clause in the next section. Here we come to the atomic clause for definition SF, but must first address the connection with interpretations via definition TA. As from page 115, we say $\mathsf{I}[h^n]\langle \mathsf{a}_1 \ldots \mathsf{a}_n\rangle$ is the thing the function $\mathsf{I}[h^n]$ associates with input $\langle \mathsf{a}_1 \ldots \mathsf{a}_n\rangle$. Then for any interpretation I and variable assignment d, with constant c, variable x, and complex term $h^n t_1 \ldots t_n$,

TA (c) $\mathsf{I}_\mathsf{d}[c] = \mathsf{I}[c]$

(v) $\mathsf{I}_\mathsf{d}[x] = \mathsf{d}[x]$

(f) $\mathsf{I}_\mathsf{d}[h^n t_1 \ldots t_n] = \mathsf{I}[h^n]\langle \mathsf{I}_\mathsf{d}[t_1] \ldots \mathsf{I}_\mathsf{d}[t_n]\rangle$

This is a direct restatement of the definition. To manipulate it we need rules for equality.

eq $\quad \mathfrak{t} = \mathfrak{t} \qquad \mathfrak{t} = \mathfrak{u} \Leftrightarrow \mathfrak{u} = \mathfrak{t} \qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{u} = \mathfrak{v}}{\mathfrak{t} = \mathfrak{v}} \qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$

These should remind you of results from *ND*. We will allow generalized versions so that from $\mathfrak{t} = \mathfrak{u}$, $\mathfrak{u} = \mathfrak{v}$, and $\mathfrak{v} = \mathfrak{w}$, we might move directly to $\mathfrak{t} = \mathfrak{w}$. And we will not worry much about order around the equals sign so that, for example, we could move directly from $\mathfrak{u} = \mathfrak{t}$ and $\mathfrak{A}[\mathfrak{t}]$ to $\mathfrak{A}[\mathfrak{u}]$ without first converting $\mathfrak{u} = \mathfrak{t}$ to $\mathfrak{t} = \mathfrak{u}$ as required by the rule as stated.

As in other cases, we treat clauses from TA as both axioms and rules. Observe that, effectively, an axiom $\mathfrak{A} \Leftrightarrow \mathfrak{B}$ of the sort we have seen up to now works as a "rule" by combination of (an implicit) statement of the axiom with bcnd. Similarly, an axiom $\mathfrak{t} = \mathfrak{u}$ works as a rule by combination of the axiom with eq. So, for example, we might move directly from $\mathsf{I}[\mathscr{c}] = \mathsf{m}$, by (an implicit) $\mathsf{I_d}[\mathscr{c}] = \mathsf{I}[\mathscr{c}]$ from TA(c) and then eq, to $\mathsf{I_d}[\mathscr{c}] = \mathsf{m}$. And similarly we might move directly from $\mathsf{I_d}[\mathscr{c}] = \mathsf{m}$, by (an implicit) $\mathsf{I_d}[\mathscr{h}^1\mathscr{c}] = \mathsf{I}[\mathscr{h}^1]\langle \mathsf{I_d}[\mathscr{c}]\rangle$ from TA(f) and then eq, to $\mathsf{I_d}[\mathscr{h}^1\mathscr{c}] = \mathsf{I}[\mathscr{h}^1]\langle \mathsf{m}\rangle$. This use of definition TA is further illustrated below.

Let us consider how this enables us to determine term assignments. Here is a relatively complex case. Suppose I has $\mathsf{U} = \{1, 2\}$ and,

$$\mathsf{I}[a] = 1$$

(S) $$\mathsf{I}[g^1] = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}$$

$$\mathsf{I}[f^2] = \{\langle\langle 1, 1\rangle, 1\rangle, \langle\langle 1, 2\rangle, 1\rangle, \langle\langle 2, 1\rangle, 2\rangle, \langle\langle 2, 2\rangle, 2\rangle\}$$

Let $\mathsf{d}[x] = 2$. Recall that one-tuples are equated with their members so that $\mathsf{I}[g^1]$ is officially $\{\langle\langle 1\rangle, 2\rangle, \langle\langle 2\rangle, 1\rangle\}$. Consider $\mathsf{I_d}[g^1 f^2 x g^1 a]$. We might do this on a tree as in Chapter 4,

(T)

$x^{[2]}$ $\quad a^{[1]}$ By TA(v) and TA(c)

$g^1a^{[2]}$ By TA(f)

$f^2xg^1a^{[2]}$ By TA(f)

$g^1f^2xg^1a^{[1]}$ By TA(f)

Perhaps we whip through this on the tree. But the derivation follows the very same path, with explicit appeal to the definitions at every stage. In the derivation that follows, lines (1)–(4) cover the top row by application of TA(v) and TA(c). Lines (5)–(7) are like the second row, using the assignment to a with the interpretation of g^1 to determine the assignment to g^1a. Lines (8) - (10) cover the third row. And (11)–(13) use this to reach the final result.

1.	$\mathsf{d}[x] = 2$	ins (d particular)
2.	$\mathsf{I_d}[x] = 2$	1 TA(v) (I particular)
3.	$\mathsf{I}[a] = 1$	ins
4.	$\mathsf{I_d}[a] = 1$	3 TA(c)
5.	$\mathsf{I_d}[g^1a] = \mathsf{I}[g^1]\langle 1\rangle$	4 TA(f)
6.	$\mathsf{I}[g^1]\langle 1\rangle = 2$	ins
7.	$\mathsf{I_d}[g^1a] = 2$	5,6 eq
8.	$\mathsf{I_d}[f^2xg^1a] = \mathsf{I}[f^2]\langle 2,2\rangle$	2,7 TA(f)
9.	$\mathsf{I}[f^2]\langle 2,2\rangle = 2$	ins
10.	$\mathsf{I_d}[f^2xg^1a] = 2$	8,9 eq
11.	$\mathsf{I_d}[g^1f^2xg^1a] = \mathsf{I}[g^1]\langle 2\rangle$	10 TA(f)
12.	$\mathsf{I}[g^1]\langle 2\rangle = 1$	ins
13.	$\mathsf{I_d}[g^1f^2xg^1a] = 1$	11,12 eq

As with trees, to discover that to which a complex term is assigned, we find the assignment to the parts. Beginning with assignments to the parts, we work up to the assignment to the whole. Notice that assertions about the interpretation and the variable assignment are justified by ins. And notice the way we use TA as a rule at (2) and (4), and then again at (5), (8), and (11).

> $\mathsf{d}[x] = 2$; so by TA(v), $\mathsf{I_d}[x] = 2$. And $\mathsf{I}[a] = 1$; so by TA(c), $\mathsf{I_d}[a] = 1$. Since $\mathsf{I_d}[a] = 1$, by TA(f), $\mathsf{I_d}[g^1a] = \mathsf{I}[g^1]\langle 1\rangle$; but $\mathsf{I}[g^1]\langle 1\rangle = 2$; so $\mathsf{I_d}[g^1a] = 2$. Since $\mathsf{I_d}[x] = 2$ and $\mathsf{I_d}[g^1a] = 2$, by TA(f), $\mathsf{I_d}[f^2xg^1a] = \mathsf{I}[f^2]\langle 2,2\rangle$; but $\mathsf{I}[f^2]\langle 2,2\rangle = 2$; so $\mathsf{I_d}[f^2xg^1a] = 2$. And from this, by TA(f), $\mathsf{I_d}[g^1f^2xg^1a] = \mathsf{I}[g^1]\langle 2\rangle$; but $\mathsf{I}[g^1]\langle 2\rangle = 1$; so $\mathsf{I_d}[g^1f^2xg^1a] = 1$.

With the ability to manipulate terms by TA, we can think about satisfaction and truth for arbitrary formulas without quantifiers. This brings us to SF(r). Say $\mathcal{R}^n$ is an n-place relation symbol, and $t_1 \ldots t_n$ are terms.

SF(r) $\mathsf{I_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S} \Leftrightarrow \langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$

$\mathsf{I_d}[\mathcal{R}^n t_1 \ldots t_n] \neq \mathsf{S} \Leftrightarrow \langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n]\rangle \notin \mathsf{I}[\mathcal{R}^n]$

This restates the definition from page 118. Typically we shall apply the definition just in its positive form, and generate the negative case from it (as in NB from *ND*+). Note that SF(r) works as a rule in combination with either bcnd or eq. Thus we might move directly from $\mathsf{I_d}[\mathcal{R}t] = \mathsf{S}$, by (an implicit) $\mathsf{I_d}[\mathcal{R}t] = \mathsf{S} \Leftrightarrow \langle \mathsf{I_d}[t]\rangle \in \mathsf{I}[\mathcal{R}]$ from SF(r) and then bcnd, to $\langle \mathsf{I_d}(t)\rangle \in \mathsf{I}[\mathcal{R}]$. And similarly, we might move directly from $\mathsf{I_d}[t] = \mathsf{m}$, by (the implicit statement of) SF(r) and then eq, to $\mathsf{I_d}[\mathcal{R}t] = \mathsf{S} \Leftrightarrow \langle \mathsf{m}\rangle \in \mathsf{I}[\mathcal{R}]$.

Let us expand the above interpretation and variable assignment (S) so that $\mathsf{I}[A^1] = \{2\}$ (or $\{\langle 2\rangle\}$) and $\mathsf{I}[B^2] = \{\langle 1,2\rangle, \langle 2,1\rangle\}$. Then $\mathsf{I_d}[Af^2xa] = \mathsf{S}$.

(U)

1.	$\mathsf{d}[x] = 2$	ins (d particular)
2.	$\mathsf{I_d}[x] = 2$	1 TA(v) (I particular)
3.	$\mathsf{I}[a] = 1$	ins
4.	$\mathsf{I_d}[a] = 1$	3 TA(c)
5.	$\mathsf{I_d}[f^2xa] = \mathsf{I}[f^2]\langle 2, 1\rangle$	2,4 TA(f)
6.	$\mathsf{I}[f^2]\langle 2, 1\rangle = 2$	ins
7.	$\mathsf{I_d}[f^2xa] = 2$	5,6 eq
8.	$\mathsf{I_d}[Af^2xa] = \mathsf{S} \Leftrightarrow \langle 2\rangle \in \mathsf{I}[A]$	7 SF(r)
9.	$\langle 2\rangle \in \mathsf{I}[A]$	ins
10.	$\mathsf{I_d}[Af^2xa] = \mathsf{S}$	8,9 bcnd

Again, this mirrors what we did with trees—moving through term assignments to the value of the atomic. Observe that satisfaction is not the same as truth! Insofar as d is particular, unv does not apply for the result that Af^2xa is satisfied on every variable assignment and so by TI that the formula is true. In this case, it is a simple matter to identify a variable assignment other than d on which the formula is not satisfied, and so to show that it is not true on I. Set $\mathsf{h}[x] = 1$.

(V)

1.	$\mathsf{h}[x] = 1$	ins (h particular)
2.	$\mathsf{I_h}[x] = 1$	1 TA(v) (I particular)
3.	$\mathsf{I}[a] = 1$	ins
4.	$\mathsf{I_h}[a] = 1$	3 TA(c)
5.	$\mathsf{I_h}[f^2xa] = \mathsf{I}[f^2]\langle 1, 1\rangle$	2,4 TA(f)
6.	$\mathsf{I}[f^2]\langle 1, 1\rangle = 1$	ins
7.	$\mathsf{I_h}[f^2xa] = 1$	5,6 eq
8.	$\mathsf{I_h}[Af^2xa] = \mathsf{S} \Leftrightarrow \langle 1\rangle \in \mathsf{I}[A]$	7 SF(r)
9.	$\langle 1\rangle \notin \mathsf{I}[A]$	ins
10.	$\mathsf{I_h}[Af^2xa] \neq \mathsf{S}$	8,9 bcnd
11.	$S\mathsf{d}(\mathsf{I_d}[Af^2xa] \neq \mathsf{S})$	10 exs
12.	$\mathsf{I}[Af^2xa] \neq \mathsf{T}$	11 TI

Given that it is not satisfied on the particular variable assignment h, exs and TI give the result that Af^2xa is not true. In this case, we simply *pick* the variable assignment we want: Since the formula is not satisfied on this assignment, there is an assignment on which it is not satisfied; so it is not true. To show that an open formula is not true, this is the way to go. Just as we produce particular interpretations to show that arguments are invalid, so we produce particular variable assignments to show that open formulas are not true.

> $\mathsf{h}[x] = 1$; so by TA(v), $\mathsf{I_h}[x] = 1$. And $\mathsf{I}[a] = 1$; so by TA(c), $\mathsf{I_h}[a] = 1$. So by TA(f), $\mathsf{I_h}[f^2xa] = \mathsf{I}[f^2]\langle 1, 1\rangle$; but $\mathsf{I}[f^2]\langle 1, 1\rangle = 1$; so $\mathsf{I_h}[f^2xa] = 1$. So by SF(r), $\mathsf{I_h}[Af^2xa] = \mathsf{S}$ iff $\langle 1\rangle \in \mathsf{I}[A]$; but $\langle 1\rangle \notin \mathsf{I}[A]$; so $\mathsf{I_h}[Af^2xa] \neq \mathsf{S}$. So there is a variable assignment d such that $\mathsf{I_d}[Af^2xa] \neq \mathsf{S}$; so by TI, $\mathsf{I}[Af^2xa] \neq \mathsf{T}$.

In contrast, even though it has free variables, Bxg^1x is true on this I. Say o is a metalinguistic variable that ranges over members of U. In this case, it will be necessary to make an assertion by ins that $A\mathsf{o}(\mathsf{o} = 1 \bigtriangledown \mathsf{o} = 2)$. This is clear enough, since $\mathsf{U} = \{1, 2\}$. Observe that this assertion makes implicit reference to I of which U is a part.

(W)

1.	$\mathcal{A}\mathsf{o}(\mathsf{o} = 1 \triangledown \mathsf{o} = 2)$	ins (I particular)
2.	$\mathsf{I_h}[x] = 1 \triangledown \mathsf{I_h}[x] = 2$	1 unv (h arbitrary)
3.	$\mathsf{I_h}[x] = 1$	assp
4.	$\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 1\rangle$	3 TA(f)
5.	$\mathsf{I}[g^1]\langle 1\rangle = 2$	ins
6.	$\mathsf{I_h}[g^1x] = 2$	4,5 eq
7.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S} \Leftrightarrow \langle 1, 2\rangle \in \mathsf{I}[B]$	3,6 SF(r)
8.	$\langle 1, 2\rangle \in \mathsf{I}[B]$	ins
9.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S}$	7,8 bcnd
10.	$\mathsf{I_h}[x] = 2$	assp
11.	$\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 2\rangle$	10 TA(f)
12.	$\mathsf{I}[g^1]\langle 2\rangle = 1$	ins
13.	$\mathsf{I_h}[g^1x] = 1$	11,12 eq
14.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S} \Leftrightarrow \langle 2, 1\rangle \in \mathsf{I}[B]$	10,13 SF(r)
15.	$\langle 2, 1\rangle \in \mathsf{I}[B]$	ins
16.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S}$	14,15 bcnd
17.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S}$	2,3-9,10-16 dsj
18.	$\mathcal{A}\mathsf{d}(\mathsf{I_d}[Bxg^1x] = \mathsf{S})$	17 unv
19.	$\mathsf{I}[Bxg^1x] = \mathsf{T}$	18 TI

Up to this point, by ins we have made only particular claims about an assignment or interpretation, for example that $\langle 2, 1\rangle \in \mathsf{I}[B]$ or that $\mathsf{I}[g^1]\langle 2\rangle = 1$. This is the typical use of ins. In this case, however, at (1), we make a universal claim about U: any $\mathsf{o} \in \mathsf{U}$ is equal to 1 or 2. For arbitrary h, $\mathsf{I_h}[x]$ is a metalinguistic term picking out some member of U; we instantiate the universal to it with the result that $\mathsf{I_h}[x] = 1$ or $\mathsf{I_h}[x] = 2$. When U is small, this is often helpful: By ins we identify all the members of U; then we are in a position to argue about them individually. Thus we convert the universal claim to a result about the arbitrary assignment, for application of unv and then TI.

> Since $\mathsf{U} = \{1, 2\}$, for arbitrary assignment h, $\mathsf{I_h}[x] = 1$ or $\mathsf{I_h}[x] = 2$. Suppose $\mathsf{I_h}[x] = 1$; then by TA(f), $\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 1\rangle$; but $\mathsf{I}[g^1]\langle 1\rangle = 2$; so $\mathsf{I_h}[g^1x] = 2$; so by SF(r), $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$ iff $\langle 1, 2\rangle \in \mathsf{I}[B]$; but $\langle 1, 2\rangle \in \mathsf{I}[B]$; so $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$. Suppose $\mathsf{I_h}[x] = 2$; then by TA(f), $\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 2\rangle$; but $\mathsf{I}[g^1]\langle 2\rangle = 1$; so $\mathsf{I_h}[g^1x] = 1$; so by SF(r), $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$ iff $\langle 2, 1\rangle \in \mathsf{I}[B]$; but $\langle 2, 1\rangle \in \mathsf{I}[B]$; so $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$. In either case then $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$; and since h is arbitrary, for any assignment d, $\mathsf{I_d}[Bxg^1x] = \mathsf{S}$; so by TI, $\mathsf{I}[Bxg^1x] = \mathsf{T}$.

To show that a formula is not true, we need only find an assignment on which it is not satisfied. To show that a formula is true, we show that it is satisfied on every variable assignment. For this, in the above case with free variables, we have been forced to reason individually about each of the possible assignments to x. This is doable when U is small. We will have to consider other options when it is larger!

E7.11. Consider an I and d such that $\mathsf{U} = \{1, 2\}$,

$\mathsf{I}[a] = 1$

$\mathsf{I}[g^1] = \{\langle 1, 1\rangle, \langle 2, 1\rangle\}$

$\mathsf{I}[f^2] = \{\langle\langle 1, 1\rangle, 2\rangle, \langle\langle 1, 2\rangle, 1\rangle, \langle\langle 2, 1\rangle, 1\rangle, \langle\langle 2, 2\rangle, 2\rangle\}$

where $\mathsf{d}[x] = 1$ and $\mathsf{d}[y] = 2$. Produce metalinguistic derivations and informal reasoning to determine the assignment $\mathsf{I_d}$ for each of the following.

*a. a

b. g^1y

*c. g^1g^1x

d. f^2g^1ax

e. $f^2g^1af^2yx$

E7.12. Augment the interpretation and variable assignment for E7.11 so that $\mathsf{I}[A^1] = \{1\}$ and $\mathsf{I}[B^2] = \{\langle 1, 2\rangle, \langle 2, 2\rangle\}$. Produce (variable assignments as necessary with) metalinguistic derivations and informal reasoning to demonstrate each of the following.

a. $\mathsf{I_d}[Ax] = \mathsf{S}$

*b. $\mathsf{I}[Byx] \neq \mathsf{T}$

c. $\mathsf{I}[Bg^1ay] \neq \mathsf{T}$

d. $\mathsf{I}[Aa] = \mathsf{T}$

e. $\mathsf{I}[\sim Bxg^1x] = \mathsf{T}$

7.3.4 Quantifiers

We are finally ready to think more generally about validity and truth for quantifier forms. For this, we will complete our metalinguistic system by adding the quantifier clause to definition SF.

SF($\forall$) $\mathsf{I_d}[\forall x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$ $\quad$ $\mathsf{I_d}[\forall x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

This is a simple statement of the definition from page 118. We treat the metalinguistic variable 'o' as implicitly restricted to the members of U (for any $\mathsf{o} \in \mathsf{U}\ldots$). You should think about this in relation to trees: From $\mathsf{I_d}[\forall x\mathcal{P}]$ there are branches with $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}]$ for each object $\mathsf{o} \in \mathsf{U}$. The universal is satisfied when each branch is satisfied; not satisfied when some branch is unsatisfied. That is what is happening above. We have the derived clause too.

SF′(∃) $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$ $\quad\quad$ $\mathsf{I_d}[\exists x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

The existential is satisfied when some branch is satisfied; not satisfied when every branch is not satisfied. For the positive form,

(X)

1.	$\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S}$	abb (I, d arbitrary)
2.	$\mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S}$	SF(∼)
3.	$\mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S})$	SF(∀)
4.	$\mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S}) \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$	SF(∼)
5.	$\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$	1,2,3,4 bcnd

By abb, $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S}$; by SF(∼) iff $\mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S}$; by SF(∀), iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S}$; by SF(∼), iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$. So $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S}$ iff there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$.

Recall that we were not able to use trees to demonstrate validity in the quantificational case because there were too many interpretations to have trees for all of them, and because universes may be too large to have branches for all their members. But this is not a special difficulty for us now. For a simple case, let us show that $\vDash \forall x(Ax \to Ax)$.

(Y)

1.	$\nvDash \forall x(Ax \to Ax)$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\forall x(Ax \to Ax)] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\forall x(Ax \to Ax)] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathcal{S}\mathsf{d}(\mathsf{J_d}[\forall x(Ax \to Ax)] \neq \mathsf{S})$	3 TI
5.	$\mathsf{J_h}[\forall x(Ax \to Ax)] \neq \mathsf{S}$	4 exs (h particular)
6.	$\mathcal{S}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Ax] \neq \mathsf{S})$	5 SF(∀)
7.	$\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Ax] \neq \mathsf{S}$	6 exs (m particular)
8.	$\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S} \vartriangle \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$	7 SF(→)
9.	$\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$	8 cnj
10.	$\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$	8 cnj
11.	$\perp$	9,10 bot
12.	$\vDash \forall x(Ax \to Ax)$	1-11 neg

If $\forall x(Ax \to Ax)$ is not valid, there has to be *some* I on which it is not true. If $\forall x(Ax \to Ax)$ is not true on some I, there has to be some d on which it is not satisfied. And if the universal is not satisfied, there has to be some $\mathsf{o} \in \mathsf{U}$ for which the corresponding "branch" is not satisfied. But this is impossible—for we cannot have a branch where this is so.

Suppose $\nvDash \forall x(Ax \to Ax)$; then by QV, there is some I such that $\mathsf{I}[\forall x(Ax \to Ax)] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\forall x(Ax \to Ax)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{J_d}[\forall x(Ax \to Ax)] \neq \mathsf{S}$. Let h be a particular assignment of this sort; then $\mathsf{J_h}[\forall x(Ax \to Ax)] \neq \mathsf{S}$; so by SF(∀), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Ax] \neq \mathsf{S}$. Let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Ax] \neq \mathsf{S}$; so by SF(→), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$ and $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$. But this is impossible; reject the assumption: $\vDash \forall x(Ax \to Ax)$.

Notice, again, that the general strategy is to instantiate metalinguistic existential quantifiers as quickly as possible. Contradictions tend to arise at the level of atomic expressions and individuals.

Here is a case that is similar, but somewhat more involved. We show $\forall x(Ax \to Bx), \exists xAx \vDash \exists zBz$. Here is a start:

	1.	$\forall x(Ax \to Bx), \exists xAx \nvDash \exists zBz$	assp
	2.	$S\mathsf{I}(\mathsf{I}[\forall x(Ax \to Bx)] = \mathsf{T} \vartriangle \mathsf{I}[\exists xAx] = \mathsf{T} \vartriangle \mathsf{I}[\exists zBz] \neq \mathsf{T})$	1 QV
	3.	$\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T} \vartriangle \mathsf{J}[\exists xAx] = \mathsf{T} \vartriangle \mathsf{J}[\exists zBz] \neq \mathsf{T}$	2 exs (J particular)
	4.	$\mathsf{J}[\exists zBz] \neq \mathsf{T}$	3 cnj
	5.	$S\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\exists zBz] \neq \mathsf{S})$	4 TI
	6.	$\mathsf{J}_{\mathsf{h}}[\exists zBz] \neq \mathsf{S}$	5 exs (h particular)
	7.	$\mathsf{J}[\exists xAx] = \mathsf{T}$	3 cnj
	8.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\exists xAx] = \mathsf{S})$	7 TI
	9.	$\mathsf{J}_{\mathsf{h}}[\exists xAx] = \mathsf{S}$	8 unv
(Z)	10.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Ax] = \mathsf{S})$	9 SF′(∃)
	11.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax] = \mathsf{S}$	10 exs (m particular)
	12.	$\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$	3 cnj
	13.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall x(Ax \to Bx)] = \mathsf{S})$	12 TI
	14.	$\mathsf{J}_{\mathsf{h}}[\forall x(Ax \to Bx)] = \mathsf{S}$	13 unv
	15.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Ax \to Bx] = \mathsf{S})$	14 SF(∀)
	16.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax \to Bx] = \mathsf{S}$	15 unv
	17.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax] \neq \mathsf{S} \triangledown \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Bx] = \mathsf{S}$	16 SF(→)
	18.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Bx] = \mathsf{S}$	17,11 dsj
	19.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(z\vert\mathsf{o})}[Bz] \neq \mathsf{S})$	6 SF′(∃)
	20.	$\mathsf{J}_{\mathsf{h}(z\vert\mathsf{m})}[Bz] \neq \mathsf{S}$	19 unv

Note again the way we work with the metalinguistic quantifiers: We begin with the conclusion, because it is the one that requires a particular variable assignment; the premises can then be instantiated to that same assignment. Similarly, with that particular variable assignment on the table, we focus on the second premise, because it is the one that requires an instantiation to a particular individual. The other premise and the conclusion then come in later with universal quantifications that go onto the same thing. Also, $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$ contradicts $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$; this justifies dsj at (18). However $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$ at (18) does not contradict $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] \neq \mathsf{S}$ at (20). There would have been a contradiction if the variable had been the same. But it is not. However, with the distinct variables, we can bring out the contradiction by "forcing the result into the interpretation" as follows:

21.	$\mathsf{h}(x\|\mathsf{m})[x] = \mathsf{m}$	ins
22.	$\mathsf{J}_{\mathsf{h}(x\|\mathsf{m})}[x] = \mathsf{m}$	21 TA(v)
23.	$\mathsf{J}_{\mathsf{h}(x\|\mathsf{m})}[Bx] = \mathsf{S} \Leftrightarrow \langle\mathsf{m}\rangle \in \mathsf{J}[B]$	22 SF(r)
24.	$\langle\mathsf{m}\rangle \in \mathsf{J}[B]$	23,18 bcnd
25.	$\mathsf{h}(z\|\mathsf{m})[z] = \mathsf{m}$	ins
26.	$\mathsf{J}_{\mathsf{h}(z\|\mathsf{m})}[z] = \mathsf{m}$	25 TA(v)
27.	$\mathsf{J}_{\mathsf{h}(z\|\mathsf{m})}[Bz] = \mathsf{S} \Leftrightarrow \langle\mathsf{m}\rangle \in \mathsf{J}[B]$	26 SF(r)
28.	$\langle\mathsf{m}\rangle \notin \mathsf{J}[B]$	27,20 bcnd
29.	$\perp$	24,28 bot
30.	$\forall x(Ax \to Bx), \exists x Ax \vDash \exists z Bz$	1-29 neg

The assumption that the argument is not valid leads to the result that there is some interpretation J and $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{m} \in \mathsf{J}[B]$ and $\mathsf{m} \notin \mathsf{J}[B]$; so there can be no such interpretation, and the argument is quantificationally valid. Observe that, although we do not know anything else about h, simple inspection reveals that $\mathsf{h}(x|\mathsf{m})$ assigns object m to x. So we allow ourselves to assert it at (21) by ins; and similarly at (25). This pattern of moving from facts about satisfaction to facts about the interpretation is typical.

With the order of a few lines slightly rearranged toward the end, here is the informal reasoning:

> Suppose $\forall x(Ax \to Bx)$, $\exists x Ax \nvDash \exists z Bz$; then by QV, there is some I such that $\mathsf{I}[\forall x(Ax \to Bx)] = \mathsf{T}$ and $\mathsf{I}[\exists x Ax] = \mathsf{T}$ but $\mathsf{I}[\exists z Bz] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$ and $\mathsf{J}[\exists x Ax] = \mathsf{T}$ but $\mathsf{J}[\exists z Bz] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J}_{\mathsf{d}}[\exists z Bz] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J}_{\mathsf{h}}[\exists z Bz] \neq \mathsf{S}$. Since $\mathsf{J}[\exists x Ax] = \mathsf{T}$, by TI, for any d, $\mathsf{J}_{\mathsf{d}}[\exists x Ax] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\exists x Ax] = \mathsf{S}$; so by SF′(∃) there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax] = \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$. Since $\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$, by TI, for any d, $\mathsf{J}_{\mathsf{d}}[\forall x(Ax \to Bx)] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\forall x(Ax \to Bx)] = \mathsf{S}$; so by SF(∀), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Bx] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Bx] = \mathsf{S}$; so by SF(→), either $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$ or $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$; but $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$, so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$; $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[x] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$ iff $\langle\mathsf{m}\rangle \in \mathsf{J}[B]$; so $\langle\mathsf{m}\rangle \in \mathsf{J}[B]$. But since $\mathsf{J}_{\mathsf{h}}[\exists z Bz] \neq \mathsf{S}$, by SF′(∃), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(z|\mathsf{o})}[Bz] \neq \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] \neq \mathsf{S}$; $\mathsf{h}(z|\mathsf{m})[z] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[z] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] = \mathsf{S}$ iff $\langle\mathsf{m}\rangle \in \mathsf{J}[B]$; so $\langle\mathsf{m}\rangle \notin \mathsf{J}[B]$. This is impossible; reject the assumption: $\forall x(Ax \to Bx)$, $\exists x Ax \vDash \exists z Bz$.

Observe again the repeated use of the pattern that moves from truth through TI to satisfaction, so that SF gets a grip, and the pattern that moves through satisfaction to the interpretation. These should be nearly automatic.

Here is an example that is particularly challenging in the way metalinguistic quantifier rules apply. We show $\exists x \forall y Axy \vDash \forall y \exists x Axy$. For this, you should carefully work through the derivation (AA) in the upper box on page 349. When multiple quantifiers come off, variable assignments once modified are simply modified again—just as with trees. Observe again that we instantiate the metalinguistic existential quantifiers before universals. Also, the different existential quantifiers go to *different* individuals, to respect the requirement that individuals from exs be *new*. The key to this derivation is getting out *both* metalinguistic existentials for m and n *before* applying the corresponding universals—and what makes the derivation difficult is seeing that this needs to be done. Strictly, the variable assignment at (15) is the same as the one at (17), only the names are variants of one another. Thus we observe by ins that the assignments are the same, and apply eq for the contradiction.

Another approach would have been to push for contradiction at the level of the interpretation. Something along these lines would have been required if the conclusion had been, say, $\forall w \exists z Azw$ and so (17) $\mathsf{J}_{\mathsf{h}(w|\mathsf{m},z|\mathsf{n})}[Azw] \neq \mathsf{S}$; then insofar as they involve *different* atomic formulas and different assignments (15) and (17) would not themselves contradict. Even so, we might have continued as at (AB) in the lower box on the following page.

In the case we have been given, though, this is not necessary. With the original conclusion $\forall y \exists x Axy$, here is the informal version:

> Suppose $\exists x \forall y Axy \not\vDash \forall y \exists x Axy$; then by QV there is some I such that $\mathsf{I}[\exists x \forall y Axy] = \mathsf{T}$ and $\mathsf{I}[\forall y \exists x Axy] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\exists x \forall y Axy] = \mathsf{T}$ and $\mathsf{J}[\forall y \exists x Axy] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J}_{\mathsf{d}}[\forall y \exists x Axy] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J}_{\mathsf{h}}[\forall y \exists x Axy] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(y|\mathsf{o})}[\exists x Axy] \neq \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(y|\mathsf{m})}[\exists x Axy] \neq \mathsf{S}$. Since $\mathsf{J}[\exists x \forall y Axy] = \mathsf{T}$, by TI for any d, $\mathsf{J}_{\mathsf{d}}[\exists x \forall y Axy] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\exists x \forall y Axy] = \mathsf{S}$; so by SF$'$($\exists$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[\forall y Axy] = \mathsf{S}$; let n be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x|\mathsf{n})}[\forall y Axy] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x|\mathsf{n},y|\mathsf{o})}[Axy] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{n},y|\mathsf{m})}[Axy] = \mathsf{S}$. Since $\mathsf{J}_{\mathsf{h}(y|\mathsf{m})}[\exists x Axy] \neq \mathsf{S}$, by SF$'$($\exists$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(y|\mathsf{m},x|\mathsf{o})}[Axy] \neq \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(y|\mathsf{m},x|\mathsf{n})}[Axy] \neq \mathsf{S}$; but $\mathsf{h}(y|\mathsf{m}, x|\mathsf{n})$ is the same assignment as $\mathsf{h}(x|\mathsf{n}, y|\mathsf{m})$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{n},y|\mathsf{m})}[Axy] \neq \mathsf{S}$. This is impossible; reject the assumption: $\exists x \forall y Axy \vDash \forall y \exists x Axy$.

Try reading that to your roommate or parents! If you have followed to this stage, you have accomplished something significant. These are important results, given that we wondered in Chapter 4 how this sort of thing could be done at all.

Here is a last trick that can sometimes be useful. Suppose we are trying to show $\forall x Px \vDash Pa$. We will come to a stage where we want to use the premise to instantiate a variable o to the thing that is $\mathsf{J}_{\mathsf{h}}[a]$. So we might move directly from $A\mathsf{o}(\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Px] = \mathsf{S})$ to $\mathsf{J}_{\mathsf{h}(x|\mathsf{J}_{\mathsf{h}}[a])}[Px] = \mathsf{S}$ by unv. But this is ugly, and hard to follow. An alternative is allow a rule (def) that defines m as a metalinguistic term for the *same* object as $\mathsf{J}_{\mathsf{h}}[a]$. This new term is not separately declared arbitrary or particular, but rather inherits its status from the original. The result is as follows:

(AA)

1.	$\exists x\forall yAxy \not\models \forall y\exists xAxy$	assp
2.	$S\mathsf{I}(\mathsf{I}[\exists x\forall yAxy] = \mathsf{T} \vartriangle \mathsf{I}[\forall y\exists xAxy] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\exists x\forall yAxy] = \mathsf{T} \vartriangle \mathsf{J}[\forall y\exists xAxy] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\forall y\exists xAxy] \neq \mathsf{T}$	3 cnj
5.	$S\mathsf{d}(\mathsf{J_d}[\forall y\exists xAxy] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J_h}[\forall y\exists xAxy] \neq \mathsf{S}$	5 exs (h particular)
7.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(y\mid\mathsf{o})}[\exists xAxy] \neq \mathsf{S})$	6 SF($\forall$)
8.	$\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m})}[\exists xAxy] \neq \mathsf{S}$	7 exs (m particular)
9.	$\mathsf{J}[\exists x\forall yAxy] = \mathsf{T}$	3 cnj
10.	$A\mathsf{d}(\mathsf{J_d}[\exists x\forall yAxy] = \mathsf{S})$	9 TI
11.	$\mathsf{J_h}[\exists x\forall yAxy] = \mathsf{S}$	10 unv
12.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{o})}[\forall yAxy] = \mathsf{S})$	11 SF′($\exists$)
13.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n})}[\forall yAxy] = \mathsf{S}$	12 exs (n particular)
14.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{o})}[Axy] = \mathsf{S})$	13 SF($\forall$)
15.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[Axy] = \mathsf{S}$	14 unv
16.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m},x\mid\mathsf{o})}[Axy] \neq \mathsf{S})$	8 SF′($\exists$)
17.	$\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m},x\mid\mathsf{n})}[Axy] \neq \mathsf{S}$	16 unv
18.	$\mathsf{h}(y\mid\mathsf{m}, x\mid\mathsf{n}) = \mathsf{h}(x\mid\mathsf{n}, y\mid\mathsf{m})$	ins
19.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[Axy] \neq \mathsf{S}$	17,18 eq
20.	$\perp$	15,19 bot
21.	$\exists x\forall yAxy \models \forall y\exists xAxy$	1-20 neg

(AB)

*17.	$\mathsf{J}_{\mathsf{h}(w\mid\mathsf{m},z\mid\mathsf{n})}[Azw] \neq \mathsf{S}$	
18.	$\mathsf{h}(x\mid\mathsf{n}, y\mid\mathsf{m})[x] = \mathsf{n}$	ins
19.	$\mathsf{h}(x\mid\mathsf{n}, y\mid\mathsf{m})[y] = \mathsf{m}$	ins
20.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[x] = \mathsf{n}$	18 TA(v)
21.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[y] = \mathsf{m}$	19 TA(v)
22.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[Axy] = \mathsf{S} \Leftrightarrow \langle\mathsf{n},\mathsf{m}\rangle \in \mathsf{I}[A]$	20,21 SF(r)
23.	$\langle\mathsf{n},\mathsf{m}\rangle \in \mathsf{I}[A]$	22,15 bcnd
24.	$\mathsf{h}(w\mid\mathsf{m}, z\mid\mathsf{n})[z] = \mathsf{n}$	ins
25.	$\mathsf{h}(w\mid\mathsf{m}, z\mid\mathsf{n})[w] = \mathsf{m}$	ins
26.	$\mathsf{J}_{\mathsf{h}(w\mid\mathsf{m},z\mid\mathsf{n})}[z] = \mathsf{n}$	24 TA(v)
27.	$\mathsf{J}_{\mathsf{h}(w\mid\mathsf{m},z\mid\mathsf{n})}[w] = \mathsf{m}$	25 TA(v)
28.	$\mathsf{J}_{\mathsf{h}(w\mid\mathsf{m},z\mid\mathsf{n})}[Azw] = \mathsf{S} \Leftrightarrow \langle\mathsf{n},\mathsf{m}\rangle \in \mathsf{I}[A]$	26,27 SF(r)
29.	$\langle\mathsf{n},\mathsf{m}\rangle \notin \mathsf{I}[A]$	28, *17 bcnd
30.	$\perp$	23,29 bot
31.	$\exists x\forall yAxy \models \forall w\exists zAzw$	1-30 neg

(AC)

1.	$\forall x Px \nvDash Pa$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\forall x Px] = \mathsf{T} \vartriangle \mathsf{I}[Pa] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\forall x Px] = \mathsf{T} \vartriangle \mathsf{J}[Pa] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[Pa] \neq \mathsf{T}$	3 cnj
5.	$\mathcal{S}\mathsf{d}(\mathsf{J_d}[Pa] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J_h}[Pa] \neq \mathsf{S}$	5 exs (h particular)
7.	$\mathsf{J_h}[a] = \mathsf{m}$	def
8.	$\mathsf{J_h}[Pa] = \mathsf{S} \Leftrightarrow \langle\mathsf{m}\rangle \in \mathsf{J}[P]$	7 SF(r)
9.	$\langle\mathsf{m}\rangle \notin \mathsf{J}[P]$	8,6 bcnd
10.	$\mathsf{J}[\forall x Px] = \mathsf{T}$	3 cnj
11.	$\mathcal{A}\mathsf{d}(\mathsf{J_d}[\forall x Px] = \mathsf{S})$	10 TI
12.	$\mathsf{J_h}[\forall x Px] = \mathsf{S}$	11 unv
13.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{o})}[Px] = \mathsf{S})$	12 SF($\forall$)
14.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[Px] = \mathsf{S}$	13 unv
15.	$\mathsf{h}(x\mid\mathsf{m})[x] = \mathsf{m}$	ins
16.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[x] = \mathsf{m}$	15 TA(v)
17.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[Px] = \mathsf{S} \Leftrightarrow \langle\mathsf{m}\rangle \in \mathsf{J}[P]$	16 SF(r)
18.	$\langle\mathsf{m}\rangle \in \mathsf{J}[P]$	17,14 bcnd
19.	$\perp$	9,18 bot
20.	$\forall x Px \vDash Pa$	1-19 neg

The result adds a couple lines, but is perhaps easier to follow. Though an interpretation is not specified, we can be sure that $\mathsf{J_h}[a]$ is some particular member of U; we simply let m designate that individual, and instantiate the universal to it. Again the contradiction appears as we force results into the interpretation.

> Suppose $\forall x Px \nvDash Pa$; then by QV, there is some I such that $\mathsf{I}[\forall x Px] = \mathsf{T}$ and $\mathsf{I}[Pa] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\forall x Px] = \mathsf{T}$ and $\mathsf{J}[Pa] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J_d}[Pa] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J_h}[Pa] \neq \mathsf{S}$; let $\mathsf{m} = \mathsf{J_h}[a]$; then by SF(r), $\mathsf{J_h}[Pa] = \mathsf{S}$ iff $\langle\mathsf{m}\rangle \in \mathsf{J}[P]$; so $\langle\mathsf{m}\rangle \notin \mathsf{J}[P]$. Since $\mathsf{J}[\forall x Px] = \mathsf{T}$, by TI, for any d, $\mathsf{J_d}[\forall x Px] = \mathsf{S}$; so $\mathsf{J_h}[\forall x Px] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x\mid\mathsf{o})}[Px] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[Px] = \mathsf{S}$; $\mathsf{h}(x\mid\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[x] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x\mid\mathsf{m})}[Px] = \mathsf{S}$ iff $\langle\mathsf{m}\rangle \in \mathsf{J}[P]$; so $\langle\mathsf{m}\rangle \in \mathsf{J}[P]$. This is impossible; reject the assumption: $\forall x Px \vDash Pa$.

Since we can instantiate $\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{o})}[Px] = \mathsf{S})$ to any object, we can instantiate it to the one that happens to be $\mathsf{J_h}[a]$. The extra name streamlines the process. One can always do without the name. But there is no harm introducing it when it will help.

At this stage, we have the tools for proof of the following theorems that will be useful for later chapters.

*T7.6. For any I and $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\forall x\mathcal{P}] = \mathsf{T}$.

> Hint: For one direction, if $\mathcal{P}$ is satisfied on the arbitrary assignment, you may conclude that it is satisfied on one like $\mathsf{h}(x\mid\mathsf{m})$. For the other direction, if you can instantiate o to any object, you can instantiate it to the thing that is $\mathsf{h}[x]$. But by

ins, h with *this* assigned to x, just *is* h. So after substitution, you can end up with the very same assignment as the one with which you started.

T7.7. Each of the following conditions obtains.

(a) $\mathsf{I_d}[(\forall x : \mathcal{B})\mathcal{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I_{d(x|o)}}[\mathcal{B}] \neq \mathsf{S}$ or $\mathsf{I_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$.

(b) $\mathsf{I_d}[(\exists x : \mathcal{B})\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I_{d(x|o)}}[\mathcal{B}] = \mathsf{S}$ and $\mathsf{I_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$.

Demonstration of these results is straightforward with definition RQ from page 296.

T7.6 is interesting insofar as it underlies principles like Gen in *AD* and $\forall$I in *ND*. We further explore this link in following chapters. T7.7 applies to the restricted quantifiers introduced in Chapter 6. Reasoning with restricted quantifiers is streamlined by their derived semantic conditions.

E7.13. Produce metalinguistic derivations and informal reasoning to demonstrate each of the following.

*a. $\vDash \forall x(Ax \rightarrow {\sim}{\sim}Ax)$

b. $\vDash {\sim}\exists x(Ax \wedge {\sim}Ax)$

*c. $Pa \vDash \exists xPx$

d. $\forall x(Ax \wedge Bx) \vDash \forall yBy$

e. $\forall yPy \vDash \forall xPf^1x$

f. $\exists yAy \vDash \exists x(Ax \vee Bx)$

g. ${\sim}\forall x(Ax \rightarrow Dx) \vDash \exists x(Ax \wedge {\sim}Dx)$

h. $\forall x(Ax \rightarrow Bx), \forall x(Bx \rightarrow Cx) \vDash \forall x(Ax \rightarrow Cx)$

i. $\forall x\forall yAxy \vDash \forall y\forall xAxy$

j. $\forall x\exists y(Ay \rightarrow Bx) \vDash \forall x(\forall yAy \rightarrow Bx)$

*E7.14. Provide demonstrations for T7.6–T7.7 in the informal style. Hint: You may or may not decide that a metalinguistic derivation will be helpful.

Metalinguistic Quick Reference (quantificational)

DEFINITIONS:

TA (c) $\mathsf{I_d}[c] = \mathsf{I}[c]$

(v) $\mathsf{I_d}[x] = \mathsf{d}[x]$

(f) $\mathsf{I_d}[h^n t_1 \dots t_n] = \mathsf{I}[h^n]\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle$

SF (s) $\mathsf{I_d}[\mathcal{S}] = \mathsf{S} \Leftrightarrow \mathsf{I}[\mathcal{S}] = \mathsf{T}$

(r) $\mathsf{I_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S} \Leftrightarrow \langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$

($\sim$) $\mathsf{I_d}[\sim\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S}$

$\mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S}$

($\rightarrow$) $\mathsf{I_d}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[\mathcal{P} \rightarrow \mathcal{Q}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\forall$) $\mathsf{I_d}[\forall x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathscr{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I_d}[\forall x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathscr{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

SF$'$ ($\vee$) $\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\wedge$) $\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\leftrightarrow$) $\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S})$

$\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S})$

($\exists$) $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathscr{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I_d}[\exists x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathscr{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

TI $\mathsf{I}[\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathscr{A}\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \Leftrightarrow \mathscr{S}\mathsf{d}(\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S})$

QV $\neg\mathscr{S}\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \dots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \dots \mathcal{P}_n \vDash \mathcal{Q}$

$\mathscr{S}\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \dots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \dots \mathcal{P}_n \nvDash \mathcal{Q}$

abb As before, *abbreviation* allows $\mathfrak{A}[\mathcal{P}'] \Leftrightarrow \mathfrak{A}[\mathcal{P}]$ where $\mathcal{P}'$ abbreviates $\mathcal{P}$.

RULES:

All the rules from the sentential metalinguistic reference (page 333) plus:

unv $\dfrac{\mathscr{A}\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ of any type $\qquad$ $\dfrac{\mathfrak{A}[\mathfrak{u}]}{\mathscr{A}\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary and new

qn $\neg\mathscr{A}\mathfrak{t}\mathfrak{A} \Leftrightarrow \mathscr{S}\mathfrak{t}\neg\mathfrak{A}$ $\qquad$ $\neg\mathscr{S}\mathfrak{t}\mathfrak{A} \Leftrightarrow \mathscr{A}\mathfrak{t}\neg\mathfrak{A}$

eq $\mathfrak{t} = \mathfrak{t}$ $\qquad$ $\mathfrak{t} = \mathfrak{u} \Leftrightarrow \mathfrak{u} = \mathfrak{t}$ $\qquad$ $\dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{u} = \mathfrak{v}}{\mathfrak{t} = \mathfrak{v}}$ $\qquad$ $\dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$

def Defines one metalinguistic term $\mathfrak{t}$ by another $\mathfrak{u}$ so that $\mathfrak{t} = \mathfrak{u}$.

On the Semantics of Variables

Ours is the standard quantifier semantics, essentially due to Tarski's 1933, "The Concept of Truth in Formalized Languages." At the start of section 2.3 we suggested that variables are like pronouns. Correspondingly, as emphasized in section 5.3.1, bound variables function as placeholders—there is no semantic difference between,

$$\exists x(x < \emptyset) \qquad \text{and} \qquad \exists y(y < \emptyset)$$

Similarly one might think $\emptyset < x$ and $\emptyset < y$ say the same thing. But with $\mathsf{d}[x] = 1$ and $\mathsf{d}[y] = 0$, $\bar{\mathsf{N}}_{\mathsf{d}}[\emptyset < x] = \mathsf{S}$ and $\bar{\mathsf{N}}_{\mathsf{d}}[\emptyset < y] \neq \mathsf{S}$. So the formulas get *different* evaluations. In response, K. Fine and others suggest alternative accounts to preserve the status of variables as mere placeholders (Fine, "The Role of Variables," see also Button and Walsh, *Philosophy and Model Theory*, Chapter 1).

It is not clear that we have intuitions about satisfaction that do not come from the semantics itself. So one might respond, "Well, that is the way *satisfaction* works." But allow that the placeholder intuition applies generally. Button and Walsh prefer an account that substitutes constants for free variables. A quantified sentence $\forall x\mathcal{P}$ is evaluated in terms of *sentences* $\mathcal{P}^x_c$. This does not work if there are objects to which no constant is assigned. One option is to extend $\mathcal{L}$ by the addition of some constant c_{o} for each $\mathsf{o} \in \mathsf{U}$. Another option adds only as many constants as there are variables in $\mathcal{P}$, considering $\mathcal{P}^x_c$ for each of the possible *interpretations* of c.

As applied to sentences, all the options give the same results for truth and validity. Insofar as it applies exclusively to sentences, the approach with constants bypasses formulas where variables have anything but a placeholder role. (And derivations might be developed to bypass free variables too—compare note 3 on page 471 of Chapter 10.) But it is not clear that we need abandon the traditional approach in order to preserve the role of variables as placeholders. As a start,

> Consider a sequence $x_1, x_2, \ldots$ of metavariables and a function k that assigns to each an object from U. For some $\mathcal{P}$ with variables $z_a \ldots z_b$ (in the order of their first appearance in $\mathcal{P}$), let m be a map that takes $z_a \ldots z_b$ in that order to $x_1 \ldots x_n$. So $\mathsf{m}[z_a] = x_1$ and so forth. Note that m is syntactically defined. Given some $\mathcal{P}$ with its m, proceed very much as usual: If c is a constant, $\mathsf{I}_{\mathsf{k}}[c] = \mathsf{I}[c]$; if z is a variable, then $\mathsf{I}_{\mathsf{k}}[z] = \mathsf{k}[\mathsf{m}(z)]$; if t is $h^n t_1 \ldots t_n$ then $\mathsf{I}_{\mathsf{k}}[t] = \mathsf{I}[h^n]\langle \mathsf{I}_{\mathsf{k}}[t_1] \ldots \mathsf{I}_{\mathsf{k}}[t_n]\rangle$. $\mathsf{I}_{\mathsf{k}}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$ iff $\langle \mathsf{I}_{\mathsf{k}}[t_1] \ldots \mathsf{I}_{\mathsf{k}}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; $\mathsf{I}_{\mathsf{k}}[\sim\mathcal{A}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{k}}[\mathcal{A}] \neq \mathsf{S}$; $\mathsf{I}_{\mathsf{k}}[\mathcal{A} \to \mathcal{B}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{k}}[\mathcal{A}] \neq \mathsf{S}$ or $\mathsf{I}_{\mathsf{k}}[\mathcal{B}] = \mathsf{S}$; and $\mathsf{I}_{\mathsf{k}}[\forall z\mathcal{A}] = \mathsf{S}$ iff for every $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{k}(\mathsf{m}(z)|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$. An assignment to $\mathcal{X}$ is relative to some $\mathcal{P}$ that gives the context of which it is a part and might perspicuously be indicated $\mathsf{I}_{\mathsf{k}}[\mathcal{X}/\mathcal{P}]$.

Treat variables as marking "slots" in a formula. Taken separately, $x < \emptyset$ and $y < \emptyset$ get the same evaluation—both x and y map to x_1 and so are assigned the same individual. But they mark different slots in $x < \emptyset \to y < \emptyset$ and so may be assigned different individuals. In effect, we group together variable assignments that supply all the same objects to the slots. Given this, variables reappear as placeholders.

7.3.5 Invalidity

We already have in hand concepts required for showing invalidity. Difficulties are mainly strategic and practical. As usual, for invalidity, the idea is to produce an interpretation and show that it makes the premises true and the conclusion not.

Here is a case parallel to one you worked with trees in homework from E4.15. We show $\forall xPf^1x \nvDash \forall xPx$. For the interpretation J set, $\mathsf{U} = \{1, 2\}$, $\mathsf{J}[P] = \{1\}$, $\mathsf{J}[f^1] = \{\langle 1, 1\rangle, \langle 2, 1\rangle\}$. We want to take advantage of the particular features of this interpretation to show that it makes the premise true and the conclusion not. Begin as follows:

	1.	$\mathsf{h}(x\vert 2)[x] = 2$	ins (h arbitrary)
	2.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[x] = 2$	1 TA(v) (J particular)
	3.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[Px] = \mathsf{S} \Leftrightarrow \langle 2\rangle \in \mathsf{J}[P]$	2 SF(r)
	4.	$\langle 2\rangle \notin \mathsf{J}[P]$	ins
(AD)	5.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[Px] \neq \mathsf{S}$	3,4 bcnd
	6.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert \mathsf{o})}[Px] \neq \mathsf{S})$	5 exs
	7.	$\mathsf{J}_{\mathsf{h}}[\forall xPx] \neq \mathsf{S}$	6 SF(∀)
	8.	$S\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall xPx] \neq \mathsf{S})$	7 exs
	9.	$\mathsf{J}[\forall xPx] \neq \mathsf{T}$	8 TI

Another option would have been to assume $\mathsf{J}[\forall xPx] = \mathsf{T}$ and work to a contradiction.

1.	$\mathsf{J}[\forall xPx] = \mathsf{T}$	assp (J particular)
2.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall xPx] = \mathsf{S})$	1 TI
3.	$\mathsf{J}_{\mathsf{h}}[\forall xPx] = \mathsf{S}$	2 unv (h arbitrary)
4.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert \mathsf{o})}[Px] = \mathsf{S})$	3 SF(∀)
5.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[Px] = \mathsf{S}$	4 unv
6.	$\mathsf{h}(x\vert 2)[x] = 2$	ins
7.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[x] = 2$	6 TA(v)
8.	$\mathsf{J}_{\mathsf{h}(x\vert 2)}[Px] = \mathsf{S} \Leftrightarrow \langle 2\rangle \in \mathsf{J}[P]$	7 SF(r)
9.	$\langle 2\rangle \in \mathsf{J}[P]$	5,8 bcnd
10.	$\langle 2\rangle \notin \mathsf{J}[P]$	ins
11.	$\perp$	9,10 bot
12.	$\mathsf{J}[\forall xPx] \neq \mathsf{T}$	1-11 neg

This takes extra lines, but may feel more natural insofar as it works down from the whole to the parts, as we have done for validity. The first version goes up from the parts to the whole, as we did showing invalidity for sentential forms. A choice between the two is a matter of style, not correctness.

Now to show that the premise is true, one option is to reason individually about each member of U. When the universe is small, this is always possible and sometimes necessary. Thus the argument is straightforward but tedious by methods we have seen before. Continuing from (AD),

10.	$\mathcal{A}\mathsf{o}(\mathsf{o} = 1 \triangledown \mathsf{o} = 2)$	ins
11.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 1 \triangledown \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 2$	10 unv (h, m arbitrary)
12.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 1$	assp
13.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = \mathsf{J}[f^1]\langle 1\rangle$	12 TA(f)
14.	$\vert\ \mathsf{J}[f^1]\langle 1\rangle = 1$	ins
15.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = 1$	13,14 eq
16.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S} \Leftrightarrow \langle 1\rangle \in \mathsf{J}[P]$	15 SF(r)
17.	$\vert\ \langle 1\rangle \in \mathsf{J}[P]$	ins
18.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S}$	16,17 bcnd
19.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 2$	assp
20.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = \mathsf{J}[f^1]\langle 2\rangle$	19 TA(f)
21.	$\vert\ \mathsf{J}[f^1]\langle 2\rangle = 1$	ins
22.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = 1$	20,21 eq
23.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S} \Leftrightarrow \langle 1\rangle \in \mathsf{J}[P]$	22 SF(r)
24.	$\vert\ \langle 1\rangle \in \mathsf{J}[P]$	ins
25.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S}$	23,24 bcnd
26.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S}$	11,12-18,19-25 dsj
27.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Pf^1x] = \mathsf{S})$	26 unv
28.	$\mathsf{J}_{\mathsf{h}}[\forall xPf^1x] = \mathsf{S}$	27 SF(∀)
29.	$\mathcal{A}\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall xPf^1x] = \mathsf{S})$	28 unv
30.	$\mathsf{J}[\forall xPf^1x] = \mathsf{T}$	29 TI
31.	$\mathsf{J}[\forall xPf^1x] = \mathsf{T} \vartriangle \mathsf{J}[\forall xPx] \neq \mathsf{T}$	30,9 cnj
32.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\forall xPf^1x] = \mathsf{T} \vartriangle \mathsf{I}[\forall xPx] \neq \mathsf{T})$	31 exs
33.	$\forall xPf^1x \not\vDash \forall xPx$	32 QV

$\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}$ has to be some member of U, so we instantiate the universal at (10) to it, and reason about the cases individually. This reflects what we have done before.

But on this interpretation, no matter what o may be, $\mathsf{I}[f^1]\langle\mathsf{o}\rangle = 1$. And, rather than the simple generalization about the universe, we might have generalized by ins about the interpretation of the function symbol itself. Thus we might have substituted for lines (10)–(26) as follows:

10.	$\vert\ \mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$	ins (h, m arbitrary)
11.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$	10 TA(v)
12.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = \mathsf{J}[f^1]\langle\mathsf{m}\rangle$	11 TA(f)
13.	$\vert\ \mathcal{A}\mathsf{o}(\mathsf{J}[f^1]\langle\mathsf{o}\rangle) = 1$	ins
14.	$\vert\ \mathsf{J}[f^1]\langle\mathsf{m}\rangle = 1$	13 unv
15.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1x] = 1$	12,14 eq
16.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S} \Leftrightarrow \langle 1\rangle \in \mathsf{J}[P]$	15 SF(r)
17.	$\vert\ \langle 1\rangle \in \mathsf{J}[P]$	ins
18.	$\vert\ \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Pf^1x] = \mathsf{S}$	16,17 bcnd

and pick up with (27) after. This is better! Before, we obtained the result when $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}$ was 1 and again when it was 2. But, in either case, the reason for the result is that the function has output 1. So this version avoids the cases by reasoning directly about the result from the function. Here is the informal version on this latter strategy:

For an arbitrary assignment h, $\mathsf{h}(x|2)[x] = 2$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x|2)}[x] = 2$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x|2)}[Px] = \mathsf{S}$ iff $\langle 2\rangle \in \mathsf{J}[P]$; but $\langle 2\rangle \notin \mathsf{J}[P]$; so $\mathsf{J}_{\mathsf{h}(x|2)}[Px] \neq \mathsf{S}$; so there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Px] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{J}_{\mathsf{h}}[\forall x Px] \neq \mathsf{S}$; so there is an assignment d such that $\mathsf{J}_{\mathsf{d}}[\forall x Px] \neq \mathsf{S}$; so by TI, $\mathsf{J}[\forall x Px] \neq \mathsf{T}$.

For arbitrary h and m, $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[x] = \mathsf{m}$; so by TA(f), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[f^1x] = \mathsf{J}[f^1]\langle \mathsf{m}\rangle$; but for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}[f^1]\langle \mathsf{o}\rangle = 1$; so $\mathsf{J}[f^1]\langle \mathsf{m}\rangle = 1$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[f^1x] = 1$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Pf^1x] = \mathsf{S}$ iff $\langle 1\rangle \in \mathsf{J}[P]$; but $\langle 1\rangle \in \mathsf{J}[P]$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Pf^1x] = \mathsf{S}$; so since m is arbitrary, for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Pf^1x] = \mathsf{S}$; so by SF($\forall$), $\mathsf{J}_{\mathsf{h}}[\forall x Pf^1x] = \mathsf{S}$; and since h is arbitrary, for any assignment d, $\mathsf{J}_{\mathsf{d}}[\forall x Pf^1x] = \mathsf{S}$; so by TI, $\mathsf{J}[\forall x Pf^1x] = \mathsf{T}$.

So there is an interpretation I such that $\mathsf{I}[\forall x Pf^1x] = \mathsf{T}$ and $\mathsf{I}[\forall x Px] \neq \mathsf{T}$; so by QV, $\forall x Pf^1x \not\models \forall x Px$.

Reasoning about cases is possible, and sometimes necessary, when the universe is small. But it is often convenient to organize your reasoning by generalizations about the interpretation as above. Such generalizations are required when the universe is large.

Here is a case that requires such generalizations insofar as the universe U has infinitely many members. Reasoning with $\mathscr{L}_{\mathrm{NT}}$, we show $\forall x\forall y(x \neq y \to Sx \neq Sy) \not\models \exists x(Sx = \emptyset)$. First note that no interpretation with finite U makes the premise true and conclusion false. To see this, let $\mathsf{I}[\emptyset]$ be some object o_0, and suppose successor connects it to just finitely many objects—so for some n there is a sequence,

$$\mathsf{o}_0 \longrightarrow \mathsf{o}_1 \longrightarrow \mathsf{o}_2 \longrightarrow \mathsf{o}_3 \longrightarrow \mathsf{o}_4 \longrightarrow \mathsf{o}_5 \longrightarrow \cdots \longrightarrow \mathsf{o}_\mathsf{n}$$

So $\mathsf{I}[S]$ includes $\langle \mathsf{o}_0, \mathsf{o}_1\rangle$, $\langle \mathsf{o}_1, \mathsf{o}_2\rangle$, $\langle \mathsf{o}_2, \mathsf{o}_3\rangle$, and so forth. But the interpretation of a function symbol is a total function; so $\mathsf{I}[S]$ pairs some object with o_n. This object cannot be any of o_1 through o_n, or the premise is violated insofar as some one thing is the successor of both o_n and the object before it. And if the conclusion is false no successor is equal to zero—so the object cannot be o_0. So successor connects o_n to an object other than any of $\mathsf{o}_0 \ldots \mathsf{o}_\mathsf{n}$—and so connects o_0 to more than n objects. Reject the assumption: there is no finite n such that successor connects o_0 to just n objects. But, as should be obvious by consideration of a standard interpretation of the symbols, the argument is not valid. To show this, let the interpretation be N, where,

$\mathsf{U} = \{0, 1, 2, \ldots\}$

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[S] = \{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \ldots\}$

$\mathsf{N}[=] = \{\langle 0, 0\rangle, \langle 1, 1\rangle, \langle 2, 2\rangle, \ldots\}$

First we show that $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$. Note that we might have specified the interpretation for equality by saying something like, $A\mathsf{o}A\mathsf{p}(\langle \mathsf{o}, \mathsf{p}\rangle \in \mathsf{N}[=] \Leftrightarrow \mathsf{o} = \mathsf{p})$.

Similarly, the interpretation of S is such that no o has a successor equal to zero—$\mathcal{A}\mathsf{o}(\mathsf{N}[S]\langle\mathsf{o}\rangle \neq \mathsf{0})$. We will simply appeal to these facts by ins in the following:

	1.	$\mathsf{N}[\emptyset] = \mathsf{0}$	ins (N particular)
	2.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[\emptyset] = \mathsf{0}$	1 TA(c) (h, m arbitrary)
	3.	$\mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$	ins
	4.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$	3 TA(v)
	5.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx] = \mathsf{N}[S]\langle\mathsf{m}\rangle$	4 TA(f)
	6.	$\mathsf{N}[S]\langle\mathsf{m}\rangle = \mathsf{q}$	def
	7.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx] = \mathsf{q}$	5,6 eq
	8.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx = \emptyset] = \mathsf{S} \Leftrightarrow \langle\mathsf{q}, \mathsf{0}\rangle \in \mathsf{N}[=]$	7,2 SF(r)
	9.	$\mathcal{A}\mathsf{o}(\mathsf{N}[S]\langle\mathsf{o}\rangle \neq \mathsf{0})$	ins
(AE)	10.	$\mathsf{N}[S]\langle\mathsf{m}\rangle \neq \mathsf{0}$	9 unv
	11.	$\mathsf{q} \neq \mathsf{0}$	10,6 eq
	12.	$\mathcal{A}\mathsf{o}\mathcal{A}\mathsf{p}(\langle\mathsf{o}, \mathsf{p}\rangle \in \mathsf{N}[=] \Leftrightarrow \mathsf{o} = \mathsf{p})$	ins
	13.	$\langle\mathsf{q}, \mathsf{0}\rangle \in \mathsf{N}[=] \Leftrightarrow \mathsf{q} = \mathsf{0}$	12 unv
	14.	$\langle\mathsf{q}, \mathsf{0}\rangle \notin \mathsf{N}[=]$	13,11 bcnd
	15.	$\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx = \emptyset] \neq \mathsf{S}$	8,14 bcnd
	16.	$\mathcal{A}\mathsf{o}(\mathsf{N}_{\mathsf{h}(x\vert\mathsf{o})}[Sx = \emptyset] \neq \mathsf{S})$	15 unv
	17.	$\mathsf{N}_{\mathsf{h}}[\exists x(Sx = \emptyset)] \neq \mathsf{S}$	16 SF′(∃)
	18.	$\mathcal{S}\mathsf{d}(\mathsf{N}_{\mathsf{d}}[\exists x(Sx = \emptyset)] \neq \mathsf{S})$	17 exs
	19.	$\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$	18 TI

Most of this is as usual. What is interesting is that at (9) we assert that no o is such that $\langle\mathsf{o}, \mathsf{0}\rangle \in \mathsf{N}[S]$. This should be obvious from the specification of $\mathsf{N}[S]$. And at (12) we assert by ins that for any o and p in U, $\langle\mathsf{o}, \mathsf{p}\rangle \in \mathsf{N}[=]$ iff $\mathsf{o} = \mathsf{p}$. Again, this should be clear from the initial (automatic) specification of $\mathsf{N}[=]$. In this case, there is no way to reason individually about *each* member of U, on the pattern of what we have been able to do with two-member universes. But we do not have to, as the general facts are sufficient for the result.

> Consider arbitrary h and m. $\mathsf{N}[\emptyset] = \mathsf{0}$; so by TA(c), $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[\emptyset] = \mathsf{0}$. But $\mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$; so by TA(f), $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx] = \mathsf{N}[S]\langle\mathsf{m}\rangle$; let $\mathsf{N}[S]\langle\mathsf{m}\rangle = \mathsf{q}$; then $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx] = \mathsf{q}$. From these, by SF(r), $(*)$ $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx = \emptyset] = \mathsf{S}$ iff $\langle\mathsf{q}, \mathsf{0}\rangle \in \mathsf{N}[=]$. For any $\mathsf{o} \in \mathsf{U}$, $\mathsf{N}[S]\langle\mathsf{o}\rangle \neq \mathsf{0}$; so $\mathsf{N}[S]\langle\mathsf{m}\rangle \neq \mathsf{0}$; so $\mathsf{q} \neq \mathsf{0}$; but for any $\mathsf{o}, \mathsf{p} \in \mathsf{U}$, $\langle\mathsf{o}, \mathsf{p}\rangle \in \mathsf{N}[=]$ iff $\mathsf{o} = \mathsf{p}$; so $\langle\mathsf{q}, \mathsf{0}\rangle \in \mathsf{N}[=]$ iff $\mathsf{q} = \mathsf{0}$; so $\langle\mathsf{q}, \mathsf{0}\rangle \notin \mathsf{N}[=]$; so with $(*)$, $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{m})}[Sx = \emptyset] \neq \mathsf{S}$; and since m is arbitrary, for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{N}_{\mathsf{h}(x\vert\mathsf{o})}[Sx = \emptyset] \neq \mathsf{S}$; so by SF′(∃), $\mathsf{N}_{\mathsf{h}}[\exists x(Sx = \emptyset)] \neq \mathsf{S}$; so there is an assignment d such that $\mathsf{N}_{\mathsf{d}}[\exists x(Sx = \emptyset)] \neq \mathsf{S}$; so by TI, $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$.

Given what we have already seen, this should be straightforward. Demonstration that $\mathsf{N}[\forall x \forall y(x \neq y \to Sx \neq Sy)] = \mathsf{T}$, and so that the argument is not valid, is left as an exercise. Hint: In addition to facts about equality, you may find it helpful to assert $\mathcal{A}\mathsf{o}\mathcal{A}\mathsf{p}(\mathsf{o} \neq \mathsf{p} \Rightarrow \mathsf{N}[S]\langle\mathsf{o}\rangle \neq \mathsf{N}[S]\langle\mathsf{p}\rangle)$. Be sure that you understand this before you assert it! Of course, we have here something that could never have been accomplished with trees insofar as the universe is infinite.

Recall that the interpretation of equality is the same across all interpretations. Thus our general assertion is possible in case of the arbitrary interpretation, and we are positioned to prove some last theorems.

T7.8. $\vDash (t = t)$

Hint: By ins for any I and any $\mathsf{o} \in \mathsf{U}$, $\langle \mathsf{o}, \mathsf{o} \rangle \in \mathsf{N}[=]$. Given this, the argument is easy.

*T7.9. $\vDash (x_i = y) \to (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$

Hint: If you have trouble with this, try showing a simplified version: $\vDash (x = y) \to (h^1 x = h^1 y)$.

T7.10. $\vDash (x_i = y) \to (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \to \mathcal{R}^n x_1 \ldots y \ldots x_n)$

Hint: If you have trouble with this, try showing a simplified version: $\vDash (x = y) \to (Rx \to Ry)$.

At this stage, we have introduced a method for reasoning about semantic definitions. As you continue to work with the definitions, it should become increasingly clear how they fit together into a coherent (and pleasing) whole. In following chapters, we will leave the metalinguistic derivation system behind as we encounter further definitions in diverse contexts. But from this chapter you should have gained a solid grounding in the *sort* of thing we will want to do.

E7.15. Produce interpretations (with, if necessary, variable assignments) and then metalinguistic derivations and informal reasoning to show each of the following.

a. $\exists x Px \nvDash Pa$

*b. $\nvDash f^1 g^1 x = g^1 f^1 x$

c. $\forall x Ax \to C \nvDash \forall x(Ax \to C)$

d. $\exists x Fx, \exists y Gy \nvDash \exists z(Fz \wedge Gz)$

e. $\forall x \exists y Axy \nvDash \exists y \forall x Axy$

*E7.16. Provide demonstrations for T7.8 and simplified versions of T7.9, T7.10 in the informal style. Hint: You may or may not decide that a metalinguistic derivation would be helpful. Challenge: can you show the theorems in their general form?

Theorems of Chapter 7

T7.1s $\mathscr{P}, \mathscr{P} \to \mathscr{Q} \vDash_s \mathscr{Q}$

T7.2s $\vDash_s \mathscr{P} \to (\mathscr{Q} \to \mathscr{P})$

T7.3s $\vDash_s (\mathscr{O} \to (\mathscr{P} \to \mathscr{Q})) \to ((\mathscr{O} \to \mathscr{P}) \to (\mathscr{O} \to \mathscr{Q}))$

T7.4s $\vDash_s (\sim\mathscr{Q} \to \sim\mathscr{P}) \to [(\sim\mathscr{Q} \to \mathscr{P}) \to \mathscr{Q}]$

T7.1 $\mathscr{P}, \mathscr{P} \to \mathscr{Q} \vDash \mathscr{Q}$

T7.2 $\vDash \mathscr{P} \to (\mathscr{Q} \to \mathscr{P})$

T7.3 $\vDash (\mathscr{O} \to (\mathscr{P} \to \mathscr{Q})) \to ((\mathscr{O} \to \mathscr{P}) \to (\mathscr{O} \to \mathscr{Q}))$

T7.4 $\vDash (\sim\mathscr{Q} \to \sim\mathscr{P}) \to [(\sim\mathscr{Q} \to \mathscr{P}) \to \mathscr{Q}]$

T7.5 There is no interpretation I and formula $\mathscr{P}$ such that $\mathsf{I}[\mathscr{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$.

T7.6 For any I and $\mathscr{P}$, $\mathsf{I}[\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\forall x \mathscr{P}] = \mathsf{T}$.

T7.7 Each of the following conditions obtains.

(a) $\mathsf{I_d}[(\forall x : \mathscr{B})\mathscr{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{B}] \neq \mathsf{S}$ or $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$.

(b) $\mathsf{I_d}[(\exists x : \mathscr{B})\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{B}] = \mathsf{S}$ and $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$.

T7.8 $\vDash (t = t)$

T7.9 $\vDash (x_i = y) \to (h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n)$

T7.10 $\vDash (x_i = y) \to (\mathscr{R}^n x_1 \dots x_i \dots x_n \to \mathscr{R}^n x_1 \dots y \dots x_n)$

E7.17. Show that $\mathsf{N}[\forall x \forall y(x \neq y \to Sx \neq Sy)] = \mathsf{T}$, and so complete the demonstration that $\forall x \forall y(x \neq y \to Sx \neq Sy) \nvDash \exists x(Sx = \emptyset)$. You may simply assert that $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$ with justification, "from the text."

*E7.18. Here is an interpretation to show $\nvDash \exists x \forall y[(Axy \wedge \sim Ayx) \to (Axx \leftrightarrow Ayy)]$.

$\mathsf{U} = \{1, 2, 3, \dots\}$

$\mathsf{I}[A] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m} \leq \mathsf{n}$ and m is odd, or $\mathsf{m} < \mathsf{n}$ and m is even$\}$

So $\mathsf{I}[A]$ has members,

$\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle, \dots \quad \langle 2, 3\rangle, \langle 2, 4\rangle, \langle 2, 5\rangle, \dots$

$\langle 3, 3\rangle, \langle 3, 4\rangle, \langle 3, 5\rangle, \dots \quad \langle 4, 5\rangle, \langle 4, 6\rangle, \langle 4, 7\rangle, \dots$

and so forth. Try to understand *why* this works, and why $\leq$ or $<$ will not work by themselves. Then find an interpretation where U has $\leq$ four members and use it to demonstrate that $\nvDash \exists x \forall y[(Axy \wedge \sim Ayx) \rightarrow (Axx \leftrightarrow Ayy)]$. Hint: This is challenging.

E7.19. Consider $\mathcal{L}_{NT}$ and the axioms of Robinson Arithmetic as in Chapter 6 (page 300). (a) Use the standard interpretation N to show $\mathrm{Q} \nvDash \sim\forall x \forall y[(x \times y) = (y \times x)]$. And (b) using N from below, show $\mathrm{Q} \nvDash \forall x \forall y[(x \times y) = (y \times x)]$. You need only complete parts not worked in the answer to this exercise. For N*, let $\mathsf{U}^* = \mathbb{N} \cup \{a\}$ for some object a that is not a number; assign 0 to $\emptyset$ in the usual way; then,

S	
i	$i+1$
a	a

$+$	j	a
i	$i+j$	a
a	a	a

$\times$	0	$j \neq 0$	a
0	0	0	a
$i \neq 0$	0	$i \times j$	a
a	0	a	a

So, for example, from the top row of the '+' table, $\langle\langle i, j\rangle, i + j\rangle$, $\langle\langle i, a\rangle, a\rangle \in \mathsf{N}^*[+]$. Hint: This is no different than you have done before, only with premises the axioms of Q. Also notice that N* is the same as N for $\mathsf{m}, \mathsf{n} \in \mathbb{N}$ so that reasoning about N* partially coincides with reasoning about N. This lets you collapse some of the work: So, for example, when variables are assigned to some $\mathsf{m}, \mathsf{n} \in \mathsf{U}^*$, there are cases for (i) $\mathsf{m}, \mathsf{n} \in \mathbb{N}$, (ii) $\mathsf{m} \in \mathbb{N}, \mathsf{n} = \mathsf{a}$, (iii) $\mathsf{m} = \mathsf{a}, \mathsf{n} \in \mathbb{N}$, (iv) $\mathsf{m} = \mathsf{a}, \mathsf{n} = \mathsf{a}$. By itself (i) is sufficient for a result about N.

This result (together with T10.5) is sufficient to show that Robinson Arithmetic is not *negation complete*—there are sentences $\mathcal{P}$ of $\mathcal{L}_{NT}$ such that Q proves neither $\mathcal{P}$ nor $\sim\mathcal{P}$.

E7.20. For each of the following concepts, explain in an essay of about two pages, so that (high-school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The difference between satisfaction and truth.

b. The definitions SF(r) and SF($\forall$).

c. The way your reasoning works. For this you can provide an example of some reasonably complex but clean bits of reasoning, (a) for validity, and (b) for invalidity. Then explain to Hannah how your reasoning works. That is, provide her a commentary on what you have done, so that she could understand.

Chapter 8

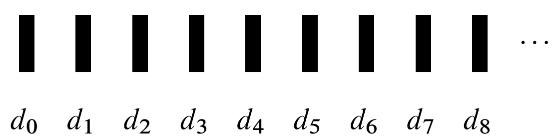
Mathematical Induction

In Chapter 1 (page 12), we distinguished *deductive* from *inductive* arguments. As described there, in a deductive argument, conclusions are supposed to be *guaranteed* by premises. In an inductive argument, conclusions are merely made probable or plausible. Typical cases of inductive arguments involve generalization from cases. Thus, for example, one might reason from the premise that every crow we have ever seen is black, to the conclusion that all crows are black. The premise does not *guarantee* the conclusion, but it does give it some probability or plausibility. Similarly, mathematical induction involves a sort of generalization. But mathematical induction is a *deductive* argument form. The conclusion of a valid argument by mathematical induction is *guaranteed* by its premises. So mathematical induction is to be distinguished from the sort of induction described in Chapter 1.

In this chapter, I begin with a general characterization of mathematical induction, and turn to a series of examples. Some of the examples will matter for things to come. But the primary aim is to gain facility with this crucial argument form. After a general characterization in section 8.1, there are some introductory examples (section 8.2) then cases of special interest for Part III (section 8.3) and for Part IV (section 8.4).

8.1 General Characterization

Arguments by mathematical induction apply to objects that are arranged in *series*. The conclusion of an argument by mathematical induction is that all the elements of the series are of a certain sort. For cases with which we will be concerned, the elements of a series are ordered by natural numbers: there is an initial member, one after that, and so forth (we may thus think of a series as a *function* from the numbers to the members). Consider, for example, a series of dominoes:



This series is ordered spatially. So d_0 is the initial domino, d_1 the next, and so forth. Alternatively, we might think of the series as defined by a function $\mathscr{D}$ from the natural numbers to the dominoes, with $\mathscr{D}(0) = d_0$, $\mathscr{D}(1) = d_1$, and so forth—where this ordering is merely *exhibited* by the spatial arrangement.

Suppose we are interested in showing that *all* the dominoes fall, and consider the following two claims:

(i) The first domino falls.

(ii) For any domino, if all the ones prior to it fall, then it falls.

By itself, (i) does not tell us that all the dominoes fall. For all we know, there might be some flaw in the series so that for some k dominoes prior to d_k fall, but d_k does not. Perhaps the space between d_{k-1} and d_k is too large. In this case, under ordinary circumstances, neither d_k nor any of the dominoes after it fall. Claim (ii) tells us that there is no such flaw in the series—if all the dominoes up to d_k fall, then d_k falls. But (ii) is not, by itself, sufficient for the conclusion that all the dominoes fall. From the fact that the dominoes are so arranged, it does not follow that *any* of the dominoes fall. Perhaps you do the arrangement, and are so impressed with your work, that you leave the setup forever as a memorial!

However, given both (i) and (ii), it is safe to conclude that all the dominoes fall. There are a couple of ways to see this. First, we can reason from one domino to the next. By (i), the first domino falls. This means that all the dominoes prior to the second domino fall. So by (ii), the second falls. But this means all the dominoes prior to the third fall. So by (ii), the third falls. So all the dominoes prior to the fourth fall. And so forth. Thus we reach the conclusion that each domino falls. Here is another way to make the point: Suppose not every member of the series falls. Then there must be some *least* member d_a of the series which does not fall. This d_a cannot be the first member of the series, since by (i) the first member of the series falls. And since d_a is the least member of the series which does not fall, all the members of the series prior to it *do* fall. So by (ii), d_a falls. This is impossible; reject the assumption: every member of the series falls.

Suppose we have some reason for accepting (i) that the first domino falls—perhaps you push it with your finger. Suppose further, that we have some "special reason" for moving from the premise that all the dominoes prior to an arbitrary d_k fall, to the conclusion that d_k falls—perhaps the setup satisfies some rule that adaquetly relates each dominoe d_k to the ones before. Then we might attempt to show that all the dominoes fall as follows:

	a.	d_0 falls	prem (d_0 particular)
	b.	all the dominoes prior to d_k fall	assp (d_k arbitrary)
		⋮	
(A)	c.	d_k falls	"special reason"
	d.	if all the dominoes prior to d_k fall, then d_k falls	b-c cnd
	e.	for any domino, if all the dominoes prior to it fall, then it falls	d unv
	f.	every domino falls	a,e induction

(a) is (i) and (e) is (ii); the conclusion that every domino falls follows from (a) and (e) by mathematical induction. In this case, (a) is given; d_0 falls because you push it. In order to obtain (e), for arbitrary d_k we reason from the assumption at (b) to the conclusion that d_k falls, and then move to (e) by cnd and unv. This is in fact how we reason. However, all the moves are automatic once we complete the subderivation—the moves by cnd to get (d), by unv to get (e), and by mathematical induction to get (f) are automatic once we reach (c). In practice, then, those steps are usually left implicit and omitted. Having gotten (a) and, from the assumption that all the dominoes prior to d_k fall reached the conclusion that d_k falls, we move directly to the conclusion that all the dominoes fall.

Thus we arrive at a general form for arguments by mathematical induction. Suppose we want to show that $\mathscr{P}$ holds for each member of some series. Then an argument from mathematical induction goes as follows:

(B) *Basis*: Show that $\mathscr{P}$ holds for the first member of the series.

Assp: Assume, for arbitrary k, that $\mathscr{P}$ holds for every member of the series prior to the k^{th} member.

Show: Show that $\mathscr{P}$ holds for the k^{th} member of the series.

Indct: Conclude that $\mathscr{P}$ holds for every member of the series.

In the domino case, for the *basis* we show (i). At the *assp* (assumption) step, we assume that all the dominoes prior to d_k fall. In the *show* step, we would complete the subderivation with the conclusion that domino d_k falls. From this, moves by cnd to the conditional statement, and by unv to its generalization, are omitted and we move directly to the conclusion that all the dominoes fall. Notice that the assumption is nothing more than a standard assumption for the (suppressed) application of cnd.

Perhaps the "special reason" is too special, and it is not clear how we might generally reason from the assumption that some $\mathscr{P}$ holds for every member of a series prior to the k^{th}, to the conclusion that it holds for the k^{th}. For our purposes, the key is that such reasoning is possible in contexts characterized by *recursive definitions*. As we have seen, a recursive definition always moves from the parts to the whole. There are some *basic* elements, and some rules for combining elements to form further elements. In general, it is a fallacy (the fallacy of *composition*) to move directly from characteristics of parts, to characteristics of a whole. From the fact that the bricks are small, it does not follow that a building made from them is small. But there are cases where facts about parts, together with the way they are arranged, are sufficient for conclusions about wholes. If the bricks are hard, it may be that the building is hard. And similarly with recursive definitions.

To see how this works, let us turn to another example. We show that every *term* of a certain language has an odd number of symbols. Recall that the recursive definition TR tells us how terms are formed from others. Variables and constants are terms; and if $\hbar^n$ is a n-place function symbol and $t_1 \dots t_n$ are n terms, then $\hbar^n t_1 \dots t_n$ is a term. On tree diagrams, across the top (row 0) are variables and constants—terms with no function symbols; in the next row are terms constructed out of them, and for any $n \geq 1$, terms in row n are constructed out of terms from earlier rows. Let this *series* of rows be our series for mathematical induction. Every term must appear in some row of a tree. We consider a series whose first element consists of terms which appear in the top row of a tree, whose second element consists of terms which appear in the next, and so forth. Let $\mathscr{L}_t$ be a language with variables and constants as usual, but just two function symbols, a two-place function symbol f^2 and a four-place function symbol g^4. We show, by induction on the rows in which terms appear, that the total number of symbols in any term t of this language is odd. Here is the argument:

(C) *Basis*: If t appears in the top row, then it is a variable or a constant; in this case, t consists of just one variable or constant symbol; so the total number of symbols in t is odd.

Assp: For any i such that $0 \leq i < k$, the total number of symbols in any t appearing in row i is odd.

Show: The total number of symbols in any t appearing in row k is odd.

If t appears in row k, then it is of the form $f^2 t_1 t_2$ or $g^4 t_1 t_2 t_3 t_4$ where $t_1 \dots t_4$ appear in rows prior to k. So there are two cases.

(f) Suppose t is $f^2 t_1 t_2$. Let a be the total number of symbols in t_1, and b be the total number of symbols in t_2; then the total number of symbols in t is $(a+b)+1$: all the symbols in t_1, all the symbols in t_2, plus the symbol f^2. Since t_1 and t_2 each appear in rows prior to k, by assumption, both a and b are odd. But the sum of two odds is an even, and the sum of an even plus one is odd; so $(a+b)+1$ is odd; so the total number of symbols in t is odd.

(g) Suppose t is $g^4 t_1 t_2 t_3 t_4$. Let a be the total number of symbols in t_1, b be the total number of symbols in t_2, c be the total number of symbols in t_3, and d be the total number of symbols in t_4; then the total number of symbols in t is $[(a+b)+(c+d)]+1$. Since $t_1 \dots t_4$ each appear in rows prior to k, by assumption a, b, c, and d are all odd. But the sum of two odds is an even; the sum of two evens is an even, and the sum of an even plus one is odd; so $[(a+b)+(c+d)]+1$ is odd; so the total number of symbols in t is odd.

In either case, then, if t appears in row k, the total number of symbols in t is odd.

Indct: For any term t in $\mathcal{L}_t$, the total number of symbols in t is odd.

Notice that this argument is *entirely structured by the recursive definition for terms*. The definition TR includes clauses (v) and (c) for terms that appear in the top row. In the basis stage, we show that all such terms consist of an odd number of symbols. Then, for (suppressed) application of cnd and unv we assume that all terms prior to an arbitrary row k have an odd number of symbols. After that, the *show* line simply announces what we plan to do. Observe the way reasoning for the show part works:

$$
\begin{tikzcd}
\text{item at stage } k \arrow[d] & & \text{result at stage } k \\
\text{items at stages prior to } k \arrow[rr, "\textit{inductive assumption}"] & & \text{results at stages prior to } k \arrow[u]
\end{tikzcd}
$$

By the recursive definition, items at stage k result from items at stages prior to k. The inductive assumption applies to the items at stages prior to k, and so gives a result for those items. And with the recursive definition we put those results together for a conclusion about stage k. Over and over, you will be able to reason according to this pattern. And our argument proceeds in just this way: The sentence after *show* says how terms at stage k derive from ones before—if $f^2t_1t_2$ appears in row k, t_1 and t_2 must appear in previous rows; then by the assumption they have an odd number of symbols; and since the number of symbols in the parts are odd, the number of symbols in the whole is odd. And similarly for $g^4t_1t_2t_3t_4$. So any term in row k has an odd number of symbols. Then by induction it follows that every term in this language $\mathcal{L}_t$ consists of an odd number of symbols.

Returning to the domino analogy, the basis is like (i), where we show that the first member of the series falls—terms appearing in the top row always have an odd number of symbols. Then, for arbitrary k, we assume that all the members of the series prior to the k^{th} fall—that terms appearing in rows prior to the k^{th} always have an odd number of symbols. We then reason that, given this, the k^{th} member falls—terms constructed out of others which, by assumption, have an odd number of symbols must themselves have an odd number of symbols. From this, (ii) follows by cnd and unv, and the general conclusion by mathematical induction.

The argument works for the same reasons as before: Insofar as a variable or constant is regarded as a single element of the vocabulary, it is automatic that variables and constants have an odd number of symbols. So terms in the top row have an odd number of symbols. Given this expressions in the next row of a tree, as f^2xc, or g^4xycz, must have an odd number of symbols—one function symbol, plus two or four variables and constants. But if terms from rows zero and one of a tree have an odd number of symbols, by reasoning from the show step, terms constructed out of *them* must have an odd number of symbols as well. And so forth. So terms in all the rows have an odd number of symbols. Here is the other way to think about it: Suppose some terms in $\mathcal{L}_t$ have an even number of symbols; then there must be a least row a where such terms appear. From the basis, this row a is not the top row. But since a is the least row at which terms have an even number of symbols, terms at all the earlier rows must have an odd number of symbols. But then, by reasoning as in the show step, terms at row a have an odd number of symbols. Reject the assumption: no terms in $\mathcal{L}_t$ have an even number of symbols.

In practice, for this sort of case it is common to reason, not based on the row in which a term appears, but on the number of function symbols in the term. This differs in detail, but not in effect, from what we have done. In our trees, it may be that a term in row two, combining one from row zero and another from row one, has two function symbols, as f^2xf^2ab, or it may be that a term in row two, combining terms from row one, has three function symbols, as $f^2f^2xyf^2ab$, or five, as $g^4f^2xyf^2abf^2zwf^2cd$, and so forth. However, it remains that the total number of function symbols in each of some terms $\mathscr{s}_1 \dots \mathscr{s}_n$ is fewer than the total number of function symbols in $\mathscr{h}^n\mathscr{s}_1 \dots \mathscr{s}_n$; for the latter includes all the function symbols in $\mathscr{s}_1 \dots \mathscr{s}_n$ plus $\mathscr{h}^n$. Thus we may consider the series: terms with no function symbols, terms with one function symbol, and so forth—and be sure that for any $n > 0$, terms at stage n are constructed of ones before. Here is a sketch of the argument modified along these lines:

(D) *Basis*: If t has no function symbols, then it is a variable or a constant; in this case, t consists of just the one variable or constant symbol; so the total number of symbols in t is odd.

Assp: For any i such that $0 \leq i < k$, the total number of symbols in t with i function symbols is odd.

Show: The total number of symbols in t with k function symbols is odd.

If t has k function symbols, then it is of the form $f^2t_1t_2$ or $g^4t_1t_2t_3t_4$ where $t_1 \dots t_4$ have less than k function symbols. So there are two cases.

(f) Suppose t is $f^2t_1t_2$. [As before…] the total number of symbols in t is odd.

(g) Suppose t is $g^4t_1t_2t_3t_4$. [As before…] the total number of symbols in t is odd.

In either case, then, if t has k function symbols, then the total number of symbols in t is odd.

Indct: For any term t in $\mathcal{L}_t$, the total number of symbols in t is odd.

Here is the key point: If $f^2t_1t_2$ has k function symbols, the number of function symbols in t_1 and t_2 combined has to be $k - 1$; and since the number of function symbols in t_1 and in t_2 must individually be less than or equal to their combined total, the number of function symbols in t_1 and the number of function symbols in t_2 must also be less than k. And similarly for $g^4 t_1 t_2 t_3 t_4$. That is why the inductive assumption applies to $t_1 \dots t_4$, and reasoning in the cases can proceed as before.

8.2 Preliminary Examples

Let us turn now to a series of examples, meant to illustrate mathematical induction in a variety of contexts. Some of the examples have to do with our subject matter. But some do not. For now, the primary aim is to gain facility with the argument form. As you work through the cases, think about *why* the induction works. At first, examples may be difficult to follow. But they should be more clear by the end.

8.2.1 Case

First, a case where the conclusion may seem too obvious even to merit argument. We show that any (official) formula $\mathcal{P}$ of a quantificational language has an equal number of left and right parentheses. Again, the relevant definition FR is recursive. Its basis clause specifies formulas without operator symbols; these occur across the top row of our trees. FR then includes clauses which say how complex formulas are constructed out of those that are less complex. We take as our series, formulas with no operator symbols, formulas with one operator symbol, and so forth; thus the argument is by induction on the *number of operator symbols*. As in the above case with terms, this orders formulas so that we can use facts from the recursive definition in our reasoning. Let us say $\mathrm{L}(\mathcal{P})$ is the number of left parentheses in $\mathcal{P}$, and $\mathrm{R}(\mathcal{P})$ is the number of right parentheses in $\mathcal{P}$. Our goal is to show that for any formula $\mathcal{P}$, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

(E) *Basis*: If $\mathcal{P}$ has no operator symbols, then $\mathcal{P}$ is a sentence letter $\mathcal{S}$ or an atomic $\mathcal{R}^n t_1 \dots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. In either case, $\mathcal{P}$ has no parentheses. So $\mathrm{L}(\mathcal{P}) = 0$ and $\mathrm{R}(\mathcal{P}) = 0$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Assp: For any i such that $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Show: For every $\mathcal{P}$ with k operator symbols, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $(\mathcal{A} \rightarrow \mathcal{B})$, or $\forall x \mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A})$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A})$. By the inductive assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$. So $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A}) = \mathrm{R}(\mathcal{P})$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

($\rightarrow$) Suppose $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$. Then $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A}) + \mathrm{L}(\mathcal{B}) + 1$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A}) + \mathrm{R}(\mathcal{B}) + 1$. By assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$, and $\mathrm{L}(\mathcal{B}) = \mathrm{R}(\mathcal{B})$. So $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A}) + \mathrm{L}(\mathcal{B}) + 1 = \mathrm{R}(\mathcal{A}) + \mathrm{R}(\mathcal{B}) + 1 = \mathrm{R}(\mathcal{P})$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Induction Schemes

Schemes for mathematical induction sometimes appear in different forms. But for our purposes, these amount to the same thing. Suppose a series of objects, and consider the following:

I.

(a) Show that $\mathscr{P}$ holds for the first member
(b) Assume that $\mathscr{P}$ holds for members $< k$
(c) Show $\mathscr{P}$ holds for member k
(d) Conclude $\mathscr{P}$ holds for every member

This is the form as we have seen it.

II.

(a) Show that $\mathscr{P}$ holds for the first member
(b) Assume that $\mathscr{P}$ holds for members $\leq j$
(c) Show $\mathscr{P}$ holds for member $j + 1$
(d) Conclude $\mathscr{P}$ holds for every member

This comes to the same thing if we think of j as $k - 1$. Then $\mathscr{P}$ holds for members $\leq j$ just in case it holds for members $< k$.

III.

(a) Show that $\mathscr{Q}$ holds for the first member
(b) Assume that $\mathscr{Q}$ holds for member j
(c) Show $\mathscr{Q}$ holds for member $j + 1$
(d) Conclude $\mathscr{Q}$ holds for every member

This comes to the same thing if we think of j as $k - 1$ and $\mathscr{Q}$ as the proposition that $\mathscr{P}$ holds for members $\leq j$.

And similarly the other forms follow from ours. So though in a given context one form may be more convenient than another, the forms are equivalent—or at least they are equivalent for sequences corresponding to the natural numbers.

Our form of induction (I) is known as "strong induction," for its relatively strong inductive assumption, and the third as "weak." The second is a sometimes-encountered blend of the other two. In PA the weak form is mirrored by axiom PA7; we use that axiom to prove a theorem like (II) in T13.11ai.

It turns out that mathematical induction can be applied not only to sequences corresponding to the natural numbers but also to sequences indexed by infinite ordinals. Though we wave in that direction in section 11.4, our main concerns will be restricted to series ordered by the natural numbers. The infinite ordinals are a topic for a course in set theory.

Still, a remark for the interested: The first infinite ordinal ω is the number of the series $0, 1, 2, \ldots$. But there is no finite number n such that $n + 1 = \omega$—for any finite n, $n + 1$ is just *another* member of the series. So for a sequence ordered by infinite ordinals, our assumption that $\mathscr{P}$ holds for all the members $< k$ might hold though there *is* no $j = k - 1$ as in the second and third cases. So the equivalence between the forms breaks down for series that are so ordered.

($\forall$) Suppose $\mathcal{P}$ is $\forall x\mathcal{A}$. Then as in the case for ($\sim$), $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A})$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A})$. By assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$. So $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A}) = \mathrm{R}(\mathcal{P})$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

If $\mathcal{P}$ has k operator symbols, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Indct: For any formula $\mathcal{P}$, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

No doubt, you already knew that the numbers of left and right parentheses match. But, presumably, you knew it by reasoning of *this very sort*. Atomic formulas have no parentheses, and so an equal number of left and right parentheses; after that, parentheses are always added in pairs; so, no matter how complex a formula may be, there is never a left parenthesis without a right to match. Reasoning by mathematical induction may thus seem perfectly natural! All we have done is to make explicit the various stages that are required to reach the conclusion. But it is important to make the stages explicit, for many cases are not so obvious. Notice again: We understand formulas at stage k in terms of formulas from stages before—and so to which the assumption applies—and then put the results together for a conclusion about stage k.

Here are some closely related problems:

*E8.1. For any (official) formula $\mathcal{P}$ of a quantificational language, where $\mathrm{A}(\mathcal{P})$ is the number of its atomic formulas, and $\mathrm{B}(\mathcal{P})$ is the number of its arrow symbols, show that $\mathrm{A}(\mathcal{P}) = \mathrm{B}(\mathcal{P}) + 1$. Hint: Argue by induction on the number of operator symbols in $\mathcal{P}$. For the basis, when $\mathcal{P}$ has no operator symbols, it is an atomic, so that $\mathrm{A}(\mathcal{P}) = 1$ and $\mathrm{B}(\mathcal{P}) = 0$. Then, as above, you will have cases for $\sim$, $\rightarrow$, and $\forall$. The hardest case is when $\mathcal{P}$ is of the form $(\mathcal{A} \rightarrow \mathcal{B})$.

E8.2. Consider now expressions which allow abbreviations ($\vee$), ($\wedge$), ($\leftrightarrow$), and ($\exists$). Where $\mathrm{A}(\mathcal{P})$ is the number of atomic formulas in $\mathcal{P}$ and $\mathrm{B}(\mathcal{P})$ is the number of its binary operators, show that $\mathrm{A}(\mathcal{P}) = \mathrm{B}(\mathcal{P}) + 1$. Hint: Now you have seven cases: ($\sim$), ($\rightarrow$), and ($\forall$) as before, but also cases for ($\vee$), ($\wedge$), ($\leftrightarrow$), and ($\exists$). This suggests the beauty of reasoning just about the minimal language!

8.2.2 Case

Many applications of mathematical induction occur in mathematics. It will be helpful to have a couple of examples of this sort. These should be illuminating—at least if you do not get bogged down in the details of the arithmetic! The series of odd positive integers is $1, 3, 5, 7, \ldots$ where the n^{th} odd number is $2n - 1$. (The n^{th} even number is $2n$; to find the n^{th} odd, go to the even just above it, and come down one.) Let $\mathrm{S}(n)$ be the sum of the first n odd positive integers. So $\mathrm{S}(1) = 1$, $\mathrm{S}(2) = 1 + 3 = 4$, $\mathrm{S}(3) = 1 + 3 + 5 = 9$, $\mathrm{S}(4) = 1 + 3 + 5 + 7 = 16$ and, in general,

$$\mathrm{S}(n) = 1 + 3 + \cdots + (2n - 1)$$

We consider the series of these sums, S(1), S(2), and so forth, and show that, for any $n \geq 1$, $S(n) = n^2$. Observe that $S(1) = 1$, and for $n > 1$, $S(n) = S(n-1) + (2n-1)$. The sum of all the odd numbers up to the n^{th} odd number is equal to the sum of all the odd numbers up to the $(n-1)^{th}$ odd number plus the n^{th} odd number—and since the n^{th} odd number is $2n-1$, $S(n) = S(n-1) + (2n-1)$. This gives us the required recursive connection between a member of the series and one before. Given this, the argument is straightforward. We argue by induction on the series of sums.

(F) *Basis*: If $n = 1$ then $S(n) = 1$ and $n^2 = 1$; so $S(n) = n^2$.

Assp: For any i, $1 \leq i < k$, $S(i) = i^2$.

Show: $S(k) = k^2$. As above, $S(k) = S(k-1)+(2k-1)$. But since $k-1 < k$, by the inductive assumption, $S(k-1) = (k-1)^2$; so $S(k) = (k-1)^2 + (2k-1) = (k^2 - 2k + 1) + (2k-1) = k^2$. So $S(k) = k^2$.

Indct: For any n, $S(n) = n^2$.

As is often the case in mathematical arguments, the k^{th} element is completely determined by the one immediately before; so we do not need to consider any more than this one way that elements at stage k are determined by those at earlier stages.[1] Surely this is an interesting result—though you might have wondered about it after testing initial cases, we have a demonstration that it holds for every n.

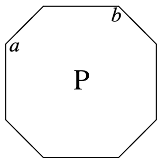
*E8.3. Let $S(n)$ be the sum of the first n even positive integers; that is $S(n) = 2+4+\cdots+2n$. So $S(1) = 2$, $S(2) = 2+4 = 6$, $S(3) = 2+4+6 = 12$, and so forth. Show by mathematical induction that for any $n \geq 1$, $S(n) = n(n+1)$.

E8.4. Let $S(n)$ be the sum of the first n positive integers; that is $S(n) = 1+2+3+\cdots+n$. So $S(1) = 1$, $S(2) = 1+2 = 3$, $S(3) = 1+2+3 = 6$, and so forth. Show by mathematical induction that for any $n \geq 1$, $S(n) = n(n+1)/2$.

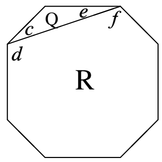
8.2.3 Case

Now a case from geometry. Where a polygon is *convex* iff each of its interior angles is less than 180°, we show that the sum of the interior angles in any convex polygon with n sides is equal to $(n-2)180°$. Let us consider polygons with three sides, polygons with four sides, polygons with five sides, and so forth. The key is that when $n > 3$, any n-sided polygon may be regarded as one with $n-1$ sides combined with a triangle. Thus given an n-sided polygon P,

[1] Thus arguments by induction in arithmetic and geometry are often conveniently cast according to the third "weak" induction scheme from the induction schemes reference on page 368. But, as above, our standard scheme applies as well.

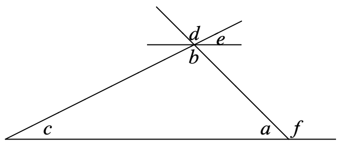


Construct a line connecting opposite ends of a pair of adjacent sides.



The result is a triangle Q and a figure R with $n-1$ sides, where $a = c+d$ and $b = e+f$. The sum of the interior angles of P is the same as the sum of the interior angles of Q plus the sum of the interior angles of R. Once we realize this, our argument by mathematical induction is straightforward. For any convex n-sided polygon P, we show that the sum of the interior angles of P, S(P) = $(n-2)180°$. The argument is by induction on the number n of sides of the polygon.

(G) *Basis*: If $n = 3$, then P is a triangle; but by reasoning as follows,



By definition, $a + f = 180°$; but $b = d$ and if the horizontal lines are parallel, $c = e$ and $d + e = f$; so $a + (b + c) = a + (d + e) = a + f = 180°$.

the sum of the angles in a triangle is 180°. So S(P) = 180°. But $(3-2)180° = 180°$. So S(P) = $(n-2)180°$.

Assp: For any i, $3 \leq i < k$, every P with i sides has S(P) = $(i-2)180°$.

Show: For every P with k sides, S(P) = $(k-2)180°$.

For P with k sides, construct a line connecting opposite ends of a pair of adjacent sides; the result divides P into a triangle Q and polygon R with $k-1$ sides such that S(P) = S(Q) + S(R). Q is a triangle, so S(Q) = 180°. Since $k-1 < k$, the inductive assumption applies to R; so S(R) = $((k-1)-2)180°$. So S(P) = $180° + ((k-1)-2)180° = (1+k-1-2)180° = (k-2)180°$. So S(P) = $(k-2)180°$.

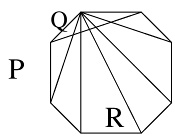
Indct: For any n-sided polygon P, S(P) = $(n-2)180°$.

Perhaps reasoning in the basis brings back good (or bad!) memories of high school geometry. But you do not have to worry about that.

In this case, the sum of the angles of a figure with n sides is completely determined once we are given the sum of the angles for a figure with $n-1$ sides. So we do not need to consider any more than this one way that elements at stage k are determined by those at earlier stages. It is worth noting however that we do not have to see a k-sided polygon as composed of a triangle and a figure with $k-1$ sides. For consider *any* diagonal of a k-sided polygon; it divides the figure into two, each with $< k$ sides. So the inductive assumption applies to each figure. So we might reason about the angles of a k-sided figure as the sum of angles of these arbitrary parts, as in the exercise that follows.

*E8.5. Using the fact that for $k > 3$ any diagonal of a k-sided polygon divides it into two polygons with $< k$ sides, show by mathematical induction that the sum of the interior angles of any n-sided convex polygon P, S(P) = $(n - 2)180°$. Hint: If a figure has k sides, then for some a such that both a and $k - a$ are at least two (> 1), a diagonal divides it into a figure Q with $a + 1$ sides (a sides from P, plus the diagonal), and a figure R with $(k - a) + 1$ sides (the remaining sides from P, plus the diagonal). From $k - a > 1$, $k > a + 1$; and from $a > 1$, $k + a > k + 1$ so that $k > (k - a) + 1$. So the inductive assumption applies to both Q and R.

E8.6. Where P is a convex polygon with n sides, and D(P) is the number of its diagonals (where a *diagonal* is a line from one vertex to another that is not a side), show by mathematical induction that any P with $n \geq 3$ sides is such that D(P) = $n(n - 3)/2$.



Hint: For P with k sides, connecting the vertices of adjacent sides divides P into a triangle Q and a convex figure R with $k - 1$ sides. Then the diagonals are all the diagonals of R, plus the base of the triangle, plus $k - 3$ lines from vertices not belonging to the triangle to the apex of the triangle (P has k vertices, and diagonals from the apex go to all but 3 of them). Also, in case your algebra is rusty, $(k - 1)(k - 4) = k^2 - 5k + 4$.

8.2.4 Case

Finally we take up a couple of cases of real interest for our purposes—though we limit consideration just to sentential forms. We have seen cases structured by the recursive definitions TR and FR. Here is one that uses ST. Say a formula is in *normal* form iff its only operators are $\vee$, $\wedge$, and $\sim$, and the only instances of $\sim$ are immediately prefixed to atomics (of course, any normal form is an abbreviation of a formula whose only operators are $\rightarrow$ and $\sim$). Where $\mathcal{P}$ is a normal form, let $\mathcal{P}'$ be like $\mathcal{P}$ except that $\vee$ and $\wedge$ are interchanged and, for a sentence letter $\mathcal{S}$, $\mathcal{S}$ and $\sim\mathcal{S}$ are interchanged. Thus, for example, if $\mathcal{P}$ is an atomic A, then $\mathcal{P}'$ is $\sim A$, if $\mathcal{P}$ is $(A \vee (\sim B \wedge C))$, then $\mathcal{P}'$ is $(\sim A \wedge (B \vee \sim C))$. We show that if $\mathcal{P}$ is in normal form, then $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. Thus, for the case we have just seen,

$$\mathsf{I}[\sim(A \vee (\sim B \wedge C))] = \mathsf{T} \quad \text{iff} \quad \mathsf{I}[(\sim A \wedge (B \vee \sim C))] = \mathsf{T}$$

So the result works like a generalized semantic version of DeM in combination with DN: When you push a negation into a normal form, $\wedge$ flips to $\vee$, $\vee$ flips to $\wedge$, and atomics switch between $\mathcal{S}$ and $\sim\mathcal{S}$. Our argument is by induction on the number of operators in a formula $\mathcal{P}$.

(H) *Basis*: Suppose $\mathcal{P}$ has no operators and is in normal form. Then $\mathcal{P}$ is an atomic $\mathcal{S}$; so $\sim\mathcal{P} = \sim\mathcal{S}$ and $\mathcal{P}' = \sim\mathcal{S}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{S}] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. So if $\mathcal{P}$ has no operators then if it is in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols then if it is in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Show: If $\mathcal{P}$ has k operator symbols then if it is in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Suppose $\mathcal{P}$ is in normal form and has k operator symbols. Then $\mathcal{P}$ is $\sim\mathcal{S}$, $\mathcal{A} \vee \mathcal{B}$, or $\mathcal{A} \wedge \mathcal{B}$ where $\mathcal{S}$ is atomic and $\mathcal{A}$ and $\mathcal{B}$ are normal forms with less than k operator symbols. So there are three cases.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{S}$. Then $\sim\mathcal{P}$ is $\sim\sim\mathcal{S}$, and $\mathcal{P}'$ is $\mathcal{S}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\sim\mathcal{S}] = \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim\mathcal{S}] \neq \mathsf{T}$; by ST($\sim$) again iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

($\vee$) $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$. Then $\sim\mathcal{P}$ is $\sim(\mathcal{A} \vee \mathcal{B})$, and $\mathcal{P}'$ is $\mathcal{A}' \wedge \mathcal{B}'$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{A} \vee \mathcal{B})] = \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\mathcal{A} \vee \mathcal{B}] \neq \mathsf{T}$; by ST$'$($\vee$) iff $\mathsf{I}[\mathcal{A}] \neq \mathsf{T}$ and $\mathsf{I}[\mathcal{B}] \neq \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}'] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}'] = \mathsf{T}$; by ST$'$($\wedge$) iff $\mathsf{I}[\mathcal{A}' \wedge \mathcal{B}'] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

($\wedge$) Homework.

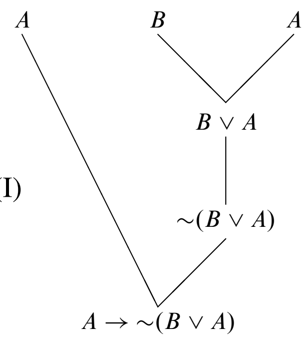
If $\mathcal{P}$ has k operator symbols then if it is in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Indct: For any $\mathcal{P}$, if it is in normal form then $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

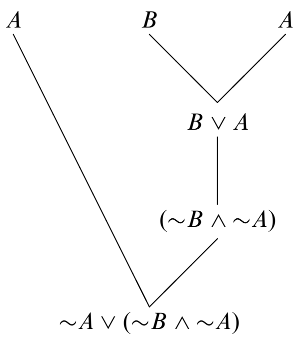
Since the thesis to be proved is a conditional, we obtain that conditional for the basis and show. Similarly, the *assumption* is a conditional that applies to formulas with less than k operator symbols *that are in normal form*. Thus, for application of the assumption at the show step, it is important not only that $\mathcal{A}$ and $\mathcal{B}$ have less than k operator symbols, but that they are in normal form. If they were not, then the inductive assumption would not apply to them. The overall pattern of the show step is as usual: In the cases, we break down to parts to which the assumption applies, apply the assumption, and put the resultant parts back together. In the second case, we assert that if $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$, then $\mathcal{P}'$ is $\mathcal{A}' \wedge \mathcal{B}'$. Here $\mathcal{A}$ and $\mathcal{B}$ may be complex. We do the conversion on $\mathcal{P}$ iff we do the conversion on its main operator, and then do the conversion on its parts. And similarly for ($\wedge$). It is this which enables us to feed into the inductive assumption. Notice that it is convenient to cast reasoning in the "collapsed" biconditional style.

Where $\mathcal{P}$ is *any* form whose operators are $\sim$, $\vee$, $\wedge$, or $\rightarrow$, we now show that $\mathcal{P}$ is equivalent to a normal form. Consider a transform $\mathcal{P}_{\mathrm{N}}$ defined as follows: For atomic $\mathcal{S}$, $\mathcal{S}_{\mathrm{N}} = \mathcal{S}$; for arbitrary formulas $\mathcal{A}$ and $\mathcal{B}$ with just those operators, $(\mathcal{A} \vee \mathcal{B})_{\mathrm{N}} = (\mathcal{A}_{\mathrm{N}} \vee \mathcal{B}_{\mathrm{N}})$, $(\mathcal{A} \wedge \mathcal{B})_{\mathrm{N}} = (\mathcal{A}_{\mathrm{N}} \wedge \mathcal{B}_{\mathrm{N}})$, and with prime defined as above, $(\mathcal{A} \rightarrow \mathcal{B})_{\mathrm{N}} = ([\mathcal{A}_{\mathrm{N}}]' \vee \mathcal{B}_{\mathrm{N}})$, and $[\sim\mathcal{A}]_{\mathrm{N}} = [\mathcal{A}_{\mathrm{N}}]'$. To see how this works, consider how you would construct $\mathcal{P}_{\mathrm{N}}$ on a tree.

(I)



For any $\mathcal{P}$ on the left, the corresponding $\mathcal{P}_{\mathrm{N}}$ appears on the right.



These trees work very much like unabbreviating trees from section 2.2.3. For each $\mathcal{P}$ on the left, $\mathcal{P}_{\mathrm{N}}$ is on the right. The conversion of a complex formula depends on the conversion of its parts. So starting with the parts, we construct the transform of the whole, one component at a time. Thus for example $(B \vee A)_{\mathrm{N}}$ is just $B \vee A$; then $[\sim(B \vee A)]_{\mathrm{N}} = [(B \vee A)_{\mathrm{N}}]' = [B \vee A]' = \sim B \wedge \sim A$. Observe that at each stage of the right-hand tree, the result is a normal form.

We show by mathematical induction on the number of operators in $\mathcal{P}$ that $\mathcal{P}_{\mathrm{N}}$ must be a normal form and that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$. For the argument it will be important, not only to use the inductive assumption, but also the result from above that for any $\mathcal{P}$ in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. In order to apply this result, it will be crucial that every $\mathcal{P}_{\mathrm{N}}$ is in normal form. Suppose the operators of $\mathcal{P}$ (and so its subformulas) are just $\sim$, $\vee$, $\wedge$, and $\to$. Here is an outline of the argument, with parts left as homework:

T8.1. For any $\mathcal{P}$ whose operators are $\sim$, $\wedge$, $\vee$, and $\to$, $\mathcal{P}_{\mathrm{N}}$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

Basis: If $\mathcal{P}$ is an atomic $\mathcal{S}$, then $\mathcal{P}_{\mathrm{N}} = \mathcal{S}$. An atomic $\mathcal{S}$ is in normal form; so $\mathcal{P}_{\mathrm{N}} = \mathcal{S}$ is in normal form. And $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

Assp: For any i, $0 \le i < k$ if $\mathcal{P}$ has i operator symbols, then $\mathcal{P}_{\mathrm{N}}$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

Show: If $\mathcal{P}$ has k operator symbols, then $\mathcal{P}_{\mathrm{N}}$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

If $\mathcal{P}$ has k operator symbols, then $\mathcal{P}$ is of the form $\sim\mathcal{A}$, $\mathcal{A} \wedge \mathcal{B}$, $\mathcal{A} \vee \mathcal{B}$, or $\mathcal{A} \to \mathcal{B}$ for formulas $\mathcal{A}$ and $\mathcal{B}$ with less than k operator symbols.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$. Then $\mathcal{P}_{\mathrm{N}} = (\mathcal{A}_{\mathrm{N}})'$. By assumption $\mathcal{A}_{\mathrm{N}}$ is in normal form; so since the prime operation converts a normal form to another normal form, $(\mathcal{A}_{\mathrm{N}})'$ is in normal form; so $\mathcal{P}_{\mathrm{N}}$ is in normal form. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{A}] \neq \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}_{\mathrm{N}}] \neq \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim(\mathcal{A}_{\mathrm{N}})] = \mathsf{T}$; since $\mathcal{A}_{\mathrm{N}}$ is in normal form, by our previous result (H), iff $\mathsf{I}[(\mathcal{A}_{\mathrm{N}})'] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$. So $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

($\wedge$) Homework.

($\vee$) Homework.

($\rightarrow$) Homework.

In any case, if $\mathscr{P}$ has k operator symbols, $\mathscr{P}_{\text{N}}$ is in normal form and $\mathsf{I}[\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}_{\text{N}}] = \mathsf{T}$.

Indct: For any $\mathscr{P}$, $\mathscr{P}_{\text{N}}$ is in normal form and $\mathsf{I}[\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}_{\text{N}}] = \mathsf{T}$.

The inductive assumption applies just to formulas with $< k$ operator symbols. So it applies just to formulas on the order of $\mathscr{A}$ and $\mathscr{B}$. The result from before applies to any formulas in normal form. So it applies to $\mathscr{A}_{\text{N}}$ once we have determined that $\mathscr{A}_{\text{N}}$ is in normal form.

E8.7. Complete induction (H) to show that every $\mathscr{P}$ in normal form is such that $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}'] = \mathsf{T}$. You should set up the whole induction with statements for the basis, assumption, and show parts. But then you may appeal to the text for parts already done, as the text appeals to homework. Hint: If $\mathscr{P} = (\mathscr{A} \wedge \mathscr{B})$ then $\mathscr{P}' = (\mathscr{A}' \vee \mathscr{B}')$.

E8.8. Complete the demonstration of T8.1 to show that any $\mathscr{P}$ with just operators $\sim$, $\wedge$, $\vee$, and $\rightarrow$ has a $\mathscr{P}_{\text{N}}$ in normal form such that $\mathsf{I}[\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}_{\text{N}}] = \mathsf{T}$. Again, you should set up the whole induction with statements for the basis, assumption, and show parts. But then you may appeal to the text for parts already done, as the text appeals to homework.

*E8.9. Show that for any $\mathscr{P}$ in normal form, $\vdash_{ND+} \sim\mathscr{P} \leftrightarrow \mathscr{P}'$. Hint: The reasoning is parallel to the semantic case, but now about what you can *derive*.

E8.10. Use the result from the previous problem to show that for any $\mathscr{P}$ whose operators are $\sim$, $\vee$, $\wedge$, and $\rightarrow$, $\mathscr{P}_{\text{N}}$ is in normal form and $\vdash_{ND+} \mathscr{P} \leftrightarrow \mathscr{P}_{\text{N}}$. Hint: Again the reasoning is parallel to the semantic case, but now about what you can derive.

8.2.5 Case

Here is a result like one we will seek later for the quantificational case. It depends on the (recursive) notion of a *derivation*. Because of their relative simplicity, we will focus on axiomatic derivations. If we were working with "derivations" of the sort described in the diagram on page 67, then we could reason by induction on the row in which a formula appears. Formulas in the top row result directly as premises or axioms, those in the next row from ones before with MP; and so forth. But our official notion of an axiomatic derivation is not this; in an official axiomatic derivation, lines are ordered, where each line is either an axiom, a premise, or follows from previous lines by a rule. But this is sufficient for us to reason about one line of an axiomatic derivation based on ones that come before; that is, we reason by induction on the *line number* of a derivation. Recall that $\vdash_{ADs} \mathcal{P}$ just in case there is a derivation of $\mathcal{P}$ in the sentential fragment of *AD* with just A1, A2, A3, and MP. We show that if $\mathcal{P}$ is a theorem of *ADs*, then $\mathcal{P}$ is a tautology: if $\vdash_{ADs} \mathcal{P}$ then $\vDash_s \mathcal{P}$. Thus we establish the (*weak*) *soundness* of *ADs*.

Suppose $\vdash_{ADs} \mathcal{P}$; then there is an *ADs* derivation $\langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \rangle$ of $\mathcal{P}$ from no premises, with $\mathcal{A}_n = \mathcal{P}$. By induction on the line numbers of this derivation, we show that for any j, $\vDash_s \mathcal{A}_j$. The case when $j = n$ is the desired result.

(J) *Basis*: Since $\langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \rangle$ is a derivation from no premises, $\mathcal{A}_1$ can only be an instance of A1, A2, or A3.

(A1) Say $\mathcal{A}_1$ is an instance of A1 and so of the form $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$. Suppose $\nvDash_s \mathcal{A}_1$; then $\nvDash_s \mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$; so by SV, there is an I such that $\mathsf{I}[\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{Q} \to \mathcal{P}] \neq \mathsf{T}$; from the latter, by ST($\to$), $\mathsf{J}[\mathcal{Q}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s \mathcal{A}_1$.

(A2) Similarly.

(A3) Similarly.

Assp: For any i, $1 \leq i < k$, $\vDash_s \mathcal{A}_i$.

Show: $\vDash_s \mathcal{A}_k$.

$\mathcal{A}_k$ is either an axiom or arises from previous lines by MP. If $\mathcal{A}_k$ is an axiom then, as in the basis, $\vDash_s A_k$. So suppose $\mathcal{A}_k$ arises from previous lines by MP. In this case, the picture is something like this:

$a.$	$\mathcal{B} \to \mathcal{C}$	
$b.$	$\mathcal{B}$	
$k.$	$\mathcal{C}$	a,b MP

where $a, b < k$ and $\mathcal{C}$ is $\mathcal{A}_k$. Suppose $\nvDash_s \mathcal{A}_k$; then $\nvDash_s \mathcal{C}$ and by SV there is some I such that $\mathsf{I}[\mathcal{C}] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\mathcal{C}] \neq \mathsf{T}$. But by assumption, $\vDash_s \mathcal{B}$ and $\vDash_s \mathcal{B} \to \mathcal{C}$; so by SV, for any I, $\mathsf{I}[\mathcal{B}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B} \to \mathcal{C}] = \mathsf{T}$; so $\mathsf{J}[\mathcal{B}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{B} \to \mathcal{C}] = \mathsf{T}$; from the latter, by ST($\to$), $\mathsf{J}[\mathcal{B}] \neq \mathsf{T}$ or $\mathsf{J}[\mathcal{C}] = \mathsf{T}$; so $\mathsf{J}[\mathcal{C}] = \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s \mathcal{A}_k$.

Indct: For any line j of the derivation $\vDash_s \mathcal{A}_j$.

We might have continued as above for (A2) and (A3). Alternatively, since we have already done the work, we might have appealed directly to T7.2s, T7.3s, and T7.4s for (A1), (A2), and (A3) respectively. From the case when $\mathcal{A}_j = \mathcal{P}$ we have $\vDash_s \mathcal{P}$. This result is a precursor to one we will obtain in Chapter 10. There, we will show *strong soundness* for the complete system *AD*, if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. This tells us that our derivation system can never lead us astray. There is no situation where a derivation moves from premises that are true to a conclusion that is not. Still, what we have is interesting in its own right: It is a first connection between the syntactic notions associated with derivations, and the semantic notions of validity and truth (and it reflects informal reasoning sketched on page 201).

E8.11. Consider the system A^* for exercise E3.5 and take MP in its primitive form. Show by mathematical induction that A^* is weakly sound. That is, show that if $\vdash_{A^*} \mathcal{P}$ then $\vDash_s \mathcal{P}$.

E8.12. Modify your argument for E8.11 to show that A^* is strongly sound. That is, modify the argument to show that if $\Gamma \vdash_{A^*} \mathcal{P}$ then $\Gamma \vDash_s \mathcal{P}$. You may appeal to reasoning from the previous problem where it is applicable. Hint: When premises are allowed, $\mathcal{A}_j$ is either an axiom, a premise, or arises by a rule. So there is one additional case in the basis; but that case is trivial—if all of the premises are true, and $\mathcal{A}_j$ is a premise, then $\mathcal{A}_j$ cannot be false. And your reasoning for the show will be modified; now the assumption gives you $\Gamma \vDash_s \sim(\mathcal{B} \wedge \sim\mathcal{C})$ and $\Gamma \vDash_s \mathcal{B}$ and your goal is to show $\Gamma \vDash_s \mathcal{C}$.

E8.13. Modify table T($\sim$) to a T$'$($\sim$) that has $\mathsf{I}[\sim\mathcal{P}] = \mathsf{F}$ both when $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; let table T($\rightarrow$) and so ST($\rightarrow$) remain as before. Say a formula is *select* iff it is true on every interpretation given the revised tables. Show by mathematical induction that every consequence of MP with A1 and A2 alone is select. Then by a table show that the A3 instance $(\sim B \rightarrow \sim A) \rightarrow [(\sim B \rightarrow A) \rightarrow B]$ is not select. It follows that there is no derivation of that formula from A1 and A2 alone (this is an *independence* result of the sort discussed in section 11.3). Hint: Your induction may be a simple modification of argument (J) from above.

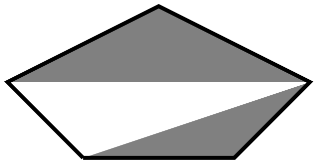
E8.14. Let $\mathsf{I}[\mathcal{S}] = \mathsf{T}$ for every sentence letter $\mathcal{S}$. Where $\mathcal{P}$ is any sentential formula whose only operators are $\rightarrow$, $\wedge$, $\vee$, and $\leftrightarrow$, show by induction on the number of operators in $\mathcal{P}$ that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. Use this result to show that $\nvDash_s \sim\mathcal{P}$.

E8.15. Where t is a term of $\mathcal{L}_q$, let $X(t)$ be the sum of all the superscripts in t and $Y(t)$ be the number of symbols in t. So, for example, if t is z, then $X(t) = 0$ and $Y(t) = 1$; if t is $g^1 f^2 cx$, then $X(t) = 3$ and $Y(t) = 4$. By induction on the number of function symbols in t, show that for any t in $\mathcal{L}_q$, $X(t) + 1 = Y(t)$.

E8.16. For $n \geq 1$, let $S(n) = 1/2 + 1/4 + \cdots + 1/2^n$. Show by mathematical induction that $S(n) = 1 - 1/2^n$, and so that $S(n)$ approaches 1 as n approaches infinity.

E8.17. Show by mathematical induction that for any integer $n \geq 0$, 3^n is odd—that is that for any $n \geq 0$, there is some a such that $3^n = 2a - 1$.

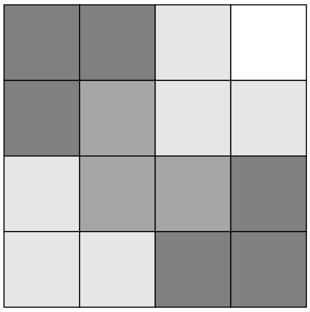
E8.18. Show by mathematical induction that for any $n \geq 3$, an n-sided convex polygon P may be decomposed into $n - 2$ triangles (where a triangle is "decomposed" into itself). So, for example, a five-sided figure decomposes into three triangles.



E8.19. If a Hershey bar has n squares, show by mathematical induction that it takes $n - 1$ breaks (along the lines) to divide it into its individual squares.

E8.20. Show by mathematical induction that at a recent convention the number of logicians who shook hands an odd number of times is even. Assume that 0 is even. Hints: Reason by induction on the number of handshakes at the convention. At any stage n, let $O(n)$ be the number of people who have shaken hands an odd number of times. Your task is to show that for any n, $O(n)$ is even. You will want to consider cases for what happens to $O(n)$ when (i) someone who has already shaken hands an odd number of times shakes with someone who has shaken an odd number of times; (ii) someone who has already shaken hands an even number of times shakes with someone who has shaken an even number of times; and (iii) someone who has already shaken hands an odd number of times shakes with someone who has shaken an even number of times.

E8.21. For any $n \geq 1$, given a $2^n \times 2^n$ checkerboard with any one square deleted, show by mathematical induction that it is possible to cover the board with 3-square L-shaped pieces. For example, a 4×4 board with a corner deleted could be covered as follows:



Hint: The basis is easy—a 2×2 board with one square missing is covered by a single L-shaped piece. The trick is to see how an arbitrary 2^k board with one square missing can be constructed out of an L-shaped piece and 2^{k-1} size boards with a square missing.

E8.22. Say $\mathcal{P}$ is in *disjunctive normal form* iff it is of the sort,

$$(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_a) \vee (\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_b) \vee \ldots \vee (\mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_c)$$

where each $\mathcal{A}_i, \mathcal{B}_i, \ldots, \mathcal{C}_i$ is an atomic or negated atomic (omitting inner parentheses for the extended conjunctions and disjunction). Allow that a disjunctive normal form may reduce to a single disjunct and its conjunctions to a single conjunct—so an atomic or negated atomic is already a disjunctive normal form. Show that for any normal form $\mathcal{P}_{\mathrm{N}}$ there is a disjunctive normal form $\mathcal{P}_{\mathrm{D}}$ such that $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{D}}] = \mathsf{T}$. Hint: This is straightforward except for the case where $\mathcal{P}_{\mathrm{N}}$ is $\mathcal{B}_{\mathrm{N}} \wedge \mathcal{C}_{\mathrm{N}}$. In this case, by assumption there are disjunctive normal forms $\mathcal{B}_{\mathrm{D}}$ and $\mathcal{C}_{\mathrm{D}}$ such that $\mathsf{I}[\mathcal{B}_{\mathrm{N}}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{B}_{\mathrm{D}}] = \mathsf{T}$, and $\mathsf{I}[\mathcal{C}_{\mathrm{N}}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{C}_{\mathrm{D}}] = \mathsf{T}$. Since they are in disjunctive normal form, $\mathcal{B}_{\mathrm{D}}$ and $\mathcal{C}_{\mathrm{D}}$ are of the sort $\mathcal{B}_1 \vee \mathcal{B}_2 \vee \ldots \vee \mathcal{B}_b$ and $\mathcal{C}_1 \vee \mathcal{C}_2 \vee \ldots \vee \mathcal{C}_c$ for some $\mathcal{B}_1 \ldots \mathcal{B}_b$ and $\mathcal{C}_1 \ldots \mathcal{C}_c$ that are conjunctions of atomics and negated atomics. Consider a grid as follows:

	$\mathcal{B}_1$	$\mathcal{B}_2$	$\ldots$	$\mathcal{B}_b$
$\mathcal{C}_1$	$\mathcal{B}_1 \wedge \mathcal{C}_1$	$\mathcal{B}_2 \wedge \mathcal{C}_1$		$\mathcal{B}_b \wedge \mathcal{C}_1$
$\mathcal{C}_2$	$\mathcal{B}_1 \wedge \mathcal{C}_2$	$\mathcal{B}_2 \wedge \mathcal{C}_2$		$\mathcal{B}_b \wedge \mathcal{C}_2$
$\vdots$				
$\mathcal{C}_c$	$\mathcal{B}_1 \wedge \mathcal{C}_c$	$\mathcal{B}_2 \wedge \mathcal{C}_c$		$\mathcal{B}_b \wedge \mathcal{C}_c$

And let $\mathcal{P}_{\mathrm{D}}$ be the disjunction of conjuncts from the grid. (Effectively, this is like repeatedly applying Dist to $\mathcal{B}_{\mathrm{D}} \wedge \mathcal{C}_{\mathrm{D}}$.) Now you should be able to show that $\mathcal{P}_{\mathrm{D}}$ is a disjunctive normal form, and that $\mathcal{P}_{\mathrm{N}}$ is equivalent to $\mathcal{P}_{\mathrm{D}}$.

E8.23. Limit attention to sentential forms whose only operators are $\sim$ and $\leftrightarrow$. Show that under any (sub)formula on a truth table with at least four rows is an even number of Ts and Fs. Hints: Reason by induction on the number of operators in $\mathcal{P}$ where $\mathcal{P}$ is a (sub)formula on a table with at least four rows. Then by construction of the table, it is immediate that atomics have an even number of Ts and Fs. The show step has cases for $\sim$ and $\leftrightarrow$. The former is easy, the latter is not. Here is a trick that may help (which I learned from a student): Let each T be assigned an even number and each F an odd; assign $\mathcal{A} \leftrightarrow \mathcal{B}$ the sum of the numbers assigned to $\mathcal{A}$ and $\mathcal{B}$; then consider the sum of the numbers in *columns* of your table.

E8.24. After a few days studying mathematical logic, Zeno hits upon what he thinks is conclusive proof that all is one. He argues by mathematical induction that all the members of any n-tuple are identical:

Basis: If A is a 1-tuple, then it is of the sort $\langle \mathsf{o} \rangle$, and every member of $\langle \mathsf{o} \rangle$ is identical. So every member of A is identical.

Assp: For any i, $1 \le i < k$, all the members of any i-tuple are identical.

Show: All the members of any k-tuple are identical.

If A is a k-tuple, then it is of the form $\langle \mathsf{o}_1, \ldots, \mathsf{o}_{k-1}, \mathsf{o}_k \rangle$. But both $\langle \mathsf{o}_1, \ldots, \mathsf{o}_{k-2}, \mathsf{o}_{k-1} \rangle$ and $\langle \mathsf{o}_1, \ldots, \mathsf{o}_{k-2}, \mathsf{o}_k \rangle$ are $k-1$ tuples; so by the inductive assumption, all their members are identical; but these have o_1 in common and together include all the members of A; so all the members of A are identical to o_1 and so to one another.

Indct: All the members of any n-tuple A are identical.

Given this, he considers the n-tuple consisting of you and Mount Rushmore, and concludes that you are identical; similarly for you and Donald Trump, and so forth. What is the matter with Zeno's reasoning? Hint: Does the reasoning at the show stage apply to *arbitrary* k?

8.3 Further Examples (for Part III)

We continue our series of examples, moving now to quantificational cases, and to some theorems that will be useful especially if you go on to consider Part III.

8.3.1 Case

For variables x and v, where v does not appear in term t, it should be obvious that $[t^x_v]^v_x = t$. If we replace every instance of x with v, and then all the instances of v with x, we get back to where we started. The restriction that v not appear in t is required to prevent putting back instances of x where there were none in the original—as fxv^x_v is fvv, but then fvv^v_x is fxx. We demonstrate that when v does not appear in t, $[t^x_v]^v_x = t$ more rigorously by a simple induction on the number of function symbols in t.

(K) *Basis*: If t has no function symbols then it is a variable or a constant. Suppose v does not appear in t. If t is a variable or a constant other than x, then $t^x_v = t$ (nothing is replaced); and since v does not appear in t, $t^v_x = t$ (nothing is replaced); so $[t^x_v]^v_x = t^v_x = t$. If t is the variable x, then $t^x_v = v$; and $v^v_x = x$; so $[t^x_v]^v_x = v^v_x = x = t$. So if v does not appear in t then $[t^x_v]^v_x = t$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols and v does not appear in t, then $[t^x_v]^v_x = t$.

Show: If t has k function symbols and v does not appear in t, then $[t^x_v]^v_x = t$.

If t has k function symbols, then it is of the form, $h^n s_1 \dots s_n$ for some function symbol h^n and terms $s_1 \dots s_n$ each of which has $< k$ function symbols. Suppose v does not appear in t; then v does not appear in any of $s_1 \dots s_n$; so the inductive assumption applies to $s_1 \dots s_n$; so by assumption $[s_1{}^x_v]^v_x = s_1$ and … and $[s_n{}^x_v]^v_x = s_n$. But $[t^x_v]^v_x = [h^n s_1 \dots s_n{}^x_v]^v_x$; and since replacements only occur within the terms, this is $h^n[s_1{}^x_v]^v_x \dots [s_n{}^x_v]^v_x$; and by assumption this is $h^n s_1 \dots s_n = t$. So $[t^x_v]^v_x = t$.

Indct: For any term t, if v does not appear in t, $[t^x_v]^v_x = t$.

Consider a concrete application of reasoning for the show stage: Substitutions applied to f^2xb, say, do not affect the function symbol, but rather "distribute" onto the individual terms x and b; so we find $[f^2xb^x_v]^v_x$ if we combine the function symbol with $[x^x_v]^v_x$ and $[b^x_v]^v_x$; but $[x^x_v]^v_x = x$ and $[b^x_v]^v_x = b$; so $[f^2xb^x_v]^v_x$ is just f^2xb. It is also worthwhile to note the place where it matters that v is not a variable in t: In the basis case where t is a variable other than x, $t^v_x = t$ insofar as nothing is replaced; but suppose t is v; then $t^v_x = x \neq t$, and we do not achieve the desired result.

This result can be extended to one with application to formulas. If v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$. We require the restriction that v is not free in $\mathcal{P}$ for the same reason as before: If v were free in $\mathcal{P}$, we might end up with instances of x where there are none in the original—as Rxv^x_v is Rvv, but then Rvv^v_x is Rxx. And we need the restriction that v is free for x in $\mathcal{P}$ so that once we have $\mathcal{P}^x_v$, instances of x will go back for all the instances of v. So for example, $\forall vRxv^x_v$ is $\forall vRvv$, but then remains the same when x is substituted for free instances of v. Here is the basic structure of the argument, with parts left for homework:

*T8.2. For variables x and v, if v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \ldots t_n$. Suppose v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$. (i) If $\mathcal{P}$ is $\mathcal{S}$ then it has no variables; so $\mathcal{P}^x_v = \mathcal{P}$ and $\mathcal{P}^v_x = \mathcal{P}$. So $[\mathcal{P}^x_v]^v_x = \mathcal{P}^v_x = \mathcal{P}$. (ii) Say $\mathcal{P}$ is $\mathcal{R}^n t_1 \ldots t_n$. Since v is not free in $\mathcal{P}$, v does not appear at all in $\mathcal{P}$ or its terms; so by the previous result (K), $[t_1{}^x_v]^v_x = t_1$ and ... and $[t_n{}^x_v]^v_x = t_n$. So $[\mathcal{P}^x_v]^v_x = [\mathcal{R}^n t_1 \ldots t_n{}^x_v]^v_x = \mathcal{R}^n [t_1{}^x_v]^v_x \ldots [t_n{}^x_v]^v_x = \mathcal{R}^n t_1 \ldots t_n = \mathcal{P}$. So if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$ then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Assp: For any i, $0 \le i < k$, any $\mathcal{P}$ with i operator symbols is such that if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Show: Any $\mathcal{P}$ with k operator symbols is such that if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $(\mathcal{A} \to \mathcal{B})$, or $\forall w \mathcal{A}$ for some variable w and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. Suppose v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$. Then $[\mathcal{P}^x_v]^v_x = [(\sim\mathcal{A})^x_v]^v_x = \sim([\mathcal{A}^x_v]^v_x)$. Since v is not free in $\mathcal{P}$, v is not free in $\mathcal{A}$; and since v is free for x in $\mathcal{P}$, v is free for x in $\mathcal{A}$. So the assumption applies to $\mathcal{A}$ and... [homework].

($\to$) Homework.

($\forall$) $\mathcal{P}$ is $\forall w \mathcal{A}$. Either x is free in $\mathcal{P}$ or not. (i) If x is not free in $\mathcal{P}$, then $\mathcal{P}^x_v = \mathcal{P}$ and since v is not free in $\mathcal{P}$, $\mathcal{P}^v_x = \mathcal{P}$; so $[\mathcal{P}^x_v]^v_x = \mathcal{P}^v_x = \mathcal{P}$. (ii) Suppose x is free in $\mathcal{P} = \forall w \mathcal{A}$. Then x is other than w; and since v is free for x in $\mathcal{P}$, v is other than w; so the quantifier does not affect the replacements, and $[\mathcal{P}^x_v]^v_x$ is $\forall w([\mathcal{A}^x_v]^v_x)$. Since v is not free in $\mathcal{P}$ and is not w, v is not free in $\mathcal{A}$; and since v is free for x in $\mathcal{P}$, v is free for x in $\mathcal{A}$. So the inductive assumption applies to $\mathcal{A}$; so $[\mathcal{A}^x_v]^v_x = \mathcal{A}$; so $[\mathcal{P}^x_v]^v_x = \forall w([\mathcal{A}^x_v]^v_x) = \forall w \mathcal{A} = \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Indct: For any $\mathcal{P}$, if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

There are a few things to note about this argument. First, again, we have to be careful that the formulas $\mathcal{A}$ and $\mathcal{B}$ of which $\mathcal{P}$ is composed are in fact of the sort to which the inductive assumption applies. In this case, the requirement is not only that $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operator symbols, but that they satisfy the additional assumptions, that v is not free in $\mathcal{P}$ but is free for x in $\mathcal{P}$. It is easy to see that this condition obtains in the cases for $\sim$ and $\to$, but it is relatively complicated in the case for $\forall$, where there is interaction with another quantifier. Observe also that we cannot assume that the arbitrary quantifier has the same variable as x or v. In fact, it is because the variable may be different that we are able to reason the way we do. Finally, observe that the arguments of this section for (K) and T8.2 are a "linked pair" in the sense that the result of the first for terms is required for the basis of the next for formulas. This pattern repeats in the next cases, including the theorem immediately following.

*T8.3. Where constant c does not appear in formula $\mathcal{P}$, $[\mathcal{P}^x_c]^c_v = \mathcal{P}^x_v$.

*E8.25. Provide a complete argument for T8.2, completing cases for ($\sim$) and ($\to$). You should set up the complete induction, but may appeal to the text at parts that are already completed, just as the text appeals to homework.

*E8.26. Show T8.3. Hint: You will need arguments parallel to (K) and then T8.2.

8.3.2 Case

This example develops another pair of linked results which may seem obvious. Even so, the reasoning is instructive, and we will need the results for things to come. First,

T8.4. For any interpretation I, variable assignments d and h, and term t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

If variable assignments agree at least on assignments to the variables in t, then corresponding term assignments agree on the assignment to t. The reasoning, as one might expect, is by induction on the number of function symbols in t.

Basis: If t has no function symbols, then it is a variable x or a constant c. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t. (i) Say t is a constant c; then by TA(c), $\mathsf{I_d}[c] = \mathsf{I}[c]$ and $\mathsf{I}[c] = \mathsf{I_h}[c]$. So $\mathsf{I_d}[t] = \mathsf{I_d}[c] = \mathsf{I}[c] = \mathsf{I_h}[c] = \mathsf{I_h}[t]$. (ii) Say t is a variable x; then $\mathsf{d}[x] = \mathsf{h}[x]$; and by TA(v), $\mathsf{I_d}[x] = \mathsf{d}[x]$ and $\mathsf{h}[x] = \mathsf{I_h}[x]$. So $\mathsf{I_d}[t] = \mathsf{I_d}[x] = \mathsf{d}[x] = \mathsf{h}[x] = \mathsf{I_h}[x] = \mathsf{I_h}[t]$. So if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, and $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Show: If t has k function symbols, and $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

If t has k function symbols, then it is of the form $h^n s_1 \ldots s_n$ for some function symbol h^n and terms $s_1 \ldots s_n$ with $< k$ function symbols. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t; then $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in $s_1 \ldots s_n$; so the inductive assumption applies to $s_1 \ldots s_n$; so $\mathsf{I_d}[s_1] = \mathsf{I_h}[s_1]$ and ... and $\mathsf{I_d}[s_n] = \mathsf{I_h}[s_n]$. So with two applications of TA(f), $\mathsf{I_d}[t] = \mathsf{I_d}[h^n s_1 \dots s_n] = \mathsf{I}[h^n]\langle \mathsf{I_d}[s_1] \dots \mathsf{I_d}[s_n]\rangle = \mathsf{I}[h^n]\langle \mathsf{I_h}[s_1] \dots \mathsf{I_h}[s_n]\rangle = \mathsf{I_h}[h^n s_1 \dots s_n] = \mathsf{I_h}[t]$. So if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Indct: For any t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

It should be clear that we follow our usual pattern to complete the show step: The assumption gives us information about the parts—in this case, about assignments to $s_1 \dots s_n$; from this, with TA, we move to a conclusion about the whole term t. Notice again that it is important to show that the parts are of the right sort for the inductive assumption to apply: It matters that $s_1 \dots s_n$ have $< k$ function symbols, and that $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $s_1 \dots s_n$. Perhaps the overall result is intuitively obvious: If there is no difference in assignments to relevant variables then, by the way things build from the parts to the whole, there is no difference in assignments to the whole terms. Our demonstration merely makes explicit how this result follows from the definitions.

We now turn to a result that is very similar, except that it applies to formulas. In this case, T8.4 is essential for reasoning in the basis.

*T8.5. For any interpretation I, variable assignments d and h, and formula $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

The argument, as you should expect, is by induction on the number of operator symbols in the formula $\mathcal{P}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \dots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x free in $\mathcal{P}$. (i) Say $\mathcal{P}$ is a sentence letter $\mathcal{S}$; then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{S}] = \mathsf{S}$; by SF(s) iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; by SF(s) again iff $\mathsf{I_h}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. (ii) Say $\mathcal{P}$ is $\mathcal{R}^n t_1 \dots t_n$; then since every variable in $\mathcal{P}$ is free, we have $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $\mathcal{P}$; so $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $t_1 \dots t_n$; so by T8.4, $\mathsf{I_d}[t_1] = \mathsf{I_h}[t_1]$ and … and $\mathsf{I_d}[t_n] = \mathsf{I_h}[t_n]$. So $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; by SF(r) iff $\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; iff $\langle \mathsf{I_h}[t_1] \dots \mathsf{I_h}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; by SF(r) iff $\mathsf{I_h}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. So if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x free in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, if $\mathcal{P}$ has i operator symbols and $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Show: If $\mathcal{P}$ has k operator symbols and $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, or $\forall v \mathcal{A}$ for variable v and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then since $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, and every variable free in $\mathcal{A}$ is free in $\mathcal{P}$, $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable in $\mathcal{A}$; so the inductive assumption applies to $\mathcal{A}$. $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$) iff $\mathsf{I_d}[\mathcal{A}] \neq \mathsf{S}$; by assumption iff $\mathsf{I_h}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_h}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

($\rightarrow$) Homework.

($\forall$) Suppose $\mathcal{P}$ is $\forall v\mathcal{A}$. Then since $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable in $\mathcal{A}$ with the possible exception of v; so for arbitrary $\mathsf{o} \in \mathsf{U}$, $\mathsf{d}(v|\mathsf{o})[x] = \mathsf{h}(v|\mathsf{o})[x]$ for every free variable x in $\mathcal{A}$. Since the assumption applies to arbitrary assignments, it applies to $\mathsf{d}(v|\mathsf{o})$ and $\mathsf{h}(v|\mathsf{o})$; so for any $\mathsf{o} \in \mathsf{U}$, by assumption, $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{h}(v|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$.

Now suppose $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ but $\mathsf{I_h}[\mathcal{P}] \neq \mathsf{S}$; then $\mathsf{I_d}[\forall v\mathcal{A}] = \mathsf{S}$ but $\mathsf{I_h}[\forall v\mathcal{A}] \neq \mathsf{S}$; from the latter, by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{I}_{\mathsf{h}(v|\mathsf{o})}[\mathcal{A}] \neq \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{I}_{\mathsf{h}(v|\mathsf{m})}[\mathcal{A}] \neq \mathsf{S}$; so, with the inductive assumption as above, $\mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathcal{A}] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{I_d}[\forall v\mathcal{A}] \neq \mathsf{S}$. This is impossible; reject the assumption: if $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$, then $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. And similarly [by homework] in the other direction.

If $\mathcal{P}$ has k operator symbols and $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Indct: For any $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$ then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Notice again that it is important to make sure the inductive assumption applies. First, in the ($\forall$) case, we are careful to distinguish the arbitrary variable of quantification v from x of the assumption. Then, for the quantifier case, the condition that d and h agree on assignments to all the free variables in $\mathcal{A}$ is *not* satisfied merely because they agree on assignments to all the free variables in $\mathcal{P}$. We solve the problem by switching to assignments $\mathsf{d}(v|\mathsf{o})$ and $\mathsf{h}(v|\mathsf{o})$, which must agree on all the free variables in $\mathcal{A}$. (Why?) Reasoning in the quantifier case is more involved than we have seen so far. But you should be in a position to bear down and follow each step.

From T8.5 it is a short step to three quick corollaries that will be useful for things to come. First a result that should remind you of A5 from *AD*,

*T8.6. $\vDash \forall x(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P} \rightarrow \forall x\mathcal{Q})$ where x is not free in $\mathcal{P}$.

Homework.

Second, a result the proof of which was promised in Chapter 4 (page 122). If a *sentence* $\mathcal{P}$ is satisfied on any variable assignment, then it is satisfied on every variable assignment, and so true.

T8.7. For any interpretation I and sentence $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff there is some assignment d such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

Consider some sentence $\mathcal{P}$ and interpretation I. (i) Suppose $\mathsf{I}[\mathcal{P}] = \mathsf{T}$; then by TI, $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ for any d; so there is an assignment d such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose there is some assignment d such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$, but $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$. From the latter, by TI, there is some assignment h such that $\mathsf{I_h}[\mathcal{P}] \neq \mathsf{S}$; but if $\mathcal{P}$ is a sentence, it has no free variables; so (vacuously) every assignment agrees with h in its assignment to free variables in $\mathcal{P}$; in particular d agrees with h in its assignment to every free variable in $\mathcal{P}$; so by T8.5, $\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S}$. This is impossible; reject the assumption: if $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ then $\mathsf{I}[\mathcal{P}] = \mathsf{T}$.

In effect, the reasoning is as sketched in Chapter 4. Whether $\forall x\mathcal{P}$ is satisfied by d does not depend on the particular object d assigns to x—for the quantifier "overrides" the assignment from d. The key is contained in reasoning for the ($\forall$) case of T8.5, which "exempts" a quantified variable from ones on which assignments must agree. Given this, the move to T8.7 is straightforward.

Finally, as we have emphasized, the ($\sim$) and ($\rightarrow$) clauses of definition SF apply to satisfaction, not truth. Even so, for *sentences* of a quantificational language we recover simple truth conditions as from ST. Reasoning appeals most naturally to T8.7, though we may think of this as another corollary to T8.5.

*T8.8. For any interpretation I and sentences $\mathcal{P}$ and $\mathcal{Q}$,

(i) $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$

(ii) $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$.

Homework.

As a quick consequence of this last theorem, we obtain corresponding results for $\wedge$, $\vee$, and $\leftrightarrow$. Thus, for the sentential operators, sentences of a quantificational language obey the same truth conditions as ones from sentential languages.

*E8.27. Provide a complete argument for T8.5, completing the case for ($\rightarrow$), and expanding the other direction for ($\forall$). You should set up the complete induction, but may appeal to the text at parts that are already completed, as the text appeals to homework.

*E8.28. Show T8.6 and both parts of T8.8.

E8.29. Show that for any interpretation I and *sentence* $\mathcal{P}$, either $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$. Hint: This is not an argument by induction, but rather another quick corollary to T8.5; you can begin by supposing the result is false and show that the assumption is impossible.

8.3.3 Case

Here is another pair of results, with reasoning like we have already seen.

*T8.9. For any formula $\mathcal{P}$, term t, constant c, and distinct variables v and x, $[\mathcal{P}^v_t]^c_x$ is the same formula as $[\mathcal{P}^c_x]^v_{t^c_x}$.

Notice that $[\mathcal{P}^v_t]^c_x$ might be different from $[\mathcal{P}^c_x]^v_t$—for if t contains an instance of c, that instance of c is replaced in the first case, but not in the second. The proof breaks into two parts. (i) By induction on the number of function symbols in an arbitrary term r, we show that $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$. Given this, (ii) by induction on the number of operator symbols in an arbitrary formula $\mathcal{P}$, we show that $[\mathcal{P}^v_t]^c_x = [\mathcal{P}^c_x]^v_{t^c_x}$. Only part (i) is completed here; (ii) is left for homework. Suppose $v \neq x$.

Basis: If r has no function symbols, then it is either v, c, or some other constant or variable.

(v) Suppose r is v. Then r^v_t is t; so $[r^v_t]^c_x = t^c_x$. But r^c_x is v; so $[r^c_x]^v_{t^c_x} = v^v_{t^c_x} = t^c_x$. So $[r^v_t]^c_x = t^c_x = [r^c_x]^v_{t^c_x}$.

(c) Suppose r is c. Then r^v_t is c and $[r^v_t]^c_x$ is x. But r^c_x is x; and, since $v \neq x$, $[r^c_x]^v_{t^c_x}$ is x. So $[r^v_t]^c_x = x = [r^c_x]^v_{t^c_x}$.

(oth) Suppose r is some variable or constant other than v or c. Then $[r^v_t]^c_x = r^c_x = r$. Similarly, $[r^c_x]^v_{t^c_x} = r^v_{t^c_x} = r$. So $[r^v_t]^c_x = r = [r^c_x]^v_{t^c_x}$.

Assp: For any i, $0 \leq i < k$, if r has i function symbols, then $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

Show: If r has k function symbols, then $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

If r has k function symbols, then it is of the form, $h^n s_1 \dots s_n$ for some function symbol h^n and terms $s_1 \dots s_n$ each of which has $< k$ function symbols; so by assumption, $[{s_1}^v_t]^c_x = [{s_1}^c_x]^v_{t^c_x}$ and … and $[{s_n}^v_t]^c_x = [{s_n}^c_x]^v_{t^c_x}$. So $[r^v_t]^c_x = [h^n s_1 \dots {s_n}^v_t]^c_x = h^n[{s_1}^v_t]^c_x \dots [{s_n}^v_t]^c_x = h^n[{s_1}^c_x]^v_{t^c_x} \dots [{s_n}^c_x]^v_{t^c_x} = [h^n s_1 \dots {s_n}^c_x]^v_{t^c_x} = [r^c_x]^v_{t^c_x}$; so $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

Indct: For any r, $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

You will find this result useful when you turn to the final proof of T8.9. That argument is a straightforward induction on the number of operator symbols in $\mathcal{P}$. For the case where $\mathcal{P}$ is of the form $\forall w \mathcal{A}$, notice that v is either w or it is not. On the one hand, if v is w, then $\mathcal{P} = \forall w \mathcal{A}$ has no free instances of v so that $\mathcal{P}^v_t = \mathcal{P}$, and $[\mathcal{P}^v_t]^c_x = \mathcal{P}^c_x$; but, similarly, $\mathcal{P}^c_x$ has no free instances of v, so $[\mathcal{P}^c_x]^v_{t^c_x} = \mathcal{P}^c_x$. On the other hand, if v is a variable other than w, then $[\mathcal{P}^v_t]^c_x = \forall w([\mathcal{A}^v_t]^c_x)$ and $[\mathcal{P}^c_x]^v_{t^c_x} = \forall w([\mathcal{A}^c_x]^v_{t^c_x})$ and you will be able to use the inductive assumption.

*E8.30. Complete the proof of T8.9 by showing by induction on the number of operator symbols in an arbitrary formula $\mathcal{P}$ that if v is distinct from x, then $[\mathcal{P}^v_t]^c_x = [\mathcal{P}^c_x]^v_{t^c_x}$.

8.3.4 Case

We conclude this section with a result that depends on the one just before. Where $\Delta = \langle \mathcal{D}_1 \ldots \mathcal{D}_n \rangle$ is an *AD* derivation, and $\Phi = \{\mathcal{F}_1, \mathcal{F}_2 \ldots\}$ is a set of formulas, for some constant a and variable x, say $\Delta^a_x = \langle \mathcal{D}_1{}^a_x \ldots \mathcal{D}_n{}^a_x \rangle$ and $\Phi^a_x = \{\mathcal{F}_1{}^a_x, \mathcal{F}_2{}^a_x \ldots\}$. By induction on the line numbers in Δ, we show,

*T8.10. If Δ is an *AD* derivation from Φ, and x is a variable that does not appear in Δ, then for any constant a, Δ^a_x is an *AD* derivation from Φ^a_x.

Suppose $\Delta = \langle \mathcal{D}_1 \ldots \mathcal{D}_n \rangle$ is an *AD* derivation from Φ, a a constant, and x a variable that does not appear in Δ.

Basis: $\mathcal{D}_1$ is either a member of Φ or an axiom.

(prem) If $\mathcal{D}_1$ is a member of Φ, then $\mathcal{D}_1{}^a_x$ is a member of Φ^a_x; so $\langle \mathcal{D}_1{}^a_x \rangle$ is a derivation from Φ^a_x.

(A1) If $\mathcal{D}_1$ is an instance of A1, then it is of the form, $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$; so $\mathcal{D}_1{}^a_x$ is $[\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})]^a_x = \mathcal{P}^a_x \to (\mathcal{Q}^a_x \to \mathcal{P}^a_x)$; but this is an instance of A1; so if $\mathcal{D}_1$ is an instance of A1, then $\mathcal{D}_1{}^a_x$ is an instance of A1, and $\langle \mathcal{D}_1{}^a_x \rangle$ is a derivation from Φ^a_x.

(A2) Homework.

(A3) Homework.

(A4) If $\mathcal{D}_1$ is an instance of A4, then it is of the form, $\forall v \mathcal{P} \to \mathcal{P}^v_t$, for some variable v and term t that is free for v in $\mathcal{P}$. So $\mathcal{D}_1{}^a_x = [\forall v \mathcal{P} \to \mathcal{P}^v_t]^a_x = [\forall v \mathcal{P}]^a_x \to [\mathcal{P}^v_t]^a_x$. But since a is a constant, $[\forall v \mathcal{P}]^a_x = \forall v [\mathcal{P}^a_x]$. And since x does not appear in Δ, $x \neq v$; so by T8.9, $[\mathcal{P}^v_t]^a_x = [\mathcal{P}^a_x]^v_{t^a_x}$. So $\mathcal{D}_1{}^a_x = \forall v [\mathcal{P}^a_x] \to [\mathcal{P}^a_x]^v_{t^a_x}$; and since x is new to Δ and t is free for v in $\mathcal{P}$, t^a_x is free for v in $\mathcal{P}^a_x$; so $\forall v [\mathcal{P}^a_x] \to [\mathcal{P}^a_x]^v_{t^a_x}$ is an instance of A4; so if $\mathcal{D}_1$ is an instance of A4, then $\mathcal{D}_1{}^a_x$ is an instance of A4, and $\langle \mathcal{D}_1{}^a_x \rangle$ is a derivation from Φ^a_x.

(A5) Homework.

(eq) If $\mathcal{D}_1$ is an equality axiom, A6, A7, or A8, then it includes no constants; so $\mathcal{D}_1 = \mathcal{D}_1{}^a_x$; so $\mathcal{D}_1{}^a_x$ is an equality axiom, and $\langle \mathcal{D}_1{}^a_x \rangle$ is a derivation from Φ^a_x.

Assp: For any i, $1 \leq i < k$, $\langle \mathcal{D}_1{}^a_x \ldots \mathcal{D}_i{}^a_x \rangle$ is a derivation from Φ^a_x.

Show: $\langle \mathcal{D}_1{}^a_x \ldots \mathcal{D}_k{}^a_x \rangle$ is a derivation from Φ^a_x.

$\mathcal{D}_k$ is a member of Φ, an axiom, or arises from previous lines by MP or Gen. If $\mathcal{D}_k$ is a member of Φ or an axiom then, by reasoning as in the basis, $\langle \mathcal{D}_1{}^a_x \ldots \mathcal{D}_k{}^a_x \rangle$ is a derivation from Φ^a_x. So two cases remain.

(MP) Homework.

(Gen) If $\mathcal{D}_k$ arises by Gen, then there are some lines in Δ,

$$\begin{array}{lll} i & \mathcal{P} & \\ & \vdots & \\ k & \forall v \mathcal{P} & i \text{ Gen} \end{array}$$

where $i < k$ and $\mathcal{D}_k = \forall v \mathcal{P}$. By assumption $\mathcal{P}^a_x$ is a member of the derivation $\langle \mathcal{D}_1{}^a_x \dots \mathcal{D}_{k-1}{}^a_x \rangle$ from Φ^a_x; so $\forall v[\mathcal{P}^a_x]$ follows in this new derivation by Gen; but since a is a constant, this is $[\forall v \mathcal{P}]^a_x$. So $\langle \mathcal{D}_1{}^a_x \dots \mathcal{D}_k{}^a_x \rangle$ is a derivation from Φ^a_x.

$\langle \mathcal{D}_1{}^a_x \dots \mathcal{D}_k{}^a_x \rangle$ is a derivation from Φ^a_x.

Indct: For any n, $\langle \mathcal{D}_1{}^a_x \dots \mathcal{D}_n{}^a_x \rangle$ is a derivation from Φ^a_x.

The reason this works is that none of the justifications change: switching x for a leaves each line justified for the same reasons as before. The only sticking point may be the case for A4. But we did the real work for this in T8.9. Given this, the rest is straightforward. Observe that this theorem is generally available: from VC, a quantificational language has infinitely many variables; but derivations are finitely long; so there must always *be* variables that do not appear in a derivation Δ.

*E8.31. Finish the cases for A2, A3, A5, and MP to complete the proof of T8.10. You should set up the complete demonstration, but may refer to the text for cases completed there, as the text refers cases to homework.

E8.32. Where $\Phi = \{Ab\}$ and Δ is as follows,

1.	$\forall x{\sim}Ax \to {\sim}Ab$	A4
2.	$(\forall x{\sim}Ax \to {\sim}Ab) \to ({\sim}{\sim}Ab \to {\sim}\forall x{\sim}Ax)$	T3.13
3.	${\sim}{\sim}Ab \to {\sim}\forall x{\sim}Ax$	2,1 MP
4.	$Ab \to {\sim}{\sim}Ab$	T3.11
5.	$Ab \to {\sim}\forall x{\sim}Ax$	4,3 T3.2
6.	Ab	prem
7.	${\sim}\forall x{\sim}Ax$	5,6 MP
8.	$\exists x Ax$	7 abv

apply the method of T8.10 to show that Δ^b_y is a derivation from Φ^b_y. Do any of the justifications change? Explain.

E8.33. Set $\mathsf{U} = \{1\}$, $\mathsf{I}[\mathcal{S}] = \mathsf{T}$ for every sentence letter $\mathcal{S}$, $\mathsf{I}[\mathcal{R}^1] = \{1\}$ for every $\mathcal{R}^1$; $\mathsf{I}[\mathcal{R}^2] = \{\langle 1, 1\rangle\}$ for every $\mathcal{R}^2$; and in general, $\mathsf{I}[\mathcal{R}^n] = \{\langle 1, \dots, 1\rangle\}$. Notice that $\mathsf{I}[c]$ can only be 1 for every constant c, and $\mathsf{I}[h^n] = \{\langle\langle 1, \dots, 1\rangle, 1\rangle\}$ for every function symbol h^n. Where $\mathcal{P}$ is any formula whose only operators are $\to$, $\wedge$, $\vee$,

$\leftrightarrow$, $\forall$, and $\exists$, show by induction on the number of operators in $\mathcal{P}$ that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. Use this result to show that $\nvDash \sim\mathcal{P}$. Hint: This is a quantificational version of E8.14; this time you will want to show first that for any term t, $\mathsf{I_d}[t] = 1$; and with this that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

8.4 Additional Examples (for Part IV)

Again, our primary motivation in this section is to practice doing mathematical induction. This final series of examples develops some results about Q that will be particularly useful if you go on to consider Part IV. As we have already mentioned (page 306, and compare E7.19), many true generalizations are not provable in Robinson Arithmetic. However, we shall be able to show that Q is generally adequate for some interesting classes of results. As you work through these results, you may find it convenient to refer to the final Chapter 8 theorems reference on page 405.

First Theorems of Chapter 8

T8.1 For any $\mathcal{P}$ whose operators are $\sim$, $\vee$, $\wedge$, and $\rightarrow$, $\mathcal{P}_{\mathrm{N}}$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}_{\mathrm{N}}] = \mathsf{T}$.

T8.2 For variables x and v, if v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

T8.3 Where constant c does not appear in formula $\mathcal{P}$, $[\mathcal{P}^x_c]^c_v = \mathcal{P}^x_v$.

T8.4 For any interpretation I, variable assignments d and h, and term t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

T8.5 For any interpretation I, variable assignments d and h, and formula $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

T8.6 $\vDash \forall x(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P} \rightarrow \forall x\mathcal{Q})$ where x is not free in $\mathcal{P}$.

T8.7 For any interpretation I and sentence $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff there is some assignment d such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

T8.8 For any interpretation I and sentences $\mathcal{P}$ and $\mathcal{Q}$, (i) $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$; and (ii) $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$. Corollary: Similarly for $\wedge$, $\vee$, and $\leftrightarrow$.

T8.9 For any formula $\mathcal{P}$, term t, constant c, and distinct variables v and x, $[\mathcal{P}^v_t]^c_x$ is the same formula as $[\mathcal{P}^c_x]^v_{t^c_x}$.

T8.10 If Δ is an *AD* derivation from Φ, and x is a variable that does not appear in Δ, then for any constant a, Δ^a_x is an *AD* derivation from Φ^a_x.

First, we shall string together a series of results sufficient to show that Q correctly *decides* atomic sentences of $\mathcal{L}_{NT}$: For any atomic sentence $\mathcal{P}$ and the standard interpretation N, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then Q $\vdash_{ND+} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then Q $\vdash_{ND+} \sim\mathcal{P}$. Include among the atomic sentences equalities $s = t$, but also the inequalities, $s \leq t$ and $s < t$.[2] Observe that if $\mathcal{P}$ is atomic and a sentence, its terms s and t have no variables.

As a preliminary, let $\overline{\mathsf{n}}$ abbreviate, $\overbrace{S \ldots S}^{\mathsf{n}\ Ss} \emptyset$. So, for example, $\overline{2}$ is $SS\emptyset$, and $\overline{0}$ is just $\emptyset$. Any such $\overline{\mathsf{n}}$ is a *numeral*. Observe that $S\overline{2}$, say, is just $SSS\emptyset$. Then it is easy to see that,

T8.11. For any $\mathsf{n} \in \mathsf{U}$ and assignment d, $\mathsf{N_d}[\overline{\mathsf{n}}] = \mathsf{n}$.

By induction on the value of n. Consider an arbitrary assignment d.

Basis: By TA(c), $\mathsf{N_d}[\emptyset] = \mathsf{N}[\emptyset] = 0$; but this is just to say, $\mathsf{N_d}[\overline{0}] = 0$.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, $\mathsf{N_d}[\overline{\mathsf{i}}] = \mathsf{i}$.

Show: $\mathsf{N_d}[\overline{\mathsf{k}}] = \mathsf{k}$. Where $\mathsf{k} > 0$, $\overline{\mathsf{k}}$ is the same numeral as $S\overline{\mathsf{k}-1}$; and by assumption, $\mathsf{N_d}[\overline{\mathsf{k}-1}] = \mathsf{k}-1$. So $\mathsf{N_d}[\overline{\mathsf{k}}]$ is $\mathsf{N_d}[S\overline{\mathsf{k}-1}]$; by TA(f), this is $\mathsf{N}[S]\langle \mathsf{N_d}[\overline{\mathsf{k}-1}]\rangle$; by assumption this is $\mathsf{N}[S]\langle \mathsf{k}-1\rangle$; which is $(\mathsf{k}-1)+1$; which is k. So $\mathsf{N_d}[\overline{\mathsf{k}}] = \mathsf{k}$.

Indct: For any n and d, $\mathsf{N_d}[\overline{\mathsf{n}}] = \mathsf{n}$.

If the assignment to $\overline{\mathsf{k}-1}$ is $\mathsf{k}-1$, then the assignment to $S\overline{\mathsf{k}-1}$ is the successor of $\mathsf{k}-1$ which is k.[3] Typically, I shall treat this result as "common knowledge" and assert (or suppose) $\mathsf{N_d}[\overline{\mathsf{n}}] = \mathsf{n}$ without explicit appeal to T8.11.

8.4.1 Case

We begin with some simple results for the addition and multiplication of numerals.

T8.12. For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a}+\mathsf{b} = \mathsf{c}$, then Q $\vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

By induction on the value of b. Recall that by T6.48, Q $\vdash_{ND+} t + \emptyset = t$ and from T6.49, Q $\vdash_{ND+} t + Ss = S(t + s)$. Further, as above, we depend on the general fact that, so long as $\mathsf{a} > 0$, $S\overline{\mathsf{a}-1}$ is the same numeral as $\overline{\mathsf{a}}$.

[2] Of course, the inequalities are abbreviations, $\exists u(u + s = t)$ and $\exists u(Su + s = t)$ and as such, not atomic. However trees to construct abbreviated formulas have inequalities in the top row of their formula part—and, as for the identification of grammatical parts with other abbreviations, the notion is thus applied in a derived sense.

[3] Again, insofar as reasoning applies the assumption just to $\mathsf{k}-1$, it would have been natural to apply scheme III from the induction schemes reference (assume for m, show for $\mathsf{m}+1$); however we get the same effect by applying our usual assumption to $\mathsf{k}-1$ (see note 1 on page 370).

Basis: Suppose $\mathsf{b} = 0$ and $\mathsf{a} + \mathsf{b} = \mathsf{c}$; then $\mathsf{a} = \mathsf{c}$; but by T6.48, $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{0} = \overline{\mathsf{a}}$; so $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$ if $\mathsf{a} + \mathsf{i} = \mathsf{c}$, then $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{i}} = \overline{\mathsf{c}}$.

Show: If $\mathsf{a} + \mathsf{k} = \mathsf{c}$, then $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{k}} = \overline{\mathsf{c}}$.

Suppose $\mathsf{a} + \mathsf{k} = \mathsf{c}$. Since $\mathsf{k} > \mathsf{i}$, $\mathsf{k} > 0$ and so $\mathsf{c} > 0$; let $\mathsf{k} - 1 = \mathsf{m}$ and $\mathsf{c} - 1 = \mathsf{d}$; then $\overline{\mathsf{k}}$ is the same as $S\overline{\mathsf{m}}$, $\overline{\mathsf{c}}$ is the same as $S\overline{\mathsf{d}}$, and $\mathsf{a} + \mathsf{m} = \mathsf{d}$. From the latter, by assumption $Q \vdash_{ND+} (\overline{\mathsf{a}} + \overline{\mathsf{m}}) = \overline{\mathsf{d}}$; by T6.49, $Q \vdash_{ND+} (\overline{\mathsf{a}} + S\overline{\mathsf{m}}) = S(\overline{\mathsf{a}} + \overline{\mathsf{m}})$; so by =E, $Q \vdash_{ND+} (\overline{\mathsf{a}} + S\overline{\mathsf{m}}) = S\overline{\mathsf{d}}$; and this is just to say $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{k}} = \overline{\mathsf{c}}$.

Indct: For any a, b, and c, if $\mathsf{a} + \mathsf{b} = \mathsf{c}$, then $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

Corollary: if $\mathsf{a} + 1 = \mathsf{b}$ then $Q \vdash_{ND+} S\overline{\mathsf{a}} = \overline{\mathsf{b}}$. Suppose $\mathsf{a} + 1 = \mathsf{b}$; then as above, $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{1} = \overline{\mathsf{b}}$; but by T6.53, $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{1} = S\overline{\mathsf{a}}$; so by =E, $Q \vdash_{ND+} S\overline{\mathsf{a}} = \overline{\mathsf{b}}$.

The basic idea for this theorem is simple: From the basis, $Q \vdash_{ND+} \overline{\mathsf{a}} + \overline{0} = \overline{\mathsf{a}}$; then given the assumption for one value of b, we use T6.49 to get the next. Observe that a, b, and c are numbers—objects in the universe—and we informally manipulate them to conclude that, say, $\mathsf{a} = \mathsf{c}$ from $\mathsf{b} = 0$ and $\mathsf{a} + \mathsf{b} = \mathsf{c}$. In contrast, $\overline{\mathsf{a}}$, $\overline{\mathsf{k}}$, and $\overline{\mathsf{c}}$ are *numerals* of the sort $S \ldots S\emptyset$ and, say, $\overline{\mathsf{a}} + \overline{\mathsf{k}} = \overline{\mathsf{c}}$ is a sentence of $\mathcal{L}_{NT}$ which we show follows from the axioms of Q. It is not as though we somehow forget how to do arithmetic! Rather we understand arithmetic, and show how Q is related to it. Note the (slight) typographical difference between '$+$' in the object language and '+' to express the function.

*T8.13. For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} \times \mathsf{b} = \mathsf{c}$ then $Q \vdash_{ND+} \overline{\mathsf{a}} \times \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

By induction on the value of b. Hint: Let $\mathsf{k} - 1 = \mathsf{m}$ and $\mathsf{c} - \mathsf{a} = \mathsf{d}$. By assumption you should be able to obtain $Q \vdash_{ND+} \overline{\mathsf{a}} \times \overline{\mathsf{m}} = \overline{\mathsf{d}}$; then you will be able to apply T6.51 and T8.12 for the desired result.

*E8.34. Provide an argument to show T8.13.

8.4.2 Case

We now obtain a series of results for atomics and negated atomics whose terms are numerals. First, without mathematical induction it is easy to see that Q proves true atomic sentences with numerals as terms. Recall that $s \leq t$ is $\exists u(u + s = t)$ and $s < t$ is $\exists u(Su + s = t)$ for u not in s or t.

T8.14. For any $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, (i) if $\mathsf{a} = \mathsf{b}$ then $Q \vdash_{ND+} \overline{\mathsf{a}} = \overline{\mathsf{b}}$; (ii) if $\mathsf{a} \leq \mathsf{b}$ then $Q \vdash_{ND+} \overline{\mathsf{a}} \leq \overline{\mathsf{b}}$; and (iii) if $\mathsf{a} < \mathsf{b}$ then $Q \vdash_{ND+} \overline{\mathsf{a}} < \overline{\mathsf{b}}$.

(i) If $a = b$ then $Q \vdash_{ND+} \overline{a} = \overline{b}$: Suppose $a = b$; then $\overline{a}$ is the same term as $\overline{b}$; and by =I, $Q \vdash_{ND+} \overline{a} = \overline{b}$.

(ii) If $a \leq b$ then $Q \vdash_{ND+} \overline{a} \leq \overline{b}$: Suppose $a \leq b$; then there is some d that is the *difference* between them, such that $d + a = b$; so by T8.12, $Q \vdash_{ND+} \overline{d} + \overline{a} = \overline{b}$; so by ∃I, $Q \vdash_{ND+} \exists u(u + \overline{a} = \overline{b})$; and by abv, $Q \vdash_{ND+} \overline{a} \leq \overline{b}$.

(iii) If $a < b$ then $Q \vdash_{ND+} \overline{a} < \overline{b}$: Suppose $a < b$; then there is some d such that $(d + 1) + a = b$; so by T8.12, $Q \vdash_{ND+} \overline{d+1} + \overline{a} = \overline{b}$; but $\overline{d+1}$ is the same term as $S\overline{d}$; so $Q \vdash_{ND+} S\overline{d} + \overline{a} = \overline{b}$; so by ∃I, $Q \vdash_{ND+} \exists u(Su + \overline{a} = \overline{b})$; so by abv, $Q \vdash_{ND+} \overline{a} < \overline{b}$.

The cases for negated atomics are more interesting. Arguments are similar, though we require a preliminary for one case:

T8.15. $Q \vdash_{ND+} Sj + \overline{n} = j + S\overline{n}$.

Homework.

With this, we are ready for the results about negated atomics. Recall that according to T6.46, $Q \vdash_{ND+} St \neq \overline{0}$; and from T6.47, $Q \vdash_{ND+} St = Ss \rightarrow t = s$.

T8.16. For any $a, b \in U$, (i) if $a \neq b$, then $Q \vdash_{ND+} \overline{a} \neq \overline{b}$; (ii) if $a \not\leq b$, then $Q \vdash_{ND+} \overline{a} \not\leq \overline{b}$; and (iii) if $a \not< b$, then $Q \vdash_{ND+} \overline{a} \not< \overline{b}$.

(i) If $a \neq b$, then $Q \vdash_{ND+} \overline{a} \neq \overline{b}$: Suppose $a \neq b$. Then there is some $d > 0$ that is the difference between them, such that either $d + a = b$ or $d + b = a$. The argument is the same either way, so suppose the latter. We show that for any n, $Q \vdash_{ND+} \overline{d+n} \neq \overline{n}$; then when $n = b$, $Q \vdash_{ND+} \overline{d+b} \neq \overline{b}$ which is $Q \vdash_{ND+} \overline{a} \neq \overline{b}$.

Basis: Suppose $n = 0$. Then $d = d + n$ and $\overline{d} = \overline{d+n}$; and since $d > 0$, $\overline{d} = S\overline{d-1}$; so $S\overline{d-1} = \overline{d} = \overline{d+n}$. By T6.46, $Q \vdash_{ND+} S\overline{d-1} \neq \overline{0}$; but this is just to say $Q \vdash_{ND+} \overline{d+n} \neq \overline{n}$.

Assp: For $0 \leq i < k$, $Q \vdash_{ND+} \overline{d+i} \neq \overline{i}$.

Show: $Q \vdash_{ND+} \overline{d+k} \neq \overline{k}$.

Since $k > i$, $k > 0$; let $k - 1 = m$; then $\overline{k}$ is $S\overline{m}$ and $\overline{d+k}$ is $S\overline{d+m}$. By T6.47, $Q \vdash_{ND+} S\overline{d+m} = S\overline{m} \rightarrow \overline{d+m} = \overline{m}$; but by assumption, $Q \vdash_{ND+} \overline{d+m} \neq \overline{m}$; so by MT, $Q \vdash_{ND+} S\overline{d+m} \neq S\overline{m}$; which is to say, $Q \vdash_{ND+} \overline{d+k} \neq \overline{k}$.

Indct: For any n, $Q \vdash_{ND+} \overline{d+n} \neq \overline{n}$.

So $Q \vdash_{ND+} \overline{d+b} \neq \overline{b}$; so $Q \vdash_{ND+} \overline{a} \neq \overline{b}$.

(ii) If $\mathsf{a} \not\leq \mathsf{b}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$: Suppose $\mathsf{a} \not\leq \mathsf{b}$. Then $\mathsf{a} > \mathsf{b}$, and for some $\mathsf{d} > 0$, $\mathsf{d} + \mathsf{b} = \mathsf{a}$. By induction on n, we show that for any n, $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$; the case when $\mathsf{n} = \mathsf{b}$ gives $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$; then by $\forall$I, $\mathrm{Q} \vdash_{ND+} \forall u(u + \overline{\mathsf{a}} \neq \overline{\mathsf{b}})$; and the result follows with QN.

Basis: Suppose $\mathsf{n} = 0$; then $\mathsf{d} = \mathsf{d} + \mathsf{n}$. For $\mathsf{d} > 0$, let $\mathsf{d} - 1 = \mathsf{m}$; then $\overline{\mathsf{d}} = S\overline{\mathsf{m}}$; and $S\overline{\mathsf{m}} = \overline{\mathsf{d}} = \overline{\mathsf{d}+\mathsf{n}}$. By T6.49, $\mathrm{Q} \vdash_{ND+} j + S\overline{\mathsf{m}} = S(j + \overline{\mathsf{m}})$; and by T6.46, $\mathrm{Q} \vdash_{ND+} S(j + \overline{\mathsf{m}}) \neq \overline{0}$; so by =E, $\mathrm{Q} \vdash_{ND+} j + S\overline{\mathsf{m}} \neq \overline{0}$; where this is just to say $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$.

Assp: For $0 \leq \mathsf{i} < \mathsf{k}$, $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{i}} \neq \overline{\mathsf{i}}$.

Show: $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{k}} \neq \overline{\mathsf{k}}$.

Since $\mathsf{k} > \mathsf{i}$, $\mathsf{k} > 0$; let $\mathsf{k} - 1 = \mathsf{m}$; then $\overline{\mathsf{k}} = S\overline{\mathsf{m}}$ and $\overline{\mathsf{d}+\mathsf{k}} = S\overline{\mathsf{d}+\mathsf{m}}$. By T6.47, $\mathrm{Q} \vdash_{ND+} S(j + \overline{\mathsf{d}+\mathsf{m}}) = S\overline{\mathsf{m}} \rightarrow j + \overline{\mathsf{d}+\mathsf{m}} = \overline{\mathsf{m}}$; but by assumption, $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{m}} \neq \overline{\mathsf{m}}$; so by MT, $\mathrm{Q} \vdash_{ND+} S(j + \overline{\mathsf{d}+\mathsf{m}}) \neq S\overline{\mathsf{m}}$; by T6.49, $\mathrm{Q} \vdash_{ND+} j + S\overline{\mathsf{d}+\mathsf{m}} = S(j + \overline{\mathsf{d}+\mathsf{m}})$; so by =E, $\mathrm{Q} \vdash_{ND+} j + S\overline{\mathsf{d}+\mathsf{m}} \neq S\overline{\mathsf{m}}$; but this is just to say, $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{k}} \neq \overline{\mathsf{k}}$.

Indct: For any n, $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$.

So $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{d}+\mathsf{b}} \neq \overline{\mathsf{b}}$ which is to say $\mathrm{Q} \vdash_{ND+} j + \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$. So by $\forall$I, $\mathrm{Q} \vdash_{ND+} \forall u(u + \overline{\mathsf{a}} \neq \overline{\mathsf{b}})$; and by QN, $\mathrm{Q} \vdash_{ND+} \sim\exists u(u + \overline{\mathsf{a}} = \overline{\mathsf{b}})$; which is to say, $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$.

(iii) If $\mathsf{a} \not< \mathsf{b}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \not< \overline{\mathsf{b}}$. Homework.

Supposing that $\mathrm{Q} \vdash_{ND+} \overline{0} + \overline{\mathsf{d}+\mathsf{n}} = \overline{\mathsf{d}+\mathsf{n}}$ (which you can show by an easy induction), it is possible to reconceive the result of the induction for (i) as an *instance* of that for (ii), in the case when $j = \overline{0}$.

E8.35. Provide arguments to show T8.15 and (iii) of T8.16. Hints: T8.15 is a simple induction on n; for the show, let $\mathsf{m} = \mathsf{k} - 1$; you will want the assumption in the form, $\mathrm{Q} \vdash_{ND+} Sj + \overline{\mathsf{m}} = j + S\overline{\mathsf{m}}$. For T8.16, the induction is to show $\mathrm{Q} \vdash_{ND+} Sj + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$. There is a complication, however, in the basis: From $\mathsf{a} \not< \mathsf{b}$, $\mathsf{b} + \mathsf{d} = \mathsf{a}$ for $\mathsf{d} \geq 0$; given that d might be 0, we cannot simply treat d as a successor and set $\overline{\mathsf{d}} = S\overline{\mathsf{m}}$ as above; you can solve the problem by obtaining $j + S\overline{\mathsf{d}} \neq \overline{0}$ for an application of T8.15. For the show, since $\mathsf{k} > 0$, the argument remains straightforward.

E8.36. Show that for any n, $\mathrm{Q} \vdash_{ND+} \overline{0} + \overline{\mathsf{n}} = \overline{\mathsf{n}}$, and use this to obtain the result of the induction for T8.16(i) from that for T8.16(ii).

8.4.3 Case

The results of the previous section were limited to atomics and negated atomics whose terms are numerals. We now extend results to consider atomics with terms of arbitrary complexity. The atomic sentences are of course still *sentences* so that their terms remain variable-free.

We have said a formula is *true* iff it is satisfied on every variable assignment. Let us introduce a parallel notion for terms.

AI The *assignment* to a term on an interpretation $\mathsf{I}[t] = \mathsf{n}$ iff with any d for I, $\mathsf{I_d}[t] = \mathsf{n}$.

If there were some one a such that for every $\mathsf{x} \in \mathsf{U}$, $\langle \mathsf{x}, \mathsf{a}\rangle \in \mathsf{I}[h^1]$, then for any d and assignment to x, $\mathsf{I_d}[h^1x] = \mathsf{a}$—and the assignment to h^1x, $\mathsf{I}[h^1x] = \mathsf{a}$. We meet some functions of this sort in Chapter 12; but such "constant" functions are not the norm. More relevantly, from T8.4, if assignments d and h agree on assignments to variables in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$; and if t is without variables then any assignments agree on assignments to all the variables in t; so it is automatic that, for a variable-free term, any $\mathsf{I_d}[t] = \mathsf{I_h}[t] = \mathsf{I}[t]$.

Given this, we start by establishing that Q proves the proper relation between arbitrary variable-free terms and numerals.

T8.17. For any variable-free term t of $\mathscr{L}_{NT}$, if $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND+} t = \overline{\mathsf{n}}$.

Let t be a variable-free term of $\mathscr{L}_{NT}$. By induction on the number of function symbols in t,

Basis: Suppose t has no function symbols and $\mathsf{N}[t] = \mathsf{n}$. Then t can only be the constant $\emptyset$; so $\mathsf{N}[t] = \mathsf{N}[\emptyset] = 0$; and $\mathsf{n} = 0$. But by =I, $Q \vdash_{ND+} \overline{0} = \overline{0}$; so $Q \vdash_{ND+} t = \overline{\mathsf{n}}$.

Assp: For any $i, 0 \le i < k$ if t has i function symbols and $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND+} t = \overline{\mathsf{n}}$.

Show: If t has k function symbols and $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND+} t = \overline{\mathsf{n}}$.

Suppose t has has k function symbols and $\mathsf{N}[t] = \mathsf{n}$. Then t is of the form, $S r$, $r + s$, or $r \times s$ for r, s, with $< k$ function symbols.

(S) t is Sr. Since t is variable-free, r is variable-free and $\mathsf{N}[r] = \mathsf{N_d}[r] = \mathsf{a}$ for some a. And since t is variable-free, $\mathsf{N}[t] = \mathsf{N_d}[t] = \mathsf{N_d}[Sr]$; by TA(f), this is $\mathsf{N}[S]\langle \mathsf{N_d}[r]\rangle = \mathsf{N}[S]\langle \mathsf{a}\rangle = \mathsf{a}+1$; so $\mathsf{N}[t] = \mathsf{a}+1$; so $\mathsf{a}+1 = \mathsf{n}$. By assumption $Q \vdash_{ND+} r = \overline{\mathsf{a}}$; and by =I, $Q \vdash_{ND+} Sr = Sr$; so by =E, $Q \vdash_{ND+} Sr = S\overline{\mathsf{a}}$; but since $\mathsf{a}+1 = \mathsf{n}$ by the corollary to T8.12, $Q \vdash_{ND+} S\overline{\mathsf{a}} = \overline{\mathsf{n}}$; so by =E, $Q \vdash_{ND+} Sr = \overline{\mathsf{n}}$, where this is to say $Q \vdash_{ND+} t = \overline{\mathsf{n}}$.

(+) t is $r + s$. Since t is variable-free, r and s are variable-free and $\mathsf{N}[r] = \mathsf{N_d}[r] = \mathsf{a}$ and $\mathsf{N}[s] = \mathsf{N_d}[s] = \mathsf{b}$ for some a and b. Since t is variable-free,

$\mathsf{N}[t] = \mathsf{N_d}[t] = \mathsf{N_d}[r + s]$; by TA(f), $\mathsf{N_d}[r + s] = \mathsf{N}[+]\langle \mathsf{a}, \mathsf{b}\rangle = \mathsf{a} + \mathsf{b}$; so $\mathsf{N}[t] = \mathsf{a}+\mathsf{b}$; so $\mathsf{a}+\mathsf{b} = \mathsf{n}$. By assumption, $Q \vdash_{ND+} r = \bar{\mathsf{a}}$ and $Q \vdash_{ND+} s = \bar{\mathsf{b}}$; and by =I, $Q \vdash_{ND+} r + s = r + s$; so by =E, $Q \vdash_{ND+} r + s = \bar{\mathsf{a}} + \bar{\mathsf{b}}$; but since $\mathsf{a}+\mathsf{b} = \mathsf{n}$, by T8.12, $Q \vdash_{ND+} \bar{\mathsf{a}} + \bar{\mathsf{b}} = \bar{\mathsf{n}}$; so by =E, $Q \vdash_{ND+} r + s = \bar{\mathsf{n}}$, where this is to say $Q \vdash_{ND+} t = \bar{\mathsf{n}}$.

($\times$) Similarly [by homework].

If variable-free t has k function symbols and $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND+} t = \bar{\mathsf{n}}$.

Indct: For any variable-free term t, if $\mathsf{N}[t] = \mathsf{n}$ then $Q \vdash_{ND+} t = \bar{\mathsf{n}}$.

Our intended result, that Q correctly decides atomic sentences of $\mathscr{L}_{NT}$ is not an argument by induction, but rather collects what we have done into a simple argument. The general idea is that from the truth or falsity of an atomic sentence including some terms s and t, by semantic reasoning we may obtain a result for some corresponding objects $\mathsf{a}, \mathsf{b} \in \mathbb{N}$; then, given T8.14 and T8.16, a result in Q for terms $\bar{\mathsf{a}}, \bar{\mathsf{b}}$—and finally, with T8.17, the desired result for the original atomic involving terms s and t.

T8.18. Q correctly decides atomic sentences of $\mathscr{L}_{NT}$: For any sentence $\mathcal{P}$ of the sort $s = t$, $s \leq t$, or $s < t$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $Q \vdash_{ND+} \mathcal{P}$; and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $Q \vdash_{ND+} \sim\mathcal{P}$.

Consider an atomic sentence $\mathcal{P}$ of the sort $s = t$, $s \leq t$, or $s < t$. Since $\mathcal{P}$ is a sentence, s and t are variable-free. A few selected parts are worked as examples.

(a) Suppose $\mathsf{N}[s = t] = \mathsf{T}$; then by TI, for any d, $\mathsf{N_d}[s = t] = \mathsf{S}$; so by SF(r), $\langle \mathsf{N_d}[s], \mathsf{N_d}[t]\rangle \in \mathsf{N}[=]$; so $\mathsf{N_d}[s] = \mathsf{N_d}[t]$. But since s and t are variable-free, for some a and b, $\mathsf{N_d}[s] = \mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N_d}[t] = \mathsf{N}[t] = \mathsf{b}$; so $\mathsf{a} = \mathsf{b}$; so by T8.14, $Q \vdash_{ND+} \bar{\mathsf{a}} = \bar{\mathsf{b}}$; but since $\mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}[t] = \mathsf{b}$, by T8.17, $Q \vdash_{ND+} s = \bar{\mathsf{a}}$ and $Q \vdash_{ND+} t = \bar{\mathsf{b}}$; so by =E, $Q \vdash_{ND+} s = t$.

(b) Suppose $\mathsf{N}[s = t] \neq \mathsf{T}$; then [by homework] $Q \vdash_{ND+} s \neq t$.

(c) Suppose $\mathsf{N}[s \leq t] = \mathsf{T}$; then $\mathsf{N}[\exists u(u + s = t)] = \mathsf{T}$; so by TI, for any d, $\mathsf{N_d}[\exists u(u + s = t)] = \mathsf{S}$; so by SF$'$($\exists$), for some $\mathsf{m} \in \mathsf{U}$, $\mathsf{N_{d(\mathit{u}|m)}}[u + s = t] = \mathsf{S}$. $\mathsf{d}(u|\mathsf{m})[u] = \mathsf{m}$; so by TA(v), $\mathsf{N_{d(\mathit{u}|m)}}[u] = \mathsf{m}$; since s and t are variable-free, for some a and b, $\mathsf{N_{d(\mathit{u}|m)}}[s] = \mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N_{d(\mathit{u}|m)}}[t] = \mathsf{N}[t] = \mathsf{b}$; so by TA(f), $\mathsf{N_{d(\mathit{u}|m)}}[u + s] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{a}\rangle = \mathsf{m} + \mathsf{a}$. So by SF(r), $\langle \mathsf{m} + \mathsf{a}, \mathsf{b}\rangle \in \mathsf{N}[=]$; so $\mathsf{m} + \mathsf{a} = \mathsf{b}$; so $\mathsf{a} \leq \mathsf{b}$; so by T8.14, $Q \vdash_{ND+} \bar{\mathsf{a}} \leq \bar{\mathsf{b}}$. But since $\mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}[t] = \mathsf{b}$, by T8.17, $Q \vdash_{ND+} s = \bar{\mathsf{a}}$ and $Q \vdash_{ND+} t = \bar{\mathsf{b}}$; so by =E, $Q \vdash_{ND+} s \leq t$.

(d) Suppose $\mathsf{N}[s \leq t] \neq \mathsf{T}$; then $\mathsf{N}[\exists u(u + s = t)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{N_d}[\exists u(u + s = t)] \neq \mathsf{S}$; so by SF$'$($\exists$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{N_{d(\mathit{u}|o)}}[u + s = t] \neq \mathsf{S}$; let m be an arbitrary individual of this sort; then $\mathsf{N_{d(\mathit{u}|m)}}[u + s = t] \neq \mathsf{S}$.

$\mathsf{d}(u|\mathsf{m})[u] = \mathsf{m}$; so by TA(v), $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u] = \mathsf{m}$; and since s and t are variable-free, for some a and b, $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[s] = \mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[t] = \mathsf{N}[t] = \mathsf{b}$; so by TA(f), $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u + s] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{a}\rangle = \mathsf{m} + \mathsf{a}$. So by SF(r), $\langle \mathsf{m} + \mathsf{a}, \mathsf{b}\rangle \notin \mathsf{N}[=]$; so $\mathsf{m} + \mathsf{a} \neq \mathsf{b}$; and since m is arbitrary, for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{o} + \mathsf{a} \neq \mathsf{b}$; so $\mathsf{a} \not\leq \mathsf{b}$; so by T8.16, Q $\vdash_{ND+} \overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$. But since $\mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}[t] = \mathsf{b}$, by T8.17, Q $\vdash_{ND+} s = \overline{\mathsf{a}}$ and Q $\vdash_{ND+} t = \overline{\mathsf{b}}$; so by =E, Q $\vdash_{ND+} s \not\leq t$.

(e) Suppose $\mathsf{N}[s < t] = \mathsf{T}$; then [by homework] Q $\vdash_{ND+} s < t$.

(f) Suppose $\mathsf{N}[s < t] \neq \mathsf{T}$; then [by homework] Q $\vdash_{ND+} s \not< t$.

This is an interesting result! Q is sufficient to decide arbitrary atomic sentences of basic arithmetic. So, for example, insofar as the formula $\overline{3} \times \overline{2} = \overline{2} \times \overline{3}$ is true on N, by T8.18, Q $\vdash_{ND+} \overline{3} \times \overline{2} = \overline{2} \times \overline{3}$ (compare E6.40e). And similarly for atomic sentences and their negations whose terms are arbitrarily complex.

E8.37. Complete the argument for T8.17 by completing the case for ($\times$). You should set up the entire induction, but may appeal to the text for parts that are already completed, just as the text appeals to homework.

E8.38. Complete the remaining cases of T8.18 to show that Q correctly decides atomic sentences of $\mathcal{L}_{NT}$.

8.4.4 Case

We conclude the chapter with some more examples of mathematical induction, this time working toward important results about inequality. The primary result is a version of *trichotomy*, the result that for any n, Q $\vdash_{ND+} \forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$. Again, though, we begin with preliminaries. First, a simple argument that introduces a pattern of reasoning we shall see again.

T8.19. For any n, $\vdash_{ND+} \forall x \forall y(x = Sy \to [(y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{n}}) \to (x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{n}})])$.

By induction on the value of n we show, $\vdash_{ND+} j = Sk \to [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{n}}) \to (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{n}})]$. The result follows by $\forall$I.

Basis: $\mathsf{n} = 0$. In this case, we require $\vdash_{ND+} j = Sk \to [k = \overline{0} \to j = S\overline{0}]$. But this is immediate by a couple applications of $\to$I.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, $\vdash_{ND+} j = Sk \to [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{i}}) \to (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{i}})]$.

Show: $\vdash_{ND+} j = Sk \rightarrow [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{k}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{k}})]$.
Let $\mathsf{m} = \mathsf{k} - 1$. For the derivation, see the box below.

Indct: For any n, $\vdash_{ND+} j = Sk \rightarrow [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{n}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{n}})]$.

So by $\forall$I, $\vdash_{ND+} \forall x \forall y(x = Sy \rightarrow [(y = \overline{0} \vee \ldots \vee y = \overline{\mathsf{n}}) \rightarrow (x = S\overline{0} \vee \ldots \vee x = S\overline{\mathsf{n}})])$, and the theorem is proved.

The basic idea is that we can use $j = Sk$ together with an extended version of $\vee$E on $k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{n}}$ to get the result. The induction works by obtaining the result for the first disjunct, and then showing that no matter how far we have gone, it is always possible to go to the next stage. Observe that we have not included parentheses for extended disjunctions—for it is always possible by Assoc to group disjuncts so as to justify arbitrary applications of $\vee$E and $\vee$I as above (and we prove it in E8.40). This theorem is useful for the next.

Recall that the *bounded* quantifiers $(\forall x < t)\mathcal{P}$, $(\exists x < t)\mathcal{P}$, $(\forall x \leq t)\mathcal{P}$, and $(\exists x \leq t)\mathcal{P}$, are abbreviations with associated derived introduction and exploitation rules (see page 300). Now,

T8.20. For any n, (i) $\mathrm{Q} \vdash_{ND+} (\forall x \leq \overline{\mathsf{n}})(x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n}})$ and (ii) $\mathrm{Q} \vdash_{ND+} (\forall x < \overline{\mathsf{n}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n} - 1})$.

The first disjunct $\emptyset \neq \emptyset$ in (ii) guarantees that the result is a well-formed sentence even when $\mathsf{n} = 0$. When $\mathsf{n} = 0$ the series reduces to $\emptyset \neq \emptyset$ since it contains

T8.19 (show)

1.	$j = Sk \rightarrow [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{m}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}})]$	by assp
2.	$j = Sk$	A (g, $\rightarrow$I)
3.	$k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{m}} \vee k = \overline{\mathsf{k}}$	A (g, $\rightarrow$I)
4.	$k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{m}}$	A (g, 3$\vee$E)
5.	$(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{m}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}})$	1,2 $\rightarrow$E
6.	$j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}}$	5,4 $\rightarrow$E
7.	$j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}} \vee j = S\overline{\mathsf{k}}$	6 $\vee$I
8.	$k = \overline{\mathsf{k}}$	A (g, 3$\vee$E)
9.	$j = S\overline{\mathsf{k}}$	2,8 =E
10.	$j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}} \vee j = S\overline{\mathsf{k}}$	9 $\vee$I
11.	$j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{m}} \vee j = S\overline{\mathsf{k}}$	3,4-7,8-10 $\vee$E
12.	$(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{k}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{k}})$	3-11 $\rightarrow$I
13.	$j = Sk \rightarrow [(k = \overline{0} \vee \ldots \vee k = \overline{\mathsf{k}}) \rightarrow (j = S\overline{0} \vee \ldots \vee j = S\overline{\mathsf{k}})]$	2-12 $\rightarrow$I

all the members "up" to $\mathsf{n}-1$ and there are not any; when $\mathsf{n} = 1$ the series is $\emptyset \neq \emptyset \vee x = \overline{0}$; and so forth. We work part (ii). By induction on n,

Basis: Suppose $\mathsf{n} = 0$. We need to show $Q \vdash_{ND+} (\forall x < \emptyset)(\emptyset \neq \emptyset)$. But this is easy with T6.55 and the rule ($\forall$I).

1.	$j < \emptyset$	A (g, ($\forall$I))
2.	$j \not< \emptyset$	T6.55
3.	$\perp$	1,2 $\perp$I
4.	$\emptyset \neq \emptyset$	3 $\perp$E
5.	$(\forall x < \emptyset)(\emptyset \neq \emptyset)$	1-4 ($\forall$I)

Assp: For $0 \le \mathsf{i} < \mathsf{k}$, $Q \vdash_{ND+} (\forall x < \overline{\mathsf{i}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{i}-1})$.

Show: $Q \vdash_{ND+} (\forall x < \overline{\mathsf{k}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{k}-1})$. Let $\mathsf{m} = \mathsf{k}-1$. Then by assumption $Q \vdash_{ND+} (\forall x < \overline{\mathsf{m}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{m}-1})$. See the derivation on page 401.

Indct: For any n, $Q \vdash_{ND+} (\forall x < \overline{\mathsf{n}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \ldots \vee x = \overline{\mathsf{n}-1})$.

E8.39. Complete the demonstration of T8.20 by showing part (i). Hint: The basis is easy with T6.54.

*E8.40. For extended disjunctions we have not included parentheses. Say these disjunctions are implicitly left-associated as, $((((A \vee B) \vee C) \vee D) \vee E)$. Then applications of $\vee$-rules apply directly just to the main, rightmost, operator. Where,

$$\mathcal{P}_n = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_u \vee \mathcal{A}_v \vee \ldots \vee \mathcal{A}_n$$
$$\mathcal{Q}_u^1 = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_u$$
$$\mathcal{Q}_n^v = \mathcal{A}_v \vee \ldots \vee \mathcal{A}_n$$

are each left-associated, show that our "loose" reasoning is justified by showing that for any $\mathcal{P}_n$ and any u, $1 \le u < n$, $\vdash_{ND+} \mathcal{P}_n \leftrightarrow (\mathcal{Q}_u^1 \vee \mathcal{Q}_n^v)$. Thus the left-associated $\mathcal{P}_n$ is provably equivalent to disjunctions with arbitrary main operator. Hint: The argument is by induction on the value of n. Let $m = k - 1$; then $\mathcal{P}_k = (\mathcal{P}_m \vee \mathcal{A}_k)$.

8.4.5 Case

The next theorem is a converse to T8.20 (after unabbreviation of its bounded quantifiers), and illustrates a pattern of reasoning we have already seen in application to extended disjunctions.

T8.21. For any n, (i) $Q \vdash_{ND+} \forall x[(x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n}}) \to x \le \overline{\mathsf{n}}]$ and (ii) $Q \vdash_{ND+} \forall x[(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{n}-1}) \to x < \overline{\mathsf{n}}]$.

Again I illustrate just (ii). For any n and $\mathsf{a} \leq \mathsf{n}$ we show by induction on the value of a that Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{a}-1}) \to j < \overline{\mathsf{n}}$; the case when $\mathsf{a} = \mathsf{n}$ gives Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{n}-1}) \to j < \overline{\mathsf{n}}$; and the desired result follows immediately by $\forall$I. Observe that when $\mathsf{a} = 0$ the series reduces to $\emptyset \neq \emptyset$ as before.

Basis: $\mathsf{a} = 0$. We need Q $\vdash_{ND+} \emptyset \neq \emptyset \to j < \overline{\mathsf{n}}$.

$$
\begin{array}{rl|l}
1. & \quad \emptyset \neq \emptyset & \text{A } (g, \to\text{I}) \\
2. & \quad \emptyset = \emptyset & =\text{I} \\
3. & \quad \bot & 2,1\ \bot\text{I} \\
4. & \quad j < \overline{\mathsf{n}} & 3\ \bot\text{E} \\
5. & \emptyset \neq \emptyset \to j < \overline{\mathsf{n}} & 1\text{-}4\ \to\text{I}
\end{array}
$$

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k} \leq \mathsf{n}$, Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{i}-1}) \to j < \overline{\mathsf{n}}$.

Show: For $\mathsf{k} \leq \mathsf{n}$, Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{k}-1}) \to j < \overline{\mathsf{n}}$. Let $\mathsf{k} - 1 = \mathsf{m}$.

$$
\begin{array}{rl|l}
1. & (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{m}-1}) \to j < \overline{\mathsf{n}} & \text{by assp} \\
2. & \quad \emptyset \neq \emptyset \vee j = \emptyset \vee \ldots \vee j = \overline{\mathsf{m}-1} \vee j = \overline{\mathsf{k}-1} & \text{A } (g, \to\text{I}) \\
3. & \quad\quad \emptyset \neq \emptyset \vee j = \emptyset \vee \ldots \vee j = \overline{\mathsf{m}-1} & \text{A } (g, 2\vee\text{E}) \\
4. & \quad\quad j < \overline{\mathsf{n}} & 1,3\ \to\text{E} \\
5. & \quad\quad j = \overline{\mathsf{k}-1} & \text{A } (g, 2\vee\text{E}) \\
6. & \quad\quad \overline{\mathsf{k}-1} < \overline{\mathsf{n}} & \text{T8.14 (since } \mathsf{k}-1 < \mathsf{n}) \\
7. & \quad\quad j < \overline{\mathsf{n}} & 6,5\ =\text{E} \\
8. & \quad j < \overline{\mathsf{n}} & 2,3\text{-}4,5\text{-}7\ \vee\text{E} \\
9. & (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{k}-1}) \to j < \overline{\mathsf{n}} & 2\text{-}8\ \to\text{I}
\end{array}
$$

Indct: For any $\mathsf{a} \leq \mathsf{n}$, Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{a}-1}) \to j < \overline{\mathsf{n}}$.

So Q $\vdash_{ND+} (\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{n}-1}) \to j < \overline{\mathsf{n}}$; and by $\forall$I, Q $\vdash_{ND+} \forall x[(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{n}-1}) \to x < \overline{\mathsf{n}}]$.

The next theorem does not require mathematical induction at all, but is required for our trichotomy result.

T8.22. For any n, (i) Q $\vdash_{ND+} \forall x[\overline{\mathsf{n}} \leq x \to (\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} \leq x)]$ and (ii) Q $\vdash_{ND+} \forall x[\overline{\mathsf{n}} < x \to (S\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} < x)]$.

Part (ii) is worked in the box on page 403.

With this, we are ready to obtain the result at which we have been aiming:

T8.23. For any n, (i) Q $\vdash_{ND+} \forall x(x \leq \overline{\mathsf{n}} \vee \overline{\mathsf{n}} \leq x)$ and (ii) Q $\vdash_{ND+} \forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$.

T8.20 (show)

1.	$\forall x \forall y(x = Sy \to [(y = \overline{0} \vee \ldots \vee y = \overline{m-1}) \to$ $(x = \overline{1} \vee \ldots \vee x = S\overline{m-1})])$	T8.19
2.	$(\forall x < \overline{m})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{m-1})$	by assp
3.	$j < \overline{k}$	A $(g, (\forall\mathrm{I}))$
4.	$j = \overline{0} \vee \exists y(j = Sy)$	from T6.52
5.	$j = \overline{0}$	A $(g, 4\vee\mathrm{E})$
6.	$j = \overline{0} \vee \ldots \vee j = \overline{k-1}$	5 $\vee$I
7.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{k-1}$	6 $\vee$I
8.	$\exists y(j = Sy)$	A $(g, 4\vee\mathrm{E})$
9.	$j = Sl$	A $(g, 8\exists\mathrm{E})$
10.	$\exists u(Su + j = \overline{k})$	3 abv
11.	$Sh + j = \overline{k}$	A $(g, 10\exists\mathrm{E})$
12.	$Sh + Sl = \overline{k}$	11,9 =E
13.	$Sh + Sl = S(Sh + l)$	T6.49
14.	$S(Sh + l) = \overline{k}$	12,13 =E
15.	$S(Sh + l) = S\overline{m}$	14 abv
16.	$S(Sh + l) = S\overline{m} \to Sh + l = \overline{m}$	T6.47
17.	$Sh + l = \overline{m}$	16,15 $\to$E
18.	$\exists u(Su + l = \overline{m})$	17 $\exists$I
19.	$l < \overline{m}$	18 abv
20.	$l < \overline{m}$	10,11-19 $\exists$E
21.	$\emptyset \neq \emptyset \vee l = \overline{0} \vee \ldots \vee l = \overline{m-1}$	2,20 $(\forall\mathrm{E})$
22.	$\emptyset \neq \emptyset$	A $(g, 21\vee\mathrm{E})$
23.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{k-1}$	22 $\vee$I
24.	$l = \overline{0} \vee \ldots \vee l = \overline{m-1}$	A $(g, 21\vee\mathrm{E})$
25.	$j = Sl \to [(l = \overline{0} \vee \ldots \vee l = \overline{m-1}) \to$ $(j = \overline{1} \vee \ldots \vee j = S\overline{m-1})]$	1 with $\forall$E
26.	$(l = \overline{0} \vee \ldots \vee l = \overline{m-1}) \to (j = \overline{1} \vee \ldots \vee j = S\overline{m-1})$	25,9 $\to$E
27.	$j = \overline{1} \vee \ldots \vee j = S\overline{m-1}$	26,24 $\to$E
28.	$j = \overline{1} \vee \ldots \vee j = \overline{k-1}$	27 abv
29.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{k-1}$	28 $\vee$I
30.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{k-1}$	21,22-23,24-29 $\vee$E
31.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{k-1}$	8,9-30 $\exists$E
32.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{k-1}$	4,5-7,8-31 $\vee$E
33.	$(\forall x < \overline{k})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{k-1})$	3-32 $(\forall\mathrm{I})$

The derivation is long but straightforward. From T6.52, either j is zero or it is not. If j is zero, then the result is easy. If j is the successor of some l, then $l < \overline{m}$ and the assumption applies; then the result follows with T8.19. Again parentheses for extended disjunctions are omitted.

We show (ii). By induction on the value of n, $\mathrm{Q} \vdash_{ND+} j < \overline{\mathsf{n}} \vee j = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < j$; the result immediately follows by $\forall$I.

Basis: $\mathsf{n} = 0$. We need to show that $\mathrm{Q} \vdash_{ND+} j < \overline{0} \vee j = \overline{0} \vee \overline{0} < j$.

1.	$j = \overline{0} \vee \exists y(j = Sy)$	from T6.52
2.	$j = \overline{0}$	A (g, 1$\vee$E)
3.	$j = \overline{0} \vee \overline{0} < j$	2 $\vee$I
4.	$\exists y(j = Sy)$	A (g, 1$\vee$E)
5.	$j = Sk$	A (g, 4$\exists$E)
6.	$Sk + \overline{0} = Sk$	T6.48
7.	$Sk + \overline{0} = j$	6,5 =E
8.	$\exists u(Su + \overline{0} = j)$	7 $\exists$I
9.	$\overline{0} < j$	8 abv
10.	$j = \overline{0} \vee \overline{0} < j$	9 $\vee$I
11.	$j = \overline{0} \vee \overline{0} < j$	4,5-10 $\exists$E
12.	$j = \overline{0} \vee \overline{0} < j$	1,2-3,4-11 $\vee$E
13.	$j < \overline{0} \vee j = \overline{0} \vee \overline{0} < j$	12 $\vee$I

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, $\mathrm{Q} \vdash_{ND+} j < \overline{\mathsf{i}} \vee j = \overline{\mathsf{i}} \vee \overline{\mathsf{i}} < j$.

Show: $\mathrm{Q} \vdash_{ND+} j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$. Let $\mathsf{m} = \mathsf{k} - 1$.

1.	$(\forall x < \overline{\mathsf{m}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{m}-1})$	T8.20
2.	$\forall x[(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{m}-1} \vee x = \overline{\mathsf{k}-1}) \rightarrow x < \overline{\mathsf{k}}]$	T8.21
3.	$\forall x[\overline{\mathsf{m}} < x \rightarrow (S\overline{\mathsf{m}} = x \vee S\overline{\mathsf{m}} < x)]$	T8.22
4.	$j < \overline{\mathsf{m}} \vee j = \overline{\mathsf{m}} \vee \overline{\mathsf{m}} < j$	by assp
5.	$j < \overline{\mathsf{m}}$	A (g, 4$\vee$E)
6.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{m}-1}$	1,5 ($\forall$E)
7.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{m}-1} \vee j = \overline{\mathsf{k}-1}$	6 $\vee$I
8.	$(\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{m}-1} \vee j = \overline{\mathsf{k}-1}) \rightarrow j < \overline{\mathsf{k}}$	2 $\forall$E
9.	$j < \overline{\mathsf{k}}$	8,7 $\rightarrow$E
10.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	9 $\vee$I
11.	$j = \overline{\mathsf{m}}$	A (g, 4$\vee$E)
12.	$\overline{\mathsf{m}} < \overline{\mathsf{k}}$	T8.14 ($\mathsf{m} < \mathsf{k}$)
13.	$j < \overline{\mathsf{k}}$	12,11 =E
14.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	13 $\vee$I
15.	$\overline{\mathsf{m}} < j$	A (g, 4$\vee$E)
16.	$\overline{\mathsf{m}} < j \rightarrow (S\overline{\mathsf{m}} = j \vee S\overline{\mathsf{m}} < j)$	3 $\forall$E
17.	$j = S\overline{\mathsf{m}} \vee S\overline{\mathsf{m}} < j$	16,15 $\rightarrow$E
18.	$j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	17 abv
19.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	18 $\vee$I
20.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	4,5-19 $\vee$E

Indct: For any n, Q $\vdash_{ND+}$ $j < \overline{\mathsf{n}} \vee j = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < j$.

Again, the "three-part" $\vee$E from the show is (clear but) not strictly according to the rule. And since for any n, Q $\vdash_{ND+}$ $j < \overline{\mathsf{n}} \vee j = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < j$, by $\forall$I, for any n, Q $\vdash_{ND+}$ $\forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$.

Thus the trichotomy result is established.

E8.41. Complete the demonstration of T8.23 by showing part (i) of T8.21, T8.22, and then T8.23.

T8.22(ii)

1.	$\overline{\mathsf{n}} < j$	A (g, $\to$I)
2.	$\exists u(Su + \overline{\mathsf{n}} = j)$	1 abv
3.	$Sk + \overline{\mathsf{n}} = j$	A (g, 2$\exists$E)
4.	$k = \emptyset \vee \exists y(k = Sy)$	from T6.52
5.	$k = \emptyset$	A (g, 4$\vee$E)
6.	$S\emptyset + \overline{\mathsf{n}} = j$	3,5 =E
7.	$S\emptyset + \overline{\mathsf{n}} = S\overline{\mathsf{n}}$	T8.12 ($1 + \mathsf{n} = \mathsf{Sn}$)
8.	$j = S\overline{\mathsf{n}}$	7,6 =E
9.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	8 $\vee$I
10.	$\exists y(k = Sy)$	A (g, 4$\vee$E)
11.	$k = Sl$	A (g, 10$\exists$E)
12.	$Sk + \overline{\mathsf{n}} = k + S\overline{\mathsf{n}}$	T8.15
13.	$k + S\overline{\mathsf{n}} = j$	3,12 =E
14.	$Sl + S\overline{\mathsf{n}} = j$	13,11 =E
15.	$\exists u(Su + S\overline{\mathsf{n}} = j)$	14 $\exists$I
16.	$S\overline{\mathsf{n}} < j$	15 abv
17.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	16 $\vee$I
18.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	10,11-17 $\exists$E
19.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	4,5-9,10-18 $\vee$E
20.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	2,3-19 $\exists$E
21.	$\overline{\mathsf{n}} < j \to (j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j)$	1-20 $\to$I
22.	$\forall x[\overline{\mathsf{n}} < x \to (x = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < x)]$	21 $\forall$I

From T6.52, either k is zero or it is not. If k is zero, it is a simple addition problem to show that $j = S\overline{\mathsf{n}}$ and so obtain the desired result. If k is a successor, then $S\overline{\mathsf{n}} < j$ and again we have the desired result.

8.4.6 Case

Finally, a couple of quick theorems that move from the provability of particular instances to the provability of bounded quantifications.

T8.24. For any n and formula $\mathcal{P}(x)$,

(i) if $Q \vdash_{ND+} \mathcal{P}(\overline{0})$ or ... or $Q \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}})$, then $Q \vdash_{ND+} (\exists x \leq \overline{\mathsf{n}})\mathcal{P}(x)$

(ii) if $0 \neq 0$ or $Q \vdash_{ND+} \mathcal{P}(\overline{0})$ or ... or $Q \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}-1})$, then $Q \vdash_{ND+} (\exists x < \overline{\mathsf{n}})\mathcal{P}(x)$.

In the second case, again, we include the first disjunct to keep the conditional defined in the case when $\mathsf{n} = 0$; then the conditional obtains because the antecedent does not. This theorem is nearly trivial: (i) For some $\mathsf{m} \leq \mathsf{n}$ suppose $Q \vdash_{ND+} \mathcal{P}(\overline{\mathsf{m}})$; by T8.14, $Q \vdash_{ND+} \overline{\mathsf{m}} \leq \overline{\mathsf{n}}$; so by ($\exists$I), $Q \vdash_{ND+} (\exists x \leq \overline{\mathsf{n}})\mathcal{P}(x)$. Similarly for (ii).

So if $\mathcal{P}$ is provable for some individual $\leq \mathsf{n}$ or $< \mathsf{n}$ then it is immediate that the corresponding bounded existential generalization is provable.

*T8.25. For any n and formula $\mathcal{P}(x)$,

(i) if $Q \vdash_{ND+} \mathcal{P}(\overline{0})$ and ... and $Q \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}})$, then $Q \vdash_{ND+} (\forall x \leq \overline{\mathsf{n}})\mathcal{P}(x)$

(ii) if $0 = 0$ and $Q \vdash_{ND+} \mathcal{P}(\overline{0})$ and ... and $Q \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}-1})$, then $Q \vdash_{ND+} (\forall x < \overline{\mathsf{n}})\mathcal{P}(x)$.

This time in the second case we include a trivial truth in order to keep the conditional defined when $\mathsf{n} = 0$; when $\mathsf{n} = 0$, then the antecedent is trivially true, but the consequent follows from nothing. The argument is by induction on the value of n.

So if Q proves $\mathcal{P}$ for each individual $\leq \mathsf{n}$ or $< \mathsf{n}$ then Q proves the corresponding bounded universal generalization.

*E8.42. Provide arguments to show both parts of T8.25.

E8.43. For each of the following concepts, explain in an essay of about two pages, so that (high school age) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The use of the inductive assumption in an argument from mathematical induction.

b. The reason mathematical induction works as a deductive argument form.

Final Theorems of Chapter 8

T8.11 For any $\mathsf{n} \in \mathsf{U}$ and assignment d, $\mathsf{N_d}[\overline{\mathsf{n}}] = \mathsf{n}$.

T8.12 For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} + \mathsf{b} = \mathsf{c}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$. Corollary: if $\mathsf{a} + 1 = \mathsf{b}$ then $\mathrm{Q} \vdash_{ND+} S\overline{\mathsf{a}} = \overline{\mathsf{b}}$.

T8.13 For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} \times \mathsf{b} = \mathsf{c}$ then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \times \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

T8.14 For any $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, (i) if $\mathsf{a} = \mathsf{b}$ then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} = \overline{\mathsf{b}}$; (ii) if $\mathsf{a} \leq \mathsf{b}$ then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \leq \overline{\mathsf{b}}$; and (iii) if $\mathsf{a} < \mathsf{b}$ then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} < \overline{\mathsf{b}}$.

T8.15 $\mathrm{Q} \vdash_{ND+} Sj + \overline{\mathsf{n}} = j + S\overline{\mathsf{n}}$.

T8.16 For any $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, (i) if $\mathsf{a} \neq \mathsf{b}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$; (ii) if $\mathsf{a} \not\leq \mathsf{b}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$; and (iii) if $\mathsf{a} \not< \mathsf{b}$, then $\mathrm{Q} \vdash_{ND+} \overline{\mathsf{a}} \not< \overline{\mathsf{b}}$.

T8.17 For any variable-free term t of $\mathscr{L}_{NT}$, if $\mathsf{N}[t] = \mathsf{n}$, then $\mathrm{Q} \vdash_{ND+} t = \overline{\mathsf{n}}$.

T8.18 Q correctly decides atomic sentences of $\mathscr{L}_{NT}$: For any sentence $\mathcal{P}$ of the sort $s = t$, $s \leq t$, or $s < t$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash_{ND+} \mathcal{P}$; and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash_{ND+} \sim\mathcal{P}$.

T8.19 For any n, $\vdash_{ND+} \forall x \forall y(x = Sy \to [(y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{n}}) \to (x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{n}})])$.

T8.20 For any n, (i) $\mathrm{Q} \vdash_{ND+} (\forall x \leq \overline{\mathsf{n}})(x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n}})$ and (ii) $\mathrm{Q} \vdash_{ND+} (\forall x < \overline{\mathsf{n}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n}-1})$.

T8.21 For any n, (i) $\mathrm{Q} \vdash_{ND+} \forall x([x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{n}}] \to x \leq \overline{\mathsf{n}})$ and (ii) $\mathrm{Q} \vdash_{ND+} \forall x([\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{n}-1}] \to x < \overline{\mathsf{n}})$.

T8.22 For any n, (i) $\mathrm{Q} \vdash_{ND+} \forall x[\overline{\mathsf{n}} \leq x \to (\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} \leq x)]$ and (ii) $\mathrm{Q} \vdash_{ND+} \forall x[\overline{\mathsf{n}} < x \to (S\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} < x)]$.

T8.23 For any n, (i) $\mathrm{Q} \vdash_{ND+} \forall x(x \leq \overline{\mathsf{n}} \vee \overline{\mathsf{n}} \leq x)$ and (ii) $\mathrm{Q} \vdash_{ND+} \forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$.

T8.24 For any n and formula $\mathcal{P}(x)$,

(i) if $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{0})$ or $\ldots$ or $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}})$, then $\mathrm{Q} \vdash_{ND+} (\exists x \leq \overline{\mathsf{n}})\mathcal{P}(x)$

(ii) if $0 \neq 0$ or $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{0})$ or $\ldots$ or $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}-1})$, then $\mathrm{Q} \vdash_{ND+} (\exists x < \overline{\mathsf{n}})\mathcal{P}(x)$.

T8.25 For any n and formula $\mathcal{P}(x)$,

(i) if $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{0})$ and $\ldots$ and $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}})$, then $\mathrm{Q} \vdash_{ND+} (\forall x \leq \overline{\mathsf{n}})\mathcal{P}(x)$

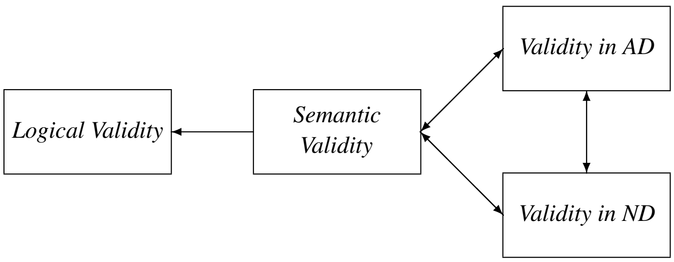
(ii) if $0 = 0$ and $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{0})$ and $\ldots$ and $\mathrm{Q} \vdash_{ND+} \mathcal{P}(\overline{\mathsf{n}-1})$, then $\mathrm{Q} \vdash_{ND+} (\forall x < \overline{\mathsf{n}})\mathcal{P}(x)$.

Part III

Classical Metalogic: S̊oundness and C̊ompleteness

Introductory

In Part I we introduced four notions of validity. In this part, we set out to show that they are interrelated as follows:



An argument is valid in *AD* iff it is valid in *ND*. And an argument is semantically valid iff it is valid in the derivation systems. So the three formal notions apply to exactly the same arguments. And if an argument is semantically valid, then it is logically valid. So any of the formal notions imply logical validity for a corresponding ordinary argument.

More carefully, in Part I, we introduced four main notions of validity. There is logical validity from Chapter 1, semantic validity from Chapter 4, and syntactic validity in the derivation systems *AD* from Chapter 3 and *ND* from Chapter 6. These notions are *independently defined*. Thus it is not immediate or obvious how they are related. We turn in this part to the task of thinking *about* these notions, and especially about how they are related. The primary result is that $\Gamma \vDash \mathscr{P}$ iff $\Gamma \vdash_{AD} \mathscr{P}$ iff $\Gamma \vdash_{ND} \mathscr{P}$ (iff $\Gamma \vdash_{ND+} \mathscr{P}$). Thus our different formal notions of validity are met by just the same arguments, and the derivation systems—defined in terms of *form*, are "faithful" to the semantic notion—defined in terms of truth. What is derivable is neither more nor less than what is semantically valid. And this is just right: If what is derivable were more than what is semantically valid, derivations could lead us from true premises to false conclusions; if it were less, not all semantically valid arguments could be identified as such by derivations. That the derivable is no *more* than what is semantically valid is *soundness* of a derivation system; that it is no *less* is *completeness*. In addition, we show that if an argument is semantically valid, then a corresponding ordinary argument is *logically valid*. Given the equivalence between the formal notions of validity, it follows that if an argument is valid in any of the formal senses, then it is logically valid. This connects the formal machinery to the notion of validity with which we began.

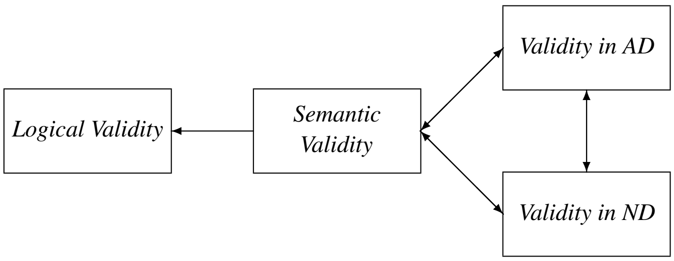
Notions of *soundness* and *completeness* appear in a variety of contexts. We have seen *sound* arguments from Chapter 1; in this part we have *sound* derivation systems, and encounter *sound* theories. Similarly, in this part we have *complete* derivation systems and shall encounter *complete* (and incomplete) theories. These notions of soundness and completeness are separately defined, and apply to different objects. This invites confusion! One option is to introduce new vocabulary. But the weight of tradition is strong. Also, in section 11.4.1 we exhibit a notion of *relative* soundness such that both the soundness of derivation systems and the soundness of theories are instances of it. And similarly, there is a *relative* completeness such that both the completeness of derivation systems and the completeness of theories are instances of it. So the different notions appear as separate instances of more general kinds. In order to indicate distinctness at the same time as we (honor tradition and) acknowledge underlying conceptual connections, I introduce a (silent) diacritical mark for each—identifying the notions with application to derivation systems with an enlarged dot, (s̊oundness, c̊ompleteness) and ones whose application is to theories with a tilde (s̃oundness, c̃ompleteness).

We begin in Chapter 9 showing that just the same arguments are valid in the derivation systems *ND* and *AD*. This puts us in a position to demonstrate in Chapter 10 the core result that the derivation systems are both s̊ound and c̊omplete—that if some premises prove $\mathscr{P}$, then those premises entail $\mathscr{P}$, and if some premises entail $\mathscr{P}$ then those premises prove $\mathscr{P}$. Chapter 11 fills out this core picture in different directions. It begins with short sections on *expressive completeness*, *unique readability*, and *independence* (these do not depend on one another or on chapters 9 or 10 and so might be worked any time). The chapter concludes with a longer section on beginning *model theory*; this builds upon the Chapter 10 discussion of relations between interpretations and formal expressions, and concludes with the identification of some c̃omplete theories—to contrast with ĩncomplete theories from Part IV.

Chapter 9

Preliminary Results

We have said that the aim of this part is to establish the following relations: An argument is valid in *AD* iff it is valid in *ND;* an argument is semantically valid iff it is valid in *AD*, and iff it is valid in *ND;* and if an argument is semantically valid, then it is logically valid.



In this chapter, we begin to develop these relations, taking up some of the simpler cases. We consider the leftmost horizontal arrow, and the rightmost vertical ones. Thus we show that quantificational (semantic) validity implies logical validity (section 9.1), that validity in *AD* implies validity in *ND* (section 9.2), that validity in *ND* implies validity in *AD* (section 9.3), and extend the results to *ND+* (section 9.4). Implications between semantic validity and the syntactical notions will wait for Chapter 10.

9.1 Semantic Validity Implies Logical Validity

Logical validity is defined for arguments in ordinary language. From LV, an argument is logically valid iff there is no consistent *story* in which all the premises are true and the conclusion is false. Quantificational validity is defined for arguments in a formal language. From QV, an argument is quantificationally valid iff there is no *interpretation* on which all the premises are true and the conclusion is not. So our task is to show how facts about formal expressions and interpretations connect with ordinary expressions and stories. In particular, where $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is an ordinary-language argument, and $\mathcal{P}'_1 \ldots \mathcal{P}'_n$, $\mathcal{Q}'$ are the formulas of a good translation, we show that if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$, then the ordinary argument $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is logically valid. The reasoning itself is straightforward. We will spend a bit more time discussing the result.

Recall our criterion of goodness for translation CG from Chapter 5 (page 135). When we identify a (sentential or quantificational) interpretation function II, we thereby identify an *intended interpretation* II_ω corresponding to any way ω that the world can be. For example, corresponding to the interpretation function,

II B: Barack is happy

M: Michelle is happy

$\mathsf{II}_\omega[B] = \mathsf{T}$ just in case Barack is happy at ω, and similarly for M. And a formal translation $\mathcal{A}'$ of some ordinary $\mathcal{A}$ is *good* only if for any ω, $\mathsf{II}_\omega[\mathcal{A}']$ has the same truth value as $\mathcal{A}$ at ω. Given this, we can show,

T9.1. For any ordinary argument $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$, with good translation consisting of II and $\mathcal{P}'_1 \ldots \mathcal{P}'_n$, $\mathcal{Q}'$, if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$, then $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is logically valid.

Consider an ordinary $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ and good translation consisting of II and $\mathcal{P}'_1 \ldots \mathcal{P}'_n$, $\mathcal{Q}'$. Suppose $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ but $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is not logically valid. From the latter, by LV, there is some consistent story ω where each of $\mathcal{P}_1 \ldots \mathcal{P}_n$ is true but $\mathcal{Q}$ is false; and since ω is consistent and $\mathcal{Q}$ is false, $\mathcal{Q}$ is not true at ω. Since $\mathcal{P}_1 \ldots \mathcal{P}_n$ are true at ω, by CG, $\mathsf{II}_\omega[\mathcal{P}'_1] = \mathsf{T}$ and … and $\mathsf{II}_\omega[\mathcal{P}'_n] = \mathsf{T}$; and since $\mathcal{Q}$ is not true at ω; by CG, $\mathsf{II}_\omega[\mathcal{Q}'] \neq \mathsf{T}$. So II_ω is an interpretation I that has each of $\mathsf{I}[\mathcal{P}'_1] = \mathsf{T}$ and … and $\mathsf{I}[\mathcal{P}'_n] = \mathsf{T}$, and $\mathsf{I}[\mathcal{Q}'] \neq \mathsf{T}$; so by QV, $\mathcal{P}'_1 \ldots \mathcal{P}'_n \nvDash \mathcal{Q}'$. This is impossible; reject the assumption: if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ then $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is logically valid.

It is that easy. If there is no interpretation where $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ are true but $\mathcal{Q}'$ is not, then there is no *intended* interpretation where $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ are true but $\mathcal{Q}'$ is not; so, by CG, there is no consistent story where the premises are true and the conclusion is not; so $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is logically valid. So if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ then $\mathcal{P}_1 \ldots \mathcal{P}_n/\mathcal{Q}$ is logically valid. This is good! It shows that we can apply our formal machinery to the arguments with which we began.

Let us make a couple of observations: First, CG is stronger than is actually required for our application of semantic validity to logical validity. CG requires a biconditional for good translation.

$$\mathcal{A} \text{ is true at } \omega \quad \Longleftrightarrow \quad \mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$$

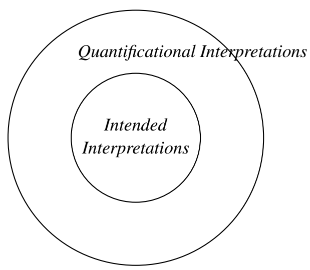
But our reasoning applies to premises just the left-to-right portion of this condition: if $\mathcal{P}$ is true at ω then $\mathsf{II}_\omega[\mathcal{P}'] = \mathsf{T}$. And for the conclusion, the reasoning goes in the opposite direction: if $\mathsf{II}_\omega[\mathcal{Q}'] = \mathsf{T}$ then $\mathcal{Q}$ is true at ω (so that if $\mathcal{Q}$ is not true at ω, then $\mathsf{II}_\omega[\mathcal{Q}'] \neq \mathsf{T}$). The biconditional from CG guarantees both. But, strictly, for premises, all we need is that truth of an ordinary expression at a story guarantees truth for the corresponding formal one at the intended interpretation. And for a conclusion, all we need is that truth of the formal expression on the intended interpretation guarantees truth of the corresponding ordinary expression at the story.

Thus we might use our methods to identify logical validity even where translations are less than completely good. Consider, for example, the following argument:

(A) Bob took a shower and got dressed
      ———————
      Bob took a shower

As discussed in Chapter 5 (page 152), where II gives S the same value as ‘Bob took a shower’ and D the same as ‘Bob got dressed’, we might agree that there are cases where $\mathsf{II}_\omega[S \wedge D] = \mathsf{T}$ but ‘Bob took a shower and got dressed’ is false. So we might agree that the right-to-left conditional is false, and the translation is not good. However, even if this is so, given our interpretation function, there is no situation where ‘Bob took a shower and got dressed’ is true but $S \wedge D$ is F at the corresponding intended interpretation. So the left-to-right conditional is sustained. So even if the translation is not good by CG, it remains possible to use our methods to demonstrate logical validity. Since it remains that if the ordinary premise is true at a story then the formal expression is true at the corresponding intended interpretation, semantic validity implies logical validity. A similar point applies to conclusions. Of course, we already knew that this argument is logically valid. But the point applies to more complex arguments as well.

Second, observe that our reasoning does not work in reverse. Logical validity for $\mathcal{P}_1 \dots \mathcal{P}_n/\mathcal{Q}$ does not imply $\mathcal{P}'_1 \dots \mathcal{P}'_n \vDash \mathcal{Q}'$. Or, put the other way around, finding a quantificational interpretation where $\mathcal{P}'_1 \dots \mathcal{P}'_n$ are true and $\mathcal{Q}'$ is not shows that $\mathcal{P}'_1 \dots \mathcal{P}'_n \nvDash \mathcal{Q}'$; however it does not show that $\mathcal{P}_1 \dots \mathcal{P}_n/\mathcal{Q}$ is not logically valid. Here is why: There may be quantificational interpretations which do not correspond to any consistent story. The situation is like this:



Intended interpretations are a subset of all interpretations. The intended interpretations correspond to consistent stories. If no interpretation whatsoever has the premises true and the conclusion not, then no intended interpretation has the premises true and conclusion not, so no consistent story makes the premises true and the conclusion not. But it may be that some unintended interpretation makes the premises true and conclusion false. Thus $\mathcal{P}'_1 \dots \mathcal{P}'_n \not\models \mathcal{Q}'$, requires that there is an interpretation with the premises true and conclusion not; but it does not require that there is an intended interpretation where the premises are true and the conclusion is not; so it does not require that there is a consistent story where $\mathcal{P}_1 \dots \mathcal{P}_n$ are true but $\mathcal{Q}$ is not; so it does not show that $\mathcal{P}_1 \dots \mathcal{P}_n / \mathcal{Q}$ is invalid.

It is easy to see why there might be unintended interpretations. Consider, first, this standard argument:

(B)
All humans are mortal
Socrates is human
———
Socrates is mortal

It is logically valid. But consider what happens when we translate into a *sentential* language. We might try an interpretation function as follows:

A: All humans are mortal

H: Socrates is human

M: Socrates is mortal

with translation, A, H/M. But, of course, there is an interpretation (row of the truth table) J on which A and H are T and M is F. So the argument is not sententially valid. Given the interpretation function, J would correspond to a story where all humans are mortal, Socrates is human, and Socrates is not mortal—but such a story is inconsistent! So interpretation J is (unintended and) not sufficient to show that the argument is logically invalid. The interpretation function matches every consistent story to an interpretation; but this leaves open that there are interpretations not matched to consistent stories. Sentential languages are sufficient to identify validity when validity results from truth functional structure; this argument is valid, but not valid because of truth functional structure.

We are in a position to expose its validity only in the quantificational case. Thus we might have,

s: Socrates

H^1: $\{\mathsf{o} \mid \mathsf{o} \text{ is human}\}$

M^1: $\{\mathsf{o} \mid \mathsf{o} \text{ is mortal}\}$

with translation $\forall x(Hx \to Mx)$, Hs/Ms. The argument is quantificationally valid. And, as above, it follows that the ordinary one is logically valid.

But related problems may arise even for quantificational languages. Thus consider,

(C) Socrates is necessarily human
$\overline{\text{Socrates is human}}$

Again, the argument is logically valid—if Socrates is human according to every consistent story, then Socrates is human according to the real story. But now, with a quantificational language, we end up with something like an additional relation symbol N^1 for $\{o \mid o \text{ is necessarily human}\}$, and translation Ns/Hs. And this is not quantificationally valid. Consider, for example, an interpretation with $\mathsf{U} = \{1\}$, $\mathsf{I}[s] = 1$, $\mathsf{I}[N] = \{1\}$, and $\mathsf{I}[H] = \{\,\}$. Then the premise is true, but the conclusion is not. Again, the problem is that the argument includes structure that our quantificational language fails to capture. As it turns out, *modal* logic is precisely an attempt to work with structure introduced by notions of possibility and necessity. Where '$\Box$' represents necessity, this argument, with translation $\Box Hs/Hs$ is valid on standard modal systems.[1]

The upshot of this discussion is that our methods are adequate when they work to identify validity. When an argument is quantificationally valid, we can be sure that it is logically valid. But not the converse. Thus quantificational invalidity does not imply logical invalidity. We should not be discouraged by this or somehow put off the logical project. Rather, we have a rationale for *expanding* the logical project. In Part I, we set up formal logic as a "tool" or "machine" to identify logical validity. Beginning with the notion of logical validity, we introduce our formal languages, learn to translate into them, and to manipulate arguments by semantical and syntactical methods. The sentential notions have some utility. But when it turns out that sentential languages miss important structure, we expand the language to include quantificational structure, developing the semantical and syntactical methods to match. And similarly, if our quantificational languages should turn out to miss important structure, we expand the language to capture that structure, and further develop the semantical and syntactical methods. As it happens, the classical quantificational logic we have seen so far is sufficient to identify validity in a wide variety of contexts—and, in particular, for arguments in mathematics. Also, controversy may be introduced as one expands beyond the classical quantificational level. So the logical project is a live one. But let us return to the kinds of validity we have already seen.

E9.1. (i) Recast the above reasoning to show directly a corollary to T9.1: If $\vDash \mathcal{Q}'$, then $\mathcal{Q}$ is necessarily true (that is, there is no consistent story where it is false). (ii) Suppose $\nvDash \mathcal{Q}'$; does it follow that $\mathcal{Q}$ is not necessarily true? Explain.

[1] Modal logics are introduced in Priest, *Non-Classical Logics*. His book is profitably read together with Roy, "Natural Derivations for Priest."

*E9.2. In Chapter 5 (page 148) we informally suggest inductive reasoning to show that our translation procedure (TP) gives the right results. Make this rigorous. That is, for an ordinary $\mathscr{P}$ and good formal translation I and $\mathscr{P}'$, show by induction on the number of ordinary truth functional operators (branching in the parse tree for $\mathscr{P}$) that $\mathscr{P}$ is true at world ω iff $\mathsf{I}_\omega[\mathscr{P}'] = \mathrm{T}$. Hint: When $\mathscr{P}$ has k operators, for some ordinary operator Op and equivalent formal expression Op', $\mathscr{P}$ is of the sort $Op(\mathscr{A}_1 \ldots \mathscr{A}_n)$ and $\mathscr{P}'$ is $Op'(\mathscr{A}'_1 \ldots \mathscr{A}'_n)$ for $\mathscr{A}_1 \ldots \mathscr{A}_n$ with $< k$ operators.

9.2 Validity in *AD* Implies Validity in *ND*

It is easy to see that if $\Gamma \vdash_{AD} \mathscr{P}$, then $\Gamma \vdash_{ND} \mathscr{P}$. Roughly, anything we can accomplish in *AD* we can accomplish in *ND* as well. If a premise appears in an *AD* derivation, that same premise can be used in *ND*. If an axiom appears in an *AD* derivation, that axiom can be derived in *ND*. And if a line is justified by MP or Gen in *AD*, that same line may be justified by rules of *ND*. So anything that can be derived in *AD* can be derived in *ND*. Officially, this reasoning is by induction on the line numbers of an *AD* derivation, and it is appropriate to work out the details more formally. The argument by mathematical induction is longer than anything we have seen so far, but the reasoning is straightforward.

*T9.2. If $\Gamma \vdash_{AD} \mathscr{P}$, then $\Gamma \vdash_{ND} \mathscr{P}$.

Suppose $\Gamma \vdash_{AD} \mathscr{P}$. Then there is an *AD* derivation $A = \langle \mathscr{Q}_1 \ldots \mathscr{Q}_n \rangle$ of $\mathscr{P}$ from premises in Γ, with $\mathscr{Q}_n = \mathscr{P}$. We show that there is a corresponding *ND* derivation N, such that if $\mathscr{Q}_i$ appears on line i of A, then $\mathscr{Q}_i$ appears, under the scope of the premises alone, on the line numbered i of N. It follows that $\Gamma \vdash_{ND} \mathscr{P}$. For any *premises* $\mathscr{Q}_a, \mathscr{Q}_b, \ldots, \mathscr{Q}_j$ in A, let N begin,

0.a	$\mathscr{Q}_a$	P
0.b	$\mathscr{Q}_b$	P
$\vdots$		
0.j	$\mathscr{Q}_j$	P

Now we reason by induction on the line numbers in A. The general plan is to *construct* a derivation N which accomplishes just what is accomplished in A. Fractional line numbers, as above, maintain the parallel between the two derivations.

Basis: $\mathscr{Q}_1$ in A is a premise or an instance of A1, A2, A3, A4, A5, A6, A7, or A8.

(prem) If $\mathscr{Q}_1$ is a premise $\mathscr{Q}_i$, continue N as follows:

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P
1	$\mathcal{Q}_i$	0.i R

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered 1 of N.

(A1) If $\mathcal{Q}_1$ is an instance of A1, then it is of the form, $\mathcal{B} \to (\mathcal{C} \to \mathcal{B})$, and we continue N as follows:

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P
1.1	$\quad \mathcal{B}$	A (g, $\to$I)
1.2	$\quad\quad \mathcal{C}$	A (g, $\to$I)
1.3	$\quad\quad \mathcal{B}$	1.1 R
1.4	$\quad \mathcal{C} \to \mathcal{B}$	1.2-1.3 $\to$I
1	$\mathcal{B} \to (\mathcal{C} \to \mathcal{B})$	1.1-1.4 $\to$I

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered 1 of N.

(A2) If $\mathcal{Q}_1$ is an instance of A2, then it is of the form, $(\mathcal{B} \to (\mathcal{C} \to \mathcal{D})) \to ((\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D}))$ and we continue N as follows:

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P
1.1	$\quad \mathcal{B} \to (\mathcal{C} \to \mathcal{D})$	A (g, $\to$I)
1.2	$\quad\quad \mathcal{B} \to \mathcal{C}$	A (g, $\to$I)
1.3	$\quad\quad\quad \mathcal{B}$	A (g, $\to$I)
1.4	$\quad\quad\quad \mathcal{C}$	1.2, 1.3 $\to$E
1.5	$\quad\quad\quad \mathcal{C} \to \mathcal{D}$	1.1, 1.3 $\to$E
1.6	$\quad\quad\quad \mathcal{D}$	1.5, 1.4 $\to$E
1.7	$\quad\quad \mathcal{B} \to \mathcal{D}$	1.3-1.6 $\to$I
1.8	$\quad (\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D})$	1.2-1.7 $\to$I
1	$(\mathcal{B} \to (\mathcal{C} \to \mathcal{D})) \to ((\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D}))$	1.1-1.8 $\to$I

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered 1 of N.

(A3) Homework.

(A4) Homework.

(A5) If $\mathcal{Q}_1$ is an instance of A5, then it is of the form $\forall x(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P} \rightarrow \forall x\mathcal{Q})$ for some variable x that is not free in $\mathcal{P}$, and we continue N as follows:

$$
\begin{array}{r|l|l}
0.a & \mathcal{Q}_a & \text{P} \\
0.b & \mathcal{Q}_b & \text{P} \\
\vdots & & \\
0.j & \mathcal{Q}_j & \text{P} \\
1.1 & \quad \underline{\forall x(\mathcal{P} \rightarrow \mathcal{Q})} & \text{A } (g, \rightarrow\text{I}) \\
1.2 & \quad\quad \underline{\mathcal{P}} & \text{A } (g, \rightarrow\text{I}) \\
1.3 & \quad\quad \mathcal{P} \rightarrow \mathcal{Q} & 1.1\ \forall\text{E} \\
1.4 & \quad\quad \mathcal{Q} & 1.3, 1.2 \rightarrow\text{E} \\
1.5 & \quad\quad \forall x\mathcal{Q} & 1.4\ \forall\text{I} \\
1.6 & \quad \mathcal{P} \rightarrow \forall x\mathcal{Q} & 1.2\text{-}1.5 \rightarrow\text{I} \\
1 & \forall x(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P} \rightarrow \forall x\mathcal{Q}) & 1.1\text{-}1.6 \rightarrow\text{I}
\end{array}
$$

x is sure to be free for x in $\mathcal{P} \rightarrow \mathcal{Q}$; so (1.3) meets the constraint on $\forall$E. In addition, x is sure to be free for x in $\mathcal{Q}$ and x is not free in $\forall x\mathcal{Q}$; further x is not free in $\forall x(\mathcal{P} \rightarrow \mathcal{Q})$ and we are given that x is not free in $\mathcal{P}$, so x is not free in any undischarged assumption; so the restrictions are met for $\forall$I at (1.5). So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered 1 of N.

(A6) Homework.

(A7) If $\mathcal{Q}_1$ is an instance of A7, then it is $(x_i = y) \rightarrow (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$ for some variables $x_1 \ldots x_n$ and y and function symbol h^n, and we continue N as follows:

$$
\begin{array}{r|l|l}
0.a & \mathcal{Q}_a & \text{P} \\
0.b & \mathcal{Q}_b & \text{P} \\
\vdots & & \\
0.j & \mathcal{Q}_j & \text{P} \\
1.1 & \quad \underline{x_i = y} & \text{A } (g, \rightarrow\text{I}) \\
1.2 & \quad h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots x_i \ldots x_n & =\text{I} \\
1.3 & \quad h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n & 1.2, 1.1 =\text{E} \\
1 & (x_i = y) \rightarrow (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n) & 1.1\text{-}1.3 \rightarrow\text{I}
\end{array}
$$

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered 1 of N.

(A8) Homework.

Assp: For any i, $1 \le i < k$, if $\mathcal{Q}_i$ appears on line i of A, then $\mathcal{Q}_i$ appears, under the scope of the premises alone, on the line numbered i of N.

Show: If $\mathcal{Q}_k$ appears on line k of A, then $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered k of N.

$\mathcal{Q}_k$ in A is a premise, an axiom, or arises from previous lines by MP or Gen. If $\mathcal{Q}_k$ is a premise or an axiom then, by reasoning as in the basis (with line numbers adjusted to $k.n$) if $\mathcal{Q}_k$ appears on line k of A, then $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered k of N. So suppose $\mathcal{Q}_k$ arises by MP or Gen.

(MP) If $\mathcal{Q}_k$ arises from previous lines by MP, then A is as follows:

$$\begin{array}{lll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ & \vdots & \\ j & \mathcal{B} & \\ & \vdots & \\ k & \mathcal{C} & i, j \text{ MP} \end{array}$$

where $i, j < k$ and $\mathcal{Q}_k$ is $\mathcal{C}$. By assumption, then, there are lines in N,

$$\begin{array}{l|l} i & \mathcal{B} \rightarrow \mathcal{C} \\ & \vdots \\ j & \mathcal{B} \end{array}$$

So we simply continue derivation N:

$$\begin{array}{l|ll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ & \vdots & \\ j & \mathcal{B} & \\ & \vdots & \\ k & \mathcal{C} & i, j \rightarrow\text{E} \end{array}$$

So $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered k of N.

(Gen) If $\mathcal{Q}_k$ arises from previous lines by Gen, then A is as follows:

$$\begin{array}{lll} i & \mathcal{B} & \\ & \vdots & \\ k & \forall x \mathcal{B} & i \text{ Gen} \end{array}$$

where $i < k$, and $\mathcal{Q}_k$ is $\forall x \mathcal{B}$. By assumption N has a line i,

$$\begin{array}{l|l} & \vdots \\ i & \mathcal{B} \\ & \vdots \end{array}$$

under the scope of the premises alone. So we continue N as follows:

$$\begin{array}{l|ll} i & \mathcal{B} & \\ & \vdots & \\ k & \forall x \mathcal{B} & i\ \forall\text{I} \end{array}$$

Since i and k are under the scope of the premises alone, x is not free in an undischarged assumption. Further, since there is no change of variables, we can be sure that x is free for every free instance of x in $\mathcal{B}$, and that x is not free in $\forall x \mathcal{B}$. So the restrictions are met on $\forall$I. So $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered k of N.

In any case then, $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered k of N.

Indct: For any line j of A, $\mathcal{Q}_j$ appears, under the scope of the premises alone, on the line numbered j of N.

So $\Gamma \vdash_{ND} \mathcal{Q}_n$, where this is just to say $\Gamma \vdash_{ND} \mathcal{P}$. So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

Notice the way we use line numbers, $i.1$, $i.2$, $\ldots$, $i.n$, i in N to make good on the claim that for each $\mathcal{Q}_i$ in A, $\mathcal{Q}_i$ appears on the line numbered i of N—where the line numbered i may or may not be the i^{th} line of N. We need this parallel between the line numbers when it comes to cases for MP and Gen. With the parallel, we are in a position to use line numbers from justifications in derivation A for the specification of derivation N.

Given an *AD* derivation, what we have done shows that there exists an *ND* derivation by showing how to construct it. We can see how this works by considering an application. Thus, for example, consider the following derivation of T3.2:

	1.	$\mathcal{B} \to \mathcal{C}$	prem
	2.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
	3.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2,1 MP
(D)	4.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
	5.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	4,3 MP
	6.	$\mathcal{A} \to \mathcal{B}$	prem
	7.	$\mathcal{A} \to \mathcal{C}$	5,6 MP

Let this be derivation A; we will follow the method of our induction to construct a corresponding *ND* derivation N. The first step is to list the premises. Premises appear on lines (1) and (6) and we begin,

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.6	$\mathcal{A} \to \mathcal{B}$	P

Now to the induction itself. The first line of A is a premise. Looking back to the basis case of the induction, we see that we are instructed to produce the line numbered 1 by reiteration. So that is what we do:

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.6	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R

This may strike you as somewhat pointless! But, again, we need $\mathcal{B} \to \mathcal{C}$ on the line numbered 1 in order to maintain the parallel between the derivations. So our recipe requires this simple step.

Numeral and Number

Numerals designate *numbers*. Thus the numeral (symbol) '1' designates the number (object) 1. Set aside (for philosophy of mathematics) the question what sort of object numbers are supposed to be. Whatever numbers turn out to be, one often encounters an ambiguity between numbers and the numerals that designate them. This does not usually lead to trouble. So, for example, in this text we have said that sentence letters of $\mathcal{L}_s$ are "Roman italics $A \dots Z$ with or without positive integer subscripts"; but sentence letters are *symbols* and their subscripts are *numerals*—integers are not even candidates for the subscript role. And in a mathematical induction we might say, "for any i such that $0 \leq i < k$, $\mathcal{Q}_i$ has such-and-such feature"; then $\mathcal{Q}_i$ is a symbol with a numeral subscript, but i and k of the inequality are numbers—*less than* is a relation on *numbers* not *numerals*.

A derivation is a syntactical object, with numerals to mark the lines. Thus when we refer to a line i of some derivation, there are different options: (i) i is a metavariable mapping to a numeral marking the line—this is the natural understanding of, say, schematic descriptions of the derivation rules in Chapter 6. (ii) Variable i is assigned a number, and thereby associated with the numeral $\langle i \rangle$ that marks the line—this is the natural understanding of the induction where i is a number, and we reason that $\mathcal{Q}_i$ has such-and-such feature. (iii) Another option is that i is assigned a number to identify the i^{th} line of a derivation. In ordinary cases, the i^{th} line is the same as the one marked $\langle i \rangle$.

In the current discussion, the natural understanding is (ii)—this is so especially because the i^{th} line may be other than the one marked $\langle i \rangle$. Without wholesale application of the bracket notation, where it is important to make this point I say the line is "numbered" i.

Line 2 of A is an instance of A1, and the induction therefore tells us to get it "by reasoning as in the basis." Looking then to the case for A1 in the basis, we continue on that pattern as follows:

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.6	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R
2.1	$\quad \mathcal{B} \to \mathcal{C}$	A (g, $\to$I)
2.2	$\quad\quad \mathcal{A}$	A (g, $\to$I)
2.3	$\quad\quad \mathcal{B} \to \mathcal{C}$	2.1 R
2.4	$\quad \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2-2.3 $\to$I
2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	2.1-2.4 $\to$I

Notice that this reasoning for the show step now applies to line 2, so that the line numbers are 2.1, 2.2, 2.3, 2.4, 2 instead of 1.1, 1.2, 1.3, 1.4, 1 as for the basis. Also, what we have added follows *exactly* the pattern from the recipe in the induction, given the relevant instance of A1.

Line 3 is justified by 2,1 MP. Again, by the recipe from the induction, we continue,

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.6	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R
2.1	$\quad \mathcal{B} \to \mathcal{C}$	A (g, $\to$I)
2.2	$\quad\quad \mathcal{A}$	A (g, $\to$I)
2.3	$\quad\quad \mathcal{B} \to \mathcal{C}$	2.1 R
2.4	$\quad \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2-2.3 $\to$I
2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	2.1-2.4 $\to$I
3	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2,1 $\to$E

Notice that the line numbers of the justification are identical to those in the justification from A. And similarly, we are in a position to generate each line in A. Thus, for example, line 4 of A is an instance of A2. So we would continue with lines 4.1–4.8 and 4 to generate the appropriate instance of A2. As it turns out, the resultant *ND* derivation is not very efficient. But it is a derivation, and our point is merely to show that some *ND* derivation of the same result exists. So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

*E9.3. Set up the above induction for T9.2, and complete the unfinished cases to show that if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.4. (i) Where A is the derivation (D) from above, complete the process of finding the corresponding derivation N. Hint: If you follow the recipe correctly, the result should have exactly 21 lines. (ii) This derivation N is not very efficient. See if you can find an *ND* derivation to show $\mathcal{A} \to \mathcal{B}$, $\mathcal{B} \to \mathcal{C} \vdash_{ND} \mathcal{A} \to \mathcal{C}$ that takes fewer than 10 lines.

E9.5. Extend system A^* as described for E3.5 to an $A^\star$ that has $\sim$, $\wedge$, and $\exists$ primitive with axioms and rules as follows:

$A^\star$
- A1. $\mathcal{P} \to (\mathcal{P} \wedge \mathcal{P})$
- A2. $(\mathcal{P} \wedge \mathcal{Q}) \to \mathcal{P}$
- A3. $(\mathcal{O} \to \mathcal{P}) \to [\sim(\mathcal{P} \wedge \mathcal{Q}) \to \sim(\mathcal{Q} \wedge \mathcal{O})]$
- A4. $\mathcal{P}^x_t \to \exists x \mathcal{P}$ where t is free for x in $\mathcal{P}$
- MP. $\sim(\mathcal{P} \wedge \sim\mathcal{Q}), \mathcal{P} \vdash_{A^\star} \mathcal{Q}$
- $\exists$R. $\mathcal{P} \to \mathcal{Q} \vdash_{A^\star} \exists x \mathcal{P} \to \mathcal{Q}$ where x is not free in $\mathcal{Q}$

Produce a complete demonstration to show that if $\Gamma \vdash_{A^\star} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

9.3 Validity in *ND* Implies Validity in *AD*

Perhaps the result we have just attained is obvious: Insofar as the resources of *ND* seem to exceed the resources of *AD*, whenever $\Gamma \vdash_{AD} \mathcal{P}$, we expect $\Gamma \vdash_{ND} \mathcal{P}$. But the other direction may be less clear. Insofar as *AD* may seem to have fewer resources than *ND*, one might wonder whether it is the case that if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. But, in fact, it is possible to do in *AD* whatever can be done in *ND*. To show this, we need a couple of preliminary results. I begin with an important result known as the *deduction theorem*, turn to some substitution theorems, and finally to the intended result that whatever is provable in *ND* is provable in *AD*.

9.3.1 Deduction Theorem

According to the deduction theorem—subject to an important restriction—if there is an *AD* derivation of $\mathcal{Q}$ from the members of some set of sentences Δ plus $\mathcal{P}$, then there is an *AD* derivation of $\mathcal{P} \rightarrow \mathcal{Q}$ from the members of Δ alone: if $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$ then $\Delta \vdash_{AD} \mathcal{P} \rightarrow \mathcal{Q}$. In practice, this lets us reason just as we do with $\rightarrow$I.

(E)

	members of Δ	
a.	$\mid$ $\mathcal{P}$	
b.	$\mid$ $\mathcal{Q}$	
c.	$\mathcal{P} \rightarrow \mathcal{Q}$	a-b deduction theorem

At (b), there is a derivation of $\mathcal{Q}$ from the members of Δ plus $\mathcal{P}$. At (c), the assumption is discharged to indicate a derivation of $\mathcal{P} \rightarrow \mathcal{Q}$ from the members of Δ alone. By the deduction theorem, if there is a derivation of $\mathcal{Q}$ from Δ plus $\mathcal{P}$, then there is a derivation of $\mathcal{P} \rightarrow \mathcal{Q}$ from Δ. Here is the restriction: The discharge of an auxiliary assumption $\mathcal{P}$ is legitimate just in case no application of Gen under its scope generalizes on a variable free in $\mathcal{P}$. The effect is like that of the *ND* restriction on $\forall$I—here, though, the restriction is not on Gen, but rather on the discharge of auxiliary assumptions. In the one case, an assumption available for discharge is one such that no application of Gen under its scope is to a variable free in the assumption; in the other, we cannot apply $\forall$I to a variable free in an undischarged assumption (so that, effectively, every assumption is always available for discharge).

Again, our strategy is to show that given one derivation, it is possible to construct another. In this case, as indicated on the following page, we begin with an *AD* derivation (A), with premises $\Delta \cup \{\mathcal{P}\}$ and conclusion $\mathcal{Q}_n = \mathcal{Q}$. Treating $\mathcal{P}$ as an auxiliary premise, with scope as indicated in (B), we set out to show that there is an *AD* derivation (C), with premises in Δ alone, and lines numbered 1, 2, … corresponding to 1, 2, … in (A).

(A)		(B)		(C)	
1.	$\mathcal{Q}_1$	1.	$\mathcal{Q}_1$	1.	$\mathcal{P} \to \mathcal{Q}_1$
2.	$\mathcal{Q}_2$	2.	$\mathcal{Q}_2$	2.	$\mathcal{P} \to \mathcal{Q}_2$
	$\vdots$		$\vdots$		$\vdots$
	$\mathcal{P}$		$\mathcal{P}$ (assumption)		$\mathcal{P} \to \mathcal{P}$
	$\vdots$		$\vdots$		$\vdots$
n.	$\mathcal{Q}_n$	n.	$\mathcal{Q}_n$	n.	$\mathcal{P} \to \mathcal{Q}_n$

That is, we construct a derivation with premises in Δ such that for any formula $\mathcal{A}$ on line i of the first derivation, $\mathcal{P} \to \mathcal{A}$ appears on the line numbered i of the constructed derivation. The line numbered n of the resultant derivation is the desired result, so $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

T9.3. If $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$, and no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, then $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$. *Deduction Theorem.*

Suppose $A = \langle \mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_n \rangle$ is an *AD* derivation of $\mathcal{Q}$ from $\Delta \cup \{\mathcal{P}\}$, where $\mathcal{Q}$ is $\mathcal{Q}_n$ and no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$. By induction on the line numbers in derivation A, we show there is a derivation C with premises only in Δ, such that for any line i of A, $\mathcal{P} \to \mathcal{Q}_i$ appears on the line numbered i of C. The case when $i = n$ gives the desired result, that $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

Basis: $\mathcal{Q}_1$ of A is an axiom, a member of Δ, or $\mathcal{P}$ itself.

(i) If $\mathcal{Q}_1$ is an axiom or a member of Δ, then begin C as follows:

1.1	$\mathcal{Q}_1$	axiom / premise
1.2	$\mathcal{Q}_1 \to (\mathcal{P} \to \mathcal{Q}_1)$	A1
1	$\mathcal{P} \to \mathcal{Q}_1$	1.2, 1.1 MP

(ii) $\mathcal{Q}_1$ is $\mathcal{P}$ itself. By T3.1, $\vdash_{AD} \mathcal{P} \to \mathcal{P}$; which is to say $\mathcal{P} \to \mathcal{Q}_1$; so begin derivation C,

1	$\mathcal{P} \to \mathcal{P}$	T3.1

In either case, $\mathcal{P} \to \mathcal{Q}_1$ appears on the line numbered 1 of C with premises in Δ alone.

Assp: For any i, $1 \leq i < k$, $\mathcal{P} \to \mathcal{Q}_i$ appears on the line numbered i of C, with premises in Δ alone.

Show: $\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered k of C, with premises in Δ alone.

$\mathcal{Q}_k$ of A is a member of Δ, an axiom, $\mathcal{P}$ itself, or arises from previous lines by MP or Gen. If $\mathcal{Q}_k$ is a member of Δ, an axiom, or $\mathcal{P}$ itself then, by reasoning as in the basis, $\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered k of C from premises in Δ alone. So two cases remain.

(MP) If $\mathcal{Q}_k$ arises from previous lines by MP, then there are lines in derivation A of the sort,

$$\begin{array}{lll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ & \vdots & \\ j & \mathcal{B} & \\ & \vdots & \\ k & \mathcal{C} & i,j \text{ MP} \end{array}$$

where i, $j < k$ and $\mathcal{Q}_k$ is $\mathcal{C}$. By assumption, there are lines in C,

$$\begin{array}{ll} i & \mathcal{P} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \\ & \vdots \\ j & \mathcal{P} \rightarrow \mathcal{B} \end{array}$$

So continue derivation C as follows:

$$\begin{array}{rll} i & \mathcal{P} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) & \\ & \vdots & \\ j & \mathcal{P} \rightarrow \mathcal{B} & \\ & \vdots & \\ k.1 & [\mathcal{P} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})] \rightarrow [(\mathcal{P} \rightarrow \mathcal{B}) \rightarrow (\mathcal{P} \rightarrow \mathcal{C})] & \text{A2} \\ k.2 & (\mathcal{P} \rightarrow \mathcal{B}) \rightarrow (\mathcal{P} \rightarrow \mathcal{C}) & k.1, i \text{ MP} \\ k & \mathcal{P} \rightarrow \mathcal{C} & k.2, j \text{ MP} \end{array}$$

So $\mathcal{P} \rightarrow \mathcal{Q}_k$ appears on the line numbered k of C, with premises in Δ alone.

(Gen) If $\mathcal{Q}_k$ arises from a previous line by Gen, then there are lines in derivation A of the sort,

$$\begin{array}{lll} i & \mathcal{B} & \\ & \vdots & \\ k & \forall x \mathcal{B} & i \text{ Gen} \end{array}$$

where $i < k$ and $\mathcal{Q}_k$ is $\forall x \mathcal{B}$. Either line k is under the scope of $\mathcal{P}$ in derivation A or not.

(i) If line k is not under the scope of $\mathcal{P}$, then $\forall x \mathcal{B}$ in A follows from Δ alone. So continue C as follows:

$$\begin{array}{rll} k.1 & \mathcal{Q}_1 & \text{exactly as in } A \text{ but with prefix} \\ k.2 & \mathcal{Q}_2 & k \text{ for numeric references} \\ & \vdots & \\ k.k & \forall x \mathcal{B} & \\ k.k+1 & \forall x \mathcal{B} \rightarrow (\mathcal{P} \rightarrow \forall x \mathcal{B}) & \text{A1} \\ k & \mathcal{P} \rightarrow \forall x \mathcal{B} & k.k+1, k.k \text{ MP} \end{array}$$

Since each of the lines in A up to k is derived from Δ alone, we have $\mathcal{P} \rightarrow \mathcal{Q}_k$ on the line numbered k of C from premises in Δ alone.

(ii) If line k is under the scope of $\mathcal{P}$, we depend on the assumption, and continue C as follows:

$$
\begin{array}{lll}
i & \mathcal{P} \to \mathcal{B} & \text{(by inductive assumption)} \\
 & \vdots & \\
k & \mathcal{P} \to \forall x \mathcal{B} & i \text{ T3.29}
\end{array}
$$

If line k in derivation A is under the scope of $\mathcal{P}$ then, since no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, x is not free in $\mathcal{P}$; so the restriction on T3.29 is met. So we have $\mathcal{P} \to \mathcal{Q}_k$ on the line numbered k of C, from premises in Δ alone.

$\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered k of C, with premises in Δ alone.

Indct: For any i, $\mathcal{P} \to \mathcal{Q}_i$ appears on the line numbered i of C, from premises in Δ alone.

For the last, and most interesting, case: Outside the scope of $\mathcal{P}$ each of the lines in A, including $\forall x \mathcal{B}$, is already derived from Δ alone; so with A1, $\mathcal{P} \to \forall x \mathcal{B}$ from Δ alone. Under the scope of $\mathcal{P}$, the restriction guarantees that x is not free in $\mathcal{P}$, so with T3.29 and $\mathcal{P} \to \mathcal{B}$ from Δ alone, $\mathcal{P} \to \forall x \mathcal{B}$ from Δ alone.

So given an *AD* derivation of $\mathcal{Q}$ from $\Delta \cup \{\mathcal{P}\}$, where no application of Gen under the scope of assumption $\mathcal{P}$ is to a variable free in $\mathcal{P}$, there is sure to be an *AD* derivation of $\mathcal{P} \to \mathcal{Q}$ from Δ. Notice that T3.29 and T3.32 abbreviate sequences which include applications of Gen. So the restriction on Gen for the deduction theorem applies to applications of these results as well. (Some other theorems from T3.28–T3.38 require Gen, but derivation for theorems of the sort $\vdash_{AD} \mathcal{A}$ may be moved to the start, and so outside the scope of $\mathcal{P}$. So they remain available.)

As a sample application of the deduction theorem (DT), let us consider another derivation of T3.2. In this case, $\Delta = \{\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C}\}$, and we argue as follows:

(F)

1.	$\mathcal{A} \to \mathcal{B}$	prem
2.	$\mathcal{B} \to \mathcal{C}$	prem
3.	$\quad\mid \mathcal{A}$	assp (g, DT)
4.	$\quad\mid \mathcal{B}$	1,3 MP
5.	$\quad\mid \mathcal{C}$	2,4 MP
6.	$\mathcal{A} \to \mathcal{C}$	3-5 DT

At line (5) we have established that $\Delta \cup \{\mathcal{A}\} \vdash_{AD} \mathcal{C}$; it follows from the deduction theorem that $\Delta \vdash_{AD} \mathcal{A} \to \mathcal{C}$. But we should be careful: This is not an *AD* derivation of $\mathcal{A} \to \mathcal{C}$ from $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$. And it is not an abbreviation in the sense that we have seen so far—we do not appeal to a result whose derivation could be inserted at that very stage. Rather, what we have is a demonstration, via the deduction theorem, that there *exists* an *AD* derivation of $\mathcal{A} \to \mathcal{C}$ from the premises. If there is any abbreviating, the entire derivation abbreviates, or indicates the existence of, another. Our proof of the deduction theorem shows us that, given a derivation of $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$, it is possible to *construct* a derivation for $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

Let us see how this works in the example. Lines 1–5 become our derivation A, with $\Delta = \{\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C}\}$. For each $\mathcal{Q}_i$ in derivation A, the induction tells us how to derive $\mathcal{A} \to \mathcal{Q}_i$ from Δ alone. Thus $\mathcal{Q}_i$ on the first line is a member of Δ, and reasoning from the basis tells us to use A1 as follows:

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2, 1.1 MP

to get $\mathcal{A}$ arrow the form on line 1 of A. Notice that we are again using fractional line numbers to make lines in derivation A correspond to lines in the constructed derivation. One may wonder why we bother getting $\mathcal{A} \to \mathcal{Q}_1$. And again, the answer is that our "recipe" calls for this ingredient at stages connected to MP and Gen. Similarly, we can use A1 to get $\mathcal{A}$ arrow the form on line (2).

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2, 1.1 MP
2.1	$\mathcal{B} \to \mathcal{C}$	prem
2.2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	A1
2	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2, 2.1 MP

The form on line (3) is $\mathcal{A}$ itself. If we wanted a derivation in the primitive system, we could repeat the steps in our derivation of T3.1. But we will simply continue, as in the induction,

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2, 1.1 MP
2.1	$\mathcal{B} \to \mathcal{C}$	prem
2.2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	A1
2	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2, 2.1 MP
3	$\mathcal{A} \to \mathcal{A}$	T3.1

to get $\mathcal{A}$ arrow the form on line (3) of A. The form on line (4) arises from lines (1) and (3) by MP; reasoning in our show step tells us to continue,

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2, 1.1 MP
2.1	$\mathcal{B} \to \mathcal{C}$	prem
2.2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	A1
2	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2, 2.1 MP
3	$\mathcal{A} \to \mathcal{A}$	T3.1
4.1	$(\mathcal{A} \to (\mathcal{A} \to \mathcal{B})) \to ((\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B}))$	A2
4.2	$(\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B})$	4.1, 1 MP
4	$\mathcal{A} \to \mathcal{B}$	4.2, 3 MP

using A2 to get $\mathcal{A} \to \mathcal{B}$. Notice that the original justification from lines (1) and (3) dictates the appeal to (1) at line (4.2) and to (3) at line (4). The form on line (5) arises from lines (2) and (4) by MP; so, finally, we continue,

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2, 1.1 MP
2.1	$\mathcal{B} \to \mathcal{C}$	prem
2.2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	A1
2	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2, 2.1 MP
3	$\mathcal{A} \to \mathcal{A}$	T3.1
4.1	$(\mathcal{A} \to (\mathcal{A} \to \mathcal{B})) \to ((\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B}))$	A2
4.2	$(\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B})$	4.1, 1 MP
4	$\mathcal{A} \to \mathcal{B}$	4.2, 3 MP
5.1	$(\mathcal{A} \to (\mathcal{B} \to \mathcal{C})) \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))$	A2
5.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	5.1, 2 MP
5	$\mathcal{A} \to \mathcal{C}$	5.2, 4 MP

And we have the *AD* derivation which our proof of the deduction theorem told us there would be. Notice that this derivation is not very efficient. We did it in seven lines (without appeal to T3.1) in Chapter 3. What our proof of the deduction theorem tells us is that there is sure to be some derivation—where there is no expectation that the guaranteed derivation is particularly elegant or efficient.

Here is a last example which makes use of the deduction theorem. First, an alternate derivation of T3.3:

(G)

1.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
2.	$\mid \mathcal{B}$	assp (g, DT)
3.	$\mid\mid \mathcal{A}$	assp (g, DT)
4.	$\mid\mid \mathcal{B} \to \mathcal{C}$	1,3 MP
5.	$\mid\mid \mathcal{C}$	4,2 MP
6.	$\mid \mathcal{A} \to \mathcal{C}$	3-5 DT
7.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	2-6 DT

In Chapter 3 we proved T3.3 in five lines (with an appeal to T3.2). But perhaps this version is relatively intuitive, coinciding as it does with strategies from *ND*. In this case, there are two applications of DT, and reasoning from the induction therefore applies twice: First, at line (5), there is an *AD* derivation of $\mathcal{C}$ from $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\} \cup \{\mathcal{A}\}$. By reasoning from the induction, then, there is an *AD* derivation from just $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\}$ with $\mathcal{A}$ arrow each of the forms on lines 1–5. So there is a derivation of $\mathcal{A} \to \mathcal{C}$ from $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\}$. But then reasoning from the induction applies again. By reasoning from the induction applied to this *new* derivation, there is a derivation from just $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$ with $\mathcal{B}$ arrow each of the forms in it. So there is a derivation of $\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$ from just $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$. So the first derivation, lines 1–5 above, is replaced by another by the reasoning from DT.

Then *it* is replaced by another, again given the reasoning from DT. The result is an *AD* derivation of the desired result.

Here are a couple more cases, where the latter at least may inspire a certain affection for the deduction theorem:

T9.4. $\vdash_{AD} \mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B}))$

T9.5. $\vdash_{AD} (\mathcal{A} \to \mathcal{C}) \to [(\mathcal{B} \to \mathcal{C}) \to ((\mathcal{A} \vee \mathcal{B}) \to \mathcal{C})]$

The deduction theorem streamlines reasoning for many results in *AD*. And, towards a demonstration that *AD* accomplishes whatever is accomplished in *ND*, with the deduction theorem we shall be able to show that *AD* mimics *ND* rules requiring subderivations.

E9.6. (i) Making use of the deduction theorem, prove T9.4 and T9.5. (ii) Having done so, see if you can prove them in the style of Chapter 3, without any appeal to DT.

E9.7. By the method of our proof of the deduction theorem, convert the above derivation (G) for T3.3 into an official *AD* derivation. Hint: As described above, the method of the induction applies twice: first to lines 1–5, and then to the new derivation. The result should be derivations with 13, and then 37 lines.

E9.8. Consider the axiomatic system $A^\star$ from E9.5, and produce a demonstration of the deduction theorem for it. That is show that if $\Delta \cup \{\mathcal{P}\} \vdash_{A^\star} \mathcal{Q}$ and no application of ∃R under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, then $\Delta \vdash_{A^\star} \mathcal{P} \to \mathcal{Q}$. Because $A^\star$ extends A^*, you may appeal to any of the A^* theorems from E3.5.

9.3.2 Substitution Theorems

Allowing, as it does, substitution of arbitrary terms into arbitrary formulas, the *ND* rule =E applies in contexts where the *AD* axioms A7 and A8 do not. Again, then, *ND* may seem to have resources that *AD* lacks. As a basis for showing that the resources of *AD* match those of *ND*, we turn now to some substitution results.

Say a complex term r is *free* in an expression $\mathcal{P}$ just in case no variable in r is bound; then where $\mathcal{T}$ is any term or formula, let $\mathcal{T}^{r}/\!\!/_{s}$ be $\mathcal{T}$ where at most one free instance of r is replaced by term s. Having shown in T3.38 that $\vdash_{AD} (q_i = s) \to (\mathcal{R}^n q_1 \dots q_i \dots q_n \to \mathcal{R}^n q_1 \dots s \dots q_n)$, one might think we have proved that $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/\!\!/_{s})$ for any atomic formula $\mathcal{A}$ and any terms r and s. But *this is not so*. Similarly, having shown in T3.37 that $\vdash_{AD} (q_i = s) \to (h^n q_1 \dots q_i \dots q_n = h^n q_1 \dots s \dots q_n)$, one might think we have proved that

$\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$ for any terms r, s, and t. But this is not so. In each case, the difficulty is that the replaced term r might be a *component* of the other terms $q_1 \dots q_n$, and so might not be any of $q_1 \dots q_n$. What we have shown is only that it is possible to replace any of the whole terms, $q_1 \dots q_n$. Thus, $(x = y) \to (f^1g^1x = f^1g^1y)$ is not an instance of T3.37 because we do not replace g^1x but rather a component of it.

However, as one might expect, it is possible to replace terms in basic parts; use the result to make replacements in terms of which *they* are parts; and so forth, all the way up to wholes. Both $(x = y) \to (g^1x = g^1y)$ and $(g^1x = g^1y) \to (f^1g^1x = f^1g^1y)$ are instances of T3.37. (Be clear about these examples in your mind.) From these, with T3.2 it follows that $\vdash_{AD} (x = y) \to (f^1g^1x = f^1g^1y)$. This example suggests a method for obtaining the more general results: Using T3.37, we work from equalities at the level of the parts, to equalities at the level of the whole. For the case of terms, the proof is by induction on the number of function symbols in an arbitrary term t.

T9.6. For arbitrary terms r, s, and t, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

Basis: If t has no function symbols, then t is a variable or a constant. Then either (i) $t^{r}/\!\!/_{s} = t$ (nothing is replaced) or (ii) $r = t$ and $t^{r}/\!\!/_{s} = s$ (all of t is replaced). In the first case, by T3.33, $\vdash_{AD} t = t$; which is to say, $\vdash_{AD} (t = t^{r}/\!\!/_{s})$; so with A1, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$. In the second case, $(r = s) \to (t = t^{r}/\!\!/_{s})$ is the same as $(r = s) \to (r = s)$; so by T3.1, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

Assp: For any i, $0 \le i < k$, if t has i function symbols, then $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

Show: If t has k function symbols, then $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

If t has k function symbols, then t is of the form $h^n q_1 \dots q_n$ for terms $q_1 \dots q_n$ with $< k$ function symbols. If all of t is replaced, or no part of t is replaced, then reason as in the basis. So suppose r is some subcomponent of t; then for some q_i, $t^{r}/\!\!/_{s}$ is $h^n q_1 \dots q_i{}^{r}/\!\!/_{s} \dots q_n$. By assumption, $\vdash_{AD} (r = s) \to (q_i = q_i{}^{r}/\!\!/_{s})$; and by T3.37, $\vdash_{AD} (q_i = q_i{}^{r}/\!\!/_{s}) \to (h^n q_1 \dots q_i \dots q_n = h^n q_1 \dots q_i{}^{r}/\!\!/_{s} \dots q_n)$; so by T3.2, $\vdash_{AD} (r = s) \to (h^n q_1 \dots q_i \dots q_n = h^n q_1 \dots q_i{}^{r}/\!\!/_{s} \dots q_n)$; but this is to say, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

Indct: For any terms r, s, and t, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

We might think of this result as a further strengthened or generalized version of the *AD* axiom A7. Where A7 lets us replace just one of the variables $x_1 \dots x_n$ in $h^n x_1 \dots x_n$ by a variable y, and T3.37 one of the terms $t_1 \dots t_n$ in $h^n t_1 \dots t_n$ with a term s, we are now in a position to replace an arbitrary "subterm" of $h^n t_1 \dots t_n$ with another term s.

Now we can go after a similarly strengthened version of A8. We show that for any formula $\mathcal{P}$, if s is free for the replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$. The argument is by induction on the number of operators in $\mathcal{P}$.

*T9.7. For any formula $\mathcal{P}$ and terms r and s, if s is free for any replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$.

Basis: If $\mathcal{P}$ is atomic then (i) $\mathcal{P}^{r}/\!\!/_{s} = \mathcal{P}$ (nothing is replaced) or (ii) $\mathcal{P}$ is $\mathcal{R}^n t_1 \ldots t_i \ldots t_n$ and $\mathcal{P}^{r}/\!\!/_{s}$ is $\mathcal{R}^n t_1 \ldots t_i{}^{r}/\!\!/_{s} \ldots t_n$. Suppose s is free for any replaced instance of r in $\mathcal{P}$. In the first case, by T3.1, $\vdash_{AD} \mathcal{P} \rightarrow \mathcal{P}$, which is to say $\vdash_{AD} \mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s}$; so with A1, $\vdash_{AD} r = s \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$. In the second case, by T9.6, $\vdash_{AD} (r = s) \rightarrow (t_i = t_i{}^{r}/\!\!/_{s})$; and by T3.38, $\vdash_{AD} (t_i = t_i{}^{r}/\!\!/_{s}) \rightarrow (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \rightarrow \mathcal{R}^n t_1 \ldots t_i{}^{r}/\!\!/_{s} \ldots t_n)$; so by T3.2, $\vdash_{AD} (r = s) \rightarrow (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \rightarrow \mathcal{R}^n t_1 \ldots t_i{}^{r}/\!\!/_{s} \ldots t_n)$; and this is just to say, $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols and s is free for any replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$.

Corollary to the assumption: If $\mathcal{P}$ has $< k$ operators, then $\mathcal{P}^{r}/\!\!/_{s}$ has $< k$ operators; and since s replaces only a free instance of r in $\mathcal{P}$, r is free for the replacing instance of s in $\mathcal{P}^{r}/\!\!/_{s}$; so where the outer substitution is made to sustain $[\mathcal{P}^{r}/\!\!/_{s}]^{s}/\!\!/_{r} = \mathcal{P}$, we have $\vdash_{AD} (s = r) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow [\mathcal{P}^{r}/\!\!/_{s}]^{s}/\!\!/_{r})$ as an instance of the inductive assumption, which is just, $\vdash_{AD} (s = r) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{P})$. And by T3.34, $\vdash_{AD} (r = s) \rightarrow (s = r)$; so with T3.2, $\vdash_{AD} (r = s) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{P})$.

Show: If $\mathcal{P}$ has k operator symbols and s is free for any replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$.

If $\mathcal{P}$ has k operator symbols, then $\mathcal{P}$ is of the form, $\sim\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$, or $\forall x \mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. If no replacement is made, reason as in the basis. So suppose some replacement is made and s is free for the replaced instance of r in $\mathcal{P}$.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then $\mathcal{P}^{r}/\!\!/_{s}$ is $[\sim\mathcal{A}]^{r}/\!\!/_{s}$ which is the same as $\sim[\mathcal{A}^{r}/\!\!/_{s}]$. Since s is free for the replaced instance of r in $\mathcal{P}$, it is free for that instance of r in $\mathcal{A}$; so by the corollary to the assumption, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A}^{r}/\!\!/_{s} \rightarrow \mathcal{A})$. But by T3.13, $\vdash_{AD} (\mathcal{A}^{r}/\!\!/_{s} \rightarrow \mathcal{A}) \rightarrow (\sim\mathcal{A} \rightarrow \sim[\mathcal{A}^{r}/\!\!/_{s}])$; so by T3.2, $\vdash_{AD} (r = s) \rightarrow (\sim\mathcal{A} \rightarrow \sim[\mathcal{A}^{r}/\!\!/_{s}])$; which is to say, $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$.

($\rightarrow$) Suppose $\mathcal{P}$ is $\mathcal{A} \rightarrow \mathcal{B}$. Then $\mathcal{P}^{r}/\!\!/_{s}$ is $\mathcal{A}^{r}/\!\!/_{s} \rightarrow \mathcal{B}$ or $\mathcal{A} \rightarrow \mathcal{B}^{r}/\!\!/_{s}$. (i) In the former case, since s is free for the replaced instance of r in $\mathcal{P}$, it is free for that instance of r in $\mathcal{A}$; so by the corollary to the assumption, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A}^{r}/\!\!/_{s} \rightarrow \mathcal{A})$; so we may reason as follows:

1.	$(\mathscr{r} = \mathscr{s}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{A})$	by assumption
2.	$\mid\ \mathscr{r} = \mathscr{s}$	assp (g, DT)
3.	$\mid\ \mid\ \mathcal{A} \to \mathcal{B}$	assp (g, DT)
4.	$\mid\ \mid\ \mid\ \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$	assp (g, DT)
5.	$\mid\ \mid\ \mid\ \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{A}$	1,2 MP
6.	$\mid\ \mid\ \mid\ \mathcal{A}$	5,4 MP
7.	$\mid\ \mid\ \mid\ \mathcal{B}$	3,6 MP
8.	$\mid\ \mid\ \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{B}$	4-7 DT
9.	$\mid\ (\mathcal{A} \to \mathcal{B}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{B})$	3-8 DT
10.	$(\mathscr{r} = \mathscr{s}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{B})]$	2-9 DT

So $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{B})]$; which is to say, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$. (ii) And similarly in the other case [by homework], $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{B}^{\mathscr{r}}/\!\!/_{\mathscr{s}})]$. So in either case, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

($\forall$) Suppose $\mathcal{P}$ is $\forall x\mathcal{A}$. Then a free instance of $\mathscr{r}$ in $\mathcal{P}$ remains free in $\mathcal{A}$ and $\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ is $\forall x[\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}]$. Since $\mathscr{s}$ is free for $\mathscr{r}$ in $\mathcal{P}$, $\mathscr{s}$ is free for $\mathscr{r}$ in $\mathcal{A}$; so by assumption, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$; so we may reason as follows:

1.	$(\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$	by assumption
2.	$\mid\ \mathscr{r} = \mathscr{s}$	assp (g, DT)
3.	$\mid\ \forall x\mathcal{A} \to \mathcal{A}$	A4
4.	$\mid\ \mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$	1,2 MP
5.	$\mid\ \forall x\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$	3,4 T3.2
6.	$\mid\ \forall x\mathcal{A} \to \forall x\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$	5 T3.29
7.	$(\mathscr{r} = \mathscr{s}) \to (\forall x\mathcal{A} \to \forall x\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$	2-6 DT

Notice that x is sure to be free for itself in $\mathcal{A}$, so that (3) is an instance of A4. And x is bound in $\forall x\mathcal{A}$, so (6) is an instance of T3.29. And because $\mathscr{r}$ is free in $\mathcal{P} = \forall x\mathcal{A}$, and $\mathscr{s}$ is free for $\mathscr{r}$ in $\mathcal{P}$, x cannot be a variable in $\mathscr{r}$ or $\mathscr{s}$; so the restriction on DT is met at (7). So $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\forall x\mathcal{A} \to \forall x\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$; which is to say, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

For any $\mathcal{P}$ with k operator symbols, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Indct: For any $\mathcal{P}$, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

So for any formula $\mathcal{P}$ and terms $\mathscr{r}$ and $\mathscr{s}$, if $\mathscr{s}$ is free for a replaced instance of $\mathscr{r}$ in $\mathcal{P}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P} \to \mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Some final substitution results are straightforward on the pattern of what we have just achieved. Let $\mathcal{P}^{t}/_{\mathscr{s}}$ be $\mathcal{P}$ with some, but not necessarily all, free instances of term t replaced by term $\mathscr{s}$ (as for equality rules of *ND*), and $\mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}$ be $\mathcal{O}$ with at most one instance of a subformula $\mathcal{P}$ replaced by formula $\mathcal{Q}$ (as for replacement rules of *ND*).

*T9.8. For any formula $\mathcal{P}$ and terms r and s, if s is free for the replaced instances of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \to (\mathcal{P} \to \mathcal{P}^{r}/_{s})$.

By repeated application of T9.7.

*T9.9. For any formulas $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{Q}$, if $\Gamma \vdash_{AD} \mathcal{P} \leftrightarrow \mathcal{Q}$, then $\Gamma \vdash_{AD} \mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}$.

The substitution applies to formulas rather than terms.

T9.10. For any formulas $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{Q}$, interpretation I, and variable assignment d, if $\mathsf{I_d}[\mathcal{P}] = \mathsf{I_d}[\mathcal{Q}]$ then $\mathsf{I_d}[\mathcal{O}] = \mathsf{I_d}[\mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}]$. Corollary: If $\mathsf{I_d}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}] = \mathsf{S}$.

This result is semantical rather than syntactical.

So T9.8 permits the substitution of arbitrarily many terms. T9.9 substitutes one formula for another. And T9.10 is a parallel semantic result. For T9.9, very often we shall be interested in the case when Γ is empty, and so if $\vdash_{AD} \mathcal{P} \leftrightarrow \mathcal{Q}$, then $\vdash_{AD} \mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}$.

*E9.9. Set up the above demonstration for T9.7 and complete the unfinished case to provide a complete demonstration that for any formula $\mathcal{P}$, and terms r and s, if s is free for any replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \to (\mathcal{P} \to \mathcal{P}^{r}/\!\!/_{s})$.

*E9.10. Provide a demonstration for T9.8. Hint: Reason by induction on the number of instances of r that are replaced by s in $\mathcal{P}$. Say $\mathcal{P}_i$ is $\mathcal{P}$ with i free instances of r replaced by s. Suppose s is free for the replaced instances of r in $\mathcal{P}$. Show that for any i, $\vdash_{AD} (r = s) \to (\mathcal{P} \to \mathcal{P}_i)$.

*E9.11. Prove T9.9. Hint: In the basis, when $\mathcal{O}$ is atomic, either $\mathcal{O} \neq \mathcal{P}$ and no replacement is made, or $\mathcal{O} = \mathcal{P}$ and all of $\mathcal{O}$ is replaced. For the show, when all of $\mathcal{O}$ is replaced or no part of $\mathcal{O}$ is replaced, reason as in the basis. If $\mathcal{P}$ is a proper part of $\mathcal{O}$, then the assumption applies. Also, where $\mathcal{P} \leftrightarrow \mathcal{Q}$ abbreviates $(\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P})$, you can use (abv) along with T3.20, T3.21, and T9.4 to manipulate formulas of the sort $\mathcal{P} \leftrightarrow \mathcal{Q}$.

E9.12. Show T9.10.

E9.13. Where the primitive operators are $\sim$, $\wedge$, and $\exists$, show an analog to T9.9 for derivation system $A^{\star}$ from E9.5—that for any formulas $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{Q}$, if $\Gamma \vdash_{A^{\star}} \mathcal{P} \leftrightarrow \mathcal{Q}$, then $\Gamma \vdash_{A^{\star}} \mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}$. Again you may appeal to any of the theorems from E3.5.

9.3.3 Intended Result

We are finally ready to show that if $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{AD} \mathcal{P}$. As usual, the idea is that the existence of one derivation guarantees the existence of another. In this case, we begin with a derivation in *ND*, and move to the existence of one in *AD*. Suppose $\Gamma \vdash_{ND} \mathcal{P}$; then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ, with lines $\langle \mathcal{Q}_1 \ldots \mathcal{Q}_n \rangle$ and $\mathcal{Q}_n = \mathcal{P}$. We show that there is an *AD* derivation of the same result. Our reasoning applies to a derivation A permitting DT as a rule; then given this derivation, by the deduction theorem, there is derivation in the primitive *AD*. Say derivation A *matches* N iff any $\mathcal{Q}_i$ from N appears at the same scope on the line numbered i of A; and say derivation A is *good* iff it has no application of Gen to a variable free in an undischarged auxiliary assumption (so that DT is available at any stage in A). Then, given derivation N, we show that there is a good derivation A that matches N. The argument is by induction on the line number of N, where we show that for any i, there is a good derivation A_i that matches N through line i. The case when $i = n$ is a good derivation of $\mathcal{P}$ under the scope of the premises alone, from which it follows that $\Gamma \vdash_{AD} \mathcal{P}$.

It will be helpful here (and later) to obtain a preliminary theorem,

T9.11. $\vdash_{AD} \forall v \mathcal{P}^x_v \rightarrow \forall x \mathcal{P}$ where v is not free in $\forall x \mathcal{P}$ and free for x in $\mathcal{P}$

Suppose v is not free in $\forall x \mathcal{P}$ and free for x in $\mathcal{P}$. If $x = v$, then T9.11 is just an instance of T3.1. So suppose $x \neq v$; then since v is not free in $\forall x \mathcal{P}$, v is not free in $\mathcal{P}$. Reason as follows:

1.	$\forall v \mathcal{P}^x_v \rightarrow \forall x (\mathcal{P}^x_v)^v_x$	T3.28
2.	$\forall v \mathcal{P}^x_v \rightarrow \forall x \mathcal{P}$	1 with T8.2

In this case, T3.28 requires x not free in $\forall v \mathcal{P}^x_v$ and free for v in $\mathcal{P}^x_v$: but since every free instance of x is replaced in $\mathcal{P}^x_v$, x is not free in $\mathcal{P}^x_v$ and so in $\forall v \mathcal{P}^x_v$; and since v is not free in $\mathcal{P}$, every free instance of v in $\mathcal{P}^x_v$ replaces a free instance of x, and x is free for v in $\mathcal{P}^x_v$. T8.2 requires v not free in $\mathcal{P}$ but free for x in $\mathcal{P}$: but it is given that v is free for x in $\mathcal{P}$ and, from above, v is not free in $\mathcal{P}$.

Note that the combination of T9.11 with T3.28 yields exchange of bound variables in both directions: where v is not free in $\forall x \mathcal{P}$ and free for x in $\mathcal{P}$, then $\vdash_{AD} \forall x \mathcal{P} \leftrightarrow \forall v \mathcal{P}^x_v$. Now we are ready for the main result.

*T9.12. If $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND} \mathcal{P}$; then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ. By induction on the line numbers of N, we show that for any i, there is a good *AD* derivation A_i that matches N through line i.

Basis: The first line of N is a premise, an assumption, or arises by =I. Let A_1 be the same (in the latter case with justification T3.33). Then A_1 matches N; and since there is no application of Gen under an undischarged assumption, A_1 is good.

Assp: For any i, $1 \leq i < k$, there is a good derivation A_i that matches N through line i.

Show: There is a good derivation A_k that matches N through line k. Either $\mathcal{Q}_k$ is a premise, an assumption, arises by =I, or results from previous lines by R, $\wedge$E, $\wedge$I, $\rightarrow$E, $\rightarrow$I, $\sim$E, $\sim$I, $\vee$E, $\vee$I, $\leftrightarrow$E, $\leftrightarrow$I, $\forall$E, $\forall$I, $\exists$E, $\exists$I, or =E.

(B) If $\mathcal{Q}_k$ is a premise, an assumption, or arises by =I, let A_k continue in the same way. Then, by reasoning as in the basis, A_k matches N and is good.[2]

(R) If $\mathcal{Q}_k$ arises from previous lines by R, then N looks something like this,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ & & \\ k & \mathcal{B} & i\ \text{R} \end{array}$$

where $i < k$, $\mathcal{B}$ is accessible at line k, and $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}$ appears at the same scope on the line numbered i of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{B} & \\ & & \\ k.1 & \mathcal{B} \rightarrow \mathcal{B} & \text{T3.1} \\ k & \mathcal{B} & k.1, i\ \text{MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\wedge$E) If $\mathcal{Q}_k$ arises by $\wedge$E, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k & \mathcal{B} & i\ \wedge\text{E} \end{array} \qquad \text{or} \qquad \begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k & \mathcal{C} & i\ \wedge\text{E} \end{array}$$

where $i < k$ and $\mathcal{B} \wedge \mathcal{C}$ is accessible at line k. In the first case, $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \wedge \mathcal{C}$ appears at the same scope on the line numbered i of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k.1 & (\mathcal{B} \wedge \mathcal{C}) \rightarrow \mathcal{B} & \text{T3.21} \\ k & \mathcal{B} & k.1, i\ \text{MP} \end{array}$$

[2]There may be an application of Gen in the derivation of T3.33 for =I. However, as mentioned on page 424, derivations for theorems of the sort $\vdash_{AD} \mathcal{P}$ may appear at the top, and so outside the scope of any undischarged assumptions.

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good. And similarly in the other case, by application of T3.20.

($\wedge$I) If $\mathcal{Q}_k$ arises from previous lines by $\wedge$I, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ j & \mathcal{C} & \\ & & \\ k & \mathcal{B} \wedge \mathcal{C} & i,j\ \wedge\text{I} \end{array}$$

where $i, j < k$, $\mathcal{B}$ and $\mathcal{C}$ are accessible at line k, and $\mathcal{Q}_k = \mathcal{B} \wedge \mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}$ and $\mathcal{C}$ appear at the same scope on the lines numbered i and j of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{B} & \\ j & \mathcal{C} & \\ & & \\ k.1 & \mathcal{B} \to (\mathcal{C} \to (\mathcal{B} \wedge \mathcal{C})) & \text{T9.4} \\ k.2 & \mathcal{C} \to (\mathcal{B} \wedge \mathcal{C}) & k.1, i\ \text{MP} \\ k & \mathcal{B} \wedge \mathcal{C} & k.2, j\ \text{MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\to$E) If $\mathcal{Q}_k$ arises from previous lines by $\to$E, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \to \mathcal{C} & \\ j & \mathcal{B} & \\ & & \\ k & \mathcal{C} & i,j\ \to\text{E} \end{array}$$

where $i, j < k$, $\mathcal{B} \to \mathcal{C}$ and $\mathcal{B}$ are accessible at line k, and $\mathcal{Q}_k = \mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \to \mathcal{C}$ and $\mathcal{B}$ appear at the same scope on the lines numbered i and j of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{B} \to \mathcal{C} & \\ j & \mathcal{B} & \\ & & \\ k & \mathcal{C} & i,j\ \text{MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\to$I) If $\mathcal{Q}_k$ arises by $\to$I, then N is something like this,

$$\begin{array}{r|ll}
i & \vert\ \mathcal{B} & \\
 & \vert & \\
j & \vert\ \mathcal{C} & \\
k & \mathcal{B} \to \mathcal{C} & i\text{-}j \ {\to}\mathrm{I}
\end{array}$$

where $i, j < k$, the subderivation is accessible at line k, and $\mathcal{Q}_k = \mathcal{B} \to \mathcal{C}$. By assumption A_{k-1} matches N through line $k - 1$ and is good. So $\mathcal{B}$ and $\mathcal{C}$ appear at the same scope on the lines numbered i and j of A_{k-1}; since they appear at the same scope, the parallel subderivation is accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}$ is to a variable free in $\mathcal{B}$. So let A_k continue as follows:

$$\begin{array}{r|ll}
i & \vert\ \mathcal{B} & \\
 & \vert & \\
j & \vert\ \mathcal{C} & \\
k & \mathcal{B} \to \mathcal{C} & i\text{-}j \ \mathrm{DT}
\end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen in this derivation, A_k is good.

($\sim$E) If $\mathcal{Q}_k$ arises by $\sim$E , then N is something like this (reverting to the unabbreviated form),

$$\begin{array}{r|ll}
i & \vert\ {\sim}\mathcal{B} & \\
 & \vert & \\
j & \vert\ \mathcal{C} \wedge {\sim}\mathcal{C} & \\
k & \mathcal{B} & i\text{-}j \ {\sim}\mathrm{E}
\end{array}$$

where $i, j < k$, the subderivation is accessible at line k, and $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k - 1$ and is good. So ${\sim}\mathcal{B}$ and $\mathcal{C} \wedge {\sim}\mathcal{C}$ appear at the same scope on the lines numbered i and j of A_{k-1}; since they appear at the same scope, the parallel subderivation is accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of ${\sim}\mathcal{B}$ is to a variable free in ${\sim}\mathcal{B}$. So let A_k continue as follows:

$$\begin{array}{r|ll}
i & \vert\ {\sim}\mathcal{B} & \\
 & \vert & \\
j & \vert\ \mathcal{C} \wedge {\sim}\mathcal{C} & \\
k.1 & {\sim}\mathcal{B} \to (\mathcal{C} \wedge {\sim}\mathcal{C}) & i\text{-}j \ \mathrm{DT} \\
k.2 & (\mathcal{C} \wedge {\sim}\mathcal{C}) \to \mathcal{C} & \mathrm{T3.21} \\
k.3 & (\mathcal{C} \wedge {\sim}\mathcal{C}) \to {\sim}\mathcal{C} & \mathrm{T3.20} \\
k.4 & {\sim}\mathcal{B} \to \mathcal{C} & k.1, k.2 \ \mathrm{T3.2} \\
k.5 & {\sim}\mathcal{B} \to {\sim}\mathcal{C} & k.1, k.3 \ \mathrm{T3.2} \\
k.6 & ({\sim}\mathcal{B} \to {\sim}\mathcal{C}) \to (({\sim}\mathcal{B} \to \mathcal{C}) \to \mathcal{B}) & \mathrm{A3} \\
k.7 & ({\sim}\mathcal{B} \to \mathcal{C}) \to \mathcal{B} & k.6, k.5 \ \mathrm{MP} \\
k & \mathcal{B} & k.7, k.4 \ \mathrm{MP}
\end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\sim$I) Homework.

($\vee$E) If $\mathcal{Q}_k$ arises by $\vee$E, then N is something like this,

$$
\begin{array}{l|l}
f & \mathcal{B} \vee \mathcal{C} \\
g & \quad \mathcal{B} \\
h & \quad \mathcal{D} \\
i & \quad \mathcal{C} \\
j & \quad \mathcal{D} \\
k & \mathcal{D} \qquad f,g\text{-}h,i\text{-}j \ \vee\text{E}
\end{array}
$$

where $f, g, h, i, j < k$, $\mathcal{B} \vee \mathcal{C}$ and the two subderivations are accessible at line k, and $\mathcal{Q}_k = \mathcal{D}$. By assumption A_{k-1} matches N through line $k - 1$ and is good. So the formulas at lines numbered f, g, h, i, j appear at the same scope on corresponding lines in A_{k-1}; since they appear at the same scope, $\mathcal{B} \vee \mathcal{C}$ and the corresponding subderivations are accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}$ is to a variable free in $\mathcal{B}$, and no application of Gen under the scope of $\mathcal{C}$ is to a variable free in $\mathcal{C}$. So let A_k continue as follows:

$$
\begin{array}{r|ll}
f & \mathcal{B} \vee \mathcal{C} & \\
g & \quad \mathcal{B} & \\
h & \quad \mathcal{D} & \\
i & \quad \mathcal{C} & \\
j & \quad \mathcal{D} & \\
k.1 & \mathcal{B} \to \mathcal{D} & g\text{-}h \text{ DT} \\
k.2 & \mathcal{C} \to \mathcal{D} & i\text{-}j \text{ DT} \\
k.3 & (\mathcal{B} \to \mathcal{D}) \to [(\mathcal{C} \to \mathcal{D}) \to ((\mathcal{B} \vee \mathcal{C}) \to \mathcal{D})] & \text{T9.5} \\
k.4 & (\mathcal{C} \to \mathcal{D}) \to ((\mathcal{B} \vee \mathcal{C}) \to \mathcal{D}) & k.3, k.1 \text{ MP} \\
k.5 & (\mathcal{B} \vee \mathcal{C}) \to \mathcal{D} & k.4, k.2 \text{ MP} \\
k & \mathcal{D} & k.5, f \text{ MP}
\end{array}
$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\vee$I) Homework.

($\leftrightarrow$E) Homework.

($\leftrightarrow$I) Homework.

($\forall$E) Homework.

(∀I) If $\mathcal{Q}_k$ arises by ∀I, then N looks something like this,

$$
\begin{array}{r|ll}
i & \mathcal{B}^x_v & \\
 & & \\
k & \forall x\mathcal{B} & i\ \forall\mathrm{I}
\end{array}
$$

where $i < k$, $\mathcal{B}^x_v$ is accessible at line k, and $\mathcal{Q}_k = \forall x\mathcal{B}$; further the *ND* restrictions on ∀I are met: (i) v is free for x in $\mathcal{B}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}^x_v$ appears at the same scope on the line numbered i of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows:

$$
\begin{array}{r|ll}
0.k & \forall v\mathcal{B}^x_v \rightarrow \forall x\mathcal{B} & \mathrm{T9.11} \\
 & & \\
i & \mathcal{B}^x_v & \\
 & & \\
k.1 & \forall v\mathcal{B}^x_v & i\ \mathrm{Gen} \\
k & \forall x\mathcal{B} & 0.k,\ k.1\ \mathrm{MP}
\end{array}
$$

From constraint (iii) v is not free in $\forall x\mathcal{B}$ and by (i) v is free for x in $\mathcal{B}$, so $0.k$ is an instance of T9.11. So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. This time, there is an application of Gen at $k.1$. But A_{k-1} is good; so no application of Gen in lines up to $k-1$ is to a variable free in an undischarged assumption. And since A_k matches N and by (ii) v is free in no undischarged auxiliary assumption of N, v is not free in any undischarged auxiliary assumption of A_k. There is also an application of Gen in T9.11 at $0.k$; but that derivation is under the scope of no undischarged assumptions. So A_k is good. (Notice that, in this reasoning, we appeal to each of the restrictions that apply to ∀I in N.)

(∃E) If $\mathcal{Q}_k$ arises by ∃E, then N looks something like this,

$$
\begin{array}{r|l|ll}
h & \multicolumn{2}{l}{\exists x\mathcal{B}} & \\
i & & \underline{\mathcal{B}^x_v} & \\
 & & & \\
j & & \mathcal{C} & \\
k & \multicolumn{2}{l}{\mathcal{C}} & h,i\text{-}j\ \exists\mathrm{E}
\end{array}
$$

where $h, i, j < k$, $\exists x\mathcal{B}$ and the subderivation are accessible at line k, and $\mathcal{Q}_k = \mathcal{C}$; further, the *ND* restrictions on ∃E are met: (i) v is free for x in $\mathcal{B}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\exists x\mathcal{B}$ or in $\mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So the formulas at lines numbered h, i, and j appear at the same scope on corresponding lines in A_{k-1}; since they appear at the same scope, $\exists x\mathcal{B}$ and the corresponding subderivation are accessible in

A_{k-1}. Since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}^x_v$ is to a variable free in $\mathcal{B}^x_v$. So let A_k continue as follows:

$0.k$	$\forall v {\sim}\mathcal{B}^x_v \to \forall x {\sim}\mathcal{B}$	T9.11
h	$\exists x \mathcal{B}$	
i	$\quad\mid \mathcal{B}^x_v$	
	$\quad\mid$	
j	$\quad\mid \mathcal{C}$	
$k.1$	$\mathcal{B}^x_v \to \mathcal{C}$	i-j DT
$k.2$	$\exists v \mathcal{B}^x_v \to \mathcal{C}$	$k.1$ T3.32
$k.3$	$(\forall v {\sim}\mathcal{B}^x_v \to \forall x {\sim}\mathcal{B}) \to ({\sim}\forall x {\sim}\mathcal{B} \to {\sim}\forall v {\sim}\mathcal{B}^x_v)$	T3.13
$k.4$	${\sim}\forall x {\sim}\mathcal{B} \to {\sim}\forall v {\sim}\mathcal{B}^x_v$	$k.3$, $0.k$ MP
$k.5$	$\exists x \mathcal{B} \to \exists v \mathcal{B}^x_v$	$k.4$ abv
$k.6$	$\exists v \mathcal{B}^x_v$	$k.5$, h MP
k	$\mathcal{C}$	$k.2$, $k.6$ MP

From constraint (iii), that v is not free in $\mathcal{C}$, $k.2$ meets the restriction on T3.32. By (iii) v is not free in $\exists x \mathcal{B}$ and so in $\forall x {\sim}\mathcal{B}$ and by (i) v is free for x in $\mathcal{B}$ and so in ${\sim}\mathcal{B}$, so $0.k$ is an instance of T9.11. So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. The application of T3.32 at $k.2$ includes an application Gen to v. But A_{k-1} is good; so no application of Gen in lines up to $k-1$ is to a variable free in an undischarged assumption. And since A_k matches N and by (ii) v is free in no undischarged auxiliary assumption of N, v is not free in any undischarged auxiliary assumption of A_k. There is also an application of Gen in T9.11 at $0.k$ but that derivation is under the scope of no undischarged assumptions. So A_k is good. (Notice again that we appeal to each of the restrictions that apply to $\exists$E in N.)

($\exists$I) Homework.

($=$E) Homework.

In any case, A_k matches N through line k and is good.

Indct: Derivation A matches N and is good.

So if there is an *ND* derivation to show $\Gamma \vdash_{ND} \mathcal{P}$, then there is a good matching derivation A to show the same; so with the deduction theorem, $\Gamma \vdash_{AD} \mathcal{P}$; and if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

From this theorem together with T9.2, *AD* and *ND* are equivalent; that is, $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$. Given this, we will often ignore the difference between *AD* and *ND* and simply write $\Gamma \vdash \mathcal{P}$ when there is a(n *AD* or *ND*) derivation of $\mathcal{P}$ from premises in Γ. Also given the equivalence between the systems, we are in a position to *transfer* results from one system to the other without demonstrating them directly for both.

We will come to appreciate this, especially the relative ease of operating in *ND* and of operating on *AD*.

As before, given any *ND* derivation, we can use the method of our induction to find a corresponding *AD* derivation. For a simple example, consider the following demonstration that $\sim A \rightarrow (A \wedge B) \vdash_{ND} A$:

(H)

1.	$\sim A \rightarrow (A \wedge B)$	P
2.	$\quad \sim A$	A (c, $\sim$E)
3.	$\quad A \wedge B$	1,2 $\rightarrow$E
4.	$\quad A$	3 $\wedge$E
5.	$\quad A \wedge \sim A$	4,2 $\wedge$I
6.	A	2-5 $\sim$E

Given relevant cases from the induction, the corresponding *AD* derivation is as follows:

1	$\sim A \rightarrow (A \wedge B)$	prem
2	$\quad \sim A$	assp (c, DT)
3	$\quad A \wedge B$	1,2 MP
4.1	$\quad (A \wedge B) \rightarrow A$	T3.21
4	$\quad A$	4.1, 3 MP
5.1	$\quad A \rightarrow (\sim A \rightarrow (A \wedge \sim A))$	T9.4
5.2	$\quad \sim A \rightarrow (A \wedge \sim A)$	5.1, 4 MP
5	$\quad A \wedge \sim A$	5.2, 2 MP
6.1	$\sim A \rightarrow (A \wedge \sim A)$	2-5 DT
6.2	$(A \wedge \sim A) \rightarrow A$	T3.21
6.3	$(A \wedge \sim A) \rightarrow \sim A$	T3.20
6.4	$\sim A \rightarrow A$	6.1, 6.2 T3.2
6.5	$\sim A \rightarrow \sim A$	6.1, 6.3 T3.2
6.6	$(\sim A \rightarrow \sim A) \rightarrow ((\sim A \rightarrow A) \rightarrow A)$	A3
6.7	$(\sim A \rightarrow A) \rightarrow A$	6.6, 6.5 MP
6	A	6.7, 6.4 MP

For the first two lines, we simply take over the premise and assumption from the *ND* derivation. For (3), the induction uses MP in *AD* where $\rightarrow$E appears in *ND;* so that is what we do. For (4), our induction shows that we can get the effect of $\wedge$E by appeal to T3.21 with MP. (5) in the *ND* derivation is by $\wedge$I, and, as above, we get the same effect by T9.4 with MP. (6) in the *ND* derivation is by $\sim$E. Following the strategy from the induction, we set up for application of A3 by getting the conditional by DT. As usual, the constructed derivation is not very efficient. You should be able to get the same result in just five lines by appeal to T3.21, T3.2, and then T3.7. But, again, the point is just to show that there always *is* a corresponding derivation.

*E9.14. Set up the above induction for T9.12 and complete the unfinished cases to show that if $\Gamma \vdash_{ND} \mathscr{P}$, then $\Gamma \vdash_{AD} \mathscr{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.15. Consider the following *ND* derivation and, using the method from the induction for T9.12, construct a derivation to show $\exists x(C \wedge Bx) \vdash_{AD} C$.

1.	$\exists x(C \wedge Bx)$	P
2.	$\quad C \wedge By$	A (g, 1$\exists$E)
3.	$\quad C$	2 $\wedge$E
4.	C	1,2-3 $\exists$E

Hint: Your derivation should have 12 lines.

*E9.16. Consider the system $A^\star$ from E9.5. As a preliminary to the exercise that follows, where v is not free in $\exists x\mathcal{P}$ and free for x in $\mathcal{P}$, show that $\vdash_{A^\star} \exists x\mathcal{P} \rightarrow \exists v\mathcal{P}^x_v$. Again you may appeal to any of the A^* results from E3.5. This works as an $A^\star$ analog to T9.11.

E9.17. Consider a system N^* which is like *ND* except that its only rules are $\sim$E, $\sim$I, $\wedge$E, $\wedge$I, $\exists$E, and $\exists$I, along with the system $A^\star$ from E9.5. Produce a complete demonstration that if $\Gamma \vdash_{N^*} \mathcal{P}$, then $\Gamma \vdash_{A^\star} \mathcal{P}$. You have the result of the previous exercise, DT from E9.8, and again may use any of the theorems for A^* from E3.5. Hint: You will want to modify the definition of a *good* derivation to accommodate $\exists$R.

9.4 Extending to *ND+*

ND+ adds twenty-six rules to *ND:* the ten inference rules, $\perp$I, $\perp$E, MT, HS, DS, NB, ($\forall$I), ($\forall$E), ($\exists$I), and ($\exists$E) and sixteen replacement rules, DN, Com, Assoc, Idem, Impl, Trans, DeM, Exp, Equiv, Dist, Sym, QS, QD, QP, QN, and RQN—where some of these have multiple forms. It might seem tedious to go through all the cases but, as it happens, we have already done most of the work. First, it is easy to see that,

T9.13. If $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{ND+} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND} \mathcal{P}$. Then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ. But since every rule of *ND* is a rule of *ND+*, N is a derivation in *ND+* as well. So $\Gamma \vdash_{ND+} \mathcal{P}$.

From T9.2 and T9.13, then, the situation is as follows:

$$\Gamma \vdash_{AD} \mathcal{P} \xrightarrow{9.2} \Gamma \vdash_{ND} \mathcal{P} \xrightarrow{9.13} \Gamma \vdash_{ND+} \mathcal{P}$$

If an argument is valid in *AD*, it is valid in *ND*, and in *ND+*. From T9.12, the leftmost arrow is a biconditional. Again, however, one might think that *ND+* has more resources than *ND*, so that more could be derived in *ND+* than *ND*. But this is not so.

To see this, we might begin with the closer systems *ND* and *ND+* and attempt to show that anything derivable in *ND+* is derivable in *ND*. Alternatively, we choose simply to expand the induction of the previous section to include cases for all the rules of *ND+*. The result is a demonstration that if $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. Given this, the three systems are connected in a "loop"—so that if there is a derivation in any one of the systems, there is a derivation in the others as well.

*T9.14. If $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND+} \mathcal{P}$; then there is an *ND+* derivation N of $\mathcal{P}$ from premises in Γ. We show that for any i, there is a good *AD* derivation A_i that matches N through line i.

Basis: The first line of N is a premise, an assumption, or arises by =I. Let A_1 be the same, in the latter case with justification T3.33. Then A_1 matches N; and since there is no application of Gen under an undischarged assumption, A_1 is good.

Assp: For any i, $0 \le i < k$, there is a good derivation A_i that matches N through line i.

Show: There is a good derivation of A_k that matches N through line k.

Either $\mathcal{Q}_k$ is a premise or assumption, arises by a rule of *ND*, or by the *ND+* derivation rules ⊥I, ⊥E, MT, HS, DS, NB, (∀I), (∀E), (∃I), (∃E), or by replacement rules DN, Com, Assoc, Idem, Impl, Trans, DeM, Exp, Equiv, Dist, Sym, QS, QD, QP, QN, or RQN. If $\mathcal{Q}_k$ is a premise or assumption or arises by a rule of *ND*, then by reasoning as for T9.12, there is a good derivation A_k that matches N through line k. So suppose $\mathcal{Q}_k$ arises by one of the *ND+* rules.

⊥I. If $\mathcal{Q}_k$ arises from previous lines by ⊥I, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{A} & \\ j & \sim\mathcal{A} & \\ & & \\ k & \bot & i,j\ \bot\text{I} \end{array} \qquad \text{which is the same as} \qquad \begin{array}{r|l} i & \mathcal{A} \\ j & \sim\mathcal{A} \\ & \\ k & \mathcal{Z} \wedge \sim\mathcal{Z} \end{array}$$

for some sentence $\mathcal{Z}$ of the language $\mathcal{L}$. Working on the right-hand version, $i, j < k$, $\mathcal{A}$ and $\sim\mathcal{A}$ are accessible at line k, and $\mathcal{Q}_k = \mathcal{Z} \wedge \sim\mathcal{Z}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{A}$ and $\sim\mathcal{A}$ appear at the same scope on the lines numbered i and j of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{A} & \\ j & \sim\mathcal{A} & \\ & & \\ k.1 & \sim\mathcal{A} \to (\mathcal{A} \to (\mathcal{Z} \wedge \sim\mathcal{Z})) & \text{T3.9} \\ k.2 & \mathcal{A} \to (\mathcal{Z} \wedge \sim\mathcal{Z}) & k.1,\ j\ \text{MP} \\ k & \mathcal{Z} \wedge \sim\mathcal{Z} & k.2,\ i\ \text{MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

⊥E. Homework.

MT. If $\mathcal{Q}_k$ arises from previous lines by MT, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \to \mathcal{C} & \\ j & \sim\mathcal{C} & \\ & & \\ k & \sim\mathcal{B} & i,j \text{ MT} \end{array}$$

where $i, j < k$, $\mathcal{B} \to \mathcal{C}$ and $\sim\mathcal{C}$ are accessible at line k, and $\mathcal{Q}_k = \sim\mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \to \mathcal{C}$ and $\sim\mathcal{C}$ appear at the same scope on the lines numbered i and j of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows:

$$\begin{array}{r|ll} i & \mathcal{B} \to \mathcal{C} & \\ j & \sim\mathcal{C} & \\ & & \\ k.1 & (\mathcal{B} \to \mathcal{C}) \to (\sim\mathcal{C} \to \sim\mathcal{B}) & \text{T3.13} \\ k.2 & \sim\mathcal{C} \to \sim\mathcal{B} & k.1, i \text{ MP} \\ k & \sim\mathcal{B} & k.2, j \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

HS. Homework.

DS. Homework.

NB. Homework.

(∀I). If $\mathcal{Q}_k$ arises from previous lines by (∀I), then N is something like this,

$$\begin{array}{r|l|l} i & \underline{\mathcal{B}^x_v} & \\ & & \\ j & \mathcal{C}^x_v & \\ k & (\forall x : \mathcal{B})\mathcal{C} & i\text{-}j\ (\forall\text{I}) \end{array} \quad \text{which is the same as} \quad \begin{array}{r|l} i & \underline{\mathcal{B}^x_v} \\ & \\ j & \mathcal{C}^x_v \\ k & \forall x(\mathcal{B} \to \mathcal{C}) \end{array}$$

Working on the right-hand version, $i, j < k$, the subderivation is accessible at k, and $\mathcal{Q}_k$ is $\forall x(\mathcal{B} \to \mathcal{C})$; further, the restrictions on (∀I) are met: (i) v is free for x in $\mathcal{B}$ and $\mathcal{C}$, (ii) v is not free in any undischarged assumption, and (iii) v is not free in $\forall x(\mathcal{B} \to \mathcal{C})$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}^x_v$ and $\mathcal{C}^x_v$ appear at the same scope on the lines numbered i and j of A_{k-1}; since they appear at the same scope, the parallel subderivation is accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}^x_v$ is to a variable free in $\mathcal{B}^x_v$; so let A_k continue as follows:

$0.k$	$\forall v(\mathcal{B} \to \mathcal{C})^x_v \to \forall x(\mathcal{B} \to \mathcal{C})$	T9.11
i	$\quad\mathcal{B}^x_v$	
j	$\quad\mathcal{C}^x_v$	
$k.1$	$(\mathcal{B} \to \mathcal{C})^x_v$	i-j DT
$k.2$	$\forall v(\mathcal{B} \to \mathcal{C})^x_v$	$k.1$ Gen
k	$\forall x(\mathcal{B} \to \mathcal{C})$	$0.k, k.2$ MP

From constraint (iii) v is not free in $\forall x(\mathcal{B} \to \mathcal{C})$ and by (i) v is free for x in $(\mathcal{B} \to \mathcal{C})$, so $0.k$ is an instance of T9.11. So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. This time, there is an application of Gen at $k.2$. But A_{k-1} is good; so no application of Gen in lines up to $k-1$ is to a variable free in an undischarged assumption. And since A_k matches N and by (ii) v is free in no undischarged auxiliary assumption of N, v is not free in any undischarged auxiliary assumption of A_k. There is also an application of Gen at $0.k$; but that derivation is under the scope of no undischarged assumptions. So A_k is good.

($\forall$E). Homework.

($\exists$I). Homework.

($\exists$E). Homework.

rep. If $\mathcal{Q}_k$ arises from a replacement rule *rep* of the form $\mathcal{C} \lhd\rhd \mathcal{D}$, then N is something like this,

i	$\mathcal{B}$		or	i	$\mathcal{B}$	
k	$\mathcal{B}^{\mathcal{C}}/\!\!/_{\mathcal{D}}$	i *rep*		k	$\mathcal{B}^{\mathcal{D}}/\!\!/_{\mathcal{C}}$	i *rep*

where $i < k$, $\mathcal{B}$ is accessible at line k, and, in the first case, $\mathcal{Q}_k = \mathcal{B}^{\mathcal{C}}/\!\!/_{\mathcal{D}}$. By assumption A_{k-1} matches N through line $k-1$ and is good. But by T6.43, T6.11–T6.30, T6.32–T6.38, and T6.45, $\vdash_{ND} \mathcal{C} \leftrightarrow \mathcal{D}$; so with T9.12, $\vdash_{AD} \mathcal{C} \leftrightarrow \mathcal{D}$; so by T9.9, $\vdash_{AD} \mathcal{B} \leftrightarrow \mathcal{B}^{\mathcal{C}}/\!\!/_{\mathcal{D}}$. Call an arbitrary particular result of this sort, *Tx*, and augment A_k as follows:

$0.k$	$\mathcal{B} \leftrightarrow \mathcal{B}^{\mathcal{C}}/\!\!/_{\mathcal{D}}$	*Tx*
i	$\mathcal{B}$	
k	$\mathcal{B}^{\mathcal{C}}/\!\!/_{\mathcal{D}}$	$0.k, i$ T3.24

So $\mathcal{Q}_k$ appears at the same scope on the line numbered k of A_k; so A_k matches N through line k. There may be applications of Gen in the derivation of *Tx;* but that derivation is under the scope of no undischarged assumption. And under the scope of any undischarged assumptions, there is no new application of Gen. So A_k is good. And similarly in the other case starting with T6.12 to obtain $\vdash_{ND} \mathcal{D} \leftrightarrow \mathcal{C}$ from $\vdash_{ND} \mathcal{C} \leftrightarrow \mathcal{D}$.

In any case, A_k matches N through line k and is good.

Indct: Derivation A matches N and is good.

That is it! The key is that work we have already done collapses cases for all the replacement rules into one. So each of the derivation systems, *AD*, *ND*, and *ND+* is equivalent to the others. That is, $\Gamma \vdash_{AD} \mathcal{P}$ iff $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{ND+} \mathcal{P}$. And that is what we set out to show.

*E9.18. Set up the above induction for T9.14 and complete the unfinished cases to show that if $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.19. Extend the system N^* from E9.17 to an $N^\star$ that has rules $\sim$E, $\sim$I, $\wedge$E, $\wedge$I, $\exists$E, $\exists$I, along with MT and the replacement rule Com (for $\wedge$). Augment your argument from E9.17 to produce a complete demonstration that if $\Gamma \vdash_{N^\star} \mathcal{P}$ then $\Gamma \vdash_{A^\star} \mathcal{P}$. In addition to E9.17, you may appeal to any of the theorems from E3.5 along with the substitution result from E9.13.

E9.20. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The reason semantic validity implies logical validity, but not the other way around.

b. The notion of a *constructive* proof by mathematical induction.

c. The equivalence between derivation systems *AD*, *ND*, and *ND+*.

Theorems of Chapter 9

T9.1 For any ordinary argument $\mathcal{P}_1 \dots \mathcal{P}_n / \mathcal{Q}$, with good translation consisting of II and $\mathcal{P}'_1 \dots \mathcal{P}'_n$, $\mathcal{Q}'$, if $\mathcal{P}'_1 \dots \mathcal{P}'_n \vDash \mathcal{Q}'$, then $\mathcal{P}_1 \dots \mathcal{P}_n / \mathcal{Q}$ is logically valid.

T9.2 If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

T9.3 If $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$, and no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, then $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$. *Deduction Theorem.*

T9.4 $\vdash_{AD} \mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B}))$.

T9.5 $\vdash_{AD} (\mathcal{A} \to \mathcal{C}) \to [(\mathcal{B} \to \mathcal{C}) \to ((\mathcal{A} \vee \mathcal{B}) \to \mathcal{C})]$.

T9.6 For arbitrary terms r, s, and t, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

T9.7 For any formula $\mathcal{P}$ and terms r and s, if s is free for any replaced instance of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \to (\mathcal{P} \to \mathcal{P}^{r}/\!\!/_{s})$.

T9.8 For any formula $\mathcal{P}$ and terms r and s, if s is free any replaced instances of r in $\mathcal{P}$, then $\vdash_{AD} (r = s) \to (\mathcal{P} \to \mathcal{P}^{r}/_{s})$.

T9.9 For any formulas $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{Q}$, if $\Gamma \vdash_{AD} \mathcal{P} \leftrightarrow \mathcal{Q}$, then $\Gamma \vdash_{AD} \mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}$.

T9.10 For any formulas $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{Q}$, interpretation I, and variable assignment d, if $\mathsf{I_d}[\mathcal{P}] = \mathsf{I_d}[\mathcal{Q}]$ then $\mathsf{I_d}[\mathcal{O}] = \mathsf{I_d}[\mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}]$.

Corollary: If $\mathsf{I_d}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathcal{O} \leftrightarrow \mathcal{O}^{\mathcal{P}}/\!\!/_{\mathcal{Q}}] = \mathsf{S}$.

T9.11 $\vdash_{AD} \forall v \mathcal{P}^{x}_{v} \to \forall x \mathcal{P}$ where v is not free in $\forall x \mathcal{P}$ and free for x in $\mathcal{P}$.

T9.12 If $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

T9.13 If $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{ND+} \mathcal{P}$.

T9.14 If $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

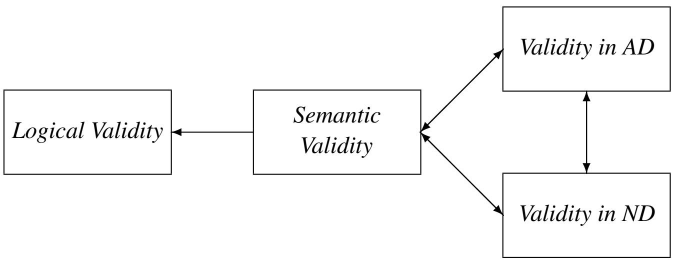
And from T9.2, T9.13, and T9.14,

$\Gamma \vdash_{AD} \mathcal{P}$ iff $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{ND+} \mathcal{P}$.

Chapter 10

Main Results

We have introduced four notions of validity, and started to think about their interrelations. In Chapter 9, we showed that if an argument is semantically valid, then it is logically valid, and that an argument is valid in *AD* iff it is valid in *ND*. We turn now to the relation between these derivation systems and semantic validity. This completes the project of demonstrating that the different notions of validity are related as follows:



Since *AD* and *ND* are equivalent, it is not necessary separately to establish the relations between *AD* and semantic validity, and between *ND* and semantic validity. Because it is relatively easy to reason about *AD*, we mostly reason about a system like *AD* to establish that an argument is valid in *AD* iff it is semantically valid. From the equivalence between *AD* and *ND* it then follows that an argument is valid in *ND* iff it is semantically valid.

The project divides into two parts. First, we take up the arrows from right to left, and show that if an argument is valid in *AD*, then it is semantically valid: if $\Gamma \vdash_{AD} \mathscr{P}$, then $\Gamma \vDash \mathscr{P}$. Thus our derivation system is *s̊ound* (recall from page 408 that diacritical marks distinguish notions of soundness and completeness). If a derivation system is s̊ound, it never leads from premises that are true on an interpretation, to a conclusion that is not (section 10.1). Second, moving in the other direction, we show that if an argument is semantically valid, then it is valid in *AD:* if $\Gamma \vDash \mathscr{P}$, then $\Gamma \vdash_{AD} \mathscr{P}$. Thus our derivation system is *c̊omplete*. If a derivation system is c̊omplete, there is a derivation from the premises to the conclusion for every argument that is semantically valid. The argument for c̊ompleteness divides into sentential (section 10.2), basic quantificational (section 10.3), and full quantificational (section 10.4) versions.

10.1 S̊oundness

An arbitrary derivation system *DS* is *s̊ound* when its provable results are semantically valid: if $\Gamma \vdash_{DS} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. It is easy to construct derivation systems that are not s̊ound. An obvious example is the preliminary system *NP* from Chapter 6—for, as we showed in table (D) of Chapter 6 (page 201), R2 makes it possible to go from a true premise to a false conclusion. Or consider a derivation system like *AD* but without the restriction on A4 that the substituted term t be free for the variable x in formula $\mathcal{P}$. Given this, we might reason as follows:

(A)

1.	$\forall x \exists y {\sim}(x = y)$	prem
2.	$\forall x \exists y {\sim}(x = y) \to \exists y {\sim}(y = y)$	"A4"
3.	$\exists y {\sim}(y = y)$	1,2 MP

The y is not free for x in $\exists y {\sim}(x = y)$; so line (2) is not an instance of A4. And it is a good thing: Consider any interpretation with at least two elements in U. Then it is true that for every x there is some y not identical to it. So the premise is true. But there is no y in U that is not identical to itself. So the conclusion is not true. So the true premise leads to a conclusion that is not true. So the derivation system is not s̊ound.

We would like to show that *AD* is s̊ound—that there is no sequence of moves, no matter how complex or clever, that would lead from premises that are true to a conclusion that is not true. The argument itself is straightforward: Suppose $\Gamma \vdash_{AD} \mathcal{P}$; then there is an *AD* derivation $A = \langle \mathcal{Q}_1 \ldots \mathcal{Q}_n \rangle$ of $\mathcal{P}$ with $\mathcal{Q}_n = \mathcal{P}$. By induction on line numbers in A, we show that for any i, $\Gamma \vDash \mathcal{Q}_i$. The case when $i = n$ is the desired result. So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. This general strategy should by now be familiar. However, for the case involving A4, it will be helpful to obtain a pair of preliminary results.

10.1.1 Switching Theorems

In this section, we develop a couple theorems which link substitutions into terms and formulas with substitutions in variable assignments. The results are a matched pair, with a first result for terms that feeds into the basis clause for a result about formulas. Perhaps the hardest part is not so much the proofs of the theorems, as understanding what the theorems say. Let us turn to the first.

Suppose we have some terms t and r with interpretation I and variable assignment d. Say $\mathsf{I_d}[r] = \mathsf{o}$. Then the first proposition is this: term t is assigned the same object on $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}$, as t^x_r is assigned on $\mathsf{I_d}$. Intuitively, this is because the same object is fed into the x-place of the term in each case. With t and $\mathsf{d}(x|\mathsf{o})$,

$$\text{(B)}\qquad \begin{array}{rl} t: & h^n \dots x \dots \\ & \qquad\;\; | \\ \mathsf{d}(x|\mathsf{o}): & \quad \dots \mathsf{o} \dots \end{array}$$

object o is the input to the "slot" occupied by x. But we are given that $\mathsf{I_d}[r] = \mathsf{o}$. So with t^x_r and d,

$$\text{(C)}\qquad \begin{array}{rl} t^x_r: & h^n \dots r \dots \\ & \qquad\;\; | \\ \mathsf{d}: & \quad \dots \mathsf{o} \dots \end{array}$$

object o is the input into the "slot" that was occupied by x. So if $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t] = \mathsf{I_d}[t^x_r]$. In the one case, we guarantee that object o goes into the x-place by meddling with the variable assignment. In the other, we get the same result by meddling with the term. Be sure you are clear about this in your own mind. This will be our first result.

*T10.1. For any interpretation I, variable assignment d, with terms t and r, if $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t] = \mathsf{I_d}[t^x_r]$.

For arbitrary terms t and r with interpretation I and variable assignment d, suppose $\mathsf{I_d}[r] = \mathsf{o}$. By induction on the number of function symbols in t, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t] = \mathsf{I_d}[t^x_r]$.

Basis: If t has no function symbols, then it is a constant or a variable. Either t is the variable x or it is not. (i) Suppose t is a constant or variable other than x; then $t^x_r = t$ (no replacement is made); but d and $\mathsf{d}(x|\mathsf{o})$ assign just the same things to variables other than x; so they assign just the same things to any variable in t; so by T8.4, $\mathsf{I_d}[t] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$. So $\mathsf{I_d}[t^x_r] = \mathsf{I_d}[t] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$. (ii) If t is x, then t^x_r is r (all of t is replaced by r); so $\mathsf{I_d}[t^x_r] = \mathsf{I_d}[r] = \mathsf{o}$. But t is x; so $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[x]$; by TA(v) this is $\mathsf{d}(x|\mathsf{o})[x]$; which is just o. So $\mathsf{I_d}[t^x_r] = \mathsf{o} = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Assp: For any i, $0 \le i < k$, for t with i function symbols, $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Show: If t has k function symbols, then $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

If t has k function symbols, then it is of the form, $h^n s_1 \dots s_n$ where $s_1 \dots s_n$ have $< k$ function symbols. In this case, $t^x_r = [h^n s_1 \dots s_n]^x_r = h^n s_1{}^x_r \dots s_n{}^x_r$; and by assumption, $\mathsf{I_d}[s_1{}^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_1]$ and … and $\mathsf{I_d}[s_n{}^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_n]$. So $\mathsf{I_d}[t^x_r] = \mathsf{I_d}[h^n s_1{}^x_r \dots s_n{}^x_r]$; by TA(f), this is $\mathsf{I}[h^n]\langle \mathsf{I_d}[s_1{}^x_r] \dots \mathsf{I_d}[s_n{}^x_r]\rangle = \mathsf{I}[h^n]\langle \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_1] \dots \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_n]\rangle$; and by TA(f) again, this is $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[h^n s_1 \dots s_n] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$. So $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Indct: For any t, $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Since the "switching" leaves assignments to the parts the same, the assignment to the whole remains the same as well.

Similarly, suppose we have term r with interpretation I and variable assignment d, where $\mathsf{I_d}[r] = \mathsf{o}$ as before. Suppose r is free for variable x in formula $\mathcal{Q}$. Then the second proposition is that $\mathcal{Q}$ is satisfied on $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}$ iff $\mathcal{Q}^x_r$ is satisfied on I_d. Again, intuitively, this is because the same object is fed into the x-place of the formula in each case. With $\mathcal{Q}$ and $\mathsf{d}(x|\mathsf{o})$,

(D)
$$\begin{array}{rl} \mathcal{Q}\text{:} & \mathcal{Q}\ldots x\ldots \\ & \qquad | \\ \mathsf{d}(x|\mathsf{o})\text{:} & \quad\ldots \mathsf{o}\ldots \end{array}$$

object o is the input to the "slot" occupied by x. But $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$. So with $\mathcal{Q}^x_r$ and d,

(E)
$$\begin{array}{rl} \mathcal{Q}^x_r\text{:} & \mathcal{Q}\ldots r\ldots \\ & \qquad | \\ \mathsf{d}\text{:} & \quad\ldots \mathsf{o}\ldots \end{array}$$

object o is the input into the "slot" that was occupied by x. So if $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$ (and r is free for x in $\mathcal{Q}$), then $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$. This is our second result, which draws directly upon the first.

T10.2. For any interpretation I, variable assignment d, term r, and formula $\mathcal{Q}$, if $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$, and r is free for x in $\mathcal{Q}$, then $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

By induction on the number of operator symbols in $\mathcal{Q}$,

Basis: Suppose r is free for x in $\mathcal{Q}$ and $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$. If $\mathcal{Q}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$. In the first case, $\mathcal{Q}^x_r = \mathcal{S}^x_r = \mathcal{S}$. So $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{S}] = \mathsf{S}$; by SF(s), iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; by SF(s) again, iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$. In the second case, $\mathcal{Q}^x_r = [\mathcal{R}^n t_1 \ldots t_n]^x_r = \mathcal{R}^n t_1{}^x_r \ldots t_n{}^x_r$. So $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\mathcal{R}^n t_1{}^x_r \ldots t_n{}^x_r] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{I}_\mathsf{d}[t_1{}^x_r] \ldots \mathsf{I}_\mathsf{d}[t_n{}^x_r]\rangle \in \mathsf{I}[\mathcal{R}^n]$; since $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$, by T10.1, iff $\langle \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t_1] \ldots \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; by SF(r), iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, if $\mathcal{Q}$ has i operator symbols, r is free for x in $\mathcal{Q}$, and $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$, then $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

Show: If $\mathcal{Q}$ has k operator symbols, r is free for x in $\mathcal{Q}$, and $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$, then $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

Suppose r is free for x in $\mathcal{Q}$ and $\mathsf{I}_\mathsf{d}[r] = \mathsf{o}$. If $\mathcal{Q}$ has k operator symbols, then $\mathcal{Q}$ is of the form $\sim\mathcal{B}$, $\mathcal{B} \to \mathcal{C}$, or $\forall v\, \mathcal{B}$ for variable v and formulas $\mathcal{B}$ and $\mathcal{C}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{Q}$ is $\sim\mathcal{B}$. Then $\mathcal{Q}^x_r = [\sim\mathcal{B}]^x_r = \sim[\mathcal{B}^x_r]$. Since r is free for x in $\mathcal{Q}$, r is free for x in $\mathcal{B}$; so the assumption applies to $\mathcal{B}$. $\mathsf{I}_\mathsf{d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{d}[\sim\mathcal{B}^x_r] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{I}_\mathsf{d}[\mathcal{B}^x_r] \neq \mathsf{S}$; by assumption iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{B}] \neq \mathsf{S}$; by SF($\sim$), iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{B}] = \mathsf{S}$; iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

($\to$) Homework.

($\forall$) Suppose $\mathcal{Q}$ is $\forall v\, \mathcal{B}$. Either there are free occurrences of x in $\mathcal{Q}$ or not.

(i) Suppose there are no free occurrences of x in $\mathcal{Q}$. Then $\mathcal{Q}^x_r$ is just $\mathcal{Q}$ (no replacement is made). But since d and $\mathsf{d}(x|\mathsf{o})$ make just the same assignments to variables other than x, they make just the same assignments to all the variables free in $\mathcal{Q}$; so by T8.5, $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$. So $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$; iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

(ii) Suppose there are free occurrences of x in $\mathcal{Q}$. Then x is some variable other than v, and $\mathcal{Q}^x_r = [\forall v \mathcal{B}]^x_r = \forall v[\mathcal{B}^x_r]$.

First, since r is free for x in $\mathcal{Q}$, r is free for x in $\mathcal{B}$, and v is not a variable in r; from this, for any $\mathsf{m} \in \mathsf{U}$, the variable assignments d and $\mathsf{d}(v|\mathsf{m})$ agree on assignments to variables in r; so by T8.4, $\mathsf{I_d}[r] = \mathsf{I_{d(v|m)}}[r]$; so $\mathsf{I_{d(v|m)}}[r] = \mathsf{o}$; so the requirement of the assumption is met for the assignment $\mathsf{d}(v|\mathsf{m})$ and, as an instance of the assumption, for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{I_{d(v|m)}}[\mathcal{B}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(v|m,x|o)}}[\mathcal{B}] = \mathsf{S}$.

Now suppose $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$ but $\mathsf{I_d}[\mathcal{Q}^x_r] \neq \mathsf{S}$; then $\mathsf{I_{d(x|o)}}[\forall v \mathcal{B}] = \mathsf{S}$ but $\mathsf{I_d}[\forall v \mathcal{B}^x_r] \neq \mathsf{S}$. From the latter, by SF($\forall$), there is some $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{I_{d(v|m)}}[\mathcal{B}^x_r] \neq \mathsf{S}$; so by the above result, $\mathsf{I_{d(v|m,x|o)}}[\mathcal{B}] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{I_{d(x|o)}}[\forall v \mathcal{B}] \neq \mathsf{S}$; this is impossible. And similarly [by homework] in the other direction. So $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$.

If $\mathcal{Q}$ has k operator symbols, r is free for x in $\mathcal{Q}$, and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Indct: For any $\mathcal{Q}$, if r is free for x in $\mathcal{Q}$ and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Perhaps the quantifier case looks more difficult than it is. The key point is that since r is free for x in $\mathcal{Q}$, changes in the assignment to v do not affect the assignment to r. Thus the assumption applies to $\mathcal{B}$ for variable assignments that differ in their assignments to v. This lets us "take the quantifier off," apply the assumption, and then "put the quantifier back on" in the usual way. Another way to make this point is to see how the argument fails when r is not free for x in $\mathcal{Q} = \forall v \mathcal{B}$. If r is not free for x in $\mathcal{Q}$, then a change in the assignment to v may affect the assignment to r. In this case, although $\mathsf{I_d}[r] = \mathsf{o}$, $\mathsf{I_{d(v|m)}}[r]$ might be something else. So there is no reason to think that substituting r for x will have the same effect as assigning o to x. As we shall see, this restriction corresponds directly to the one on axiom A4.

*E10.1. Complete the cases for ($\to$) and ($\forall$) to complete the demonstration of T10.2. You should set up the complete demonstration, but for cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

10.1.2 $\mathring{\text{S}}$oundness

We are now ready for our main proof of $\mathring{\text{s}}$oundness for *AD*. Actually, all the parts are already on the table. It is simply a matter of pulling them together into a complete demonstration.

*T10.3. If $\Gamma \vdash_{AD} \mathscr{P}$, then $\Gamma \vDash \mathscr{P}$. *Soundness.*

Suppose $\Gamma \vdash_{AD} \mathscr{P}$. Then there is an *AD* derivation $A = \langle \mathscr{Q}_1 \ldots \mathscr{Q}_n \rangle$ of $\mathscr{P}$ from premises in Γ, with $\mathscr{Q}_n = \mathscr{P}$. By induction on the line numbers in A, we show that for any i, $\Gamma \vDash \mathscr{Q}_i$. The case when $i = n$ is the desired result.

Basis: The first line of A is a premise or an axiom. So $\mathscr{Q}_1$ is either a member of Γ or an instance of A1, A2, A3, A4, A5, A6, A7, or A8. The cases for A1–A3, A5–A8 are treated together.

(prem) Suppose $\mathscr{Q}_1$ is a member of Γ and $\Gamma \nvDash \mathscr{Q}_1$, then by QV there is some I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathscr{Q}_1] \neq \mathsf{T}$; but since $\mathsf{I}[\Gamma] = \mathsf{T}$ and $\mathscr{Q}_1 \in \Gamma$, $\mathsf{I}[\mathscr{Q}_1] = \mathsf{T}$. This is impossible; reject the assumption: $\Gamma \vDash \mathscr{Q}_1$.

(Ax) Suppose $\mathscr{Q}_1$ is an instance of A1, A2, A3, A5, A6, A7, or A8 and $\Gamma \nvDash \mathscr{Q}_1$. Then by QV, there is some I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathscr{Q}_1] \neq \mathsf{T}$. But by T7.2, T7.3, T7.4, T8.6, T7.8, T7.9, and T7.10, $\vDash \mathscr{Q}_1$; so by QV, $\mathsf{I}[\mathscr{Q}_1] = \mathsf{T}$. This is impossible; reject the assumption: $\Gamma \vDash \mathscr{Q}_1$.

(A4) If $\mathscr{Q}_1$ is an instance of A4, then it is of the form $\forall x \mathscr{B} \to \mathscr{B}^x_r$ where term r is free for variable x in formula $\mathscr{B}$. Suppose $\Gamma \nvDash \mathscr{Q}_1$. Then by QV, there is an I such that $\mathsf{I}[\Gamma] = \mathsf{T}$, but $\mathsf{I}[\forall x \mathscr{B} \to \mathscr{B}^x_r] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{I_d}[\forall x \mathscr{B} \to \mathscr{B}^x_r] \neq \mathsf{S}$; so by SF($\to$), $\mathsf{I_d}[\forall x \mathscr{B}] = \mathsf{S}$ but $\mathsf{I_d}[\mathscr{B}^x_r] \neq \mathsf{S}$; from the first of these, by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{B}] = \mathsf{S}$; so where $\mathsf{I_d}[r] = \mathsf{m}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{B}] = \mathsf{S}$; so, since r is free for x in formula $\mathscr{B}$, by T10.2, $\mathsf{I_d}[\mathscr{B}^x_r] = \mathsf{S}$. This is impossible; reject the assumption: $\Gamma \vDash \mathscr{Q}_1$.

Assp: For any i, $1 \leq i < k$, $\Gamma \vDash \mathscr{Q}_i$.

Show: $\Gamma \vDash \mathscr{Q}_k$.

$\mathscr{Q}_k$ is either a premise, an axiom, or arises from previous lines by MP or Gen. If $\mathscr{Q}_k$ is a premise or an axiom then as in the basis $\Gamma \vDash \mathscr{Q}_k$. So suppose $\mathscr{Q}_k$ arises by MP or Gen.

(MP) Homework.

(Gen) If $\mathscr{Q}_k$ arises by Gen, then A is something like this,

i	$\mathscr{B}$	
	$\vdots$	
k	$\forall x \mathscr{B}$	i Gen

where $i < k$ and $\mathscr{Q}_k = \forall x \mathscr{B}$. Suppose $\Gamma \nvDash \mathscr{Q}_k$; then $\Gamma \nvDash \forall x \mathscr{B}$; so by QV, there is some I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\forall x \mathscr{B}] \neq \mathsf{T}$. By assumption, $\Gamma \vDash \mathscr{B}$; so with $\mathsf{I}[\Gamma] = \mathsf{T}$, by QV, $\mathsf{I}[\mathscr{B}] = \mathsf{T}$; so by T7.6, $\mathsf{I}[\forall x \mathscr{B}] = \mathsf{T}$. This is impossible; reject the assumption: $\Gamma \vDash \mathscr{Q}_k$.

$\Gamma \vDash \mathscr{Q}_k$.

Indct: For any n, $\Gamma \vDash \mathscr{Q}_n$.

So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. So *AD* is sound. And since *AD* is sound, with theorems T9.2, T9.13, and T9.14 it follows that *ND* and *ND+* are sound as well.

*E10.2. Complete the case for (MP) to round out the T10.3 demonstration that *AD* is sound. You should set up the complete demonstration, but for cases completed in the text, you may simply refer to the text, as the text refers cases to homework. Hint: T8.8 may smooth your reasoning.

E10.3. Consider the derivation system $A^\star$ from E9.5 and provide a complete demonstration that it is sound. Notice that (A1)–(A3) and MP are the same as A^* from E8.11, and you demonstrated the soundness of A^* from E8.11 and E8.12 (and given T8.8, your sentential reasoning for those exercises converts directly to the quantificational case). You may appeal to prior exercises and theorems as appropriate.

10.1.3 Consistency

The proof of soundness is the main result we set out to achieve in this section. But before we go on, it is worth pausing to make an application to *consistency*. Say a set Δ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Delta \vdash \mathcal{A}$ and $\Delta \vdash \sim\mathcal{A}$. Consistency is thus defined in terms of *derivations* rather than semantic notions. But we show,

T10.4. If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ (a *model* for Γ), then Γ is consistent.

Suppose there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ but Γ is inconsistent. From the latter, there is a formula $\mathcal{A}$ such that $\Gamma \vdash \mathcal{A}$ and $\Gamma \vdash \sim\mathcal{A}$; so by T10.3, $\Gamma \vDash \mathcal{A}$ and $\Gamma \vDash \sim\mathcal{A}$. But $\mathsf{M}[\Gamma] = \mathsf{T}$; so by QV, $\mathsf{M}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{M}[\sim\mathcal{A}] = \mathsf{T}$; from the second of these and T8.8, $\mathsf{M}[\mathcal{A}] \neq \mathsf{T}$. This is impossible; reject the assumption: if there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$, then Γ is consistent.

This is an interesting and important theorem. Suppose we want to show that some set of formulas is inconsistent. For this, it is enough to *derive* a contradiction from the set. But suppose we want to show that there is no way to derive a contradiction. Merely failing to find a derivation does not show that there is not one! But, with soundness, we can demonstrate that there is no such derivation by finding a model for the set.

Similarly, if we want to show that $\Gamma \vdash \mathcal{A}$, it is enough to *produce* the derivation. But suppose we want to show that $\Gamma \nvdash \mathcal{A}$. Merely failing to find a derivation does not show that there is not one! Still, given soundness, we can demonstrate that there is no derivation by finding a model on which the premises are true and the conclusion is not.

T10.5. If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{A}] \neq \mathsf{T}$, then $\Gamma \nvdash \mathcal{A}$.

Suppose there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{A}] \neq \mathsf{T}$; then by QV, $\Gamma \nvDash \mathcal{A}$; so by T10.3 (read from right to left), $\Gamma \nvdash \mathcal{A}$.

Again, the result is useful. In chapters 4 and 7 we showed $\Gamma \nvDash \mathcal{A}$ by finding interpretations with the premises true and conclusion not; with T10.5 it immediately follows that the premises do not prove the conclusions, $\Gamma \nvdash \mathcal{A}$. Suppose, for example, we want to show that $\sim\forall x Px \nvdash \sim Pa$. You may be unable to find a derivation, and be able to point out flaws in a friend's attempt. But we show that there is no derivation by finding a model on which $\sim\forall x Px$ is true and $\sim Pa$ is not. And this is easy. We did it with trees in Chapter 4 (page 126), but we can do it in the style of Chapter 7 as well. Let $\mathsf{U} = \{1, 2\}$ with $\mathsf{M}[a] = 1$ and $\mathsf{M}[P] = \{1\}$.

(i) For arbitrary h, $\mathsf{h}(x|2)[x] = 2$; so by TA(v), $\mathsf{M}_{\mathsf{h}(x|2)}[x] = 2$; so by SF(r), $\mathsf{M}_{\mathsf{h}(x|2)}[Px] = \mathsf{S}$ iff $\langle 2\rangle \in \mathsf{M}[P]$; but $\langle 2\rangle \notin \mathsf{M}[P]$; so $\mathsf{M}_{\mathsf{h}(x|2)}[Px] \neq \mathsf{S}$; so for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{M}_{\mathsf{h}(x|\mathsf{o})}[Px] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{M}_{\mathsf{h}}[\forall x Px] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{M}_{\mathsf{h}}[\sim\forall x Px] = \mathsf{S}$; and since h is arbitrary, for any assignment d, $\mathsf{M}_{\mathsf{d}}[\sim\forall x Px] = \mathsf{S}$; so by TI, $\mathsf{M}[\sim\forall x Px] = \mathsf{T}$. (ii) $\mathsf{M}[a] = 1$; so for arbitrary h, by TA(c) $\mathsf{M}_{\mathsf{h}}[a] = 1$; so by SF(r), $\mathsf{M}_{\mathsf{h}}[Pa] = \mathsf{S}$ iff $\langle 1\rangle \in \mathsf{M}[P]$; but $\langle 1\rangle \in \mathsf{M}[P]$; so $\mathsf{M}_{\mathsf{h}}[Pa] = \mathsf{S}$; so by SF($\sim$), $\mathsf{M}_{\mathsf{h}}[\sim Pa] \neq \mathsf{S}$; so by TI, $\mathsf{M}[\sim Pa] \neq \mathsf{T}$. So $\mathsf{M}[\sim\forall x Px] = \mathsf{T}$ and $\mathsf{M}[\sim Pa] \neq \mathsf{T}$.

So by T10.5, $\sim\forall x Px \nvdash \sim Pa$.

If there is a model on which all the members of Γ are true and $\mathcal{A}$ is not, then it is not the case that $\Gamma \vDash \mathcal{A}$. So, with $\mathring{s}$oundness, there cannot be a derivation of $\mathcal{A}$ from Γ. For a more substantive example, E7.19, which tells us that Q does not *entail* certain results by finding an interpretation on which the axioms are true and the result is not, gives us that Q does not *prove* the results.

E10.4. (a) Show that $\{\exists x Ax, \sim Aa\}$ is consistent. (b) Show that $\forall x(Ax \to Bx)$, $\sim Ba \nvdash \sim\exists x Ax$.

10.2 Sentential $\mathring{C}$ompleteness

An arbitrary derivation system *DS* is *$\mathring{c}$omplete* when semantically valid results are provable: if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash_{DS} \mathcal{P}$. It is easy to construct derivation systems that are *not* $\mathring{c}$omplete. Again, the preliminary system *NP* from Chapter 6 is an easy example—because each rule starts with an input, there is no $\mathcal{P}$ such that $\vdash_{NP} \mathcal{P}$; so any tautology is a formula such that that $\vDash \mathcal{P}$ without $\vdash_{NP} \mathcal{P}$. Or consider a derivation system like *AD* but without A1. It is easy to see that such a system is $\mathring{s}$ound, and so that derivations without A1 do not go astray—all we have to do is leave the case for A1 out of the proof of $\mathring{s}$oundness. But, as will appear from our section 11.3 discussion of independence (together with E11.8), there is no derivation of the A1 instance $A \to (B \to A)$ from the other axioms. So without A1, there is a formula $\mathcal{P}$ such that $\vDash \mathcal{P}$, but for which there is no derivation. So the derivation system would not be complete. We turn now to showing that our derivation systems are in fact complete. Given this, with soundness, we have $\Gamma \vDash \mathcal{P}$ iff $\Gamma \vdash \mathcal{P}$, so that our derivation systems deliver just the results they are supposed to.

Completeness for a system like *AD* was first proved by Kurt Gödel in his 1930 doctoral dissertation. While the proof of soundness is straightforward given methods we have used before, the demonstration of completeness applies those methods in new and interesting ways. The version of the proof that we will consider is the standard one, essentially due to L. Henkin.[1] An interesting feature of these proofs is that they are not constructive. So far, for the deduction theorem and such, we have been able to show that there are certain derivations by showing how to *construct* them. However, just as it is possible to prove an existential $\exists x P x$ without finding an a such that Pa, so we shall be able to prove that there are certain derivations without a construction of them. As we shall see in Part IV, a constructive proof of completeness for our full predicate logic is impossible. So this is the only way to go.

The proof of completeness is more involved than any we have encountered so far. Each of the parts is comparable to what has gone before; but there are enough parts that it is possible to lose the forest for the trees. I thus propose to come at the argument three times: In this section, we will prove sentential completeness, that for expressions in a sentential language, if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$—this should enable us to grasp the overall shape of the argument without interference from too many details. We will then consider a basic version of the quantificational argument. And finally, after addressing a few complications, put it all together for the full version. Notation and theorem numbers are organized to preserve parallels between the cases.

10.2.1 Basic Idea

The basic idea is straightforward: Let us restrict ourselves to an arbitrary sentential language $\mathcal{L}_s$ and to sentential semantic rules. Derivations are automatically restricted to sentential rules by the restricted language. For formulas in this language, our goal is to show that if $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. We can see how this works with just a couple of preliminaries.

We begin with a definition and a theorem. As before, let us say,

Con A set Δ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Delta \vdash \mathcal{A}$ and $\Delta \vdash \sim\mathcal{A}$.

So consistency is a syntactical notion. A set of formulas is consistent just in case there is no way to derive a contradiction from it. Now for the theorem,

[1] Henkin, "Completeness of the First-Order Calculus." Kurt Gödel, "Die Vollständigkeit der Axiome des Logischen Funktionenkalküls." English translation in van Heijenoort, *From Frege to Gödel*, reprint in Feferman et al., *Gödel's Collected Works: Volume I*.

T10.6$_s$. For any set of formulas Δ and sentence $\mathcal{P}$, if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

Suppose $\Delta \nvdash \sim\mathcal{P}$, but $\Delta \cup \{\mathcal{P}\}$ is not consistent. From the latter, there is some $\mathcal{A}$ such that $\Delta \cup \{\mathcal{P}\} \vdash \mathcal{A}$ and $\Delta \cup \{\mathcal{P}\} \vdash \sim\mathcal{A}$. So by DT, $\Delta \vdash \mathcal{P} \to \mathcal{A}$ and $\Delta \vdash \mathcal{P} \to \sim\mathcal{A}$; by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by T3.2, $\Delta \vdash \sim\sim\mathcal{P} \to \mathcal{A}$, and $\Delta \vdash \sim\sim\mathcal{P} \to \sim\mathcal{A}$; but by A3, $\vdash (\sim\sim\mathcal{P} \to \sim\mathcal{A}) \to [(\sim\sim\mathcal{P} \to \mathcal{A}) \to \sim\mathcal{P}]$; so by two instances of MP, $\Delta \vdash \sim\mathcal{P}$. But this is impossible; reject the assumption: if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

The idea is simple: If $\Delta \cup \{\mathcal{P}\}$ is inconsistent, then by reasoning as for $\sim$I in *ND*, $\sim\mathcal{P}$ follows from Δ alone; transposing, if $\sim\mathcal{P}$ cannot be derived from Δ alone, then $\Delta \cup \{\mathcal{P}\}$ is consistent. Notice that insofar as the language is sentential, derivations do not include any applications of Gen, so the applications of DT are sure to meet the restriction on Gen.

In the last section, we saw that any set with a model is consistent. Now suppose we knew the converse, that any consistent set of formulas Σ' has a model.

($*$) For any consistent set of formulas Σ', there is an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

This sets up the key connection between syntactic and semantic notions, between consistency on the one hand, and truth on the other, that we will need for c̊ompleteness. Schematically, then, with ($*$) we have the following:

1.	$\Gamma \cup \{\sim\mathcal{P}\}$ has a model	$\Longrightarrow$	$\Gamma \nvDash_s \mathcal{P}$	
2.	$\Gamma \cup \{\sim\mathcal{P}\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathcal{P}\}$ has a model	($*$)
3.	$\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathcal{P}$	

Where $\Gamma \cup \{\sim\mathcal{P}\} = \Sigma'$, (2) is just ($*$). (1) is by simple semantic reasoning: Suppose $\Gamma \cup \{\sim\mathcal{P}\}$ has a model; then there is some M such that $\mathsf{M}[\Gamma \cup \{\sim\mathcal{P}\}] = \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\sim\mathcal{P}] = \mathsf{T}$; from the latter, by ST($\sim$), $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by SV, $\Gamma \nvDash_s \mathcal{P}$. (3) is by straightforward syntactic reasoning: Suppose $\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent; then by T10.6$_s$, reading from right to left, $\Gamma \vdash \sim\sim\mathcal{P}$; but by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by MP, $\Gamma \vdash \mathcal{P}$. And (1)–(3) together yield the c̊ompleteness result:

Suppose $\Gamma \vDash_s \mathcal{P}$; then by (1), reading from right to left, $\Gamma \cup \{\sim\mathcal{P}\}$ does not have a model; so by (2), again from right to left, $\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent; so by (3), $\Gamma \vdash \mathcal{P}$. So if $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$.

And this is what we want. Of course, knowing that there is some way to derive $\mathcal{P}$ is not the same as knowing what that way is. All the same, ($*$) tells us that there must exist a model of a certain sort, from which it follows that there must exist a derivation.

And the work of our demonstration of c̊ompleteness reduces to a demonstration of (∗).

So we need to show that every consistent set of formulas Σ' has an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$. Here is the basic idea: We show that any consistent Σ' is a subset of a corresponding "big" set Σ'' specified in such a way that it must have a model M'—which in turn is a model for the smaller Σ'. Following the arrows,

$$\begin{tikzcd}
\Sigma'' \arrow[dr] & \\
& \mathsf{M}' \arrow[dl] \\
\Sigma' \arrow[uu] &
\end{tikzcd}$$

Given a consistent Σ', we show that there is the big set Σ''. From this we show that there must be an M' that is a model not only for Σ'' but for Σ' as well. So if Σ' is consistent, then it has a model. We proceed through a series of theorems to show that this can be done.

10.2.2 Gödel Numbering

In constructing our big sets, we will want to consider sentences, for inclusion or exclusion, serially—one after another. For this, we need to "line them up" for consideration. Thus, where an *enumeration* of some objects sorts them into a series with a first member, a second member, and so forth, in this section we show,

T10.7$_\text{s}$. There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all sentences in $\mathcal{L}_s$.

> The proof is by construction. We develop a method by which the sentences can be lined up. The method is interesting in its own right, and foreshadows methods from Part IV on Gödel's ĩncompleteness theorem for arithmetic.

In section 2.2.1, we required that any sentential language $\mathcal{L}_s$ has countably many sentence letters, which can be ordered into a series, $\mathcal{S}_0$, $\mathcal{S}_1, \ldots$. Assume some such series. We want to show that the *sentences* of $\mathcal{L}_s$ can be so ordered as well. Begin by assigning to each symbol $\mathscr{s}$ in the language an integer $g[\mathscr{s}]$, called its *Gödel number*.

a. $g[(] = 3$

b. $g[)] = 5$

c. $g[\sim] = 7$

d. $g[\rightarrow] = 9$

e. $g[\mathcal{S}_n] = 11 + 2n$

So, for example, $g[\mathscr{S}_0] = 11$ and $g[\mathscr{S}_4] = 11 + 2 \times 4 = 19$. Clearly each symbol gets a unique Gödel number, and Gödel numbers for individual symbols are > 1 and odd.

Now we are in a position to assign Gödel numbers to expressions as follows: Where $\mathscr{s}_0, \mathscr{s}_1, \ldots, \mathscr{s}_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\mathscr{s}_0]} \times 3^{g[\mathscr{s}_1]} \times 5^{g[\mathscr{s}_2]} \times \cdots \times p_n{}^{g[\mathscr{s}_n]}$$

where $2, 3, 5, \ldots, p_n$ are the first n prime numbers. So, for example, $g[\sim\sim\mathscr{S}_0] = 2^7 \times 3^7 \times 5^{11}$; similarly, $g[\sim(\mathscr{S}_0 \rightarrow \mathscr{S}_4)] = 2^7 \times 3^3 \times 5^{11} \times 7^9 \times 11^{19} \times 13^5 = 15463, 36193, 79608, 90364, 71042, 41201, 87066, 87500, 00000$—a very big integer! All the same, it is an integer, and it is clear that every expression is assigned some integer.

Further, different expressions get different Gödel numbers. It is a theorem of arithmetic that every integer > 1 is uniquely factored into primes (see the arithmetic for Gödel numbering and more arithmetic for Gödel numbering references on pages 458 and 468). So a given integer can correspond to at most one expression: Given a Gödel number, we can find its unique prime factorization; then if there are seven 2s in the factorization, the first symbol is $\sim$; if there are seven 3s, the second symbol is $\sim$; if there are eleven 5s, the third symbol is $\mathscr{S}_0$; and so forth. Notice that numbers for individual *symbols* are odd, where numbers for *expressions* always have a multiplier of two and so are even (where the number for an atomic comes out odd when it is thought of as a symbol, but even when it is thought of as an expression).

The point is not that this is a practical, or a fun, procedure. Rather, the point is that we have natural numbers associated with each expression of the language. Given this, we can take the set of all sentences, and *order* its members according to their Gödel numbers—so that there is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all sentences. And this is what was to be shown.

E10.5. Find Gödel numbers for each of the following. Treat the first as a simple symbol. (For the last, you need not do the calculation!)

$\mathscr{S}_7$ $\qquad$ $\sim\mathscr{S}_0$ $\qquad$ $\mathscr{S}_0 \rightarrow \sim(\mathscr{S}_1 \rightarrow \sim\mathscr{S}_0)$

E10.6. Determine the objects that have the following Gödel numbers:

49 $\qquad$ 1944 $\qquad$ $2^7 \times 3^3 \times 5^{11} \times 7^9 \times 11^7 \times 13^{13} \times 17^5$

E10.7. (i) Is every positive integer a Gödel number? Explain. (ii) Explain how this does or does not matter for the result that there is an enumeration of all formulas in $\mathcal{L}_{\mathscr{s}}$.

Some Arithmetic Relevant to Gödel Numbering

Say an integer i has a "representation as a product of primes" if there are some primes $p_a, p_b, \ldots, p_j$ such that $p_a \times p_b \times \cdots \times p_j = i$. We understand a single prime p to be its own representation.

G1. Every integer > 1 has at least one representation as a product of primes.

Basis: 2 is prime and so is its own representation; so the first integer > 1 has a representation as a product of primes.

Assp: For any i, $1 < i < k$, i has a representation as a product of primes.

Show: k has a representation as a product of primes.

If k is prime, the result is immediate; so suppose there are some $i, j < k$ such that $k = i \times j$; by assumption i has a representation as a product of primes $p_a \times \cdots \times p_b$ and j has a representation as a product of primes $q_a \times \cdots \times q_b$; so $k = i \times j = p_a \times \cdots \times p_b \times q_a \times \cdots \times q_b$ has a representation as a product of primes.

Indct: Any $i > 1$ has a representation as a product of primes.

Corollary: any integer > 1 is evenly divided by at least one prime.

G2. There are infinitely many prime numbers.

Suppose the number of primes is finite; then there is some list $p_1, p_2, \ldots, p_n$ of all the primes; consider $q = p_1 \times p_2 \times \cdots \times p_n + 1$; no p_i in the list $p_1 \ldots p_n$ divides q evenly, since each leaves remainder 1; but by the corollary to (G1), q is divided by some prime; so some prime is not on the list; reject the assumption: there are infinitely many primes.

Note: Sometimes q, calculated this way, is itself prime: when the list is $\{2\}$, $q = 2 + 1 = 3$, and 3 is prime. Similarly, $2 \times 3 + 1 = 7$, $2 \times 3 \times 5 + 1 = 31$, $2 \times 3 \times 5 \times 7 + 1 = 211$, and $2 \times 3 \times 5 \times 7 \times 11 + 1 = 2311$, where 7, 31, 211, and 2311 are all prime. But $2 \times 3 \times 5 \times 7 \times 11 \times 13 + 1 = 30031 = 59 \times 509$. So we are not always *finding* a prime not on the list, but rather only showing that there *is* a prime not on it.

G3. For any $i > 1$, if i is the product of the primes $p_1, p_2, \ldots, p_a$, then no distinct collection of primes $q_1, q_2, \ldots, q_b$ is such that i is the product of them. *Fundamental Theorem of Arithmetic*.

For a proof, see the more arithmetic for Gödel numbering reference in the corresponding part of section 10.3.

10.2.3 The Big Set

Recall that a set Δ is consistent iff there is no $\mathcal{A}$ such that Δ proves both $\mathcal{A}$ and $\sim\mathcal{A}$. Now say a set Δ is *maximal* iff for any $\mathcal{A}$ the set proves one or the other.

Max A set Δ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Delta \vdash \mathcal{A}$ or $\Delta \vdash \sim\mathcal{A}$.

Again, this is a syntactical notion. If a set is maximal, then it proves $\mathcal{A}$ or $\sim\mathcal{A}$ for any sentence $\mathcal{A}$; if it is consistent, then it does not prove both. We set out to construct a big set Σ'' from Σ', and show that Σ'' is both maximal and consistent.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7$_s$, there is an enumeration, $\mathcal{Q}_1$, $\mathcal{Q}_2, \ldots$ of all the sentences in $\mathcal{L}_s$. Consider this enumeration, and let Ω_0 be the same as Σ'. Then for any $i > 0$ let,

$$\Omega_i = \Omega_{i-1} \qquad \text{if} \qquad \Omega_{i-1} \vdash \sim\mathcal{Q}_i$$

else,

$$\Omega_i = \Omega_{i-1} \cup \{\mathcal{Q}_i\} \qquad \text{if} \qquad \Omega_{i-1} \nvdash \sim\mathcal{Q}_i$$

then,

$\Sigma'' = \bigcup_{i \geq 0} \Omega_i$—that is, Σ'' is the union of all the Ω_is

Beginning with set Σ' ($= \Omega_0$), we consider the sentences in the enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2, \ldots$ one by one, adding a sentence to the set just in case its negation is not already derivable. Σ'' contains all the members of Σ' together with all the sentences added this way. Observe that $\Sigma' \subseteq \Sigma''$. One might think of the Ω_is as constituting a big "sack" of formulas, and the $\mathcal{Q}_i$s as coming along on a conveyor belt: For a given $\mathcal{Q}_i$, if there is no way to derive its negation from formulas already in the sack, we throw the $\mathcal{Q}_i$ in; otherwise, we let it go on by. Of course, this is not a procedure we could complete in finite time. Rather, we give a *logical* condition which specifies, for any sentence $\mathcal{Q}_i$ in the language, whether it is to be included in Σ'' or not. The important point is that some Σ'' meeting these conditions *exists*.

As an example, suppose $\Sigma' = \{\sim A \to B\}$ and consider an enumeration which begins $A, \sim A, B, \sim B, \ldots$. Then,

$\Omega_0 = \Sigma'$; so $\Omega_0 = \{\sim A \to B\}$.

$\mathcal{Q}_1 = A$, and $\Omega_0 \nvdash \sim A$; so $\Omega_1 = \{\sim A \to B\} \cup \{A\} = \{\sim A \to B, A\}$.

(F) $\mathcal{Q}_2 = \sim A$, and $\Omega_1 \vdash \sim\sim A$; so Ω_2 is unchanged; so $\Omega_2 = \{\sim A \to B, A\}$.

$\mathcal{Q}_3 = B$, and $\Omega_2 \nvdash \sim B$; so $\Omega_3 = \{\sim A \to B, A\} \cup \{B\} = \{\sim A \to B, A, B\}$.

$\mathcal{Q}_4 = \sim B$, and $\Omega_3 \vdash \sim\sim B$; so Ω_4 is unchanged; so $\Omega_4 = \{\sim A \to B, A, B\}$.

So we include $\mathcal{Q}_i$ each time its negation is not proved. Ultimately, we will use this set to construct a model. For now, though, the point is simply to understand the condition under which a sentence is included or excluded from the set and, with this, to think about its nature.

We now show that if Σ' is consistent, then Σ'' is maximal and consistent. Perhaps the first is obvious: We guarantee that Σ'' is maximal by including $\mathcal{Q}_i$ as a member whenever $\sim\mathcal{Q}_i$ is not already a consequence. The other is not much more difficult.

T10.8$_s$. If Σ' is consistent, then Σ'' is maximal and consistent.

The proof comes to the demonstration of three results. Given the assumption that Σ' is consistent, we show, (a) Σ'' is maximal; (b) each Ω_i is consistent; and use this to show (c), Σ'' is consistent. Suppose Σ' is consistent.

(a) Σ'' is maximal. Suppose otherwise. Then there is some sentence $\mathcal{Q}_i$ such that both $\Sigma'' \nvdash \mathcal{Q}_i$ and $\Sigma'' \nvdash \sim\mathcal{Q}_i$. For this i, by construction, each member of Ω_{i-1} is in Σ''; so if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$ then $\Sigma'' \vdash \sim\mathcal{Q}_i$; but $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_i = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$; and by construction again, $\mathcal{Q}_i \in \Sigma''$; so $\Sigma'' \vdash \mathcal{Q}_i$. This is impossible; reject the assumption: Σ'' is maximal.

(b) Each Ω_i is consistent. By induction on the series of Ω_is,

Basis: $\Omega_0 = \Sigma'$ and Σ' is consistent; so Ω_0 is consistent.

Assp: For any i, $0 \leq i < k$, Ω_i is consistent.

Show: Ω_k is consistent.

Ω_k is either Ω_{k-1} or $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$. Suppose the former; by assumption, Ω_{k-1} is consistent; so Ω_k is consistent. Suppose the latter; then by construction, $\Omega_{k-1} \nvdash \sim\mathcal{Q}_k$; so by T10.6$_s$, $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$ is consistent; so Ω_k is consistent. So, either way, Ω_k is consistent.

Indct: For any i, Ω_i is consistent.

(c) Σ'' is consistent. Suppose Σ'' is not consistent; then there is some $\mathcal{A}$ such that $\Sigma'' \vdash \mathcal{A}$ and $\Sigma'' \vdash \sim\mathcal{A}$. Consider derivations *D1* and *D2* of these results, and the premises $\mathcal{Q}_i \ldots \mathcal{Q}_j$ of these derivations. Where $\mathcal{Q}_j$ is the last of these premises in the enumeration of sentences, by the construction of Σ'', each of $\mathcal{Q}_i \ldots \mathcal{Q}_j$ must be a member of Ω_j; so *D1* and *D2* are derivations from Ω_j; so Ω_j is inconsistent. But by (b) Ω_j is consistent. This is impossible; reject the assumption: Σ'' is consistent.

Observe that there is something to show at (c). The concern is that members of a sequence might individually be consistent, but the union of them all not. Consider the following example:

$\Pi_0 =$ {*a* has finitely many members}

$\Pi_1 =$ {*a* has finitely many members, *a* has at least 1 member}

$\Pi_2 =$ {*a* has finitely many members, *a* has at least 1 member, *a* has at least 2 members} and so forth. Intuitively, each Π_n is consistent with the proposition that a has exactly n members. But the union of them all is inconsistent—for any finite n, the proposition that a has n members is inconsistent with Π_{n+1}. We show that this cannot happen in the construction of Σ'', insofar as any inconsistency must emerge at some finite stage: Because derivations of $\mathcal{A}$ and $\sim\mathcal{A}$ have only finitely many premises, all the premises in a derivation of a contradiction must show up in some Ω_j; so if Σ'' is inconsistent, then some Ω_j is inconsistent, which is impossible. So we have what we set out to show: $\Sigma' \subseteq \Sigma''$, and if Σ' is consistent, then Σ'' is both maximal and consistent.

E10.8. (i) Suppose $\Sigma' = \{A \to \sim B\}$ and the enumeration of sentences begins A, $\sim A$, B, $\sim B, \ldots.$ What are Ω_0, Ω_1, Ω_2, Ω_3, and Ω_4? (ii) What are they when the enumeration begins B, $\sim B$, A, $\sim A, \ldots$?

10.2.4 The Model

We now construct a model M' for Σ'. The key is that the maximal and consistent set contains enough information that we can extract from it a specification for a model of the whole. In this sentential case, the specification is particularly simple.

CnsM$'$ For any atomic $\mathcal{S}$, let $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$.

Notice that there clearly exists some such interpretation M': We assign T to every sentence letter that can be derived from Σ'', and F to the others. It will not be the case that we are in a position to do all the derivations, and so to know what are all the assignments to the atomics. Still, it must be that any atomic either is or is not a consequence of Σ'', and so that there exists a corresponding interpretation M' on which those sentence letters either are or are not assigned T.

We now want to show that if Σ' is consistent, then M' is a model for Σ'—that if Σ' is consistent then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. As we shall see, this results immediately from the following theorem.

*T10.9$_s$. If Σ' is consistent, then for any sentence $\mathcal{P}$ of $\mathcal{L}_s$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Suppose Σ' is consistent. Then by T10.8$_s$, Σ'' is maximal and consistent. Now by induction on the number of operators in $\mathcal{P}$,

Basis: If $\mathcal{P}$ has no operators, then it is an atomic of the sort $\mathcal{S}$. But by the construction of M', $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$; so $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Assp: For any i, $0 \le i < k$, if $\mathcal{P}$ has i operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Show: If $\mathcal{P}$ has k operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$ or $\mathcal{A} \to \mathcal{B}$ where $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. (i) Suppose $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$; then $\mathsf{M}'[\sim\mathcal{A}] = \mathsf{T}$; so by ST($\sim$), $\mathsf{M}'[\mathcal{A}] \neq \mathsf{T}$; so by assumption, $\Sigma'' \nvdash \mathcal{A}$; so by maximality, $\Sigma'' \vdash \sim\mathcal{A}$; which is to say, $\Sigma'' \vdash \mathcal{P}$. (ii) Suppose $\Sigma'' \vdash \mathcal{P}$; then $\Sigma'' \vdash \sim\mathcal{A}$; so by consistency, $\Sigma'' \nvdash \mathcal{A}$; so by assumption, $\mathsf{M}'[\mathcal{A}] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{M}'[\sim\mathcal{A}] = \mathsf{T}$; which is to say, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$. So $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

($\rightarrow$) Suppose $\mathcal{P}$ is $\mathcal{A} \rightarrow \mathcal{B}$. (i) Suppose $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$; then $\mathsf{M}'[\mathcal{A} \rightarrow \mathcal{B}] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{M}'[\mathcal{A}] \neq \mathsf{T}$ or $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$; so by assumption, $\Sigma'' \nvdash \mathcal{A}$ or $\Sigma'' \vdash \mathcal{B}$; from the first of these, by maximality, $\Sigma'' \vdash \sim\mathcal{A}$; in either case by $\vee$I, $\Sigma'' \vdash \sim\mathcal{A} \vee \mathcal{B}$; so by Impl, $\Sigma'' \vdash \mathcal{A} \rightarrow \mathcal{B}$ where this is to say, $\Sigma'' \vdash \mathcal{P}$. (ii) Suppose $\Sigma'' \vdash \mathcal{P}$ but $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; by [homework], this is impossible: so if $\Sigma'' \vdash \mathcal{P}$, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$. So $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Indct: For any $\mathcal{P}$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

The key to this is that Σ'' is both maximal and consistent—so that we can move between the failure to prove a sentence and the proof of its negation. And the maximality and consistency of Σ'' are required to make the consequences of Σ'' match truths on M'. Thus in example (F), $\Sigma' = \{\sim A \rightarrow B\}$; so $\Sigma' \nvdash A$ and $\Sigma' \nvdash B$; if we were simply to follow our construction procedure as applied to this set, the result would have $\mathsf{M}'[A] \neq \mathsf{T}$ and $\mathsf{M}'[B] \neq \mathsf{T}$; but then $\mathsf{M}'[\sim A \rightarrow B] \neq \mathsf{T}$ and there is no model for Σ'. But Ω_4, and so Σ'', have A and B as members; so $\Sigma'' \vdash A$ and $\Sigma'' \vdash B$. So by the construction procedure, $\mathsf{M}'[A] = \mathsf{T}$ and $\mathsf{M}'[B] = \mathsf{T}$; so $\mathsf{M}'[\sim A \rightarrow B] = \mathsf{T}$. Thus it is the construction, together with the maximality and consistency of Σ'', that puts us in a position to draw the parallel between the consequences of Σ'' and what is true on M'. With this, it will be a short step to see that we have a model for Σ' and so ($*$) that we have been after.

*E10.9. Complete (ii) for the conditional case to complete the proof of T10.9_s. You should set up the entire induction, but may refer to the text for parts completed there, as the text refers to homework.

E10.10. (i) Where $\Sigma' = \{A \rightarrow \sim B\}$, and the enumeration of formulas is as in the first part of E10.8, what assignments does M' make to A and B? (ii) What assignments does it make on the second enumeration? Use a truth table to show, for each case, that the assignments result in a *model* for Σ'. Explain.

10.2.5 Final Result

The proof of sentential c̊ompleteness is now a simple matter of pulling together what we have done. First, it is a simple matter to show, T10.10$_s$. If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. $(*)$

> Suppose Σ' is consistent but $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{P} \in \Sigma'$ such that $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$. Since $\mathcal{P} \in \Sigma'$, by construction, $\mathcal{P} \in \Sigma''$; so $\Sigma'' \vdash \mathcal{P}$; so since Σ' is consistent, by T10.9$_s$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$. This is impossible; reject the assumption: if Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

That is it! Going back to the beginning of our discussion of sentential c̊ompleteness, all we needed was $(*)$, and now we have it. So the final argument is as sketched before:

T10.11$_s$. If $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Sentential C̊ompleteness.*

> Suppose $\Gamma \vDash_s \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment, that $\Gamma \vdash \sim\sim\mathcal{P}$; by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash \sim\sim\mathcal{P}$. Given this, by T10.6$_s$, $\Gamma \cup \{\sim\mathcal{P}\} = \Sigma'$ is consistent; so by T10.10$_s$, there is a model M' such that $\mathsf{M}'[\Gamma \cup \{\sim\mathcal{P}\}] = \mathsf{T}$; so $\mathsf{M}'[\Gamma] = \mathsf{T}$ and $\mathsf{M}'[\sim\mathcal{P}] = \mathsf{T}$; from the latter, by ST($\sim$), $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}'[\Gamma] = \mathsf{T}$ but $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by SV, $\Gamma \nvDash_s \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$.

Try again to get the complete picture in your mind: The key is that consistent sets always have models. If there is no derivation of $\mathcal{P}$ from Γ, then $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent; and if $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent, then it has a model—so that $\Gamma \nvDash_s \mathcal{P}$. Thus, put the other way around, if $\Gamma \vDash_s \mathcal{P}$, then there is a derivation of $\mathcal{P}$ from Γ. We get the key point, that consistent sets have models, by finding a relation between consistent, and *maximal* consistent sets. If a set is both maximal and consistent, then it contains enough information about its atomics that a model for its atomics is a model for the whole.

It is obvious that the argument is not constructive—we do not see how to show that $\Gamma \vdash \mathcal{P}$ whenever $\Gamma \vDash_s \mathcal{P}$. But it is interesting to see why. The argument turns on the *existence* of our big sets under certain conditions, and so on the existence of models. We show that the sets must exist and have certain properties, though we are not in a position to find all their members. This puts us in a position to know the existence of derivations, though we do not say what they are.[2]

E10.11. Suppose our primitive operators are $\sim$ and $\wedge$ and the derivation system is A^* from E3.5 (Chapter 3, page 76). Present a complete demonstration of c̊ompleteness for this derivation system—with all the definitions and theorems. You may simply appeal to the text for results that require no change. You have the results of E3.5 along with DT by E9.8.

[2]In fact, there are constructive approaches to sentential c̊ompleteness. See, for example, Lemma 1.13 and Proposition 1.14 of Mendelson, *Introduction to Mathematical Logic*. Our primary purpose, however, is to set up the argument for the quantificational case, where such methods do not apply.

10.3 Quantificational $\mathring{\text{C}}$ompleteness: Basic Version

As promised, the demonstration of quantificational $\mathring{\text{c}}$ompleteness is parallel to what we have seen. Return to a quantificational language and to our regular quantificational semantic and derivation notions. The goal is to show that if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Certain complications are avoided if we suppose that the language $\mathcal{L}'$ includes infinitely many constants not in Γ or $\mathcal{P}$, and does not include the '$=$' symbol for equality. The constants will be required for the construction of our big sets. And without $=$ in the language, the model specification is simplified. We will work through the basic argument in this section and, dropping constraints on the language, return to the general case in the next. If you are confused at any stage, it may help to refer back to the parallel section for the sentential case.

10.3.1 Basic Idea

As before, our main argument turns on the idea that every consistent set has a model. Thus we begin with a definition and a theorem.

Con A set Δ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Delta \vdash \mathcal{A}$ and $\Delta \vdash \sim\mathcal{A}$.

So a set of formulas is consistent just in case there is no way to derive a contradiction from it. Of course, now we are working with full quantificational languages, and so with our full quantificational derivation systems.

For the following theorem, notice that Δ is a set of *formulas*, and $\mathcal{P}$ a *sentence* (a distinction without a difference in the sentential case). As before,

T10.6. For any set of formulas Δ and sentence $\mathcal{P}$, if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

For some sentence $\mathcal{P}$, suppose $\Delta \nvdash \sim\mathcal{P}$ but $\Delta \cup \{\mathcal{P}\}$ is not consistent. From the latter, there is some formula $\mathcal{A}$ such that $\Delta \cup \{\mathcal{P}\} \vdash \mathcal{A}$ and $\Delta \cup \{\mathcal{P}\} \vdash \sim\mathcal{A}$; since $\mathcal{P}$ is a sentence, it has no free variables; so by DT, $\Delta \vdash \mathcal{P} \rightarrow \mathcal{A}$ and $\Delta \vdash \mathcal{P} \rightarrow \sim\mathcal{A}$; by T3.10, $\vdash \sim\sim\mathcal{P} \rightarrow \mathcal{P}$; so by T3.2, $\Delta \vdash \sim\sim\mathcal{P} \rightarrow \mathcal{A}$ and $\Delta \vdash \sim\sim\mathcal{P} \rightarrow \sim\mathcal{A}$; but by A3, $\vdash (\sim\sim\mathcal{P} \rightarrow \sim\mathcal{A}) \rightarrow [(\sim\sim\mathcal{P} \rightarrow \mathcal{A}) \rightarrow \sim\mathcal{P}]$; so by two instances of MP, $\Delta \vdash \sim\mathcal{P}$. This is impossible; reject the assumption: if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

Insofar as $\mathcal{P}$ is required to be a sentence, it has no free variables; so no application of Gen is to a variable free in $\mathcal{P}$; so the restriction on DT is sure to be met. So T10.6 does not apply for an arbitrary formula $\mathcal{P}$.

To the extent that T10.6 plays a direct role in our basic argument for $\mathring{\text{c}}$ompleteness, this point that it does not apply to an arbitrary formula $\mathcal{P}$ might seem to present a problem about reaching our general result, that if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$, which is supposed to apply in the arbitrary case. But there is a way around the problem. For any formula $\mathcal{P}$, let its *universal closure* $\mathcal{P}^u$ be $\mathcal{P}$ prefixed by a universal quantifier for every variable free in $\mathcal{P}$. To make $\mathcal{P}^u$ unique, for some enumeration of variables, $x_1, x_2, \ldots$ let the quantifiers be in order of ascending subscripts. So if $\mathcal{P}$ has no free variables, $\mathcal{P}^u = \mathcal{P}$; if x_1 is free in $\mathcal{P}$, then $\mathcal{P}^u = \forall x_1 \mathcal{P}$; if x_1 and x_3 are free in $\mathcal{P}$, then $\mathcal{P}^u = \forall x_1 \forall x_3 \mathcal{P}$; and so forth. So for any formula $\mathcal{P}$, $\mathcal{P}^u$ is a *sentence*. As it turns out, we will be able to argue about arbitrary formulas $\mathcal{P}$ by using their closures $\mathcal{P}^u$ as intermediaries.

Let $\sim\mathcal{P}^u$ be $\sim(\mathcal{P}^u)$ and suppose the members of $\Gamma \cup \{\sim\mathcal{P}^u\} = \Sigma'$ are formulas of $\mathcal{L}'$. Then it will be sufficient to show that any consistent set of this sort has a model.

($\star$) For any consistent set Σ' of formulas in $\mathcal{L}'$, there is an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

Again, this sets up the key connection between syntactic and semantic notions—between consistency on the one hand, and truth on the other—that we will need for completeness. Supposing ($\star$) we have the following:

1.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ has a model	$\Longrightarrow$	$\Gamma \not\models \mathcal{P}$	
2.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathcal{P}^u\}$ has a model	($\star$)
3.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathcal{P}$	

Where $\Gamma \cup \{\sim\mathcal{P}^u\} = \Sigma'$, (2) is just ($\star$). Observe that (1) and (3) switch between $\mathcal{P}^u$ and $\mathcal{P}$. Reasoning is as before except that T7.6 and A4 provide the required bridge between $\mathcal{P}^u$ and $\mathcal{P}$: (1) Suppose $\Gamma \cup \{\sim\mathcal{P}^u\}$ has a model; then there is some M such that $\mathsf{M}[\Gamma \cup \{\sim\mathcal{P}^u\}] = \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\sim\mathcal{P}^u] = \mathsf{T}$; from the latter, by T8.8, $\mathsf{M}[\mathcal{P}^u] \neq \mathsf{T}$; so by repeated application of T7.6, $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \not\models \mathcal{P}$. (3) Suppose $\Gamma \cup \{\sim\mathcal{P}^u\}$ is not consistent; then since $\mathcal{P}^u$ is a sentence, by an application of T10.6, $\Gamma \vdash \sim\sim\mathcal{P}^u$; but by T3.10, $\vdash \sim\sim\mathcal{P}^u \rightarrow \mathcal{P}^u$; so by MP, $\Gamma \vdash \mathcal{P}^u$; and by repeated applications of A4 and MP, $\Gamma \vdash \mathcal{P}$.

Now suppose $\Gamma \models \mathcal{P}$; then from (1), $\Gamma \cup \{\sim\mathcal{P}^u\}$ does not have a model; so by (2), $\Gamma \cup \{\sim\mathcal{P}^u\}$ is not consistent; so by (3), $\Gamma \vdash \mathcal{P}$. So if $\Gamma \models \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Again, it remains to show ($\star$), that every consistent set Σ' of formulas has a model. And, again, our strategy is to find a "big" set related to Σ' which can be used to specify a model for Σ'.

10.3.2 Gödel Numbering

As before, in constructing our big sets, we will want to line up expressions serially—one after another. The method merely expands our approach for the sentential case.

T10.7. There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all the sentences, terms, and the like in $\mathcal{L}'$.

The proof is again by construction: We develop a method by which all the expressions of $\mathcal{L}'$ can be lined up. Then the collection of all sentences, taken in that order, is an enumeration of all sentences; the collection of all terms, taken in that order, is an enumeration of all terms; and so forth.

Insofar as the collections of variable symbols, constant symbols, function symbols, sentence letters, and relation symbols in any quantificational language are countable, they are capable of being sorted into series, $x_0, x_1, \ldots$ and $a_0, a_1, \ldots$ and $h^n_0, h^n_1, \ldots$ and $\mathcal{R}^n_0, \mathcal{R}^n_1, \ldots$ for variables, constants, function symbols, and relation symbols respectively (where we think of sentence letters as 0-place relation symbols). Supposing that they are sorted into such series, begin by assigning to each symbol s in $\mathcal{L}'$ an integer $g[s]$ called its *Gödel number*.

a.	$g[(] = 3$	f.	$g[\forall] = 13$
b.	$g[)] = 5$	g.	$g[x_i] = 15 + 10i$
c.	$g[\sim] = 7$	h.	$g[a_i] = 17 + 10i$
d.	$g[\rightarrow] = 9$	i.	$g[h^n_i] = 19 + 10(2^n \times 3^i)$
e.	$g[=] = 11$	j.	$g[\mathcal{R}^n_i] = 21 + 10(2^n \times 3^i)$

Officially, we do not yet have '$=$' in the language, but it is easy enough to leave it out for now. So, for example, $g[x_0] = 15$, $g[x_1] = 15 + 10 \times 1 = 25$, and $g[\mathcal{R}^2_1] = 21 + 10(2^2 \times 3^1) = 141$.

To see that each symbol gets a distinct Gödel number, first notice that numbers in different categories cannot overlap: Each of (a)–(f) is obviously distinct and ≤ 13. But (g)–(j) are all greater than 13, and when divided by 10, the remainder is 5 for variables, 7 for constants 9 for function symbols, and 1 for relation symbols; so numbers for variables, constants, function symbols, and relation symbols do not overlap. Second, different symbols get different numbers within the categories. This is obvious except in cases (i) and (j). For these we need to see that each n/i combination results in a different multiplier.

Suppose this is not so, that there are some combinations n, i and m, j such that $2^n \times 3^i = 2^m \times 3^j$ but $n \neq m$ or $i \neq j$. If $n = m$ then, dividing both sides by 2^n, we get $3^i = 3^j$, so that $i = j$. So suppose $n \neq m$ and, without loss of generality, that $n > m$. Dividing each side by 2^m, we get $2^{n-m} \times 3^i = 3^j$; since $n > m$, $n - m$ is a positive integer; so 2^{n-m} is > 1 and even. So $3^i < 3^j$ and $i < j$. Dividing both sides again by 3^i, we get $2^{n-m} = 3^{j-i}$; but since $j > i$, $j - i$ is a positive integer and 3^{j-i} is odd. Reject the assumption: if $2^n \times 3^i = 2^m \times 3^j$, then $n = m$ and $i = j$.

So each n/i combination gets a different multiplier, and we conclude that each symbol gets a different Gödel number. (This result is a special case of the fundamental theorem of arithmetic treated on the following page.)

Now, as before, assign Gödel numbers to expressions as follows: Where $\mathscr{s}_0, \mathscr{s}_1, \ldots, \mathscr{s}_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\mathscr{s}_0]} \times 3^{g[\mathscr{s}_1]} \times 5^{g[\mathscr{s}_2]} \times \cdots \times p_n{}^{g[\mathscr{s}_n]}$$

where $2, 3, 5, \ldots, p_n$ are the first n prime numbers. So, for example, $g[\sim\sim\mathcal{R}^2_1 x_0 x_1] = 2^7 \times 3^7 \times 5^{141} \times 7^{15} \times 11^{25}$—a relatively large integer (one with over 130 digits)! All the same, it is an integer, and different expressions get different Gödel numbers. Given a Gödel number, we can find the corresponding expression by finding its prime factorization; then if there are seven 2s in the factorization, the first symbol is $\sim$; if there are seven 3s, the second symbol is $\sim$; if there are one hundred forty-one 5s, the third symbol is $\mathcal{R}^2_1$; and so forth. Notice that numbers for individual symbols are odd, where numbers for expressions always have a multiplier of two and so are even.

So we can take the set of all sentences, the set of all terms, or whatever, and order their members according to their Gödel numbers—so that there is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all sentences, terms, and so forth. And this is what was to be shown.

E10.12. Find Gödel numbers for each of the following. Treat the first as a simple symbol. For the last, you need not do the calculation!

$\mathcal{R}^2_3$ $\quad$ $h^1_1 x_1$ $\quad$ $\forall x_2 \mathcal{R}^2_1 a_2 x_2$

E10.13. Determine the objects that have the following Gödel numbers:

61 $\quad$ $2^{13} \times 3^{15} \times 5^3 \times 7^{15} \times 11^{11} \times 13^{15} \times 17^5$

10.3.3 The Big Set

Last time, to build our big set we added sentences to Σ' to form a Σ'' that was both maximal and consistent. From Con a set of formulas is consistent just in case there is no formula $\mathcal{A}$ such that both $\mathcal{A}$ and $\sim\mathcal{A}$ are consequences. As before, however, both the notion of maximality and our construction of Σ'' proceed in terms of *sentences*. So,

Max A set Δ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Delta \vdash \mathcal{A}$ or $\Delta \vdash \sim\mathcal{A}$.

And this time we require an additional property for our big sets. If a maximal and consistent set has some sentence $\forall x\mathcal{P}$ as a member, then it has $\mathcal{P}^x_a$ as a consequence for every constant a. (Be clear about why this is so.) But in a maximal and consistent set, the status of a universal $\forall x\mathcal{P}$ is not always reflected at the level of its instances. Thus, for example, though a set has $\mathcal{P}^x_a$ as a consequence for every constant a, it may consistently include $\sim\forall x\mathcal{P}$ as well—for it may be that a universal is falsified by

More Arithmetic Relevant to Gödel Numbering

G3. For any $i > 1$, if i is the product of the primes $p_1, p_2, \ldots, p_a$, then no distinct collection of primes $q_1, q_2, \ldots, q_b$ is such that i is the product of them. *Fundamental Theorem of Arithmetic*.

Basis: The first integer $> 1 = 2$; but the only collection of primes such that their product is equal to 2 is the collection containing just 2 itself; so no distinct collection of primes is such that 2 is the product of them.

Assp: For any i, $1 \le i < k$, if i is the product of primes $p_1 \ldots p_a$, then no distinct collection of primes $q_1 \ldots q_b$ is such that i is the product of them.

Show: k is such that if it is the product of the primes $p_1 \ldots p_a$, then no distinct collection of primes $q_1 \ldots q_b$ is such that k is the product of them.

Suppose there are distinct collections of primes $p_1 \ldots p_a$ and $q_1 \ldots q_b$ such that $k = p_1 \times \cdots \times p_a = q_1 \times \cdots \times q_b$; divide out terms common to both lists of primes; then for some subclasses of the original lists, $n = p_1 \times \cdots \times p_c = q_1 \times \cdots \times q_d$, where no member of $p_1 \ldots p_c$ is a member of $q_1 \ldots q_d$ and vice versa (of course this p_1 may be distinct from the one in the original list, and so forth). So $p_1 \neq q_1$; suppose, without loss of generality, that $p_1 > q_1$; and let $m = q_1(n/q_1 - n/p_1) = (p_1 - q_1)(n/p_1) = n - (q_1/p_1)n = n - q_1 \times p_2 \times \cdots \times p_c$.

Some preliminary results: (i) n/q_1 and n/p_1 are integers, with the first greater than the second; so their difference $n/q_1 - n/p_1$ is a positive integer; so either $n/q_1 - n/p_1 = 1$ or it has a prime factorization. (ii) Since $m = n - q_1 \times p_2 \times \cdots \times p_c$, $m < n$; and $n \le k$; so $m < k$. Further, since q_1 is prime, $q_1 > 1$; and since $n/q_1 - n/p_1$ is a positive integer, $m = q_1(n/q_1 - n/p_1) > 1$. So the inductive assumption applies to m. (iii) $(p_1 - q_1)/q_1 = p_1/q_1 - 1$; since p_1 is prime, this is no integer; so q_1 does not divide $(p_1 - q_1)$.

Either $p_1 - q_1 = 1$ or it has some prime factorization; and n/p_1 has a prime factorization, $p_2 \times \cdots \times p_c$; since $m = (p_1 - q_1)(n/p_1)$, the product of these factorization(s) is a prime factorization of m. Given the cancellation of common terms to get n, q_1 is not a member of $p_2 \times \cdots \times p_c$; and by (iii), q_1 is not a member of the factorization of $p_1 - q_1$; so q_1 is not a member of this factorization of m. But by (i) either $n/q_1 - n/p_1 = 1$ or it has a prime factorization *pf*; in the first case q_1 itself is a prime factorization of $m = q_1(n/q_1 - n/p_1)$, and in the other the product of q_1 and *pf* is a prime factorization of m; in either case, q_1 is a member of some prime factorization of m. But by (ii), the inductive assumption applies to m; so m has only one prime factorization. Reject the assumption: there are no distinct collections of primes, $p_1 \ldots p_a$ and $q_1 \ldots q_b$ such that $k = p_1 \times \cdots \times p_a = q_1 \times \cdots \times q_b$.

Indct: For any $i > 1$, if i is the product of the primes $p_1, \ldots, p_a$, then no distinct collection of primes $q_1, \ldots, q_b$ is such that i is the product of them.

some individual to which no constant is assigned. But when we come to showing by induction that there is a model for our big set, it will be important that the status of a universal *is* reflected at the level of its instances. We guarantee this by building the set to satisfy the following condition:

Scgt A set Δ of formulas is a *scapegoat* set iff for any sentence $\sim\forall x\mathcal{P}$, if $\Delta \vdash \sim\forall x\mathcal{P}$, then there is some constant a such that $\Delta \vdash \sim\mathcal{P}^x_a$.

Equivalently, Δ is a scapegoat set just in case any sentence $\exists x\mathcal{P}$ is such that if $\Delta \vdash \exists x\mathcal{P}$, then there is some constant a such that $\Delta \vdash \mathcal{P}^x_a$. In a scapegoat set, we assert the existence of a particular individual (a *scapegoat*) corresponding to any existential claim. Notice that since $\sim\forall x\mathcal{P}$ is a sentence, $\sim\mathcal{P}^x_a$ is a sentence too.

So we set out to construct from Σ' a maximal consistent scapegoat set. As before, the idea is to line the sentences up, and consider them for inclusion one by one. In addition, this time, we consider an enumeration of constants $c_1, c_2, \ldots$ not in Σ' and for any included sentence of the form $\sim\forall x\mathcal{P}$, add $\sim\mathcal{P}^x_c$ where c does not so-far appear in the construction. We have assumed that $\mathcal{L}'$ includes infinitely many constants not in Γ or $\mathcal{P}$; so there are infinitely many constants not in $\Sigma' = \Gamma \cup \{\sim\mathcal{P}^u\}$; so at each (finite) stage i of the construction, there remain constants to include.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7, there is an enumeration, $\mathcal{Q}_1, \mathcal{Q}_2$, … of all the sentences in $\mathcal{L}'$ and also an enumeration $c_1, c_2, \ldots$ of constants not in Σ'. Let $\Omega_0 = \Sigma'$. Then for any $i > 0$, let

$$\Omega_i = \Omega_{i-1} \quad \text{if} \quad \Omega_{i-1} \vdash \sim\mathcal{Q}_i$$

else,

$$\Omega_{i*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\} \quad \text{if} \quad \Omega_{i-1} \nvdash \sim\mathcal{Q}_i$$

and,

$$\Omega_i = \Omega_{i*} \quad \text{if} \quad \mathcal{Q}_i \text{ is not of the form } \sim\forall x\mathcal{P}$$

$$\Omega_i = \Omega_{i*} \cup \{\sim\mathcal{P}^x_c\} \quad \text{if} \quad \mathcal{Q}_i \text{ is of the form } \sim\forall x\mathcal{P};\ c \text{ the first constant not in } \Omega_{i*}$$

then,

$\Sigma'' = \bigcup_{i \geq 0} \Omega_i$—that is, Σ'' is the union of all the Ω_is

Beginning with set Σ' ($= \Omega_0$), we consider the sentences in the enumeration $\mathcal{Q}_1, \mathcal{Q}_2$, … one by one, adding a sentence just in case its negation is not already derivable. In addition, if $\mathcal{Q}_i$ is of the sort $\sim\forall x\mathcal{P}$, we add $\sim\mathcal{P}^x_c$ using a new constant. Observe that if c is not in Ω_{i*}, then c is not in $\sim\forall x\mathcal{P}$. Σ'' contains all the formulas in Σ', together with all the sentences added this way.

It remains to show that if Σ' is consistent, then Σ'' is a maximal consistent scapegoat set.

T10.8. If Σ' is consistent, then Σ'' is a maximal consistent scapegoat set.

Suppose Σ' is consistent. The proof comes to showing (a) Σ'' is maximal. (b) Each Ω_i is consistent. From this, (c) Σ'' is consistent. And (d) Σ'' is a scapegoat set.

(a) Σ'' is maximal. Suppose Σ'' is not maximal. Then there is some sentence $\mathcal{Q}_i$ such that both $\Sigma'' \nvdash \mathcal{Q}_i$ and $\Sigma'' \nvdash \sim\mathcal{Q}_i$. For this i, by construction, each member of Ω_{i-1} is in Σ''; so if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$ then $\Sigma'' \vdash \sim\mathcal{Q}_i$; but $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_{i^*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$; and by construction again, $\mathcal{Q}_i \in \Sigma''$; so $\Sigma'' \vdash \mathcal{Q}_i$. This is impossible; reject the assumption: Σ'' is maximal.

(b) Each Ω_i is consistent. By induction on the series of Ω_is,

Basis: $\Omega_0 = \Sigma'$ and Σ' is consistent; so Ω_0 is consistent.

Assp: For any i, $0 \le i < k$, Ω_i is consistent.

Show: Ω_k is consistent.

Ω_k is either (i) Ω_{k-1}, (ii) $\Omega_{k^*} = \Omega_{k-1} \cup \{\mathcal{Q}_k\}$, or (iii) $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\}$.

(i) Suppose Ω_k is Ω_{k-1}. By assumption, Ω_{k-1} is consistent; so Ω_k is consistent.

(ii) Suppose Ω_k is $\Omega_{k^*} = \Omega_{k-1} \cup \{\mathcal{Q}_k\}$. Then by construction, $\Omega_{k-1} \nvdash \sim\mathcal{Q}_k$; so, since $\mathcal{Q}_k$ is a sentence, by T10.6, $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$ is consistent; so $\Omega_{k^*} = \Omega_k$ is consistent.

(iii) Suppose Ω_k is $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\}$ for c not in Ω_{k^*} and so not in $\sim\forall x\mathcal{P}$. In this case, as in (ii) above, Ω_{k^*} is consistent; and by construction $\sim\forall x\mathcal{P} \in \Omega_{k^*}$; so $\Omega_{k^*} \vdash \sim\forall x\mathcal{P}$.

Suppose Ω_k is inconsistent; then there are formulas $\mathcal{A}$ and $\sim\mathcal{A}$ such that $\Omega_k \vdash \mathcal{A}$ and $\Omega_k \vdash \sim\mathcal{A}$; so $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\} \vdash \mathcal{A}$ and $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\} \vdash \sim\mathcal{A}$. But since $\sim\mathcal{P}^x_c$ is a sentence, the restriction on DT is met, and both $\Omega_{k^*} \vdash \sim\mathcal{P}^x_c \to \mathcal{A}$ and $\Omega_{k^*} \vdash \sim\mathcal{P}^x_c \to \sim\mathcal{A}$; by A3, $\vdash (\sim\mathcal{P}^x_c \to \sim\mathcal{A}) \to [(\sim\mathcal{P}^x_c \to \mathcal{A}) \to \mathcal{P}^x_c]$; so by two instances of MP, $\Omega_{k^*} \vdash \mathcal{P}^x_c$. Consider some derivation of this result; by T8.10, we can switch c for some variable v that does not occur in the derivation, and the result is a derivation; so $\Omega_{k^*}{}^c_v \vdash [\mathcal{P}^x_c]^c_v$; but since c does not occur in Ω_{k^*}, $\Omega_{k^*}{}^c_v = \Omega_{k^*}$, and since c does not appear in $\sim\forall x\mathcal{P}$, it does not appear in $\mathcal{P}$, so with T8.3, $[\mathcal{P}^x_c]^c_v = \mathcal{P}^x_v$; so $\Omega_{k^*} \vdash \mathcal{P}^x_v$; so by Gen, $\Omega_{k^*} \vdash \forall v\mathcal{P}^x_v$; since v is new it is free for x in $\mathcal{P}$ and not free in $\forall x\mathcal{P}$ and by T9.11, $\vdash \forall v\mathcal{P}^x_v \to \forall x\mathcal{P}$; so by MP, $\Omega_{k^*} \vdash \forall x\mathcal{P}$. But $\Omega_{k^*} \vdash \sim\forall x\mathcal{P}$. So Ω_{k^*} is inconsistent. This is impossible; reject the assumption: Ω_k is consistent.

Ω_k is consistent.

Indct: For any i, Ω_i is consistent.

(c) Σ'' is consistent. Suppose Σ'' is not consistent; then there is some $\mathcal{A}$ such that $\Sigma'' \vdash \mathcal{A}$ and $\Sigma'' \vdash \sim\mathcal{A}$; consider derivations *D1* and *D2* of these results, and the premises $\mathcal{P}_a \ldots \mathcal{P}_b$ of these derivations. Let j be the least index such that each of $\mathcal{P}_a \ldots \mathcal{P}_b$ is a member of Ω_j: if all of $\mathcal{P}_a \ldots \mathcal{P}_b$ are members of Σ', then $j = 0$; otherwise take all the members of $\mathcal{P}_a \ldots \mathcal{P}_b$ not in Σ' (ones added by the construction), then for $\mathcal{Q}_z$ the last of these in the enumeration of setnences, $j = z$; in either case, by the construction of Σ'', each of $\mathcal{P}_a \ldots \mathcal{P}_b$ must be a member of Ω_j. So *D1* and *D2* are derivations from Ω_j; so Ω_j is inconsistent. But by (b), Ω_j is consistent. This is impossible; reject the assumption: Σ'' is consistent.

(d) Σ'' is a scapegoat set. Suppose $\Sigma'' \vdash \mathcal{Q}_i$, for some sentence $\mathcal{Q}_i$ of the form $\sim\forall x\mathcal{P}$. By (c), Σ'' is consistent; so $\Sigma'' \nvdash \sim\sim\forall x\mathcal{P}$; which is to say, $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so, $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_{i*} = \Omega_{i-1} \cup \{\sim\forall x\mathcal{P}\}$ and $\Omega_i = \Omega_{i*} \cup \{\sim\mathcal{P}^x_c\}$; so by construction, $\sim\mathcal{P}^x_c \in \Sigma''$; so $\Sigma'' \vdash \sim\mathcal{P}^x_c$. So if $\Sigma'' \vdash \sim\forall x\mathcal{P}$, then $\Sigma'' \vdash \sim\mathcal{P}^x_c$, and Σ'' is a scapegoat set.

For (c), a premise might be some *open formula* in Σ', and so not a member of the enumeration of sentences; even so, any such premise, together with premises added in the construction, remains a member of Ω_j and the argument goes through as before. More interesting is the case (iii) for consistency. In effect, we "push" a supposed inconsistency in Ω_k back to inconsistency in Ω_{k*}—and so to contradiction with its already established consistency. Given $\Omega_{k*} \vdash \sim\forall x\mathcal{P}$, the idea is to obtain $\Omega_{k*} \vdash \forall x\mathcal{P}$ toward the contradiction. For this, having shown that $\Omega_{k*} \vdash \mathcal{P}^x_c$ for c not in Ω_{k*} or in $\mathcal{P}$, we want to generalize to show that $\Omega_{k*} \vdash \forall x\mathcal{P}$. But generalization is on variables, not constants.[3] To get the generalization we want, we first use T8.10 to replace c in the derivation with a new variable v; this gets $\mathcal{P}^x_v$ and so by Gen $\forall v\mathcal{P}^x_v$. From this, $\forall x\mathcal{P}$ follows by exchange of bound variables.

E10.14. Let $\Sigma' = \{\forall x\sim Bx, Ca\}$ and consider enumerations of sentences and constants in $\mathcal{L}'$ that begin, Ab, Ba, $\sim\forall xCx, \ldots$ and $b, c, \ldots$. What are Ω_0, Ω_{1*}, Ω_1, Ω_{2*}, Ω_2, Ω_{3*}, Ω_3?

E10.15. Suppose some $\Omega_{i-1} = \{Ac, \forall x(Ax \to Bx)\}$. Show that Ω_i is inconsistent if $\mathcal{Q}_i = \sim\forall xBx$, and we add $\sim\forall xBx$ and then $\sim Bc$ to form Ω_{i*} and Ω_i. Why cannot this happen in the construction of Σ''?

[3]In fact, subject to appropriate constraints, some treatments allow generalization on constants (for example Bergmann, Moor, and Nelson, *The Logic Book*). Restricted to *sentences*, all the same arguments come out valid. Even so, this approach has the disconcerting consequence that $\forall$I and Gen apply between expressions that are not equivalent—and so not related as by our T7.6. Compare the variable semantics reference (Chapter 7 page 353).

10.3.4 The Model

We turn now to constructing the model M' for Σ'. Again the key is that the maximal consistent scapegoat set contains enough information to extract a specification for a model of the whole. As it turns out, the construction is simplified by our assumption that '=' does not appear in the language. A quantificational interpretation has a universe, with assignments to sentence letters, constants, function symbols, and relation symbols.

CnsM$'$ Let the universe U be the set of natural numbers, $\{0, 1, \ldots\}$. Then, where a *variable-free* term consists just of function symbols and constants, consider an enumeration $t_0, t_1, \ldots$ of all the variable-free terms in $\mathcal{L}'$. If t_z is a constant, set $\mathsf{M}'[t_z] = \mathsf{z}$. If $t_z = h^n t_a \ldots t_b$ for some function symbol h^n and n variable-free terms $t_a \ldots t_b$, then let $\langle\langle \mathsf{a} \ldots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathcal{S}$, let $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. And for a relation symbol $\mathcal{R}^n$, let $\langle \mathsf{a} \ldots \mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.[4]

Thus, for example, where t_1 and t_3 from the enumeration of terms are constants and $\Sigma'' \vdash \mathcal{R}t_1t_3$, then $\mathsf{M}'[t_1] = 1$, $\mathsf{M}'[t_3] = 3$, and $\langle 1, 3\rangle \in \mathsf{M}'[\mathcal{R}]$. Given this, it should be clear *why* $\mathcal{R}t_1t_3$ comes out satisfied on M': Put generally, where $t_a \ldots t_b$ are constants, we set $\mathsf{M}'[t_a] = \mathsf{a}$ and … and $\mathsf{M}'[t_b] = \mathsf{b}$; so by TA(c), for any variable assignment d, $\mathsf{M}'_\mathsf{d}[t_a] = \mathsf{a}$ and … and $\mathsf{M}'_\mathsf{d}[t_b] = \mathsf{b}$. So by SF(r), $\mathsf{M}'_\mathsf{d}[\mathcal{R}^n t_a \ldots t_b] = \mathsf{S}$ iff $\langle \mathsf{a} \ldots \mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$; by construction, iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$. Just as in the sentential case, our idea is to make atomic sentences true on M' just in case they are proved by Σ''.

Our aim has been to show that if Σ' is consistent, then Σ' has a model. We have constructed an interpretation M', and turn now to showing that M' is a model for Σ'. As in the sentential case, the main weight is carried by a preliminary theorem. And, as in the sentential case, the key is that we can appeal to special features of Σ'', this time that it is a maximal consistent scapegoat set. Notice that $\mathcal{P}$ is a *sentence*.

T10.9. If Σ' is consistent, then for any sentence $\mathcal{P}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Suppose Σ' is consistent and $\mathcal{P}$ is a sentence of $\mathcal{L}'$. From the former, by T10.8, Σ'' is a maximal consistent scapegoat set.

We begin with a preliminary result which connects arbitrary variable-free terms to our treatment of constants in the example above: For any variable-free term t_z and variable assignment d, $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{z}$. For this, suppose t_z is a variable-free term and d is an arbitrary variable assignment. By induction on the number of function symbols in t_z,

[4] It is common to let U just be the set of variable-free terms in $\mathcal{L}'$, and the interpretation of a term be itself. There is nothing the matter with this. However, working with the natural numbers emphasizes continuity with other models we have seen, and positions us for further results.

Basis: If t_z has no function symbols, then it is a constant. In this case, by construction, $\mathsf{M}'[t_z] = \mathsf{z}$; so by TA(c), $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

Assp: For any i, $0 \le i < k$, if t_z has i function symbols, then $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

Show: If t_z has k function symbols, then $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

If t_z has k function symbols, then it is of the form $h^n t_a \dots t_b$ for function symbol h^n and variable-free terms $t_a \dots t_b$ each with $< k$ function symbols. By assumption, $\mathsf{M}'_{\mathsf{d}}[t_a] = \mathsf{a}$ and ... and $\mathsf{M}'_{\mathsf{d}}[t_b] = \mathsf{b}$; and since $t_z = h^n t_a \dots t_b$ is a variable-free term, by construction $\langle\langle \mathsf{a} \dots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$. So $\mathsf{M}'_{\mathsf{d}}[t_z]$ is $\mathsf{M}'_{\mathsf{d}}[h^n t_a \dots t_b]$; by TA(f) this is $\mathsf{M}'[h^n]\langle \mathsf{M}'_{\mathsf{d}}[t_a] \dots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle$; by assumption, this is $\mathsf{M}'[h^n]\langle \mathsf{a} \dots \mathsf{b}\rangle$; and since $\langle\langle \mathsf{a} \dots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$, this is just z. So $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

Indct: For any t_z, $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

Given this, we are ready to show, by induction on the number of operators in $\mathcal{P}$, that $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$. Recall that $\mathcal{P}$ is a sentence.

Basis: If $\mathcal{P}$ is a sentence with no operators, then it is a sentence letter $\mathcal{S}$, or an atomic $\mathcal{R}^n t_a \dots t_b$ for relation symbol $\mathcal{R}^n$ and variable-free terms $t_a \dots t_b$. In the first case, by construction, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. In the second case, by TI, $\mathsf{M}'[\mathcal{R}^n t_a \dots t_b] = \mathsf{T}$ iff for arbitrary d, $\mathsf{M}'_{\mathsf{d}}[\mathcal{R}^n t_a \dots t_b] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{M}'_{\mathsf{d}}[t_a] \dots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle \in \mathsf{M}'[\mathcal{R}^n]$; since $t_a \dots t_b$ are variable-free terms, by the above result, iff $\langle \mathsf{a} \dots \mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$; by construction, iff $\Sigma'' \vdash \mathcal{R}^n t_a \dots t_b$. In either case, then, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Assp: For any i, $0 \le i < k$ if a sentence $\mathcal{P}$ has i operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Show: If a sentence $\mathcal{P}$ has k operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form, $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, or $\forall x \mathcal{A}$, for variable x and $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Homework. Hint: Given T8.8, your reasoning may be very much as in the sentential case.

($\to$) Suppose $\mathcal{P}$ is $\mathcal{A} \to \mathcal{B}$. Homework.

($\forall$) Suppose $\mathcal{P}$ is $\forall x \mathcal{A}$. Then since $\mathcal{P}$ is a sentence, x is the only variable that could be free in $\mathcal{A}$.

(i) Suppose $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ but $\Sigma'' \nvdash \mathcal{P}$; from the latter, $\Sigma'' \nvdash \forall x \mathcal{A}$; since Σ'' is maximal, $\Sigma'' \vdash \sim\forall x \mathcal{A}$; and since Σ'' is a scapegoat set, for some constant c, $\Sigma'' \vdash \sim\mathcal{A}^x_c$; so by consistency, $\Sigma'' \nvdash \mathcal{A}^x_c$; but $\mathcal{A}^x_c$ is a sentence; so by assumption, $\mathsf{M}'[\mathcal{A}^x_c] \ne \mathsf{T}$; so by TI, for some d, $\mathsf{M}'_{\mathsf{d}}[\mathcal{A}^x_c] \ne \mathsf{S}$; but, where c is some t_a, by construction, $\mathsf{M}'[c] = \mathsf{a}$; so by TA(c), $\mathsf{M}'_{\mathsf{d}}[c] = \mathsf{a}$; so, since c is free for x in $\mathcal{A}$, by T10.2, $\mathsf{M}'_{\mathsf{d}(x|\mathsf{a})}[\mathcal{A}] \ne \mathsf{S}$; so by SF($\forall$),

$\mathsf{M}'_{\mathsf{d}}[\forall x\mathcal{A}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\forall x\mathcal{A}] \neq \mathsf{T}$; and this is just to say, $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$. But this is impossible; reject the assumption: if $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$, then $\Sigma'' \vdash \mathcal{P}$.

(ii) Suppose $\Sigma'' \vdash \mathcal{P}$ but $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; from the latter, $\mathsf{M}'[\forall x\mathcal{A}] \neq \mathsf{T}$; so by TI, there is some d such that $\mathsf{M}'_{\mathsf{d}}[\forall x\mathcal{A}] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{a} \in \mathsf{U}$ such that $\mathsf{M}'_{\mathsf{d}(x|\mathsf{a})}[\mathcal{A}] \neq \mathsf{S}$; but for variable-free term t_a, by our above result, $\mathsf{M}'_{\mathsf{d}}[t_a] = \mathsf{a}$, and since t_a is variable-free, it is free for x in $\mathcal{A}$, so by T10.2, $\mathsf{M}'_{\mathsf{d}}[\mathcal{A}^x_{t_a}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\mathcal{A}^x_{t_a}] \neq \mathsf{T}$; but $\mathcal{A}^x_{t_a}$ is a sentence; so by assumption, $\Sigma'' \nvdash \mathcal{A}^x_{t_a}$; so by the maximality of Σ'', $\Sigma'' \vdash \sim\mathcal{A}^x_{t_a}$; but t_a is free for x in $\mathcal{A}$, so by A4, $\vdash \forall x\mathcal{A} \to \mathcal{A}^x_{t_a}$; so by MT, $\Sigma'' \vdash \sim\forall x\mathcal{A}$; so by the consistency of Σ'', $\Sigma'' \nvdash \forall x\mathcal{A}$; which is to say, $\Sigma'' \nvdash \mathcal{P}$. This is impossible; reject the assumption: if $\Sigma'' \vdash \mathcal{P}$, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$.

If $\mathcal{P}$ has k operator symbols, then $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

Indct: For any sentence $\mathcal{P}$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

We are now just one step away from ($\star$). It will be easy to see that $\mathsf{M}'[\Sigma'] = \mathsf{T}$, and so to reach the final result.

E10.16. Complete the $\sim$ and $\to$ cases to complete the demonstration of T10.9. You should set up the complete demonstration, but may refer to the text for cases completed there, as the text refers cases to homework.

E10.17. Reconsider Ω_3 from E10.14 along with an enumeration of constants that begins, $a, b, c, \ldots$. Where $\mathsf{U} = \{1, 2, 3\}$, find the (partial) interpretation that results by our method, and show it makes all the members of Ω_3 true (your reasoning may be relatively informal).

10.3.5 Final Result

And now we are in a position to get the final result. With some care about the distinction between formulas and sentences, this works just as before. First,

T10.10. If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

Suppose Σ' is consistent, but $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{P} \in \Sigma'$ such that $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$. Since $\mathcal{P} \in \Sigma'$, by construction, $\mathcal{P} \in \Sigma''$, so $\Sigma'' \vdash \mathcal{P}$; so, where $\mathcal{P}^u$ is the universal closure of $\mathcal{P}$, by application of Gen as necessary, $\Sigma'' \vdash \mathcal{P}^u$; so since $\mathcal{P}^u$ is a sentence and Σ' is consistent, by T10.9, $\mathsf{M}'[\mathcal{P}^u] = \mathsf{T}$; so by applications of T7.6 as necessary, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$. This is impossible; reject the assumption: if Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

Notice that this result applies to arbitrary sets of *formulas*. We bridge between $\mathcal{P}^u$ and $\mathcal{P}$ by Gen and T7.6 (in the direction from $\mathsf{I}[\forall x\mathcal{P}] = \mathsf{T}$ to $\mathsf{I}[\mathcal{P}] = \mathsf{T}$). But now we have the $(\star)$ that we have needed for c̊ompleteness.

So that is it! All we needed for the proof of c̊ompleteness in this case with $\mathcal{L}'$ restricted was $(\star)$. And we have it. So here is the final argument. Suppose the members of Γ and $\mathcal{P}$ are formulas of $\mathcal{L}'$.

T10.11_r. If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Quantificational C̊ompleteness.* ($\mathcal{L}'$ restricted)

> Suppose $\Gamma \vDash \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment that $\Gamma \vdash {\sim}{\sim}\mathcal{P}^u$; by T3.10, $\vdash {\sim}{\sim}\mathcal{P}^u \to \mathcal{P}^u$; so by MP, $\Gamma \vdash \mathcal{P}^u$; so by repeated applications of A4 and MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash {\sim}{\sim}\mathcal{P}^u$. Given this, since ${\sim}{\sim}\mathcal{P}^u$ is a sentence, by T10.6, $\Gamma \cup \{{\sim}\mathcal{P}^u\} = \Sigma'$ is consistent; so by T10.10, there is a model M' such that $\mathsf{M}'[\Gamma \cup \{{\sim}\mathcal{P}^u\}] = \mathsf{T}$. So $\mathsf{M}'[\Gamma] = \mathsf{T}$ and $\mathsf{M}'[{\sim}\mathcal{P}^u] = \mathsf{T}$; from the latter, by T8.8, $\mathsf{M}'[\mathcal{P}^u] \neq \mathsf{T}$; so by repeated applications of T7.6, $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}'[\Gamma] = \mathsf{T}$ but $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \nvDash \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$.

This time we bridge between $\mathcal{P}^u$ and $\mathcal{P}$ by A4 and T7.6 (in the other direction, so if $\mathsf{I}[\forall x\mathcal{P}] \neq \mathsf{T}$ then $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$).

Again, you should try to get the complete picture in your mind: The key is that consistent sets always have models. If there is no derivation of $\mathcal{P}$ from Γ, then $\Gamma \cup \{{\sim}\mathcal{P}\}$ is consistent; and if $\Gamma \cup \{{\sim}\mathcal{P}\}$ is consistent, then it has a model—so that $\Gamma \nvDash \mathcal{P}$. Put the other way around, if $\Gamma \vDash \mathcal{P}$, there is a derivation of $\mathcal{P}$ from Γ. We get the key point, that consistent sets have models, by finding a relation between consistent, and maximal consistent scapegoat sets. If a set is maximal and consistent and a scapegoat set, then it contains enough information to specify a model for the whole. The model for the big set then guarantees the existence of a model M' for the original Γ. All of this is very much parallel to the sentential case.

E10.18. Consider again $A^\star$ of E9.5. Provide a complete demonstration that $A^\star$ is c̊omplete—that if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash_{A^\star} \mathcal{P}$. You may suppose the language has no symbol for equality, and there are infinitely many constants not in Γ or $\mathcal{P}$; and you may appeal to any results from the text whose demonstration remains unchanged, but should recreate parts whose demonstration is not the same (but you may simply assume $\star$-versions of T10.2 and any required Chapter 8 theorems).

> Hints: You will need to redefine consistency, maximality, and scapegoat set for the new context. Where the free variables of $\mathcal{P}$ are $x_a, x_b, \ldots, x_n$, let the closure of $\mathcal{P}$ be ${\sim}\exists x_a \exists x_b \ldots \exists x_n {\sim}\mathcal{P}$. You have DT from E9.8, and may appeal to E9.16, as well as to theorems from E3.5. It will be helpful to establish as a preliminary to the theorem that ${\sim}\mathcal{P} \vdash_{A^\star} {\sim}\exists x\mathcal{P}$; for this you will find it helpful to obtain ${\sim}\mathcal{P} \vdash_{A^\star} \mathcal{P} \to (\mathcal{Z} \wedge {\sim}\mathcal{Z})$ as an intermediate result. Also you will reach a point

where it will be helpful to have $\sim\exists x\mathcal{P} \vdash_{A^\star} \sim\mathcal{P}$ and, as a corollary (or alternative) to T7.6, that for any I and $\mathcal{P}$, I[$\sim\mathcal{P}$] = T iff I[$\sim\exists x\mathcal{P}$] = T.

10.4 Quantificational Completeness: Full Version

So far, we have shown that if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$ where the members of Γ and $\mathcal{P}$ are formulas of an $\mathcal{L}'$ which does not include '=' and has infinitely many constants not in Γ or $\mathcal{P}$. Now allow that the members of Γ and $\mathcal{P}$ are in an arbitrary quantificational language $\mathcal{L}$. Then we shall require not ($\star$) according to which a consistent set in $\mathcal{L}'$ has a model M′, but the more general,

($\star\star$) For any consistent set of formulas Σ in an arbitrary quantificational language $\mathcal{L}$, there is an interpretation M such that M[Σ] = T.

Given this, reasoning is exactly as before:

1.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ has a model	$\Longrightarrow$	$\Gamma \nvDash \mathcal{P}$	
2.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathcal{P}^u\}$ has a model	($\star\star$)
3.	$\Gamma \cup \{\sim\mathcal{P}^u\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathcal{P}$	

Reasoning for (1) and (3) remains the same. (2) is ($\star\star$). And from (1)–(3) it follows that if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Supposing that ($\star\star$) has application to arbitrary sets of formulas from $\mathcal{L}$, the result has application to arbitrary premises and conclusion from $\mathcal{L}$. So we are left with two issues relative to our reasoning from before: $\mathcal{L}$ might lack the infinitely many constants not in Γ or $\mathcal{P}$, and $\mathcal{L}$ might include equality.

10.4.1 Adding Constants

Suppose $\mathcal{L}$ does not have infinitely many constants not in Γ or $\mathcal{P}$. This can happen in different ways. Perhaps $\mathcal{L}$ simply does not have infinitely many constants. Or perhaps the constants of $\mathcal{L}$ are $a_0, a_1, \ldots$ and $\Gamma = \{\mathcal{R}a_0, \mathcal{R}a_1, \ldots\}$; then $\mathcal{L}$ has infinitely many constants, but there are not any constants in $\mathcal{L}$ that do not appear in Γ. And we need the extra constants for construction of the maximal consistent scapegoat set. To avoid this sort of worry, we simply *add* infinitely many constants to form a language $\mathcal{L}'$ out of $\mathcal{L}$.

Cns$\mathcal{L}'$ Where $\mathcal{L}$ is a language whose constants are members of $a_0, a_1, \ldots$ let $\mathcal{L}'$ be like $\mathcal{L}$ but with the addition of new constants $c_0, c_1, \ldots$.

By reasoning as in the Chapter 2 countability reference, insofar as they can be lined up, $a_0, c_0, a_1, c_1, \ldots$ the collection of constants remains countable, so that $\mathcal{L}'$ remains a perfectly legitimate quantificational language. Clearly, every formula of $\mathcal{L}$ remains a formula of $\mathcal{L}'$. Thus, where Σ is a set of formulas in language $\mathcal{L}$, let Σ' be like Σ except that its members are formulas of language $\mathcal{L}'$.

Our reasoning for ($\star$) has application to sets of the sort Σ'. That is, where $\mathscr{L}'$ has infinitely many constants not in Σ', we have been able to find a maximal consistent scapegoat set Σ'', and from this a model M' for Σ'. But given an arbitrary Σ of formulas in $\mathscr{L}$, we need that *it* has a model M. That is, we shall have to establish a bridge between Σ and Σ', and between M' and M. Thus, to obtain ($\star\star$), we show,

2a.	Σ is consistent	$\Longrightarrow$	Σ' is consistent
2b.	Σ' is consistent	$\Longrightarrow$	Σ' has a model M'
2c.	Σ' has a model M'	$\Longrightarrow$	Σ has a model M

(2b) is just ($\star$) from before. And by a sort of hypothethical syllogism, together these yield ($\star\star$). So we need (2a) and (2c).

For the first result, we need that if Σ is consistent, then Σ' is consistent. Of course, Σ and Σ' contain just the same formulas, only formulas of the one are in a language with extra constants. But there might be *derivations* in $\mathscr{L}'$ from Σ' that are not derivations in $\mathscr{L}$ from Σ. So we need to show that these extra derivations do not result in contradiction. For this, the overall idea is simple: If we can derive a contradiction from Σ' in the enriched language then, by a modified version of that very derivation, we can derive a contradiction from Σ in the reduced language. So if there is no contradiction in the reduced language $\mathscr{L}$, then there is no contradiction in the enriched language $\mathscr{L}'$. The argument is straightforward given T8.10. Let Σ be a set of formulas in $\mathscr{L}$, and Σ' those same formulas in $\mathscr{L}'$. We show,

T10.12. If Σ is consistent, then Σ' is consistent.

Suppose Σ is consistent but Σ' is not. From the latter, there is a formula $\mathcal{A}$ in $\mathscr{L}'$ such that $\Sigma' \vdash \mathcal{A}$ and $\Sigma' \vdash \sim\mathcal{A}$; so by $\wedge$I, $\Sigma' \vdash \mathcal{A} \wedge \sim\mathcal{A}$, and (‡) there is a derivation of a contradiction from Σ'. By induction on the number of new constants which appear in a derivation $\Delta = \langle \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n \rangle$, we show that no Δ is a derivation of a contradiction from Σ'.

Basis: Suppose Δ contains no new constants and Δ is a derivation of some contradiction $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ'. Since Δ contains no new constants, every member of Δ is also a formula of $\mathscr{L}$, so $\Delta = \langle \mathcal{D}_1, \mathcal{D}_2, \ldots \rangle$ is a derivation of $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ; so by $\wedge$E, $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$; so Σ is not consistent. This is impossible; reject the assumption: Δ is not a derivation of a contradiction from Σ'.

Assp: For any i, $0 \leq i < k$, if Δ contains i new constants, then it is not a derivation of a contradiction from Σ'.

Show: If Δ contains k new constants, then it is not a derivation of a contradiction from Σ'.

Suppose Δ contains k new constants and is a derivation of a contradiction $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ'. Where c is one of the new constants in Δ and x is a variable not in Δ, by T8.10, Δ^c_x is a derivation of $[\mathcal{A} \wedge \sim\mathcal{A}]^c_x$ from Σ'^c_x. But all the members of Σ' are in $\mathscr{L}$; so c does not appear in any member of Σ'; so $\Sigma'^c_x = \Sigma'$. And $[\mathcal{A} \wedge \sim\mathcal{A}]^c_x = \mathcal{A}^c_x \wedge \sim[\mathcal{A}^c_x]$. So Δ^c_x is a derivation of a contradiction from Σ'. But Δ^c_x has $k - 1$ new constants and so, by assumption, is not a derivation of a contradiction from Σ'. This is impossible; reject the assumption: Δ is not a derivation of a contradiction from Σ'.

Indct: No derivation Δ is a derivation of a contradiction from Σ'.

So with (‡) there is and is not a derivation of a contradiction from Σ'. Reject the assumption: if Σ is consistent, then Σ' is consistent.

So if we have a consistent set of formulas in $\mathscr{L}$, and convert to $\mathscr{L}'$ with additional constants, we can be sure that the converted set is consistent as well.

With the extra constants in hand, all our reasoning goes through as before to show that there is a model M' for Σ'. Officially, though, an interpretation for some formulas in $\mathscr{L}'$ is not a model for some formulas in $\mathscr{L}$: A model for formulas in $\mathscr{L}$ has assignments for its constants, function symbols, and relation symbols, where a model for $\mathscr{L}'$ has assignments for *its* constants, function symbols, and relation symbols. A model M' for Σ', then, is not the same as a model M for Σ. But it is a short step to a solution.

CnsM Let M be like M' but without assignments to constants not in $\mathscr{L}$.

M is an interpretation for language $\mathscr{L}$. M and M' have exactly the same universe, and exactly the same interpretations for all the symbols that are in $\mathscr{L}$. It turns out that the evaluation of any formula in $\mathscr{L}$ is therefore the same on M as on M'—that is, for any $\mathcal{P}$ in $\mathscr{L}$, $\mathsf{M}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$. Given the way satisfaction builds from the parts to the whole, it may be that this is obvious. However, it is worthwhile to consider a proof. Thus we need the following matched pair of theorems (in fact, we show somewhat more than is necessary, as M and M' differ only by assignments to constants). The proofs are straightforward, and mostly left as an exercise. I do just enough to get you started.

Suppose $\mathscr{L}'$ extends $\mathscr{L}$ by the addition of some constants, function symbols, sentence letters, or relation symbols, and M' is like M except that it makes assignments to the constants, function symbols, sentence letters, and relation symbols in $\mathscr{L}'$ but not in $\mathscr{L}$.

*T10.13. For any variable assignment d, and for any term t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

The argument is by induction on the number of function symbols in t. Let d be a variable assignment, and t a term in $\mathcal{L}$.

Basis: If t has no function symbols, then $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$. Homework.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, then $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

Show: If t has k function symbols, then $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

If t has k function symbols, then it is of the form, $h^n t_1 \dots t_n$ for function symbol h^n and terms $t_1 \dots t_n$ with $< k$ function symbols. Since t is in $\mathcal{L}$, h^n and $t_1 \dots t_n$ are symbols in $\mathcal{L}$. By construction, $\mathsf{M}[h^n] = \mathsf{M'}[h^n]$; and by assumption, $\mathsf{M_d}[t_1] = \mathsf{M'_d}[t_1]$ and … and $\mathsf{M_d}[t_n] = \mathsf{M'_d}[t_n]$. So with TA(f), $\mathsf{M_d}[t] = \mathsf{M_d}[h^n t_1 \dots t_n] = \mathsf{M}[h^n]\langle \mathsf{M_d}[t_1] \dots \mathsf{M_d}[t_n]\rangle = \mathsf{M'}[h^n]\langle \mathsf{M'_d}[t_1] \dots \mathsf{M'_d}[t_n]\rangle = \mathsf{M'_d}[h^n t_1 \dots t_n] = \mathsf{M'_d}[t]$.

Indct: For any t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

*T10.14. For any variable assignment d, and for any formula $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

The argument is by induction on the number of operator symbols in $\mathcal{P}$. Let d be a variable assignment, and $\mathcal{P}$ a formula in $\mathcal{L}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ in $\mathcal{L}$, or an atomic $\mathcal{R}^n t_1 \dots t_n$ for relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$ in $\mathcal{L}$. In the first case, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{S}] = \mathsf{S}$; by SF(s), iff $\mathsf{M}[\mathcal{S}] = \mathsf{T}$; by construction iff $\mathsf{M'}[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{M'_d}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$. For the second case, by construction, $\mathsf{M}[\mathcal{R}^n] = \mathsf{M'}[\mathcal{R}^n]$; and by T10.13, $\mathsf{M_d}[t_1] = \mathsf{M'_d}[t_1]$ and … and $\mathsf{M_d}[t_n] = \mathsf{M'_d}[t_n]$. So $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; by SF(r) iff $\langle \mathsf{M_d}[t_1] \dots \mathsf{M_d}[t_n]\rangle \in \mathsf{M}[\mathcal{R}^n]$; iff $\langle \mathsf{M'_d}[t_1] \dots \mathsf{M'_d}[t_n]\rangle \in \mathsf{M'}[\mathcal{R}^n]$; by SF(r) iff $\mathsf{M'_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$. In either case, then, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, and any variable assignment d, if $\mathcal{P}$ has i operator symbols, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

Show: For any variable assignment d, if $\mathcal{P}$ has k operator symbols, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$. Homework.

Indct: For any formula $\mathcal{P}$ of $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

And now we are in a position to show that M is indeed a model for Σ. In particular, it is easy to show, T10.15. If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

Suppose $\mathsf{M}'[\Sigma'] = \mathsf{T}$, but $\mathsf{M}[\Sigma] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{P} \in \Sigma$ such that $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{M_d}[\mathcal{P}] \neq \mathsf{S}$; so by T10.14, $\mathsf{M}'_{\mathsf{d}}[\mathcal{P}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; and since $\mathcal{P} \in \Sigma$, we have $\mathcal{P} \in \Sigma'$; so $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. This is impossible; reject the assumption: if $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

Finally, T10.12, T10.10, and T10.15 together yield a preliminary version of the theorem we are after.

T10.16$_{\text{p}}$. If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ without equality)

Suppose Σ is consistent; then by T10.12, Σ' is consistent; so by T10.10, Σ' has a model M'; so by T10.15, Σ has a model M.

And that is what we needed to recover the $\mathring{\text{c}}$ompleteness result for $\mathcal{L}$ without the constraint on constants. Where $\mathcal{L}$ does not include infinitely many constants not in Γ or $\mathcal{P}$, we simply add them to form $\mathcal{L}'$. Our theorems from this section ensure that the results go through as before.

*E10.19. Complete the proof of T10.13. You should set up the complete induction, but may refer to the text, as the text refers to homework.

*E10.20. Complete the proof of T10.14. As usual, you should set up the complete induction, but may refer to the text for cases completed there, as the text refers to homework.

E10.21. Adapt the demonstration of T10.11$_{\text{r}}$ for the supposition that $\mathcal{L}$ need not be the same as $\mathcal{L}'$. You may appeal to theorems from this section.

10.4.2 Accommodating Equality

Dropping the assumption that language $\mathcal{L}$ lacks the symbol '$=$' for equality results in another sort of complication. In constructing our models, where t_1 and t_3 from the enumeration of variable-free terms are constants and $\Sigma'' \vdash \mathcal{R}t_1t_3$, we set $\mathsf{M}'[t_1] = 1$, $\mathsf{M}'[t_3] = 3$, and $\langle 1, 3\rangle \in \mathsf{M}'[\mathcal{R}]$. Suppose $\mathcal{R}$ is the equal sign, '$=$'; then by our procedure, $\langle 1, 3\rangle \in \mathsf{M}'[=]$. But this is wrong! Where $\mathsf{U} = \{0, 1, \ldots\}$, the proper interpretation of '$=$' is $\{\langle 0, 0\rangle, \langle 1, 1\rangle, \ldots\}$, and $\langle 1, 3\rangle$ is not a member of this set. So our procedure does not result in the specification of a legitimate model. The procedure works fine for relation symbols other than equality. There are no restrictions on assignments to other relation symbols, so nothing stops us from specifying interpretations as above. But there is a restriction on the interpretation of '$=$'. So we cannot proceed blindly this way.

Here is the nub of a solution: Say $\Sigma'' \vdash a_1 = a_3$; then let the *set* $\{1, 3\}$ be an element of U, and let $\mathsf{M}'[a_1] = \mathsf{M}'[a_3] = \{1, 3\}$. Similarly, if $a_2 = a_4$ and $a_4 = a_5$ are consequences of Σ'', let $\{2, 4, 5\}$ be a member of U, and $\mathsf{M}'[a_2] = \mathsf{M}'[a_4] = \mathsf{M}'[a_5] = \{2, 4, 5\}$. That is, let U consist of certain *sets* of natural numbers—where these sets are specified by atomic equalities that are consequences of Σ''. Then let $\mathsf{M}'[a_z]$ be the set of which z is a member. Given this, if $\Sigma'' \vdash \mathscr{R}^n t_a \dots t_b$, then include the tuple consisting of the set assigned to t_a and ... and the set assigned to t_b, in the interpretation of $\mathscr{R}^n$. So on the above interpretation of the constants, if $\Sigma'' \vdash \mathscr{R} a_1 a_4$, then $\langle\{1, 3\}, \{2, 4, 5\}\rangle \in \mathsf{M}'[\mathscr{R}]$. And if $\Sigma'' \vdash a_1 = a_3$, then $\langle\{1, 3\}, \{1, 3\}\rangle \in \mathsf{M}'[=]$. You should see why this is so. And it is just right! If $\{1, 3\} \in \mathsf{U}$, then $\langle\{1, 3\}, \{1, 3\}\rangle$ *should* be in $\mathsf{M}'[=]$. So we respond to the problem by a revision of the specification for CnsM′.

Let us now turn to the details. Our idea has been to make atomic sentences true on M′ just in case they are proved by Σ''. We want an interpretation that preserves this feature while accommodating equality. A model consists of a universe U, along with assignments to constants, function symbols, sentence letters, and relation symbols. We take up these elements, one after another.

The universe. The elements of our universe U are to be certain sets of natural numbers.[5] Consider an enumeration $t_0, t_1, \dots$ of all the variable-free terms in $\mathscr{L}'$, and let there be a relation $\simeq$ on the set $\{0, 1, \dots\}$ of natural numbers such that $\mathsf{i} \simeq \mathsf{j}$ iff $\Sigma'' \vdash t_i = t_j$. Let [n] be the set of natural numbers which stand in the $\simeq$ relation to n—that is, $[\mathsf{n}] = \{\mathsf{z} \in \mathbb{N} \mid \mathsf{z} \simeq \mathsf{n}\}$. So $\mathsf{z} \in [\mathsf{n}]$ iff $\mathsf{z} \simeq \mathsf{n}$. Notice that if the things which stand in the $\simeq$ relation to m are the same as the ones that stand in that relation to n then [m] = [n]. The universe U of M′ is then the collection of all these sets—that is,

CnsM′ For any natural number, the universe includes the class corresponding to it.
$\mathsf{U} = \{[\mathsf{n}] \mid \mathsf{n} \in \mathbb{N}\}$.

The way this works is really quite simple. If according to Σ'', t_1 equals only itself, then the only z such that $\mathsf{z} \simeq 1$ is 1; so $[1] = \{1\}$, and this is a member of U. If, according to Σ'', t_1 equals just itself and t_2, then $1 \simeq 2$ so that $[1] = [2] = \{1, 2\}$, and this set is a member of U. If, according to Σ'', t_1 equals itself, t_2, and t_3, then $1 \simeq 2 \simeq 3$ so that $[1] = [2] = [3] = \{1, 2, 3\}$, and this set is a member of U. And so forth.

In order to make progress, it will be convenient to establish some facts about the $\simeq$ relation, and about the sets in U. Recall that $\simeq$ is a relation on the *natural numbers* which is specified relative to expressions in Σ'', so that $\mathsf{i} \simeq \mathsf{j}$ iff $\Sigma'' \vdash t_i = t_j$. First we show that $\simeq$ is *reflexive*, *symmetric*, and *transitive*.

[5]Again, it is common to let the universe be *sets of terms* in $\mathscr{L}'$ (see note 4 on page 472).

Reflexivity. For any $\mathsf{i} \in \mathbb{N}$, $\mathsf{i} \simeq \mathsf{i}$: By T3.33, $\vdash t_i = t_i$; so $\Sigma'' \vdash t_i = t_i$; so by construction, $\mathsf{i} \simeq \mathsf{i}$.

Symmetry. For any $\mathsf{i}, \mathsf{j} \in \mathbb{N}$, if $\mathsf{i} \simeq \mathsf{j}$, then $\mathsf{j} \simeq \mathsf{i}$: Suppose $\mathsf{i} \simeq \mathsf{j}$; then by construction, $\Sigma'' \vdash t_i = t_j$; but by T3.34, $\vdash t_i = t_j \to t_j = t_i$; so by MP, $\Sigma'' \vdash t_j = t_i$; so by construction, $\mathsf{j} \simeq \mathsf{i}$.

Transitivity. For any $\mathsf{i}, \mathsf{j}, \mathsf{k} \in \mathbb{N}$, if $\mathsf{i} \simeq \mathsf{j}$ and $\mathsf{j} \simeq \mathsf{k}$, then $\mathsf{i} \simeq \mathsf{k}$: Suppose $\mathsf{i} \simeq \mathsf{j}$ and $\mathsf{j} \simeq \mathsf{k}$; then by construction, $\Sigma'' \vdash t_i = t_j$ and $\Sigma'' \vdash t_j = t_k$; but by T3.35, $\vdash t_i = t_j \to (t_j = t_k \to t_i = t_k)$; so by two instances of MP, $\Sigma'' \vdash t_i = t_k$; so by construction, $\mathsf{i} \simeq \mathsf{k}$.

A relation which is reflexive, symmetric, and transitive is an *equivalence* relation. As an equivalence relation, it divides or *partitions* the members of $\{0, 1, \ldots\}$ into mutually exclusive classes such that each member of a class bears $\simeq$ to each of the others in its partition, but not to members outside the partition. More particularly, because $\simeq$ is an equivalence relation, the collections $[\mathsf{n}] = \{\mathsf{z} \in \mathbb{N} \mid \mathsf{z} \simeq \mathsf{n}\}$ in U are characterized as follows:

Self-membership. For any $\mathsf{n} \in \mathbb{N}$, $\mathsf{n} \in [\mathsf{n}]$: By reflexivity, $\mathsf{n} \simeq \mathsf{n}$; so by construction, $\mathsf{n} \in [\mathsf{n}]$. Corollary: Every natural number n is a member of at least one class.

Uniqueness. For any $\mathsf{i} \in \mathbb{N}$, i is an element of at most one class: Suppose i is an element of more than one class; then there are some m and n such that $\mathsf{i} \in [\mathsf{m}]$ and $\mathsf{i} \in [\mathsf{n}]$ but $[\mathsf{m}] \neq [\mathsf{n}]$. Since $[\mathsf{m}] \neq [\mathsf{n}]$ there is some j such that $\mathsf{j} \in [\mathsf{m}]$ and $\mathsf{j} \notin [\mathsf{n}]$, or $\mathsf{j} \in [\mathsf{n}]$ and $\mathsf{j} \notin [\mathsf{m}]$; without loss of generality, suppose $\mathsf{j} \in [\mathsf{m}]$ and $\mathsf{j} \notin [\mathsf{n}]$. Since $\mathsf{j} \in [\mathsf{m}]$, by construction, $\mathsf{j} \simeq \mathsf{m}$; and since $\mathsf{i} \in [\mathsf{m}]$, by construction $\mathsf{i} \simeq \mathsf{m}$; so by symmetry, $\mathsf{m} \simeq \mathsf{i}$; so $\mathsf{j} \simeq \mathsf{m}$ and $\mathsf{m} \simeq \mathsf{i}$ and by transitivity, $\mathsf{j} \simeq \mathsf{i}$. Since $\mathsf{i} \in [\mathsf{n}]$, by construction $\mathsf{i} \simeq \mathsf{n}$; so by transitivity again, $\mathsf{j} \simeq \mathsf{n}$; so by construction, $\mathsf{j} \in [\mathsf{n}]$. This is impossible; reject the assumption: i is an element of at most one class.

Equality. For any $\mathsf{m}, \mathsf{n} \in \mathbb{N}$, $\mathsf{m} \simeq \mathsf{n}$ iff $[\mathsf{m}] = [\mathsf{n}]$: (i) Suppose $\mathsf{m} \simeq \mathsf{n}$. Then by construction, $\mathsf{m} \in [\mathsf{n}]$; but by self-membership, $\mathsf{m} \in [\mathsf{m}]$; so by uniqueness, $[\mathsf{m}] = [\mathsf{n}]$. Suppose $[\mathsf{m}] = [\mathsf{n}]$; by self-membership, $\mathsf{m} \in [\mathsf{m}]$; so $\mathsf{m} \in [\mathsf{n}]$; so by construction, $\mathsf{m} \simeq \mathsf{n}$.

Corresponding to the relations by which they are formed, classes characterized by self-membership, uniqueness, and equality are *equivalence classes*. From self-membership and uniqueness, every natural number n is a member of exactly one such class. And from equality, $\mathsf{m} \simeq \mathsf{n}$ just when $[\mathsf{m}]$ is the very same thing as $[\mathsf{n}]$. So, for example, if $1 \simeq 1$, and $1 \simeq 2$, and $2 \simeq 1$, and $2 \simeq 2$ (and nothing else), then $[1] = [2] = \{1, 2\}$. You should be able to see that these formal specifications develop just the informal picture with which we began.

Terms. The specification for constants is simple:

CnsM′ If t_z in the enumeration of variable-free terms $t_1, t_2, \ldots$ is a constant, then $\mathsf{M}'[t_z] = [\mathsf{z}]$.

Thus, with self-membership, any constant t_z designates the equivalence class of which z is a member. In this case, we need to be sure that the specification picks out exactly one member of U for each constant. The specification would fail if the relation $\simeq$ generated classes such that some natural number was an element of no class, or some was an element of more than one. But, as we have just seen, by self-membership and uniqueness, every natural number z is a member of exactly one class. So far, so good!

CnsM′ If t_z in the enumeration of variable-free terms $t_1, t_2, \ldots$ is $\mathscr{h}^n t_a \ldots t_b$ for function symbol $\mathscr{h}^n$ and variable-free terms $t_a \ldots t_b$, then $\langle\langle[\mathsf{a}] \ldots [\mathsf{b}]\rangle, [\mathsf{z}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$.

Thus when the input to $\mathscr{h}^n$ is $\langle[\mathsf{a}] \ldots [\mathsf{b}]\rangle$, the output is $[\mathsf{z}]$. This time, we must be sure that the result is a total function—that (i) there is at least one output object for every input n-tuple, and (ii) there is at most one output object associated with any one input n-tuple. The former worry is easily dispatched. The second concern is that we might have $\langle[\mathsf{a}], [\mathsf{u}]\rangle, \langle[\mathsf{b}], [\mathsf{v}]\rangle \in \mathsf{M}'[\mathscr{h}]$ but with $[\mathsf{a}] = [\mathsf{b}]$ and $[\mathsf{u}] \neq [\mathsf{v}]$; in this case, we fail to specify a function.

(i) There is at least one output object: Corresponding to any $\mathscr{h}^n$ and $\langle[\mathsf{a}] \ldots [\mathsf{b}]\rangle$ where $[\mathsf{a}] \ldots [\mathsf{b}]$ are members of U, consider the variable-free term $\mathscr{h}^n t_a \ldots t_b$; this is some t_z from the sequence $t_1, t_2, \ldots$; so by construction, $\langle\langle[\mathsf{a}] \ldots [\mathsf{b}]\rangle, [\mathsf{z}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$. So $\mathsf{M}'[\mathscr{h}^n]$ has an output object when the input is $\langle[\mathsf{a}] \ldots [\mathsf{b}]\rangle$.

(ii) There is at most one output object: Suppose $\langle\langle[\mathsf{a}] \ldots [\mathsf{c}]\rangle, [\mathsf{u}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$ and $\langle\langle[\mathsf{d}] \ldots [\mathsf{f}]\rangle, [\mathsf{v}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$, where $\langle[\mathsf{a}] \ldots [\mathsf{c}]\rangle = \langle[\mathsf{d}] \ldots [\mathsf{f}]\rangle$. Since $\langle[\mathsf{a}] \ldots [\mathsf{c}]\rangle = \langle[\mathsf{d}] \ldots [\mathsf{f}]\rangle$, $[\mathsf{a}] = [\mathsf{d}]$ and … and $[\mathsf{c}] = [\mathsf{f}]$; so by equality, $\mathsf{a} \simeq \mathsf{d}$ and … and $\mathsf{c} \simeq \mathsf{f}$; so by construction, $\Sigma'' \vdash t_a = t_d$ and … and $\Sigma'' \vdash t_c = t_f$. Since $\langle\langle[\mathsf{a}] \ldots [\mathsf{c}]\rangle, [\mathsf{u}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$ and $\langle\langle[\mathsf{d}] \ldots [\mathsf{f}]\rangle, [\mathsf{v}]\rangle \in \mathsf{M}'[\mathscr{h}^n]$, by construction, there are some variable-free terms, $t_u = \mathscr{h}^n t_a \ldots t_c$ and $t_v = \mathscr{h}^n t_d \ldots t_f$ in the enumeration. By =I, $\Sigma'' \vdash \mathscr{h}^n t_a \ldots t_c = \mathscr{h}^n t_a \ldots t_c$; so by repeated applications of =E, $\Sigma'' \vdash \mathscr{h}^n t_a \ldots t_c = \mathscr{h}^n t_d \ldots t_f$; but this is to say $\Sigma'' \vdash t_u = t_v$; so by construction, $\mathsf{u} \simeq \mathsf{v}$; so by equality, $[\mathsf{u}] = [\mathsf{v}]$.

So there is at least one, and at most one output object for any input n-tuple and, as they should be, function symbols are well-defined.

We are now in a position to recover an analogue to the preliminary result for demonstration of T10.9: For any variable-free term t_z and variable assignment d, $\mathsf{M}'_{\mathsf{d}}[t_z] = [\mathsf{z}]$. The argument is very much as before. Suppose t_z is a variable-free term. By induction on the number of function symbols in t_z, *Basis*: If t_z has no function symbols, then it is a constant. In this case, by construction, $\mathsf{M}'[t_z] = [\mathsf{z}]$; so by TA(c), $\mathsf{M}'_\mathsf{d}[t_z] = [\mathsf{z}]$.

Assp: For any i, $0 \le i < k$, if t_z has i function symbols, then $\mathsf{M}'_\mathsf{d}[t_z] = [\mathsf{z}]$.

Show: If t_z has k function symbols, then $\mathsf{M}'_\mathsf{d}[t_z] = [\mathsf{z}]$.

If t_z has k function symbols, then it is of the form, $h^n t_a \ldots t_b$ where $t_a \ldots t_b$ have $< k$ function symbols. Since $t_z = h^n t_a \ldots t_b$ is a variable-free term, $\langle\langle[\mathsf{a}]\ldots[\mathsf{b}]\rangle, [\mathsf{z}]\rangle \in \mathsf{M}'[h^n]$; by assumption, $\mathsf{M}'_\mathsf{d}[t_a] = [\mathsf{a}]$ and $\ldots$ and $\mathsf{M}'_\mathsf{d}[t_b] = [\mathsf{b}]$. So with TA(f), $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{M}'_\mathsf{d}[h^n t_a \ldots t_b] = \mathsf{M}'[h^n]\langle \mathsf{M}'_\mathsf{d}[t_a] \ldots \mathsf{M}'_\mathsf{d}[t_b]\rangle = \mathsf{M}'[h^n]\langle[\mathsf{a}]\ldots[\mathsf{b}]\rangle = [\mathsf{z}]$; so $\mathsf{M}'_\mathsf{d}[t_z] = [\mathsf{z}]$.

Indct: For any variable-free term t_z, $\mathsf{M}'_\mathsf{d}[t_z] = [\mathsf{z}]$.

So the interpretation of any variable-free term is the equivalence class corresponding to its position in the enumeration of terms.

Atomics. The result we have just seen for terms makes the specification for atomics particularly natural. Sentence letters are easy. As before,

CnsM′ For a sentence letter $\mathcal{S}$, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$.

Then for relation symbols, the idea is as sketched above. We simply let the assignment be such as to make a variable-free atomic come out true iff it is a consequence of Σ''.

CnsM′ For a relation symbol $\mathcal{R}^n$, where $t_a \ldots t_b$ are n members of the enumeration of variable-free terms, let $\langle[\mathsf{a}]\ldots[\mathsf{b}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.

To see that the specification for relation symbols is legitimate, we need to be clear that the specification is consistent—that we do not both assert and deny that some tuple is in the extension of $\mathcal{R}^n$, and we need to be sure that $\mathsf{M}'[=]$ is as it should be—that $\mathsf{M}'[=] = \{\langle[\mathsf{n}], [\mathsf{n}]\rangle \mid [\mathsf{n}] \in \mathsf{U}\}$. The case for equality is easy. The former concern is that we might have some $[\mathsf{a}] = [\mathsf{b}]$ where $[\mathsf{a}] \in \mathsf{M}'[\mathcal{R}]$ but $[\mathsf{b}] \notin \mathsf{M}'[\mathcal{R}]$.

(i) $\langle[\mathsf{m}], [\mathsf{n}]\rangle \in \mathsf{M}[=]$ iff $[\mathsf{m}] = [\mathsf{n}]$: By equality, $[\mathsf{m}] = [\mathsf{n}]$ iff $\mathsf{m} \simeq \mathsf{n}$; by construction iff $\Sigma'' \vdash t_m = t_n$; by construction iff $\langle[\mathsf{m}], [\mathsf{n}]\rangle \in \mathsf{M}'[=]$.

(ii) The specification is consistent: Suppose that $\langle[\mathsf{a}]\ldots[\mathsf{c}]\rangle = \langle[\mathsf{d}]\ldots[\mathsf{f}]\rangle$ and $\langle[\mathsf{a}]\ldots[\mathsf{c}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$. From the first of these, $[\mathsf{a}] = [\mathsf{d}]$ and $\ldots$ and $[\mathsf{c}] = [\mathsf{f}]$; so by equality, $\mathsf{a} \simeq \mathsf{d}$ and $\ldots$ and $\mathsf{c} \simeq \mathsf{f}$; so by construction, $\Sigma'' \vdash t_a = t_d$ and $\ldots$ and $\Sigma'' \vdash t_c = t_f$. But since $\langle[\mathsf{a}]\ldots[\mathsf{c}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$; by construction, $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_c$; so by repeated applications of =E, $\Sigma'' \vdash \mathcal{R}^n t_d \ldots t_f$; so by construction $\langle[\mathsf{d}]\ldots[\mathsf{f}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$. So if $\langle[\mathsf{a}]\ldots[\mathsf{c}]\rangle = \langle[\mathsf{d}]\ldots[\mathsf{f}]\rangle$ and $\langle[\mathsf{a}]\ldots[\mathsf{c}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$, then $\langle[\mathsf{d}]\ldots[\mathsf{f}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$.

This completes the specification of M'. The specification is more complex than for the basic version, and we have had to work to demonstrate its consistency. Still, the result is a perfectly ordinary model M', with a domain, assignments to constants, assignments to function symbols, assignments to sentence letters, and assignments to relation symbols.

With this revised specification for M', the demonstration of T10.9 proceeds as before. Here is the key portion of the basis. We are showing that $\mathsf{M}'[\mathscr{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathscr{P}$.

> Suppose $\mathscr{P}$ is an atomic sentence $\mathscr{R}^n t_a \ldots t_b$. By TI, $\mathsf{M}'[\mathscr{R}^n t_a \ldots t_b] = \mathsf{T}$ iff for arbitrary d, $\mathsf{M}'_{\mathsf{d}}[\mathscr{R}^n t_a \ldots t_b] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{M}'_{\mathsf{d}}[t_a] \ldots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle \in \mathsf{M}'[\mathscr{R}^n]$; since $t_a \ldots t_b$ are variable-free terms, by the preliminary result for terms, iff $\langle [\mathsf{a}] \ldots [\mathsf{b}]\rangle \in \mathsf{M}'[\mathscr{R}^n]$; by construction, iff $\Sigma'' \vdash \mathscr{R}^n t_a \ldots t_b$. So $\mathsf{M}'[\mathscr{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathscr{P}$.

So all that happens is that we depend on the conversion from individuals to sets of individuals for both assignments to terms and assignments to relation symbols. Given this, the argument is exactly parallel to the one from before.

E10.22. Suppose the enumeration of variable-free terms begins, $a, b, f^1 a, f^1 b, \ldots$ (so these are $t_0 \ldots t_3$) and, for these terms, Σ'' proves just $a = a$, $b = b$, $f^1 a = f^1 a$, $f^1 b = f^1 b$, $a = f^1 a$, and $f^1 a = a$. (i) What objects stand in the $\simeq$ relation? (ii) What are $[0]$, $[1]$, $[2]$, and $[3]$? (iii) Given this much, what things must be in U?

E10.23. Return to the case from E10.22. Explain how $\simeq$ satisfies reflexivity, symmetry, and transitivity. Explain how U satisfies self-membership, uniqueness, and equality.

E10.24. Where Σ'' and U are as in the previous two exercises, what are $\mathsf{M}'[a]$, $\mathsf{M}'[b]$, and $\mathsf{M}'[f]$? Supposing that $\Sigma'' \vdash R^1 a$, $R^1 f^1 a$, and $R^1 f^1 b$, but $\Sigma'' \nvdash R^1 b$, what is $\mathsf{M}'[R^1]$? Given the consequences of Σ'' from E10.22, by the method, what is $\mathsf{M}'[=]$? Is this as it should be? Explain.

E10.25. Complete the demonstration of T10.9 on the revised specification for M'.

10.4.3 The Final Result

We are really done with the demonstration of c̊ompleteness. Perhaps, though, it will be helpful to draw the parts together. Begin with some basic definitions, a simple theorem, and the construction of $\mathcal{L}'$.

Con A set Δ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Delta \vdash \mathcal{A}$ and $\Delta \vdash \sim\mathcal{A}$.

Max A set Δ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Delta \vdash \mathcal{A}$ or $\Delta \vdash \sim\mathcal{A}$.

Scgt A set Δ of formulas is a *scapegoat* set iff for any sentence $\sim\forall x\mathcal{P}$, if $\Delta \vdash \sim\forall x\mathcal{P}$, then there is some constant a such that $\Delta \vdash \sim\mathcal{P}^x_a$.

T10.6 For any set of formulas Δ and sentence $\mathcal{P}$, if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

Cns$\mathcal{L}'$ Where $\mathcal{L}$ is a language whose constants are members of $a_0, a_1, \ldots$ let $\mathcal{L}'$ be like $\mathcal{L}$ but with the addition of new constants $c_0, c_1, \ldots$.

Where Σ is a set of formulas in language $\mathcal{L}$, let Σ' be like Σ except that its members are formulas of language $\mathcal{L}'$. Then we proceed in language $\mathcal{L}'$, for a maximal consistent scapegoat set Σ'' constructed from any consistent Σ'.

T10.7 There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all the sentences, terms, and the like in $\mathcal{L}'$.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7, there is an enumeration, $\mathcal{Q}_1, \mathcal{Q}_2$, $\ldots$ of all the sentences in $\mathcal{L}'$ and also an enumeration $c_1, c_2, \ldots$ of constants not in Σ'. Let $\Omega_0 = \Sigma'$. Then for any $i > 0$, let $\Omega_i = \Omega_{i-1}$ if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$. Otherwise, $\Omega_{i^*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$ if $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$. Then $\Omega_i = \Omega_{i^*}$ if $\mathcal{Q}_i$ is not of the form $\sim\forall x\mathcal{P}$, and $\Omega_i = \Omega_{i^*} \cup \{\sim\mathcal{P}^x_c\}$ if $\mathcal{Q}_i$ is of the form $\sim\forall x\mathcal{P}$, where c is the first constant not in Ω_{i^*}. Then $\Sigma'' = \bigcup_{i \geq 0} \Omega_i$.

T10.8 If Σ' is consistent, then Σ'' is a maximal consistent scapegoat set.

Given the maximal consistent scapegoat set Σ'', we turn to the model M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$: Consider an enumeration $t_0, t_1, \ldots$ of all the variable-free terms in $\mathcal{L}'$, and let $\simeq$ be the relation on the set $\{0, 1, \ldots\}$ of natural numbers such that $\mathsf{i} \simeq \mathsf{j}$ iff $\Sigma'' \vdash t_i = t_j$. Let $[\mathsf{n}] = \{\mathsf{z} \in \mathbb{N} \mid \mathsf{z} \simeq \mathsf{n}\}$.

CnsM' $\mathsf{U} = \{[\mathsf{n}] \mid \mathsf{n} \in \mathbb{N}\}$. If t_z in the enumeration of variable-free terms $t_0, t_1, \ldots$ is a constant, then $\mathsf{M}'[t_z] = [\mathsf{z}]$. If t_z is $h^n t_a \ldots t_b$ for function symbol h^n and variable-free terms $t_a \ldots t_b$, then $\langle\langle [\mathsf{a}] \ldots [\mathsf{b}]\rangle, [\mathsf{z}]\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathcal{S}$, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. For a relation symbol $\mathcal{R}^n$, where $t_a \ldots t_b$ are

n members of the enumeration of variable-free terms, $\langle[\mathsf{a}]\ldots[\mathsf{b}]\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.

This modifies the relatively simple version where $\mathsf{U} = \{0, 1, \ldots\}$. And for an enumeration of variable-free terms, if t_z is a constant, $\mathsf{M}'[t_z] = \mathsf{z}$. If $t_z = h^n t_a \ldots t_b$ for some relation symbol h^n and n variable-free terms $t_a \ldots t_b$, $\langle\langle \mathsf{a}\ldots\mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathcal{S}$, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. And for a relation symbol $\mathcal{R}^n$, $\langle \mathsf{a}\ldots\mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.

T10.9 If Σ' is consistent, then for any sentence $\mathcal{P}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

T10.10 If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

CnsM Let M be like M' but without assignments to constants not in $\mathcal{L}$.

Then we have had to connect results for Σ' in $\mathcal{L}'$ to Σ in $\mathcal{L}$, and M' for $\mathcal{L}'$ to M for $\mathcal{L}$.

T10.12 If Σ is consistent, then Σ' is consistent.

T10.15 If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

This is supported by the matched pair of theorems, T10.13 on which, if d is a variable assignment, then for any term t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$, and T10.14 on which, if d is a variable assignment, then for any formula $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

And we are in a position for the key result.

T10.16. If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ unconstrained) ($\star\star$)

From T10.12, T10.10, and T10.15.

This puts us in a position to recover the $\mathring{\text{c}}$ompleteness result. Recall that our argument runs through $\mathcal{P}^u$ the universal closure of $\mathcal{P}$.

T10.11. If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Quantificational $\mathring{C}$ompleteness.*

Suppose $\Gamma \vDash \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment that $\Gamma \vdash {\sim}{\sim}\mathcal{P}^u$; by T3.10, $\vdash {\sim}{\sim}\mathcal{P}^u \rightarrow \mathcal{P}^u$; so by MP, $\Gamma \vdash \mathcal{P}^u$; so by repeated applications of A4 and MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash {\sim}{\sim}\mathcal{P}^u$. Given this, since ${\sim}{\sim}\mathcal{P}^u$ is a sentence, by T10.6, $\Gamma \cup \{{\sim}\mathcal{P}^u\} = \Sigma$ is consistent; so by T10.16, there is a model M such that $\mathsf{M}[\Gamma \cup \{{\sim}\mathcal{P}^u\}] = \mathsf{T}$. So $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[{\sim}\mathcal{P}^u] = \mathsf{T}$; from the latter, by T8.8, $\mathsf{M}[\mathcal{P}^u] \neq \mathsf{T}$; so by repeated applications of T7.6, $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \nvDash \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$.

The sentential version had parallels to Con, Max, CnsΣ'', and CnsM$'$ along with theorems T10.6_s–T10.11_s. (The distinction between $(\star)$ and $(\star\star)$ is a distinction without a difference in the sentential case.) The basic quantificational version requires its own versions of Con, Max, and CnsΣ'', along with Scgt, T10.6–T10.11 and the simple version of CnsM$'$. For the full version, we have had to appeal also to T10.12 and T10.15 (and so T10.16), and use the relatively complex specification for CnsM$'$.

The argument works for all the same reasons as before: Consistent sets have models. If there is no derivation of $\mathcal{P}$ from Γ, then $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent; and if $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent, then it has a model—so that $\Gamma \not\models \mathcal{P}$. Put the other way around, if $\Gamma \models \mathcal{P}$, then there is a derivation of $\mathcal{P}$ from Γ. We get the key point, that consistent sets have models, by finding a relation between consistent, and maximal consistent scapegoat sets. If a set is a maximal consistent scapegoat set, then it contains enough information to specify a model for the whole. The model for the big set then guarantees the existence of a model M for the original Γ.

E10.26. Augment $A^\star$ from E9.5 (and E10.18) to an $A^\#$ which has A1–A4, MP, and $\exists$R as before with,

A5 $t = t$

A6 $r = s \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$ where s is free for the replaced instance of r in $\mathcal{P}$

Now without assumptions that the language has no symbol for equality, and has infinitely many constants not in Γ or $\mathcal{P}$, provide a complete demonstration that $A^\#$ is $\mathring{c}$omplete. Because axioms are treated together, you still have DT from E9.8. You may appeal to any results from the text or E10.18 whose demonstration remains unchanged, but should recreate parts whose demonstration is not the same (but you may simply assume #-versions of T10.2 and any required Chapter 8 theorems).

E10.27. By T10.3 and T10.11, *AD* is $\mathring{s}$ound and $\mathring{c}$omplete: $\Gamma \models \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$. Similarly $A^\#$ is $\mathring{s}$ound and $\mathring{c}$omplete just in case $\Gamma \models \mathcal{P}$ iff $\Gamma \vdash_{A^\#} \mathcal{P}$. Given this, $\Gamma \vdash_{A^\#} \mathcal{P}$ iff $\Gamma \models \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$—and so $\Gamma \vdash_{A^\#} \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$ (iff $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{ND+} \mathcal{P}$). Thus we obtain a means for demonstrating equivalence of derivation systems in addition to the "direct" approach of Chapter 9. By E10.26, if $\Gamma \models \mathcal{P}$ then $\Gamma \vdash_{A^\#} \mathcal{P}$. We lack the biconditional only because the demonstration of $\mathring{s}$oundness from E10.3 is for $A^\star$ not $A^\#$. Extend your argument from E10.3 to provide a complete demonstration that $A^\#$ is equivalent to *AD*. You may take as given that for any interpretation I, variable assignment d, formula $\mathcal{P}$, and terms r and s, if $\mathsf{I_d}[r] = \mathsf{I_d}[s]$ and s is free for the replaced instance of r in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{I_d}[\mathcal{P}^{r}/\!\!/_{s}]$.

E10.28. We have shown from T10.4 that if a set of formulas has a model, then it is consistent; and now from T10.16 that if a set of formulas is consistent, then it has a model—and one whose U is a set of sets of natural numbers. Notice that any such U is *countable* insofar as its members can be put into correspondence with the natural numbers (since the sets are disjoint, we might order them by their least elements). But by reasoning related to that in the more on countability reference (Chapter 2, page 48) the real numbers are uncountable.[6] How might this be a problem for the logic of real numbers? Hint: Think about the consequences sentences in an arbitrary Γ may have about the number of elements in U. (This exercise anticipates the Löwenheim-Skolem theorems as discussed at the end of section 11.4.2.)

E10.29. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The soundness of a derivation system, and its demonstration by mathematical induction.

b. The completeness of a derivation system, and the basic strategy for its demonstration.

c. Maximality and consistency, and the reasons for them.

d. Scapegoat sets, and the reasons for them.

[6] A real number is the *limit* of a decimal representation. But the limit of .5000 . . . is the same as that of .4999 . . . so that these are representations of the *same* real number. Explicitly the more on countability reference shows that there are uncountably many decimal representations. But the argument converts to demonstration that the real numbers themselves are uncountable if we exclude duplicate representations (say ones ending in an infinite string of 9s).

Theorems of Chapter 10

T10.1 For any interpretation I, variable assignment d, with terms t and r, if $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t] = \mathsf{I_d}[t^x_r]$.

T10.2 For any interpretation I, variable assignment d, term r, and formula $\mathcal{Q}$, if $\mathsf{I_d}[r] = \mathsf{o}$, and r is free for x in $\mathcal{Q}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

T10.3 If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. *S̊oundness.*

T10.4 If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ (a *model* for Γ), then Γ is consistent.

T10.5 If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{A}] \neq \mathsf{T}$, then $\Gamma \nvdash \mathcal{A}$.

T10.6$_s$ For any set of formulas Δ and sentence $\mathcal{P}$, if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

T10.6 For any set of formulas Δ and sentence $\mathcal{P}$, if $\Delta \nvdash \sim\mathcal{P}$, then $\Delta \cup \{\mathcal{P}\}$ is consistent.

T10.7$_s$ There is an enumeration $\mathcal{Q}_1\mathcal{Q}_2, \ldots$ of all sentences in $\mathcal{L}_s$.

T10.7 There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ of all the sentences, terms, and the like in $\mathcal{L}'$.

T10.8$_s$ If Σ' is consistent, then Σ'' is maximal and consistent.

T10.8 If Σ' is consistent, then Σ'' is a maximal consistent scapegoat set.

T10.9$_s$ If Σ' is consistent, then for any sentence $\mathcal{P}$ of $\mathcal{L}_s$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

T10.9 If Σ' is consistent, then for any sentence $\mathcal{P}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{P}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{P}$.

T10.10$_s$ If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($*$)

T10.10 If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

T10.11$_s$ If $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Sentential C̊ompleteness.*

T10.11$_r$ If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Quantificational C̊ompleteness.* ($\mathcal{L}'$ restricted)

T10.11 If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. *Quantificational C̊ompleteness.*

T10.12 If Σ is consistent, then Σ' is consistent.

T10.13 For any variable assignment d, and for any term t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M}'_\mathsf{d}[t]$.

T10.14 For any variable assignment d, and for any formula $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M}'_\mathsf{d}[\mathcal{P}] = \mathsf{S}$.

T10.15 If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

T10.16$_p$ If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ without equality)

T10.16 If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ unconstrained) ($\star\star$)

Chapter 11

More Main Results

In this chapter, we take up some results which deepen our understanding of the power and limits of logic. The first short sections restrict discussion to sentential forms, for results about *expressive completeness* (section 11.1), *unique readability* (section 11.2), and *independence* (section 11.3). The last more extended section (section 11.4) develops basic results from *model theory*, concluding with some c̃ompleteness results that serve as a background and counterpoint to Part IV. These sections are independent of one another and may be taken in any order.

11.1 Expressive Completeness

In Chapter 5 we introduced the idea of a truth functional operator, where the truth value of the whole is a function of the truth values of the parts. We exhibited operators as truth functional by tables. Thus, if some ordinary expression $\mathscr{P}$ with components $\mathscr{A}$ and $\mathscr{B}$ has table,

(A)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{P}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***F***

then it is truth functional. And we translate by an equivalent formal expression: in this case $\mathscr{A} \wedge \mathscr{B}$ does fine. Of course, not every such table, or truth function, is directly represented by one of our operators. Thus, suppose some expression $\mathscr{P}$ has the table,

(B)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{P}$
T	T	***F***
T	F	***T***
F	T	***T***
F	F	***F***

None of our operators is equivalent to this. But it takes only a little ingenuity to see that, say, $\sim(\mathscr{A} \leftrightarrow \mathscr{B})$ has the same table, and so results in a good translation. It turns out that for any table a truth functional operator may have, there is always some way to generate that table by means of our formal operators—and, in fact, by means of just the operators $\sim$ and $\rightarrow$, or just the operators $\sim$ and $\wedge$, or just the operators $\sim$ and $\vee$. It is also possible to express any truth function by means of just the operator $\uparrow$ (see Chapter 7, page 320). In this section, we prove these results. First, a result weaker than the ones announced.

T11.1. It is possible to represent any truth function by means of an expression with just the operators $\sim$, $\wedge$, and $\vee$.

The proof is by construction. Given an arbitrary truth function, we provide a recipe for constructing an expression with the same table.

Suppose we are given an arbitrary truth function, in this case with four basic sentences as on the left.

(C)

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{P}$	
1	T	T	T	T	***F***	$\mathcal{C}_1 = \mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4$
2	T	T	T	F	***F***	$\mathcal{C}_2 = \mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \sim\mathcal{S}_4$
3	T	T	F	T	***T***	$\mathcal{C}_3 = \mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4$
4	T	T	F	F	***F***	$\mathcal{C}_4 = \mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4$
5	T	F	T	T	***T***	$\mathcal{C}_5 = \mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4$
6	T	F	T	F	***F***	$\mathcal{C}_6 = \mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \sim\mathcal{S}_4$
7	T	F	F	T	***F***	$\mathcal{C}_7 = \mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4$
8	T	F	F	F	***F***	$\mathcal{C}_8 = \mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4$
9	F	T	T	T	***F***	$\mathcal{C}_9 = \sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4$
10	F	T	T	F	***F***	$\mathcal{C}_{10} = \sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \sim\mathcal{S}_4$
11	F	T	F	T	***F***	$\mathcal{C}_{11} = \sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4$
12	F	T	F	F	***T***	$\mathcal{C}_{12} = \sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4$
13	F	F	T	T	***T***	$\mathcal{C}_{13} = \sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4$
14	F	F	T	F	***F***	$\mathcal{C}_{14} = \sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \sim\mathcal{S}_4$
15	F	F	F	T	***F***	$\mathcal{C}_{15} = \sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4$
16	F	F	F	F	***F***	$\mathcal{C}_{16} = \sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4$

For a sentence $\mathcal{P}$ with basic sentences $\mathcal{S}_1 \ldots \mathcal{S}_n$, corresponding to each row j there is a *characteristic* sentence $\mathcal{C}_j = \mathcal{S}'_{j1} \wedge \ldots \wedge \mathcal{S}'_{jn}$ (with appropriate parentheses). If the interpretation of $\mathcal{S}_i$ on row j is T, then $\mathcal{S}'_{ji} = \mathcal{S}_i$; if the interpretation of $\mathcal{S}_i$ on row j is F, then $\mathcal{S}'_{ji} = \sim\mathcal{S}_i$. Then the characteristic sentence $\mathcal{C}_j$ is the conjunction of each $\mathcal{S}'_{ji}$. The characteristic sentences are true *only* on their corresponding rows. Thus $\mathcal{C}_4$ above is true only when $\mathsf{I}[\mathcal{S}_1] = \mathsf{T}$, $\mathsf{I}[\mathcal{S}_2] = \mathsf{T}$, $\mathsf{I}[\mathcal{S}_3] = \mathsf{F}$, and $\mathsf{I}[\mathcal{S}_4] = \mathsf{F}$.

Now given the characteristic sentences, where $\mathcal{P}$ is T on rows $a, b, \ldots, d$, $(\mathcal{S}_1 \wedge \sim\mathcal{S}_1) \vee (\mathcal{C}_a \vee \mathcal{C}_b \vee \ldots \vee \mathcal{C}_d)$ has the same table as $\mathcal{P}$. The first disjunct guarantees that the specification is well-defined and has the right result in the case where $\mathcal{P}$ is F on every row. Then the disjunction of $\mathcal{C}_a \ldots \mathcal{C}_d$ goes true on just the rows where $\mathcal{P}$ is true. Thus, for example, $(\mathcal{S}_1 \wedge \sim\mathcal{S}_1) \vee (\mathcal{C}_3 \vee \mathcal{C}_5 \vee \mathcal{C}_{12} \vee \mathcal{C}_{13})$, that is,

$$(\mathcal{S}_1 \wedge \sim\mathcal{S}_1) \vee [(\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4) \vee (\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4) \vee (\sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4) \vee (\sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4)]$$

has the same table as $\mathscr{P}$ above. Inserting parentheses, the resultant table is,

(D)

	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$(\mathscr{S}_1 \wedge$	$\sim\mathscr{S}_1)$	$\vee$	$[(\mathscr{C}_3$	$\vee$	$\mathscr{C}_5)$	$\vee$	$(\mathscr{C}_{12}$	$\vee$	$\mathscr{C}_{13})]$	$\mathscr{P}$
1	T	T	T	T	F	F	***F***	F	F	F	F	F	F	F	***F***
2	T	T	T	F	F	F	***F***	F	F	F	F	F	F	F	***F***
3	T	T	F	T	F	F	***T***	T	T	F	T	F	F	F	***T***
4	T	T	F	F	F	F	***F***	F	F	F	F	F	F	F	***F***
5	T	F	T	T	F	F	***T***	F	T	T	T	F	F	F	***T***
6	T	F	T	F	F	F	***F***	F	F	F	F	F	F	F	***F***
7	T	F	F	T	F	F	***F***	F	F	F	F	F	F	F	***F***
8	T	F	F	F	F	F	***F***	F	F	F	F	F	F	F	***F***
9	F	T	T	T	F	T	***F***	F	F	F	F	F	F	F	***F***
10	F	T	T	F	F	T	***F***	F	F	F	F	F	F	F	***F***
11	F	T	F	T	F	T	***F***	F	F	F	F	F	F	F	***F***
12	F	T	F	F	F	T	***T***	F	F	F	T	T	T	F	***T***
13	F	F	T	T	F	T	***T***	F	F	F	T	F	T	T	***T***
14	F	F	T	F	F	T	***F***	F	F	F	F	F	F	F	***F***
15	F	F	F	T	F	T	***F***	F	F	F	F	F	F	F	***F***
16	F	F	F	F	F	T	***F***	F	F	F	F	F	F	F	***F***

And we have constructed an expression with the same table as $\mathscr{P}$. And similarly for any truth function with which we are confronted. So given any truth function, there is a formal expression with the same table.

In a by now familiar pattern, the expressions produced by this method are not particularly elegant or efficient. Thus for the table,

(E)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{P}$
T	T	***T***
T	F	***F***
F	T	***T***
F	F	***T***

by our method we get the expression $(\mathscr{A} \wedge \sim\mathscr{A}) \vee [(\mathscr{A} \wedge \mathscr{B}) \vee (\sim\mathscr{A} \wedge \mathscr{B}) \vee (\sim\mathscr{A} \wedge \sim\mathscr{B})]$. It has the right table. But, of course, $\mathscr{A} \to \mathscr{B}$ is much simpler! The point is not that the resultant expressions are elegant or efficient, but that for any truth function, there *exists* a formal expression that works the same way.

We have shown that we can represent any truth function by an expression with operators $\sim$, $\wedge$, and $\vee$. But any such expression is an abbreviation of one whose only operators are $\sim$ and $\to$. So we can represent any truth function by an expression with just operators $\sim$ and $\to$. And we can argue for other cases. Thus, for example,

T11.2. It is possible to represent any truth function by means of an expression with just the operators $\sim$ and $\to$, with just the operators $\sim$ and $\wedge$, and with just the operators $\sim$ and $\vee$.

The first is immediate from T11.1 and abbreviation. The last is left as an exercise. For the other, reasoning is straightforward: Given T11.1, if we can show that any $\mathscr{P}$ whose operators are $\sim$, $\wedge$, and $\vee$ corresponds to a $\mathscr{P}^*$ whose operators are just $\sim$ and $\wedge$ such that $\mathscr{P}$ and $\mathscr{P}^*$ have the same table (such that $\mathsf{I}[\mathscr{P}] = \mathsf{I}[\mathscr{P}^*]$ for any

I), we will have shown that any truth function can be represented by an expression with just $\sim$ and $\wedge$. To see that this is so, where $\mathcal{P}$ is an atomic $\mathcal{S}$, set $\mathcal{P}^* = \mathcal{S}$; where $\mathcal{P}$ is $\sim\mathcal{A}$, set $\mathcal{P}^* = \sim\mathcal{A}^*$; where $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$, set $\mathcal{P}^* = \mathcal{A}^* \wedge \mathcal{B}^*$; and where $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$, set $\mathcal{P}^* = \sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)$. Suppose the only operators in $\mathcal{P}$ are $\sim$, $\wedge$, and $\vee$, and consider an arbitrary interpretation I.

Basis: Where $\mathcal{P}$ is a sentence letter $\mathcal{S}$, then $\mathcal{P}^*$ is $\mathcal{S}$. So $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{S}] = \mathsf{I}[\mathcal{P}^*]$.

Assp: For any i, $0 \le i < k$, if $\mathcal{P}$ has i operator symbols, then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

Show: If $\mathcal{P}$ has k operator symbols, then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \wedge \mathcal{B}$, or $\mathcal{A} \vee \mathcal{B}$ where $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$; then $\mathcal{P}^*$ is $\sim\mathcal{A}^*$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{A}] \neq \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] \neq \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\sim\mathcal{A}^*] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

($\wedge$) Suppose $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$; then $\mathcal{P}^*$ is $\mathcal{A}^* \wedge \mathcal{B}^*$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{A} \wedge \mathcal{B}] = \mathsf{T}$; by ST′($\wedge$), iff $\mathsf{I}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}^*] = \mathsf{T}$; by ST′($\wedge$), iff $\mathsf{I}[\mathcal{A}^* \wedge \mathcal{B}^*] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

($\vee$) Suppose $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$; then $\mathcal{P}^*$ is $\sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{A} \vee \mathcal{B}] = \mathsf{T}$; by ST′($\vee$), iff $\mathsf{I}[\mathcal{A}] = \mathsf{T}$ or $\mathsf{I}[\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] = \mathsf{T}$ or $\mathsf{I}[\mathcal{B}^*] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\sim\mathcal{A}^*] \neq \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{B}^*] \neq \mathsf{T}$; by ST′($\wedge$), iff $\mathsf{I}[\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*] \neq \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

If $\mathcal{P}$ has k operator symbols then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

Indct: For any $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

So if the operators in $\mathcal{P}$ are $\sim$, $\wedge$, and $\vee$, there is a $\mathcal{P}^*$ with just operators $\sim$ and $\wedge$ that has the same table. Since we can represent any truth function by an expression whose only operators are $\sim$, $\wedge$, and $\vee$, and we can represent any such $\mathcal{P}$ by a $\mathcal{P}^*$ whose only operators are $\sim$ and $\wedge$, we can represent any truth function by an expression with just operators $\sim$ and $\wedge$. Perhaps this result was obvious as soon as we saw that $\sim(\sim\mathcal{A} \wedge \sim\mathcal{B})$ has the same table as $\mathcal{A} \vee \mathcal{B}$. And, by similar reasoning, we can represent any truth function by expressions whose only operators are $\sim$ and $\vee$, and by expressions whose only operator is $\uparrow$. This is left for homework.

In E8.14, we showed that if the operators in $\mathcal{P}$ are limited to $\rightarrow$, $\wedge$, $\vee$, and $\leftrightarrow$ then when the interpretation of every atomic is T, the interpretation of $\mathcal{P}$ is T. Perhaps this is obvious insofar as tables always remain T in the top row. It follows that not every truth function can be represented by expressions whose only operators are $\rightarrow$, $\wedge$, $\vee$, and $\leftrightarrow$; for there is no way to represent a function that is F on the top row. Though it is more difficult to establish, we showed in E8.23 that any expression whose only operators are $\sim$ and $\leftrightarrow$ (with at least four rows in its truth table) has an even number of Ts and Fs under its main operator. It follows that not every truth function can be represented by expressions whose only operators are $\sim$ and $\leftrightarrow$.

E11.1. Use the method of this section to find expressions whose operators are $\sim$, $\wedge$, $\vee$ with tables corresponding to $\mathscr{P}_1$, $\mathscr{P}_2$, and $\mathscr{P}_3$. Then show on a table that your expression for $\mathscr{P}_1$ in fact has the same truth function as $\mathscr{P}_1$.

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	$\mathscr{P}_1$	$\mathscr{P}_2$	$\mathscr{P}_3$
T	T	T	T	F	F
T	T	F	T	T	F
T	F	T	F	T	T
T	F	F	F	F	F
F	T	T	F	F	T
F	T	F	F	T	F
F	F	T	F	F	T
F	F	F	F	F	T

E11.2. (i) Show that we can represent any truth function by expressions whose only operators are $\sim$ and $\vee$ and so complete the demonstration of T11.2. (ii) Show that we can represent any truth function by expressions whose only operator is the up arrow $\uparrow$ (see Chapter 7 page 320 and E7.6). Hint: Given what we have shown above, it is enough to show that you can represent expressions whose only operators are $\sim$ and $\rightarrow$. (iii) Without going through the full-blown argument, find expressions whose tables match those of $\sim$ and $\vee$ to show that we can do the same with just the down arrow operator $\downarrow$ mentioned in Chapter 5 page 188, note 9.

E11.3. Show that is not possible to represent arbitrary truth functions by expressions whose only operators are binary $*$ and $\circ$ with tables TFFT and FTTF respectively. Hint: Think about E8.23.

11.2 Unique Readability

Unique readability is one of those results whose conclusion may seem too obvious to merit argument. Still, it is a result upon which we depend at every stage—and its demonstration is not so simple as you might think: We show that every formula of $\mathcal{L}_s$ is parsed uniquely into its immediate subformulas. Things are set up so that this is so. But suppose that instead of FR($\rightarrow$) and ST($\rightarrow$) we had,

FR($\rightarrow$)* If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathscr{P} \rightarrow \mathscr{Q}$ is a *formula*

ST($\rightarrow$)* $\mathsf{I}[\mathscr{P} \rightarrow \mathscr{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] = \mathsf{F}$ or $\mathsf{I}[\mathscr{Q}] = \mathsf{T}$

without parentheses. For some atomics A, B, and C, suppose $\mathsf{I}[A] = \mathsf{I}[B] = \mathsf{I}[C] = \mathsf{F}$. Then $A \rightarrow B$ is a formula, and $\mathsf{I}[A \rightarrow B] = \mathsf{T}$; so $A \rightarrow B \rightarrow C$ is a formula, and $\mathsf{I}[A \rightarrow B \rightarrow C] = \mathsf{F}$. But again $B \rightarrow C$ is a formula, and $\mathsf{I}[B \rightarrow C] = \mathsf{T}$; so $A \rightarrow B \rightarrow C$ is a formula; and $\mathsf{I}[A \rightarrow B \rightarrow C] = \mathsf{T}$. So something is wrong: the one expression $A \to B \to C$, reconceived with different immediate subformulas, comes out false one way but true the other. A recursive definition like ST assigns unique values to expressions only if there are unique parts to which the definition applies. Thus, if our definitions are to yield determinate results, it is important that formulas be uniquely parsed into their parts.

So far, we have simply presupposed that this is so. Now we prove it. For the sentential case, according to unique readability, for any formula $\mathcal{P}$ of $\mathcal{L}_s$ exactly one of the following holds:

(s) $\mathcal{P}$ is a sentence letter.

($\sim$) There is a unique formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$.

($\to$) There are unique formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \to \mathcal{B})$.

Given a couple preliminary theorems, the result is straightforward. First, ignoring uniqueness,

T11.3. For any formula $\mathcal{P}$ of $\mathcal{L}_s$, exactly one of the following holds: (i) $\mathcal{P}$ is a sentence letter; (ii) there is a formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$; (iii) there are formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \to \mathcal{B})$.

First, by the closure clause to definition FR, any formula $\mathcal{P}$ results by FR(s), FR($\sim$), or FR($\to$); if FR(s), $\mathcal{P}$ is is a sentence letter; if FR($\sim$), $\mathcal{P}$ is $\sim\mathcal{A}$ for formula $\mathcal{A}$, and if FR($\to$), $\mathcal{P}$ is $(\mathcal{A} \to \mathcal{B})$ for formulas $\mathcal{A}$ and $\mathcal{B}$; in any case at least one of (i), (ii), or (iii). And, just as easy, at most one of (i), (ii), or (iii): If $\mathcal{P}$ is a sentence letter it begins with a sentence letter; if $\mathcal{P}$ is $\sim\mathcal{A}$ it begins with '$\sim$'; and if $\mathcal{P}$ is $(\mathcal{A} \to \mathcal{B})$ it begins with '('. (i) Suppose $\mathcal{P}$ is a sentence letter; then it does not begin with '$\sim$' or '('; so not (ii) and not (iii). Suppose $\mathcal{P}$ is $\sim\mathcal{A}$; then it does not begin with a sentence letter or '('; so not (i) or (iii). Suppose $\mathcal{P}$ is $(\mathcal{A} \to \mathcal{B})$; then it does not begin with a sentence letter or '$\sim$' so not (i) or (ii).

Say $\mathcal{A}$ is an *initial segment* of an expression $\mathcal{P}$ just in case there is some (possibly empty) $\mathcal{B}$ such that $\mathcal{P} = \mathcal{A}\mathcal{B}$—just in case $\mathcal{P}$ is the concatenation of $\mathcal{A}$ and $\mathcal{B}$. If $\mathcal{B}$ is a non-empty sequence so that $\mathcal{A}$ is not all of $\mathcal{P}$, then $\mathcal{A}$ is a *proper* initial segment of $\mathcal{P}$. So AB is a proper initial segment of ABC. To make progress on the uniqueness conditions, we show the following:

T11.4. If $\mathcal{P}$ is a formula of $\mathcal{L}_s$, then no proper initial segment of $\mathcal{P}$ is a formula.

Suppose $\mathcal{P}$ is a formula.

Basis: If $\mathcal{P}$ is atomic, then $\mathcal{P} = \mathcal{A}\mathcal{B}$ only if (i) $\mathcal{P} = \mathcal{A}$ and $\mathcal{B}$ is empty, or (ii) $\mathcal{P} = \mathcal{B}$ and $\mathcal{A}$ is empty. In the first case, $\mathcal{A}$ is not a proper initial segment. In the second case $\mathcal{A}$ is an (empty) proper initial segment; but from T11.3 no empty segment is a formula; so $\mathcal{A}$ is not a formula. In either case then, no proper initial segment of $\mathcal{P}$ is a formula.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then no proper initial segment of $\mathcal{P}$ is a formula.

Show: If $\mathcal{P}$ has k operator symbols, then no proper initial segment of $\mathcal{P}$ is a formula. If $\mathcal{P}$ has k operator symbols then it is $\sim\mathcal{A}$ or $(\mathcal{A} \rightarrow \mathcal{B})$ for formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$ for some formula $\mathcal{A}$. Suppose some proper initial segment of $\mathcal{P}$ is a formula; then for some formula $\mathcal{B}$ and non-empty $\mathcal{C}$, $\mathcal{P} = \sim\mathcal{A} = \mathcal{B}\mathcal{C}$. $\mathcal{B}$ is either empty or starts with '$\sim$'; so with T11.3, $\mathcal{B}$ is $\sim\mathcal{D}$ for some formula $\mathcal{D}$. So $\mathcal{P} = \sim\mathcal{A} = \sim\mathcal{D}\mathcal{C}$; dropping the initial tilde, $\mathcal{A} = \mathcal{D}\mathcal{C}$; so $\mathcal{D}$ is a proper initial segment of $\mathcal{A}$; so by assumption, $\mathcal{D}$ is not a formula. Reject the assumption: no proper initial segment of $\mathcal{P}$ is a formula.

($\rightarrow$) $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$. Suppose some proper initial segment of $\mathcal{P}$ is a formula; then for some formula $\mathcal{C}$ and non-empty $\mathcal{D}$, $\mathcal{P} = (\mathcal{A} \rightarrow \mathcal{B}) = \mathcal{C}\mathcal{D}$. $\mathcal{C}$ is either empty or starts with '('; so with T11.3, $\mathcal{C}$ is $(\mathcal{E} \rightarrow \mathcal{F})$ for some formulas $\mathcal{E}$ and $\mathcal{F}$; so $(\mathcal{A} \rightarrow \mathcal{B}) = (\mathcal{E} \rightarrow \mathcal{F})\mathcal{D}$. Observe that each of $\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}$ must have fewer than the number of operator symbols in $\mathcal{P} = (\mathcal{A} \rightarrow \mathcal{B}) = (\mathcal{E} \rightarrow \mathcal{F})\mathcal{D}$, and so $< k$ operator symbols. From $(\mathcal{A} \rightarrow \mathcal{B}) = (\mathcal{E} \rightarrow \mathcal{F})\mathcal{D}$, dropping the initial parenthesis, we get $\mathcal{A} \rightarrow \mathcal{B}) = \mathcal{E} \rightarrow \mathcal{F})\mathcal{D}$; given their position at the start, either $\mathcal{A}$ is identical to $\mathcal{E}$, or $\mathcal{A}$ overlaps $\mathcal{E}$, or $\mathcal{E}$ overlaps $\mathcal{A}$—that is, either $\mathcal{A} = \mathcal{E}$ or one is a proper initial segment of the other; suppose one is a proper initial segment of the other; then by assumption one or the other is not a formula; this is impossible. So $\mathcal{A} = \mathcal{E}$; so $\rightarrow \mathcal{B}) = \rightarrow \mathcal{F})\mathcal{D}$; so $\mathcal{B}) = \mathcal{F})\mathcal{D}$; so the last character of $\mathcal{D}$ is); so $\mathcal{B}$ overlaps at least $\mathcal{F}$); so $\mathcal{F}$ is a proper initial segment of $\mathcal{B}$; so by assumption $\mathcal{F}$ is not a formula. Reject the assumption: no proper initial segment of $\mathcal{P}$ is a formula.

If $\mathcal{P}$ has k operator symbols, then no proper initial segment of $\mathcal{P}$ is a formula.

Indct: For any formula $\mathcal{P}$, no proper initial segment of $\mathcal{P}$ is a formula.

Observe that we "add" and "subtract" from sequences so that, for example $\sim\mathcal{A} = \sim\mathcal{B}$ iff $\mathcal{A} = \mathcal{B}$. It is also worth noting the point at which parentheses matter for the ($\rightarrow$) case. At the stage where $\mathcal{B}) = \mathcal{F})\mathcal{D}$, suppose $\mathcal{D}$ is just) and there were no) between $\mathcal{F}$ and $\mathcal{D}$; then $\mathcal{B}) = \mathcal{F}\mathcal{D} = \mathcal{F})$; so $\mathcal{B} = \mathcal{F}$—and there is no contradiction. The parenthesis makes it the case that $\mathcal{F}$ must be a proper initial segment of $\mathcal{B}$, which is impossible.

And now we are ready to establish unique readability.

T11.5. For any formula $\mathcal{P}$ of $\mathcal{L}_s$, exactly one of the following holds:

(s) $\mathcal{P}$ is a sentence letter.

($\sim$) There is a unique formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$.

($\rightarrow$) There are unique formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$.

For any formula $\mathcal{P}$ of $\mathcal{L}_s$, by T11.3, exactly one of (i) $\mathcal{P}$ is a sentence letter, (ii) there is a formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$, (iii) there are formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$. We are now in a position to establish uniqueness constraints for cases (ii) and (iii) by showing that there can be just one $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$, and just one $\mathcal{A}$ and one $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$.

($\sim$) Suppose $\mathcal{P} = \sim\mathcal{A}$ and there is some formula $\mathcal{B}$ such that $\mathcal{P} = \sim\mathcal{B}$; then $\sim\mathcal{A} = \sim\mathcal{B}$; so, dropping the initial symbol, $\mathcal{A} = \mathcal{B}$. So there is a unique formula $\mathcal{A}$ such that $\mathcal{P} = \sim\mathcal{A}$.

($\rightarrow$) Suppose $\mathcal{P} = (\mathcal{A} \rightarrow \mathcal{B})$ and there are formulas $\mathcal{C}$ and $\mathcal{D}$ such that $\mathcal{P} = (\mathcal{C} \rightarrow \mathcal{D})$; then $(\mathcal{A} \rightarrow \mathcal{B}) = (\mathcal{C} \rightarrow \mathcal{D})$; so $\mathcal{A} \rightarrow \mathcal{B}) = \mathcal{C} \rightarrow \mathcal{D})$; so either $\mathcal{A} = \mathcal{C}$ or one is a proper initial segment of the other; but by T11.4, neither is a proper initial segment of the other; so $\mathcal{A} = \mathcal{C}$; so $\mathcal{B}) = \mathcal{D})$; so $\mathcal{B} = \mathcal{D}$. So there are unique formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P} = (\mathcal{A} \rightarrow \mathcal{B})$.

Thus unique readability is established.

*E11.4. Show unique readability for the terms of $\mathcal{L}_q$—that for every term t of $\mathcal{L}_q$, exactly one of the following holds:

(v) t is a variable.

(c) t is a constant.

(f) There are unique terms $s_1 \dots s_n$ and function symbol h^n such that $t = h^n s_1 \dots s_n$.

Hint: The argument is based on TR; you will want to show that no proper initial segment of a term is a term.

E11.5. Show unique readability for the formulas of $\mathcal{L}_q$—that for every formula $\mathcal{P}$ of $\mathcal{L}_q$, exactly one of the following holds:

(s) $\mathcal{P}$ is a sentence letter.

(r) There are unique terms $t_1 \dots t_n$ and relation symbol $\mathcal{R}^n$ such that $\mathcal{P} = \mathcal{R}^n t_1 \dots t_n$.

($\sim$) There is a unique formula $\mathcal{A}$ such that $\mathcal{P} = \sim\mathcal{A}$.

($\rightarrow$) There are unique formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P} = (\mathcal{A} \rightarrow \mathcal{B})$.

($\forall$) There is unique variable x and formula $\mathcal{A}$ such that $\mathcal{P} = \forall x \mathcal{A}$.

Hint: This time the argument is based on FR.

11.3 Independence

As we have seen, axiomatic systems are convenient insofar as their compact form makes reasoning about them relatively easy. In addition, axiomatic systems are attractive insofar as they exhibit a minimal *ground* or foundation for the logical systems. Given these aims, it is natural to wonder whether we could get the same results without one or more of our axioms. Say an axiom is *independent* in a derivation system just in case it cannot be derived by the other axioms and rules, and an axiom *schema* is independent just in case not all the axioms which are its instances can be derived without the schema. An axiom is independent iff it makes a difference to what can be derived: Suppose an axiom is not independent; then there is a derivation of it from other axioms and rules, and any result of the system with the axiom can be derived using the theorem in place of the axiom; so the axiom makes no difference to what can be derived. Suppose an axiom is independent, then the axiom itself is something not derived from the other axioms and rules; so the axiom makes a difference to what can be derived. In this section, we show that schemas A1, A2, and A3 of the sentential fragment of *AD* are independent of one another.

Say we want to show that A1 is independent of A2 and A3. When we showed, in Chapter 8, that the sentential part of *AD* is weakly sound, we showed that A1, A2, A3, and their consequences have a certain feature—that there is no interpretation where an axiom or consequence is false. The basic idea here is to find a sort of "interpretation" with some feature sustained by A2, A3, and their consequences, but not by all instances of A1. It follows that those instances of A1 are not among the consequences of A2 and A3, and so that A1 is independent of A2 and A3. Here is the key point: Any "interpretation" will do. In particular, consider the following tables which define a sort of numerical property for forms involving $\sim$ and $\rightarrow$:

A1($\sim$)

$\mathcal{P}$	$\sim\mathcal{P}$
0	1
1	0
2	0

A1($\rightarrow$)

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \rightarrow \mathcal{Q}$
0	0	0
0	1	2
0	2	2
1	0	0
1	1	0
1	2	2
2	0	0
2	1	0
2	2	0

Do not worry about what these tables "say"; it is sufficient that, given a numerical interpretation of the parts, we can always calculate the numerical value Num of the whole. Thus, for example,

$$\begin{array}{lll} & A^{(0)} \quad\quad B^{(2)} & \\ \text{(F)} & {\sim}A^{(1)} & \text{By A1(}{\sim}\text{) row 1} \\ & {\sim}A \to B^{(2)} & \text{By A1(}{\to}\text{) row 6} \end{array}$$

if $\mathsf{Num}[A] = \mathsf{0}$ and $\mathsf{Num}[B] = \mathsf{2}$, then $\mathsf{Num}[{\sim}A \to B] = \mathsf{2}$. The calculation is straightforward based on the tables. And similarly for sentential forms of arbitrary complexity. Say a formula is *select* iff it takes the value $\mathsf{0}$ on every numerical interpretation of its parts. (Compare the notion of sentential validity on which a formula is valid iff it is T on every interpretation of its parts.) Again, do not worry about what the tables mean. They are constructed for the special purpose of demonstrating independence: We show that every consequence of A2 and A3 is select but that the A1 instance $A \to (B \to A)$ is not. It follows that not every instance of A1 is a consequence of A2 and A3 and so that A1 is independent of A2 and A3.

To see that instances of A2 and A3 are select, but $A \to (B \to A)$ is not, all we have to do is complete the tables. For $A \to (B \to A)$ and A3,

(G)

A	B	$A \to$	$(B \to A)$
0	0	***0***	0
0	1	***0***	0
0	2	***0***	0
1	0	***2***	2
1	1	***0***	0
1	2	***0***	0
2	0	***0***	2
2	1	***0***	2
2	2	***0***	0

$\mathscr{A}$	$\mathscr{B}$	$({\sim}\mathscr{B}$	$\to$	${\sim}\mathscr{A})$	$\to$	$[({\sim}\mathscr{B}$	$\to \mathscr{A})$	$\to \mathscr{B}]$
0	0	1	0	1	***0***	1	0	0
0	1	0	2	1	***0***	0	0	2
0	2	0	2	1	***0***	0	0	2
1	0	1	0	0	***0***	1	0	0
1	1	0	0	0	***0***	0	2	0
1	2	0	0	0	***0***	0	2	0
2	0	1	0	0	***0***	1	2	0
2	1	0	0	0	***0***	0	2	0
2	2	0	0	0	***0***	0	2	0

Since it evaluates to $\mathsf{2}$ in the fourth row, the A1 instance $A \to (B \to A)$ is not select.[1] From the table, instances of A3 are select. To see that instances of A2 are select, again, it is enoungh to complete the table. For this, see table (H) on page 502. So any instance of A2 or A3 is select. But now we are in a position to show,

T11.6. In the sentential fragment of AD, axiom A1 is independent of A2 and A3.

Consider any derivation $\langle \mathcal{Q}_1, \mathcal{Q}_2 \ldots \mathcal{Q}_n \rangle$ where there are no premises, and the only axioms are instances of A2 and A3. By induction on line number, for any i, $\mathcal{Q}_i$ is select.

[1]Not every instance of A1 fails to be select. Thus for example, if $\mathscr{C}$ is select then $\mathscr{D} \to \mathscr{C}$ and so $\mathscr{C} \to (\mathscr{D} \to \mathscr{C})$ are select—the latter an instance of A1.

Basis: $\mathcal{Q}_1$ is an instance of A2 or A3, and as we have just seen, instances of A2 and A3 are select. So $\mathcal{Q}_1$ is select.

Assp: For any i, $0 \leq i < k$, $\mathcal{Q}_i$ is select.

Show: $\mathcal{Q}_k$ is select.

$\mathcal{Q}_k$ is an instance of A2 or A3 or arises from previous lines by MP. If $\mathcal{Q}_k$ is an instance of A2 or A3 then by reasoning as in the basis, $\mathcal{Q}_k$ is select. If $\mathcal{Q}_k$ arises from previous lines by MP, then the derivation has some lines,

a.	$\mathcal{B} \rightarrow \mathcal{C}$	
b.	$\mathcal{B}$	
k.	$\mathcal{C}$	a,b MP

where $a, b < k$ and $\mathcal{C}$ is $\mathcal{Q}_k$. By assumption, $\mathcal{B} \rightarrow \mathcal{C}$ and $\mathcal{B}$ are select; so they evaluate to 0 on every row. But the only case where $\mathcal{B} \rightarrow \mathcal{C}$ and $\mathcal{B}$ both evaluate to 0 is the top row of A1($\rightarrow$) where $\mathcal{C}$ evaluates to 0 as well; so if on every row $\mathsf{Num}[\mathcal{B} \rightarrow \mathcal{C}] = 0$ and $\mathsf{Num}[\mathcal{B}] = 0$, then on every row $\mathsf{Num}[\mathcal{C}] = 0$. So $\mathcal{Q}_k$ is select.

Indct: For any n, $\mathcal{Q}_n$ is select.

Since consequences of A2 and A3 are select and the A1 instance $A \rightarrow (B \rightarrow A)$ is not, that instance cannot be derived from A2 and A3; so A1 is independent of A2 and A3.

The basic strategy is like that from T10.5 where we found an interpretation with premises true and conclusion not in order to demonstrate that premises do not prove a conclusion. The difference is that the axioms are always true on interpretations with the standard tables—so only a nonstandard semantic account can separate one axiom from the others. In fact multiple tables of the sort A1($\sim$) and A1($\rightarrow$) are sufficient for the result. We pick just one option.[2] Similarly we may show,

T11.7. In the sentential fragment of *AD*, A2 is independent of A1 and A3, and A3 is independent of A1 and A2.

Homework.

Our independence results apply to the axioms of a derivation system. But independence results might also apply to axioms of a theory (like Q or PA). An important example is the demonstration that both the *continuum hypothesis* and its negation are independent of the axioms of ZFC set theory (see note 9 on page 535). Such demonstrations often require considerable creativity—and results for the continuum hypothesis were a major achievement. Still, the basic idea of such demonstrations remains the same: Independence is demonstrated by a structure on which other axioms and their consequences have some feature that the independent one lacks.

---

[2]In this case, there are 64 solutions by three-valued tables but over 500,000 three-valued table combinations. So the solutions remain a small fraction of the total.

(H)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	$(\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C}))$					
0	0	0	0	0	***0***	0	0	0
0	0	1	2	2	***0***	0	2	2
0	0	2	2	2	***0***	0	2	2
0	1	0	0	0	***0***	2	0	0
0	1	1	0	0	***0***	2	0	2
0	1	2	2	2	***0***	2	0	2
0	2	0	0	0	***0***	2	0	0
0	2	1	0	0	***0***	2	0	2
0	2	2	0	0	***0***	2	0	2
1	0	0	0	0	***0***	0	0	0
1	0	1	2	2	***0***	0	0	0
1	0	2	2	2	***0***	0	2	2
1	1	0	0	0	***0***	0	0	0
1	1	1	0	0	***0***	0	0	0
1	1	2	2	2	***0***	0	2	2
1	2	0	0	0	***0***	2	0	0
1	2	1	0	0	***0***	2	0	0
1	2	2	0	0	***0***	2	0	2
2	0	0	0	0	***0***	0	0	0
2	0	1	0	2	***0***	0	0	0
2	0	2	0	2	***0***	0	0	0
2	1	0	0	0	***0***	0	0	0
2	1	1	0	0	***0***	0	0	0
2	1	2	0	2	***0***	0	0	0
2	2	0	0	0	***0***	0	0	0
2	2	1	0	0	***0***	0	0	0
2	2	2	0	0	***0***	0	0	0

E11.6. Use the following tables to show that A2 is independent of A1 and A3. Then explain how E8.13 already shows that A3 is independent of A1 and A2 and so complete the demonstration of T11.7.

A2($\sim$)

$\mathscr{P}$	$\sim\mathscr{P}$
0	1
1	0
2	1

A2($\to$)

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \to \mathscr{Q}$
0	0	0
0	1	2
0	2	1
1	0	0
1	1	2
1	2	0
2	0	0
2	1	0
2	2	0

E11.7. Produce tables to show that the axioms of A^* from E3.5 are independent of one another. You need not demonstrate independence by induction, but should briefly explain how your tables suffice. Hint: This exercise requires considerable ingenuity (and patience). Alternatively, given requisite background, you might employ a computing device to search by "brute force" for (three-valued) tables that do the job.

*E11.8. Consider our full derivation system *AD*, and show that A1 is independent of A2–A8. You can do this by assigning 0 to every atomic equality $t_1 = t_2$, otherwise collapsing any atomic with relation symbol $\mathscr{R}$ to a single sentence letter $\mathscr{R}'$ (so any $\mathscr{P}^x_t$ is just $\mathscr{P}$), collapsing quantifiers to a single unary sentential operator, and supplementing tables A1($\sim$) and A1($\rightarrow$) with a table A1($\forall$). Find this table, and explain how the result demonstrates that A1 is independent of the other axioms.

11.4 Beginning Model Theory

Model theory investigates relations between formal expressions and models. Put this way, $\mathring{s}$oundness and $\mathring{c}$ompleteness are already results in model theory. But many interesting results of this kind are possible, so that $\mathring{s}$oundness and $\mathring{c}$ompleteness are only a beginning. Introductions to model theory (for example Manzano, *Model Theory*) typically presuppose a background including to groups, rings, and fields (such as would be part of a course in abstract algebra) and to transfinite arithmetic (such as would appear in a course on set theory). These supply examples and extend the range of results. The examples and results are beneficial given the required background, but a stumbling block without. Though I shall wave in the direction of some such results, our main discussion is restricted to more generally accessible applications and to the sets with which we are already familiar. Inevitably, notions from set theory extend what we have seen so far. However, I make every effort to explain concepts as they arise. The results remain interesting, and shall be sufficient for our purposes. After this brief introduction, we turn to some basic concepts (11.4.1), results from compactness (11.4.2), and some $\tilde{c}$ompleteness results (11.4.3).

Let Σ be a set of formulas and $\mathfrak{M}$ a class of models. This Σ and $\mathfrak{M}$ are language-relative: Σ is a set of formulas *in some language* $\mathscr{L}$, and $\mathfrak{M}$ a class of models *for that language* (mostly, though, the italicized part is left implicit). Σ is an *axiomatization* of $\mathfrak{M}$ just in case $\mathsf{M}[\Sigma] = \mathsf{T}$ iff $\mathsf{M} \in \mathfrak{M}$. So the members of $\mathfrak{M}$ are the models that make members of Σ true. $\mathfrak{M}$ is *axiomatizable* iff it is axiomatized by some Σ; $\mathfrak{M}$ is *finitely axiomatizable* iff it is axiomatized by some finite Σ. Let $\mathfrak{Md}(\Sigma)$ be the class of models axiomatized by Σ. Notice that there must be some such class. In the extreme, if Σ is empty then (vacuously) the members of Σ are true on any model and $\mathfrak{Md}(\Sigma)$ is the class of all models; if Σ is inconsistent, then by T10.4 its members are not true on any model and $\mathfrak{Md}(\Sigma)$ is the empty class.

Given these notions, we may raise questions in two directions: We might start with a set of formulas and ask about the models axiomatized by it; or we might start with some models, and ask about formulas to axiomatize them. Thus, for example, we may ask if there is a Σ such that $\mathfrak{Mod}(\Sigma)$ is,

- the class of models whose only member is N
- the class of all models with an infinite universe
- the class of all models with a finite universe

and, with vocabulary to be introduced below,

- the class of all models *isomorphic* to N
- the class of all models *elementarily equivalent* to N

The first four are among questions answered in this section. An answer to the last will have to wait for Part IV.

11.4.1 Basic Concepts

I introduce some basic concepts and prove some results about them. We begin with a short section introducing *relative* notions of soundness and completeness. Then we turn to *isomorphism* and *equivalence*, and to *submodels* and *embeddings*. A number of definitions are collected in the model theory definitions reference on pages 522–523, to which you may find it helpful to refer.

A qualification: On the usual account there is no *set* of all sets. This is a consequence of the way sets appear in an unending hierarchy such that the members of sets at any "rank" come from ranks below. The universe of an interpretation may be any set. Thus there is no set of all universes, and no set of all interpretations. Some theorists introduce *proper classes* as entities which may have sets from every rank as members. But then it is difficult to resist the suggestion that proper classes are themselves members of sets and are thus so many more members of the hierarchy of sets. We shall not enter into this controversy. For we may regard references to proper classes as an eliminable manner of speaking: By the metalanguage, we identify some feature of sets or models and talk about ones that have it. So, for example, where $\mathfrak{M}$ is the class of all models with a finite universe, $\mathsf{M} \in \mathfrak{M}$ just in case M has a finite universe. Talk of classes permits pleasingly concise specifications of that which would otherwise be more difficult (and the use of Fraktur variables reminds us of the underlying metalinguistic specification).

Relative Soundness and Completeness

We shall be able locate different concepts of soundness and completeness under the umbrella of a *relative* soundness and completeness.

As a preliminary, let $|\mathfrak{M}|$ be the set of formulas that are true on each $\mathsf{M} \in \mathfrak{M}$, so $|\mathfrak{M}| = \{\mathcal{P}|$ for every $\mathsf{M} \in \mathfrak{M}, \mathsf{M}[\mathcal{P}] = \mathsf{T}\}$. Roughly, as a class of models expands, the class of formulas true on all its members contracts. In the extreme, if $\mathfrak{M}$ is empty, then vacuously every $\mathcal{P}$ is true on each $\mathsf{M} \in \mathfrak{M}$, and $|\mathfrak{M}|$ is the set of all formulas; if $\mathfrak{M}$ is the class of all models, then the members of $|\mathfrak{M}|$ are just the tautologies, true on every M. And in general,

T11.8. If $\mathfrak{M} \subseteq \mathfrak{N}$ then $|\mathfrak{N}| \subseteq |\mathfrak{M}|$.

> Suppose $\mathfrak{M} \subseteq \mathfrak{N}$. To show that $|\mathfrak{N}| \subseteq |\mathfrak{M}|$, suppose $\mathcal{P} \in |\mathfrak{N}|$, but $\mathcal{P} \notin |\mathfrak{M}|$. From the latter, there is some $\mathsf{M} \in \mathfrak{M}$ such that $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; but since $\mathfrak{M} \subseteq \mathfrak{N}$, $\mathsf{M} \in \mathfrak{N}$; so $\mathcal{P} \notin |\mathfrak{N}|$. This is impossible; reject the assumption: if $\mathcal{P} \in |\mathfrak{N}|$, then $\mathcal{P} \in |\mathfrak{M}|$; so $|\mathfrak{N}| \subseteq |\mathfrak{M}|$.

The converse to this theorem need not obtain: it is not always the case that if $|\mathfrak{N}| \subseteq |\mathfrak{M}|$ then $\mathfrak{M} \subseteq \mathfrak{N}$. To see this, observe that adding or subtracting models from a class does not always change the set of formulas true on all its members. In particular, consider some $\mathfrak{A}$ and models $\mathsf{L}, \mathsf{M} \in \mathfrak{A}$ that make all the same formulas true (we will meet cases of this kind below). Let $\mathfrak{A}^*$ be like $\mathfrak{A}$ but without M. Then $|\mathfrak{A}^*| = |\mathfrak{A}|$ so that $|\mathfrak{A}^*| \subseteq |\mathfrak{A}|$; but $\mathfrak{A}$ has a member that $\mathfrak{A}^*$ does not, so $\mathfrak{A} \not\subseteq \mathfrak{A}^*$.

Now say Σ is *sound with respect to* $\mathfrak{M}$ just in case for any $\mathcal{P}$, if $\Sigma \vdash \mathcal{P}$ then $\mathcal{P} \in |\mathfrak{M}|$. So Σ is sound with respect to $\mathfrak{M}$ when formulas proved by Σ are true on each member of $\mathfrak{M}$. Then given $\mathring{s}$oundness, we can show that Σ is sound with respect to some models just in case they are among the models on which all the members of Σ are true.

T11.9. If derivations are $\mathring{s}$ound, then Σ is sound with respect to $\mathfrak{M}$ iff $\mathfrak{M} \subseteq \mathfrak{M}\mathfrak{d}(\Sigma)$.

> Suppose derivations are $\mathring{s}$ound.
>
> (i) Suppose Σ is sound with respect to $\mathfrak{M}$. To show that $\mathfrak{M} \subseteq \mathfrak{M}\mathfrak{d}(\Sigma)$ consider an arbitrary $\mathsf{M} \in \mathfrak{M}$; we need that $\mathsf{M} \in \mathfrak{M}\mathfrak{d}(\Sigma)$. If Σ is empty then $\mathfrak{M}\mathfrak{d}(\Sigma)$ is the class of all models and $\mathsf{M} \in \mathfrak{M}\mathfrak{d}(\Sigma)$. So suppose Σ is not empty and consider an arbitrary $\mathcal{P} \in \Sigma$; trivially $\Sigma \vdash \mathcal{P}$; and since Σ is sound with respect to $\mathfrak{M}$, $\mathcal{P} \in |\mathfrak{M}|$; from this and $\mathsf{M} \in \mathfrak{M}$, $\mathsf{M}[\mathcal{P}] = \mathsf{T}$; and since $\mathcal{P}$ is arbitrary, $\mathsf{M}[\Sigma] = \mathsf{T}$; so $\mathsf{M} \in \mathfrak{M}\mathfrak{d}(\Sigma)$.
>
> (ii) Suppose $\mathfrak{M} \subseteq \mathfrak{M}\mathfrak{d}(\Sigma)$. To show that Σ is sound with respect to $\mathfrak{M}$, consider an arbitrary $\mathcal{P}$ and suppose $\Sigma \vdash \mathcal{P}$; we need that $\mathcal{P} \in |\mathfrak{M}|$. If $\mathfrak{M}$ is empty then $|\mathfrak{M}|$ is the set of all formulas, and $\mathcal{P} \in |\mathfrak{M}|$. So suppose $\mathfrak{M}$ is not empty and consider an arbitrary $\mathsf{M} \in \mathfrak{M}$; then since $\mathfrak{M} \subseteq \mathfrak{M}\mathfrak{d}(\Sigma)$, $\mathsf{M}[\Sigma] = \mathsf{T}$; so by $\mathring{s}$oundness $\mathsf{M}[\mathcal{P}] = \mathsf{T}$; and since M is arbitrary, $\mathcal{P} \in |\mathfrak{M}|$.

So given s̊oundness from Chapter 10, Σ is sound with respect to $\mathfrak{M}$ iff $\mathfrak{M} \subseteq \mathfrak{Mod}(\Sigma)$. And with $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{Mod}(\Sigma)$, Σ is sound with respect to $\mathfrak{Mod}(\Sigma)$.

This result extends to show that s̊oundness is equivalent to a condition on relative soundness.

T11.10. Derivations are s̊ound iff every Σ is sound with respect to $\mathfrak{Mod}(\Sigma)$.

From $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{Mod}(\Sigma)$ and T11.9, if derivations are sound, then Σ is sound with respect to $\mathfrak{Mod}(\Sigma)$. For the other direction, suppose every Σ is sound with respect to $\mathfrak{Mod}(\Sigma)$; then for s̊oundness, consider arbitrary Σ and $\mathscr{P}$ and suppose $\Sigma \vdash \mathscr{P}$. If there is no model such that $\mathsf{M}[\Sigma] = \mathsf{T}$ then, trivially, $\Sigma \vDash \mathscr{P}$. So consider an arbitrary M such that $\mathsf{M}[\Sigma] = \mathsf{T}$; then $\mathsf{M} \in \mathfrak{Mod}(\Sigma)$; and since $\Sigma \vdash \mathscr{P}$ and Σ is sound with respect to $\mathfrak{Mod}(\Sigma)$, $\mathscr{P} \in |\mathfrak{Mod}(\Sigma)|$; so $\mathsf{M}[\mathscr{P}] = \mathsf{T}$; and since M is arbitrary, $\Sigma \vDash \mathscr{P}$. So derivations are s̊ound.

Say Σ is *complete with respect to* $\mathfrak{M}$ just in case for any formula $\mathscr{P}$, if $\mathscr{P} \in |\mathfrak{M}|$ then $\Sigma \vdash \mathscr{P}$. Trivially, $|\mathfrak{M}|$ is complete with respect to $\mathfrak{M}$—if $\mathscr{P} \in |\mathfrak{M}|$ then $|\mathfrak{M}| \vdash \mathscr{P}$. So specified, however, there may be no reasonable way to identify the individual formulas that are members of $|\mathfrak{M}|$. Ordinarily, we shall be interested in cases where there is some reasonable syntactic method for identifying the members of Σ (a notion to be made precise in Part IV). Examples of sets so specified are the axioms of Q and PA as developed in chapters 3 and 6.

And we have theorems like ones for relative soundness. First, given c̊ompleteness, Σ is complete with respect to some models if they include the models on which Σ is true.

T11.11. If derivations are c̊omplete and $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{M}$, then Σ is complete with respect to $\mathfrak{M}$.

Suppose derivations are c̊omplete and $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{M}$; to show that Σ is complete with respect to $\mathfrak{M}$, consider an arbitrary $\mathscr{P} \in |\mathfrak{M}|$; we need that $\Sigma \vdash \mathscr{P}$. If $\mathfrak{Mod}(\Sigma)$ is empty, then Σ is inconsistent and $\Sigma \vdash \mathscr{P}$. So suppose $\mathfrak{Mod}(\Sigma)$ is not empty and consider an arbitrary M such that $\mathsf{M}[\Sigma] = \mathsf{T}$; then $\mathsf{M} \in \mathfrak{Mod}(\Sigma)$; and since $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{M}$, $\mathsf{M} \in \mathfrak{M}$; then since $\mathscr{P} \in |\mathfrak{M}|$, $\mathsf{M}[\mathscr{P}] = \mathsf{T}$; and since this is so for for arbitrary M, $\Sigma \vDash \mathscr{P}$; so by c̊ompleteness $\Sigma \vdash \mathscr{P}$.

So given c̊ompleteness from Chapter 10, if $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{M}$ then Σ is complete with respect to $\mathfrak{M}$. And with $\mathfrak{Mod}(\Sigma) \subseteq \mathfrak{Mod}(\Sigma)$, Σ is complete with respect to $\mathfrak{Mod}(\Sigma)$.

In this case, the implication does not go the other way—it may be that Σ is complete with respect to $\mathfrak{M}$ but $\mathfrak{Mod}(\Sigma) \not\subseteq \mathfrak{M}$. To see this, let $\mathfrak{A} = \mathfrak{Mod}(\Sigma)$; then, as we have seen, Σ is complete with respect to $\mathfrak{A}$. Again consider $\mathsf{L}, \mathsf{M} \in \mathfrak{A}$ that make all the same sentences true, and let $\mathfrak{A}^*$ be like $\mathfrak{A}$ but without M; then $|\mathfrak{A}| = |\mathfrak{A}^*|$. Suppose $\mathscr{P} \in |\mathfrak{A}^*|$; then $\mathscr{P} \in |\mathfrak{A}|$; and since Σ is complete with respect to $\mathfrak{A}$, $\Sigma \vdash \mathscr{P}$; so if $\mathscr{P} \in |\mathfrak{A}^*|$ then $\Sigma \vdash \mathscr{P}$, and Σ is complete with respect to $\mathfrak{A}^*$. But $\mathfrak{A} \not\subseteq \mathfrak{A}^*$ and so $\mathfrak{Mod}(\Sigma) \not\subseteq \mathfrak{A}^*$. So Σ is complete with respect to $\mathfrak{A}^*$ but $\mathfrak{Mod}(\Sigma) \not\subseteq \mathfrak{A}^*$.

It remains, however, that T11.11 extends to show that c̊ompleteness is equivalent to a condition on relative completeness.

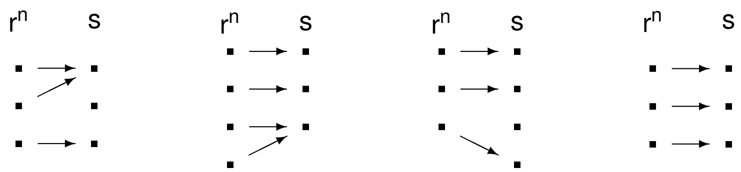
*T11.12. Derivations are c̊omplete iff every Σ is complete with respect to $\mathfrak{Md}(\Sigma)$.

Homework.

*E11.9. Show T11.12.

Isomorphism and Equivalence

In general, a total function f from r^{n} to (*into*) s maps each member of r^{n} to some member of s. Very often we have seen cases where both r and s are the universe of an interpretation and so the same set, but this is not required. f is *onto* s iff for every member of s, there is some member of r^{n} that maps to it. And f is one-to-one (1:1) iff different members of r^{n} never map to the same member of s. So, for example, each of the following satisfy the described conditions.



(total) onto 1:1 1:1, onto

Each is a total function from r^{n} to s. The first is neither onto nor 1:1; the second is onto but not 1:1; the third is 1:1 but not onto; and the last is 1:1 and onto. So a (total) 1:1 function from r^{n} onto s "matches" the members of r^{n} with the members of s.[3]

Given this, interpretations (models) are *isomorphic* when they have a sort of structural similarity that results by a 1:1 function from the universe of one onto the universe of the other.

IS For some language $\mathscr{L}$ with models L and M, L is ι-*isomorphic* to M ($\mathsf{L} \overset{\iota}{\cong} \mathsf{M}$) iff ι is a 1:1 function from the universe of L onto the universe of M and,

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$.

(f) For a function symbol h^n, $\mathsf{M}[h^n]\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle = \iota(\mathsf{L}[h^n]\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle)$.

(r) For a relation symbol $\mathcal{R}^n$, $\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle \in \mathsf{M}(\mathcal{R}^n)$ iff $\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle \in \mathsf{L}[\mathcal{R}^n]$.

[3]A function which is onto is very often called a *surjection*, a function which is 1:1 an *injection*, and one which is both 1:1 and onto a *bijection*.

If there is some ι such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{M}$, then L is *isomorphic* to M ($\mathsf{L} \cong \mathsf{M}$). For an isomorphism, the interpretation of sentence letters is the same. Then ι maps one interpretation onto the other. We might think of the two interpretations as already existing, and *finding* a function ι to exhibit them as isomorphic. Alternatively, given a model L and 1:1 function ι from its universe U_{L} onto some set U_{M}, we might think of M as resulting from application of ι to L. As we shall see, structurally, isomorphic interpretations "mirror" one another.

Here are some example isomorphic interpretations. In the first, the structural similarity between interpretations L and M is perhaps particularly obvious.

$$\text{(I)} \qquad \begin{array}{lcccc} \mathsf{U}_{\mathsf{L}}: & \text{Balto} & \text{Fido} & \text{Coco} & \text{Milo} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathsf{U}_{\mathsf{M}}: & \text{Benji} & \text{Fang} & \text{Cookie} & \text{Morris} \end{array}$$

U_{L} = {Balto, Fido, Coco, Milo}. As represented by the arrows, function ι maps these onto a disjoint set U_{M}. Then given the intended model L as below, the corresponding isomorphic interpretation is M as on the right.

$\mathsf{L}[b]$ = Balto	$\mathsf{M}[b]$ = Benji
$\mathsf{L}[c]$ = Coco	$\mathsf{M}[c]$ = Cookie
$\mathsf{L}[D]$ = {Balto, Fido}	$\mathsf{M}[D]$ = {Benji, Fang}
$\mathsf{L}[C]$ = {Coco, Milo}	$\mathsf{M}[C]$ = {Cookie, Morris}
$\mathsf{L}[P]$ = {⟨Balto, Coco⟩, ⟨Fido, Milo⟩}	$\mathsf{M}[P]$ = {⟨Benji, Cookie⟩, ⟨Fang, Morris⟩}

On model L, where Balto and Fido are dogs, and Coco and Milo are cats, and P represents pursuit, we have that every dog pursues at least one cat. So Db and Cc and $\forall x(Dx \rightarrow \exists y(Cy \wedge Pxy))$. And, supposing that Benji and Fang are dogs, and Cookie and Morris are cats, the same properties and relations are preserved on M—with only the particular individuals switched.

For a second case, let U_{L} be the same, but U_{K} the very same set, only permuted or shuffled so that each object in U_{L} has a mate in U_{K}.

$$\text{(J)} \qquad \begin{array}{lcccc} \mathsf{U}_{\mathsf{L}}: & \text{Balto} & \text{Fido} & \text{Coco} & \text{Milo} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathsf{U}_{\mathsf{K}}: & \text{Balto} & \text{Coco} & \text{Fido} & \text{Milo} \end{array}$$

So ι maps members of U_{L} to members of the very same set. Then given L as before, the corresponding isomorphic interpretation K is as follows:

$\mathsf{L}[b]$ = Balto	$\mathsf{K}[b]$ = Balto
$\mathsf{L}[c]$ = Coco	$\mathsf{K}[c]$ = Fido
$\mathsf{L}[D]$ = {Balto, Fido}	$\mathsf{K}[D]$ = {Balto, Coco}
$\mathsf{L}[C]$ = {Coco, Milo}	$\mathsf{K}[C]$ = {Fido, Milo}
$\mathsf{L}[P]$ = {⟨Balto, Coco⟩, ⟨Fido, Milo⟩}	$\mathsf{K}[P]$ = {⟨Balto, Fido⟩, ⟨Coco, Milo⟩}

This time, there is no simple way to understand $\mathsf{K}[D]$ as the set of all dogs, and $\mathsf{K}[C]$ as the set of all cats. And we cannot say that the interpretation of P reflects dogs pursuing cats. But Coco *plays the same role* in K as Fido in L; and similarly Fido plays the same role in K as Coco in L. Thus, on K, it remains that Db and Cc and that each thing in the interpretation of D stands in the relation P to at least one thing in the interpretation of C so that $\forall x(Dx \to \exists y(Cy \wedge Pxy))$—and this is just as in model L.

A final example switches to $\mathcal{L}_{NT}$ and has an infinite U. We let $\mathsf{U_N}$ be the set $\mathbb{N}$ of natural numbers, $\mathsf{U_P}$ the set $\mathbb{P}$ of positive integers, and ι be the function $\mathsf{m}+1$.

$$\text{(K)} \qquad \begin{array}{llllll} \mathsf{U_N}: & 0 & 1 & 2 & 3 & \cdots \\ & \downarrow & \downarrow & \downarrow & \downarrow & \\ \mathsf{U_P}: & 1 & 2 & 3 & 4 & \cdots \end{array}$$

Then where N is the standard interpretation for the symbols of $\mathcal{L}_{NT}$,

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$\mathsf{N}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\mathsf{N}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

we obtain P as follows:

$\mathsf{P}[\emptyset] = 1$

$\mathsf{P}[S] = \{\langle \mathsf{m}+1, \mathsf{n}+1\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$\mathsf{P}[+] = \{\langle\langle \mathsf{m}+1, \mathsf{n}+1\rangle, \mathsf{o}+1\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\mathsf{P}[\times] = \{\langle\langle \mathsf{m}+1, \mathsf{n}+1\rangle, \mathsf{o}+1\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

We build P explicitly by the rule for isomorphisms—simply finding $\iota(\mathsf{m}) = \mathsf{m}+1$ for each object in N. In this case, we cannot understand $\mathsf{P}[+]$ and $\mathsf{P}[\times]$ as the ordinary addition and multiplication functions. For example, since $\langle\langle 1, 1\rangle, 2\rangle \in \mathsf{N}[+]$, $\langle\langle 2, 2\rangle, 3\rangle \in \mathsf{P}[+]$—and, of course, $\langle\langle 2, 2\rangle, 3\rangle \notin \mathsf{N}[+]$. Nevertheless, insofar as matched objects play the same role on the different interpretations, the same *formulas* come out true on N as on P. So, for example, $S\emptyset + S\emptyset = SS\emptyset$ is true on N and, insofar as $\langle\langle 2, 2\rangle, 3\rangle \in \mathsf{P}[+]$ and $\mathsf{P}[S\emptyset] = 2$ and $\mathsf{P}[SS\emptyset] = 3$, on P as well. This is very much as for examples (I) and (J).

We shall be able to show that this sort of relation holds in general for isomorphic interpretations. That is,

EE For some language $\mathcal{L}$, models L and M are *elementarily equivalent* ($\mathsf{L} \equiv \mathsf{M}$) iff for any formula $\mathcal{P}$, $\mathsf{L}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{M}[\mathcal{P}] = \mathsf{T}$.

We show that isomorphic models are elementarily equivalent. This is straightforward given a matched pair of results, of the sort we have seen before.

T11.13. For some language $\mathcal{L}$, if models $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$ and assignments d for D and h for H are such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, then for any term t, $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$.

Suppose $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$ and corresponding assignments d and h are such that for any x, $\mathsf{h}(x) = \iota(\mathsf{d}(x))$. By induction on the number of function symbols in t:

Basis: If t has no function symbols, then it is a variable or a constant. Suppose t is a variable x; by TA(v) $\mathsf{H_h}[x]$ is $\mathsf{h}[x]$; by the assumption to the theorem this is $\iota(\mathsf{d}[x])$; and by TA(v) this is $\iota(\mathsf{D_d}[x])$; so $\mathsf{H_h}[x] = \iota(\mathsf{D_d}[x])$. Suppose t is a constant c; by TA(c), $\mathsf{H_h}[c]$ is $\mathsf{H}[c]$; since $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$, this is $\iota(\mathsf{D}[c])$; and by TA(c), this is $\iota(\mathsf{D_d}[c])$; so $\mathsf{H_h}[c] = \iota(\mathsf{D_d}[c])$.

Assp: For any i, $0 \le i < k$ if t has i function symbols, then $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$.

Show: If t has k function symbols, then $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$. If t has k function symbols, then it is of the form $h^n s_1 \dots s_n$ for relation symbol h^n and terms $s_1 \dots s_n$ with $< k$ function symbols. By assumption, $\mathsf{H_h}[s_1] = \iota(\mathsf{D_d}[s_1])$ and … and $\mathsf{H_h}[s_n] = \iota(\mathsf{D_d}[s_n])$. So $\mathsf{H_h}[t] = \mathsf{H_h}[h^n s_1 \dots s_n]$; by TA(f) this is $\mathsf{H}[h^n]\langle \mathsf{H_h}[s_1] \dots \mathsf{H_h}[s_n]\rangle$; with the assumption, this is $\mathsf{H}[h^n]\langle \iota(\mathsf{D_d}[s_1]) \dots \iota(\mathsf{D_d}[s_n])\rangle$; and since $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$, this is $\iota(\mathsf{D}[h^n]\langle \mathsf{D_d}[s_1] \dots \mathsf{D_d}[s_n]\rangle)$; by TA(f) again, this is $\iota(\mathsf{D_d}[h^n s_1 \dots s_n]) = \iota(\mathsf{D_d}[t])$. So $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$.

Indct: For any t, $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$.

So when D and H are ι-isomorphic and for any variable x, ι maps $\mathsf{d}[x]$ to $\mathsf{h}[x]$, then for any term t, ι maps $\mathsf{D_d}[t]$ to $\mathsf{H_h}[t]$.

Now we are in a position to extend the result to one for satisfaction of formulas. If D and H are ι-isomorphic, and for any variable x, ι maps $\mathsf{d}[x]$ to $\mathsf{h}[x]$, then a formula $\mathcal{P}$ is satisfied on D with d just in case it is satisfied on H with h.

*T11.14. For some language $\mathcal{L}$, if interpretations $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$ and assignments d for D and h for H and are such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, then for any formula $\mathcal{P}$, $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$.

By induction on the number of operators in $\mathcal{P}$. Suppose $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$.

Basis: Suppose $\mathcal{P}$ has no operator symbols and d and h are such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$. Then $\mathcal{P}$ is sentence letter $\mathcal{S}$ or an atomic $\mathcal{R}^n t_1 \dots t_n$ for relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. Suppose the former; then by SF(s), $\mathsf{H_h}[\mathcal{S}] = \mathsf{S}$ iff $\mathsf{H}[\mathcal{S}] = \mathsf{T}$; since $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$ iff $\mathsf{D}[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{D_d}[\mathcal{S}] = \mathsf{S}$. Suppose the latter; by SF(r), $\mathsf{H_h}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{H_h}[t_1] \dots \mathsf{H_h}[t_n]\rangle \in \mathsf{H}[\mathcal{R}^n]$; since $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$ and $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, by T11.13 iff $\langle \iota(\mathsf{D_d}[t_1]) \dots \iota(\mathsf{D_d}[t_n])\rangle \in \mathsf{H}[\mathcal{R}^n]$; since $\mathsf{D} \stackrel{\iota}{\cong} \mathsf{H}$ iff $\langle \mathsf{D_d}[t_1] \dots \mathsf{D_d}[t_n]\rangle \in \mathsf{D}[\mathcal{R}^n]$; by SF(r), iff $\mathsf{D_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, for $\mathcal{P}$ with i operator symbols and any d and h such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$.

Show: For any $\mathcal{P}$ with k operator symbols and any d and h such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, or $\forall x\mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. Suppose for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$.

($\sim$) Suppose $\mathscr{P}$ is of the form $\sim\mathcal{A}$. Then $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{H_h}[\mathcal{A}] \neq \mathsf{S}$; by assumption, iff $\mathsf{D_d}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$) iff $\mathsf{D_d}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{D_d}[\mathscr{P}] = \mathsf{S}$.

($\rightarrow$) Homework.

($\forall$) Suppose $\mathscr{P}$ is of the form $\forall x\mathcal{A}$. (i) Suppose $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$ but $\mathsf{D_d}[\mathscr{P}] \neq \mathsf{S}$; from the latter, $\mathsf{D_d}[\forall x\mathcal{A}] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{m} \in \mathsf{U_D}$ such that $\mathsf{D_{d(x|m)}}[\mathcal{A}] \neq \mathsf{S}$; but $\mathsf{d}(x|\mathsf{m})$ and $\mathsf{h}(x|\iota(\mathsf{m}))$ have all their members related by ι; so by assumption $\mathsf{H_{h(x|\iota(m))}}[\mathcal{A}] \neq \mathsf{S}$; so there is an $\mathsf{o} \in \mathsf{U_H}$ such that $\mathsf{H_{h(x|o)}}[\mathcal{A}] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{H_h}[\forall x\mathcal{A}] \neq \mathsf{S}$; so $\mathsf{H_h}[\mathscr{P}] \neq \mathsf{S}$. This is impossible; reject the assumption: if $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$, then $\mathsf{D_d}[\mathscr{P}] = \mathsf{S}$. (ii) And similarly, [by homework] in the other direction.

For d and h such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$ and $\mathscr{P}$ with k operator symbols, $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathscr{P}] = \mathsf{S}$.

Indct: For d and h such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, and any $\mathscr{P}$, $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathscr{P}] = \mathsf{S}$.

As often occurs, the most difficult case is for the quantifier. The key is that the assumption applies to $\mathsf{H_h}[\mathscr{P}]$ and $\mathsf{D_d}[\mathscr{P}]$ for *any* assignments d and h related so that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$. Supposing that d and h are so related, there is no reason to think that $\mathsf{d}(x|\mathsf{m})$ and h remain in that relation. The problem is solved with a corresponding modification to h: with $\mathsf{d}(x|\mathsf{m})$; we modify h so that the assignment to x simply is $\iota(\mathsf{m})$. Thus $\mathsf{d}(x|\mathsf{m})$ and $\mathsf{h}(x|\iota(\mathsf{m}))$ are related so that the assumption applies.

Now it is a simple matter to show that isomorphic models are elementarily equivalent.

T11.15. If $\mathsf{D} \cong \mathsf{H}$, then $\mathsf{D} \equiv \mathsf{H}$.

Suppose $\mathsf{D} \cong \mathsf{H}$ and consider an arbitrary formula $\mathscr{P}$. Since $\mathsf{D} \cong \mathsf{H}$ there is some ι such that $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$; and where d and h are related as in T11.14, $\mathsf{H_h}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathscr{P}] = \mathsf{S}$. (i) Suppose $\mathsf{D}[\mathscr{P}] = \mathsf{T}$; then by T7.6, $\mathsf{D}[\mathscr{P}^u] = \mathsf{T}$; so by T8.7, there is some d such that $\mathsf{D_d}[\mathscr{P}^u] = \mathsf{S}$; so by T11.14, $\mathsf{H_h}[\mathscr{P}^u] = \mathsf{S}$; so by T8.7, $\mathsf{H}[\mathscr{P}^u] = \mathsf{T}$; and by T7.6, $\mathsf{H}[\mathscr{P}] = \mathsf{T}$. (ii) Similarly in the other direction. And since $\mathscr{P}$ is arbitrary, $\mathsf{D} \equiv \mathsf{H}$.

Suppose $\mathsf{M} \in \mathfrak{Md}(\Sigma)$ and $\mathsf{M} \cong \mathsf{L}$; then $\mathsf{M}[\Sigma] = \mathsf{T}$ and by this theorem, $\mathsf{M} \equiv \mathsf{L}$; so $\mathsf{L}(\Sigma) = \mathsf{T}$ and $\mathsf{L} \in \mathfrak{Md}(\Sigma)$. At best, then, a set of formulas characterizes models "up to isomorphism"—if $\mathsf{M} \in \mathfrak{Md}(\Sigma)$ then $\mathfrak{Md}(\Sigma)$ includes all the models isomorphic to M.[4] This already answers the first question posed in the introduction to section 11.4: If $\mathsf{N} \in \mathfrak{Md}(\Sigma)$ then any $\mathsf{L} \cong \mathsf{N}$ is in $\mathfrak{Md}(\Sigma)$ as well; so there is no Σ such that $\mathfrak{Md}(\Sigma)$

[4] In *Reason, Truth and History*, Hilary Putnam makes this point to show that truth values of sentences are not sufficient to fix the interpretation of a language. The technical point is clear enough. It is another matter whether it bears the philosophical weight he means for it to bear!

is the class whose only member is N (that there *are* interpretations isomorphic to N is immediate from example (K)). Note also that we now have the definitions at least to *understand* the last two questions.

Given notions of isomorphism and equivalence, let us briefly return to relative soundness and completeness. This time we connect relative soundness and completeness with $\tilde{s}$oundness and $\tilde{c}$ompleteness. Associate Σ with a class $\mathfrak{I}$ of *intended* models. Then Σ is *$\tilde{s}$ound* iff it is sound with respect to $\mathfrak{I}$, so for any formula $\mathscr{P}$ if $\Sigma \vdash \mathscr{P}$, then $\mathscr{P} \in |\mathfrak{I}|$. One option is to set $\mathfrak{I} = \mathfrak{Mod}(\Sigma)$; then by T11.9, it is immediate that Σ is sound with respect to $\mathfrak{I}$. Alternatively, and more relevantly, $\mathfrak{I}$ might be independently specified; thus the intended interpretation for Q and PA is just N. Then there is an open question whether a given Σ is sound with respect to $\mathfrak{I}$. In general, from T11.9, Σ is $\tilde{s}$ound iff $\mathfrak{I} \subseteq \mathfrak{Mod}(\Sigma)$. From this, $\tilde{s}$oundness is closely related to logical soundness from Chapter 1: given $\mathring{s}$oundness, Σ entails whatever it proves—$\tilde{s}$oundness then adds the requirement that the members of Σ are true on intended models.

Σ is (*negation*) *$\tilde{c}$omplete* iff $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$ for every sentence $\mathscr{P}$ of its language. So $\tilde{c}$ompleteness is like maximality from Chapter 10 (though $\tilde{c}$ompleteness applies especially to *theories*). We show that Σ is $\tilde{c}$omplete iff it is complete relative to some $\mathfrak{M}$ whose members are elementarily equivalent.

*T11.16. Σ is $\tilde{c}$omplete iff it is complete relative to some $\mathfrak{M}$ such that for all $\mathsf{L}, \mathsf{M} \in \mathfrak{M}$, $\mathsf{L} \equiv \mathsf{M}$.

(i) Suppose Σ is $\tilde{c}$omplete. By [homework] there is an $\mathfrak{M}$ such that for all $\mathsf{L}, \mathsf{M} \in \mathfrak{M}$, $\mathsf{L} \equiv \mathsf{M}$. Hint: $\mathfrak{Mod}(\Sigma)$ is an $\mathfrak{M}$ with the required properties.

(ii) Suppose Σ is complete relative to some $\mathfrak{M}$ such that for all $\mathsf{L}, \mathsf{M} \in \mathfrak{M}$, $\mathsf{L} \equiv \mathsf{M}$. Suppose $\mathfrak{M} = \varnothing$; then $|\mathfrak{M}|$ is the set of all formulas; so by relative completeness Σ (is inconsistent and) proves $\mathscr{P}$ and $\sim\mathscr{P}$ for every $\mathscr{P}$; so for every sentence $\mathscr{P}$, $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$. Suppose $\mathfrak{M} \neq \varnothing$; then there is some $\mathsf{M} \in \mathfrak{M}$, and from E8.29 for any sentence $\mathscr{P}$, $\mathsf{M}[\mathscr{P}] = \mathsf{T}$ or $\mathsf{M}[\sim\mathscr{P}] = \mathsf{T}$. Suppose $\mathsf{M}[\mathscr{P}] = \mathsf{T}$; and consider an arbitrary $\mathsf{L} \in \mathfrak{M}$; since $\mathsf{L} \equiv \mathsf{M}$, $\mathsf{L}[\mathscr{P}] = \mathsf{T}$; and since L is arbitrary, $\mathscr{P} \in |\mathfrak{M}|$; so by relative completeness $\Sigma \vdash \mathscr{P}$; so $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$. Suppose $\mathsf{M}[\sim\mathscr{P}] = \mathsf{T}$; and consider an arbitrary $\mathsf{L} \in \mathfrak{M}$; since $\mathsf{L} \equiv \mathsf{M}$, $\mathsf{L}[\sim\mathscr{P}] = \mathsf{T}$; and since L is arbitrary, $\sim\mathscr{P} \in |\mathfrak{M}|$; so by relative completeness, $\Sigma \vdash \sim\mathscr{P}$; so $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$. In any case, then, $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$; so Σ is $\tilde{c}$omplete.

Suppose Σ is $\tilde{c}$omplete; by the reasoning for (i), the members of $\mathfrak{Mod}(\Sigma)$ are elementarily equivalent. Suppose the members of $\mathfrak{Mod}(\Sigma)$ are elementarily equivalent; by T11.11 Σ is complete with respect to $\mathfrak{Mod}(\Sigma)$; so by (ii) Σ is $\tilde{c}$omplete. So ($*$) Σ is $\tilde{c}$omplete iff the members of $\mathfrak{Mod}(\Sigma)$ are elementarily equivalent.

Now say Σ is *categorical* iff it characterizes models up to isomorphism, iff for any $\mathsf{L}, \mathsf{M} \in \mathfrak{Mod}(\Sigma)$, $\mathsf{L} \cong \mathsf{M}$. Then as another quick corollary to T11.16, if Σ is categorical, it is $\tilde{c}$omplete: Suppose Σ is categorical; and consider arbitrary $\mathsf{L}, \mathsf{M} \in \mathfrak{Mod}(\Sigma)$, since

Σ is categorical, $\mathsf{L} \cong \mathsf{M}$; so by T11.15, $\mathsf{L} \equiv \mathsf{M}$; and since L and M are arbitrary, for all $\mathsf{L}, \mathsf{M} \in \mathfrak{Mod}(\Sigma)$, $\mathsf{L} \equiv \mathsf{M}$; so from $(*)$ Σ is $\tilde{\text{c}}$omplete.

*E11.10. Suppose $\mathsf{L} \overset{\iota}{\cong} \mathsf{M}$. It is possible to manipulate definition IS to show that clauses (c) and (f) yield (intuitive) conditions as for relation symbols. So, for constants, $\mathsf{M}[c] = \iota(\mathsf{m})$ iff $\mathsf{L}[c] = \mathsf{m}$.

> If $\mathsf{M}[c] = \iota(\mathsf{m})$, then with IS(c), $\iota(\mathsf{L}[c]) = \mathsf{M}[c] = \iota(\mathsf{m})$, and since ι is 1:1, $\mathsf{L}[c] = \mathsf{m}$. And if $\mathsf{L}[c] = \mathsf{m}$, then with IS(c), $\mathsf{M}[c] = \iota(\mathsf{L}[c]) = \iota(\mathsf{m})$.

Demonstrate the related result for function symbols, that $\langle\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle, \iota(\mathsf{o})\rangle \in \mathsf{M}[h^n]$ iff $\langle\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle, \mathsf{o}\rangle \in \mathsf{L}[h^n]$. It is sufficient to restrict attention to one-place function symbols.

E11.11. (i) Explain what truth value the sentence $\sim\exists x(Dx \wedge \forall y(Cy \to Pxy))$ has on interpretation L and then M in example (I). Explain what truth value it has on K in example (J). (ii) Explain what truth value the sentence $\exists x(x + x = x)$ has on interpretations N and P in example (K). Are these results as you expect? Explain.

*E11.12. Complete the proof of T11.14 including the cases for $\to$ and $\forall$. You should set up the complete induction, but may refer to the text, as the text refers to homework. Hint for the quantifier case: Since ι is onto $\mathsf{U_H}$, for any $\mathsf{m} \in \mathsf{U_H}$ there must be some $\mathsf{n} \in \mathsf{U_D}$ such that $\iota(\mathsf{n}) = \mathsf{m}$; so $\mathsf{d}(x|\mathsf{n})$ and $\mathsf{h}(x|\mathsf{m})$ are related as the assumption requires.

*E11.13. Complete the proof of T11.16. Hint: do not forget that you can use T7.6.

Submodel and Embedding

We conclude this section with discussion of *submodels* and *elementary submodels*, then *embeddings* and *elementay embeddings*.

First submodel. For a relation $\mathsf{r^n}$ say the *restriction* of $\mathsf{r^n}$ to set s, $\mathsf{r^n} \restriction \mathsf{s} = \{\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle \mid \langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle \in \mathsf{r^n}$ and $\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle \in \mathsf{s^n}\}$. We take just the members of $\mathsf{r^n}$ that are included in $\mathsf{s^n}$. Similarly, for a function $\mathsf{f^n}$, the *restriction* of $\mathsf{f^n}$ to s, $\mathsf{f^n} \restriction \mathsf{s} = \{\langle\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle, \mathsf{a}\rangle \mid \langle\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle, \mathsf{a}\rangle \in \mathsf{f^n}$ and $\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle \in \mathsf{s^n}\}$. We take the members of $\mathsf{f^n}$ whose inputs are included in $\mathsf{s^n}$. A set s is *closed* under $\mathsf{f^n}$ just in case whenever $\langle\langle\mathsf{m_1}\ldots\mathsf{m_n}\rangle, \mathsf{a}\rangle \in \mathsf{f^n} \restriction \mathsf{s}$, then $\mathsf{a} \in \mathsf{s}$. So, for example, let $2\mathbb{N}$ be the set $\{0, 2, 4, \ldots\}$ of even numbers. Then the usual addition function plus restricted to $2\mathbb{N}$ includes pairs $\langle\langle 2, 2\rangle, 4\rangle$, $\langle\langle 2, 4\rangle, 6\rangle$, and so forth; and insofar as the sum of two evens is always even, $2\mathbb{N}$ is closed under plus. The usual successor function suc restricted to $2\mathbb{N}$ has members $\langle 0, 1\rangle$, $\langle 2, 3\rangle$, $\langle 4, 5\rangle$, and so forth; but insofar as successors of

First Theorems of Chapter 11

T11.1 It is possible to represent any truth function by means of an expression with just the operators $\sim$, $\wedge$, and $\vee$.

T11.2 It is possible to represent any truth function by means of an expression with just the operators $\sim$ and $\rightarrow$, with just the operators $\sim$ and $\wedge$, and with just the operators $\sim$ and $\vee$.

T11.3 For any formula $\mathcal{P}$ of $\mathcal{L}_s$, exactly one of the following holds: (i) $\mathcal{P}$ is a sentence letter; (ii) there is a formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$; (iii) there are formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$.

T11.4 If $\mathcal{A}$ is a formula of $\mathcal{L}_s$, then no proper initial segment of $\mathcal{A}$ is a formula.

T11.5 For any formula $\mathcal{P}$ of $\mathcal{L}_s$, exactly one of the following holds: (s) $\mathcal{P}$ is a sentence letter; ($\sim$) there is a unique formula $\mathcal{A}$ such that $\mathcal{P}$ is $\sim\mathcal{A}$; ($\rightarrow$) there are unique formulas $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$.

T11.6 In the sentential fragment of *AD*, axiom A1 is independent of A2 and A3.

T11.7 In the sentential fragment of *AD*, A2 is independent of A1 and A3, and A3 is independent of A1 and A2.

T11.8 If $\mathfrak{M} \subseteq \mathfrak{N}$ then $|\mathfrak{N}| \subseteq |\mathfrak{M}|$.

T11.9 If derivations are $\mathring{s}$ound, then Σ is sound with respect to $\mathfrak{M}$ iff $\mathfrak{M} \subseteq \mathfrak{Md}(\Sigma)$.

T11.10 Derivations are $\mathring{s}$ound iff every Σ is sound with respect to $\mathfrak{Md}(\Sigma)$.

T11.11 If derivations are $\mathring{c}$omplete and $\mathfrak{Md}(\Sigma) \subseteq \mathfrak{M}$, then Σ is complete with respect to $\mathfrak{M}$.

T11.12 Derivations are $\mathring{c}$omplete iff every Σ is complete with respect to $\mathfrak{Md}(\Sigma)$.

T11.13 For some language $\mathcal{L}$, if models $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$ and assignments d for D and h for H are such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, then for any term t, $\mathsf{H_h}[t] = \iota(\mathsf{D_d}[t])$.

T11.14 For some language $\mathcal{L}$, if interpretations $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$ and assignments d for D and h for H and are such that for any x, $\mathsf{h}[x] = \iota(\mathsf{d}[x])$, then for any formula $\mathcal{P}$, $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$.

T11.15 If $\mathsf{D} \cong \mathsf{H}$, then $\mathsf{D} \equiv \mathsf{H}$.

T11.16 Σ is $\tilde{c}$omplete iff it is complete relative to some $\mathfrak{M}$ such that for all $\mathsf{L}, \mathsf{M} \in \mathfrak{M}$, $\mathsf{L} \equiv \mathsf{M}$.

Corollary: Σ is $\tilde{c}$omplete iff the members of $\mathfrak{Md}(\Sigma)$ are elementarily equivalent.

Corollary: If Σ is categorical then Σ is $\tilde{c}$omplete.

evens are odd, $2\mathbb{N}$ is not closed under suc. For model M of some language $\mathcal{L}$, and some $\mathsf{V} \subseteq \mathsf{U}$, say V is *closed under the constants of* M just in case for any constant symbol c, $\mathsf{M}[c] \in \mathsf{V}$. And V is *closed under the functions of* M just in case for any function symbol h, V is closed under $\mathsf{M}[h]$.

Given this, the relations and functions in a *submodel* of M restrict the functions and relations of M to a subset of the universe of M.

SM For some language $\mathcal{L}$ with models L and M, L is a *submodel* of M ($\mathsf{L} \sqsubseteq \mathsf{M}$) iff,

(u) $\mathsf{U_L} \subseteq \mathsf{U_M}$, and $\mathsf{U_L}$ is closed under the constants and functions of M.

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}(c) = \mathsf{L}(c)$.

(f) For a function symbol h^n, $\mathsf{M}[h^n] \restriction \mathsf{U_L} = \mathsf{L}[h^n]$.

(r) For a relation symbol $\mathcal{R}^n$, $\mathsf{M}[\mathcal{R}^n] \restriction \mathsf{U_L} = \mathsf{L}[\mathcal{R}^n]$.

So when $\mathsf{L} \sqsubseteq \mathsf{M}$ the relations and functions of L are restrictions of the relations and functions of M to $\mathsf{U_L}$. Insofar as we require that the interpretation of any constant be a member of the universe, and that the elements in members of a function be from the universe, SM would fail to specify an interpretation apart from the requirements on $\mathsf{U_L}$ from (u). Trivially, $\mathsf{M} \sqsubseteq \mathsf{M}$; in this case, M and its submodel have the same universe. If the language lacks constants and function symbols, then the restriction to any nonempty $\mathsf{U_L} \subseteq \mathsf{U_M}$ results in a submodel (because the constraints from (u) are trivially met). If there are no constants and all the functions are such that $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \in \mathsf{f}$ only if a is among $\mathsf{m}_1 \ldots \mathsf{m}_n$ then, again, the constraints are automatically met. Otherwise, supposing that $\mathsf{U_M}$ has nonempty proper subsets (has at least two members), not all of them result in a submodel. But for any nonempty subset of $\mathsf{U_M}$ that meets the conditions from (u), there is a submodel of M that restricts the relations and functions of M to that set.

For an example, consider a language like $\mathcal{L}_{NT}^{<}$ but without S, where $\mathsf{U_M} = \mathbb{N}$ and $\mathsf{U_L} = 2\mathbb{N}$. Then $\mathsf{U_L} \subseteq \mathsf{U_M}$. Let,

$$\text{(L)}\quad \begin{aligned}
&\mathsf{M}[\emptyset] = 0\\
&\mathsf{M}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ is less than } \mathsf{n}\}\\
&\mathsf{M}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ plus } \mathsf{n} \text{ equals } \mathsf{o}\}\\
&\mathsf{M}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ times } \mathsf{n} \text{ equals } \mathsf{o}\}
\end{aligned}$$

$$\mathsf{L}[\emptyset] = 0\ ;\ \mathsf{L}[<] = \mathsf{M}[<] \restriction \mathsf{U_L};\ \mathsf{L}[+] = \mathsf{M}[+] \restriction \mathsf{U_L};\ \mathsf{L}[\times] = \mathsf{M}[\times] \restriction \mathsf{U_L}$$

So M is the standard interpretation of these symbols, and L restricts its functions and relation to $2\mathbb{N}$. Insofar as the assignment to $\emptyset$ is the same on M and L, and the sum of two evens is even, and the product of two evens is even, $\mathsf{U_L}$ is closed under the constants and functions of M; so $\mathsf{L} \sqsubseteq \mathsf{M}$. For this submodel L, it remains that $\mathsf{L}[+]$ and $\mathsf{L}[\times]$ work like (part of) the usual functions plus and times. If S were to have remained in the language (given its usual interpretation for M), then $\mathsf{L} \not\sqsubseteq \mathsf{M}$ insofar as $2\mathbb{N}$ is not closed under suc. As specified, $\mathsf{M} \not\equiv \mathsf{L}$—so, for example, $\exists x(\emptyset < x \wedge x \times x = x)$ is true on M insofar as $1 \times 1 = 1$, but not true on L insofar as $1 \notin \mathsf{U_L}$. And since the interpretations are not elementarily equivalent, by T11.15 they are not isomorphic. If we were to have omitted $\times$ from the language it would remain that $\mathsf{L} \sqsubseteq \mathsf{M}$. In this case, however, $\mathsf{M} \cong \mathsf{L}$—for $\iota(\mathsf{m}) = 2\mathsf{m}$ becomes a 1:1 function from $\mathbb{N}$ onto $2\mathbb{N}$ that maps one model into the other—and since these models are isomorphic, they are elementarily equivalent. So a submodel of M may but need not be isomorphic and/or equivalent to M.

But submodels may be restricted so that *elementary submodels* of M are elementarily equivalent to M.

ES For some language $\mathcal{L}$ with models L and M, L is an *elementary submodel* of M ($\mathsf{L} \preceq \mathsf{M}$) iff $\mathsf{L} \sqsubseteq \mathsf{M}$ and for any formula $\mathcal{P}$ of $\mathcal{L}$ and variable assignment d into $\mathsf{U_L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

So an elementary submodel is a submodel such that $\mathsf{M_d}[\mathcal{P}] = \mathsf{L_d}[\mathcal{P}]$ for any assignment d into $\mathsf{U_L}$. Observe that $\mathsf{L} \sqsubseteq \mathsf{M}$ and $\mathsf{M} \equiv \mathsf{L}$ do not imply $\mathsf{L} \preceq \mathsf{M}$. For consider (L) above without $\times$ in the language. Then $\mathsf{M} \equiv \mathsf{L}$. But any d into $2\mathbb{N}$ such that $\mathsf{d}[y] = 2$ has $\mathsf{M_d}[\exists x(x + x = y)] = \mathsf{S}$ just because $1 + 1 = 2$; but there is no $\mathsf{m} \in 2\mathbb{N}$ such that $\mathsf{m} + \mathsf{m} = 2$ so that $\mathsf{L_d}[\exists x(x + x = y)] \neq \mathsf{S}$. So L is not an elementary submodel of M. The implication does, however, go the other way: an elementary submodel of M is elementarily equivalent to M.

T11.17. If $\mathsf{L} \preceq \mathsf{M}$ then $\mathsf{L} \equiv \mathsf{M}$.

Suppose $\mathsf{L} \preceq \mathsf{M}$ and consider an arbitrary formula $\mathcal{P}$. (i) Suppose $\mathsf{L}[\mathcal{P}] = \mathsf{T}$; then by T7.6, $\mathsf{L}[\mathcal{P}^u] = \mathsf{T}$; so by T8.7, there is some d into $\mathsf{U_L}$ such that $\mathsf{L_d}[\mathcal{P}^u] = \mathsf{S}$; so since $\mathsf{L} \preceq \mathsf{M}$, $\mathsf{M_d}[\mathcal{P}^u] = \mathsf{S}$; so by T8.7, $\mathsf{M}[\mathcal{P}^u] = \mathsf{T}$; so by T7.6, $\mathsf{M}[\mathcal{P}] = \mathsf{T}$. (ii) Suppose $\mathsf{M}[\mathcal{P}] = \mathsf{T}$; then by T7.6, $\mathsf{M}[\mathcal{P}^u] = \mathsf{T}$; so by TI, $\mathsf{M_h}[\mathcal{P}^u] = \mathsf{S}$ for every assignment h into $\mathsf{U_M}$ and, in particular, $\mathsf{M_d}[\mathcal{P}^u] = \mathsf{S}$ for some assignment d into $\mathsf{U_L}$; so since $\mathsf{L} \preceq \mathsf{M}$, $\mathsf{L_d}[\mathcal{P}^u] = \mathsf{S}$; so by T8.7, $\mathsf{L}[\mathcal{P}^u] = \mathsf{T}$; so by T7.6, $\mathsf{L}[\mathcal{P}] = \mathsf{T}$. And since $\mathcal{P}$ is arbitrary, $\mathsf{L} \equiv \mathsf{M}$.

This much is clear. As we have seen (L), even without $\times$ in the language, is an example of a submodel not an elementary submodel. Trivially, for any M, $\mathsf{M} \preceq \mathsf{M}$. Beyond that, though, given the universal requirements that ES places on formulas and variable assignments, it is not easy to produce an obvious example of a submodel that is an elementary submodel. The following matched pair of theorems focus the question.

T11.18. Suppose $\mathsf{L} \sqsubseteq \mathsf{M}$ and d is a variable assignment into $\mathsf{U_L}$. Then for any term t, $\mathsf{M_d}[t] = \mathsf{L_d}[t]$.

By induction on the number of function symbols in t. Suppose $\mathsf{L} \sqsubseteq \mathsf{M}$ and d is a variable assignment into $\mathsf{U_L}$.

Basis: Suppose t has no function symbols. Then t is a variable x or a constant c. (i) Suppose t is a constant c. Then $\mathsf{M_d}[t] = \mathsf{M_d}[c]$; by TA(c) this is $\mathsf{M}[c]$; and since $\mathsf{L} \sqsubseteq \mathsf{M}$, this is $\mathsf{L}[c]$; by TA(c) again, this is $\mathsf{L_d}[c]$; which is just $\mathsf{L_d}[t]$. (ii) Suppose t is a variable x. Then $\mathsf{M_d}[t] = \mathsf{M_d}[x]$; by TA(v), this is $\mathsf{d}[x]$ and by TA(v) again, this is $\mathsf{L_d}[x]$; which is just $\mathsf{L_d}[t]$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, then $\mathsf{M_d}[t] = \mathsf{L_d}[t]$.

Show: If t has k function symbols, $\mathsf{M_d}[t] = \mathsf{L_d}[t]$. If t has k function symbols, then it is of the form $h^n s_1 \ldots s_n$ for some terms $s_1 \ldots s_n$ with $< k$ function symbols. $\mathsf{M_d}[t] = \mathsf{M_d}[h^n s_1 \ldots s_n]$; by TA(f) this is $\mathsf{M}[h^n]\langle \mathsf{M_d}[s_1] \ldots \mathsf{M_d}[s_n]\rangle$; with the assumption, this is $\mathsf{M}[h^n]\langle \mathsf{L_d}[s_1] \ldots \mathsf{L_d}[s_n]\rangle$; but since $\mathsf{L_d}[s_1] \ldots \mathsf{L_d}[s_n]$ are in $\mathsf{U_L}$, this is just $(\mathsf{M}[h^n] \restriction \mathsf{U_L})\langle \mathsf{L_d}[s_1] \ldots \mathsf{L_d}[s_n]\rangle$; and since $\mathsf{L} \sqsubseteq \mathsf{M}$, this is $\mathsf{L}[h^n]\langle \mathsf{L_d}[s_1] \ldots \mathsf{L_d}[s_n]\rangle$; and by TA(f), this is $\mathsf{L_d}[h^n s_1 \ldots s_n]$; which is just $\mathsf{L_d}[t]$.

Indct: For any term t, $\mathsf{M_d}[t] = \mathsf{L_d}[t]$.

T11.19. Suppose that $\mathsf{L} \sqsubseteq \mathsf{M}$ and that for any formula $\mathcal{P}$ and every variable assignment d into $\mathsf{U_L}$ such that $\mathsf{M_d}[\exists x \mathcal{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U_L}$ such that $\mathsf{M_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{L} \preceq \mathsf{M}$.

Suppose $\mathsf{L} \sqsubseteq \mathsf{M}$ and that for any formula $\mathcal{P}$ and every variable assignment d into $\mathsf{U_L}$ such that $\mathsf{M_d}[\exists x \mathcal{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U_L}$ such that $\mathsf{M_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$. We show by induction on the number of operators in $\mathcal{P}$ that for d any assignment into $\mathsf{U_L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$ and so that $\mathsf{L} \preceq \mathsf{M}$.

Basis: Suppose d is an assignment into $\mathsf{U_L}$. If $\mathcal{P}$ is atomic then it is either a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \ldots t_n$. (i) Suppose $\mathcal{P}$ is $\mathcal{S}$. Then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{S}] = \mathsf{S}$; by SF(s), iff $\mathsf{M}[\mathcal{S}] = \mathsf{T}$; since $\mathsf{L} \sqsubseteq \mathsf{M}$, iff $\mathsf{L}[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{L_d}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose $\mathcal{P}$ is $\mathcal{R}^n t_1 \ldots t_n$. Then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; by SF(r) iff $\langle \mathsf{M_d}[t_1] \ldots \mathsf{M_d}[t_n]\rangle \in \mathsf{M}[\mathcal{R}^n]$; since $\mathsf{L} \sqsubseteq \mathsf{M}$ and with T11.18, iff $\langle \mathsf{L_d}[t_1] \ldots \mathsf{L_d}[t_n]\rangle \in \mathsf{M}[\mathcal{R}^n]$; since $\mathsf{L_d}[t_1] \ldots \mathsf{L_d}[t_n]$ are in $\mathsf{U_L}$, iff $\langle \mathsf{L_d}[t_1] \ldots \mathsf{L_d}[t_n]\rangle \in (\mathsf{M}[\mathcal{R}^n] \restriction \mathsf{U_L})$; since $\mathsf{L} \sqsubseteq \mathsf{M}$, iff $\langle \mathsf{L_d}[t_1] \ldots \mathsf{L_d}[t_n]\rangle \in \mathsf{L}[\mathcal{R}^n]$; by SF(r) iff $\mathsf{L_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then for d any assignment into $\mathsf{U_L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

Show: If $\mathcal{P}$ has k operator symbols, then for d any assignment into $\mathsf{U_L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, or $\exists x \mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols (treating $\forall x \mathcal{P}$ as equivalent to $\sim\exists x \sim\mathcal{P}$). Let d be an assignment into $\mathsf{U_L}$.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$) iff $\mathsf{M_d}[\mathcal{A}] \neq \mathsf{S}$; by assumption iff $\mathsf{L_d}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$) iff $\mathsf{L_d}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

($\to$) Homework.

($\exists$) Suppose $\mathcal{P}$ is $\exists x\mathcal{A}$. (i) Suppose $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{M_d}[\exists x\mathcal{A}] = \mathsf{S}$; so by the assumption to the theorem, there is an $\mathsf{m} \in \mathsf{U_L}$ such that $\mathsf{M_{d(x|m)}}[\mathcal{A}] = \mathsf{S}$; since d is an assignment into $\mathsf{U_L}$, $\mathsf{d}(x|\mathsf{m})$ is an assignment into $\mathsf{U_L}$; so by assumption $\mathsf{L_{d(x|m)}}[\mathcal{A}] = \mathsf{S}$; so by SF$'$($\exists$), $\mathsf{L_d}[\exists x\mathcal{A}] = \mathsf{S}$; so $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{L_d}[\exists x\mathcal{A}] = \mathsf{S}$; so by SF$'$($\exists$), there is some $\mathsf{o} \in \mathsf{U_L}$ such that $\mathsf{L_{d(x|o)}}[\mathcal{A}] = \mathsf{S}$; so since $\mathsf{d}(x|\mathsf{o})$ is an assignment into $\mathsf{U_L}$, by assumption, $\mathsf{M_{d(x|o)}}[\mathcal{A}] = \mathsf{S}$; so by SF$'$($\exists$), $\mathsf{M_d}[\exists x\mathcal{A}] = \mathsf{S}$; so $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$. So $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

In any case, if $\mathcal{P}$ has k operator symbols, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

Indct: For any $\mathcal{P}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

So a submodel is an elementary submodel so long as existentially quantified formulas are guaranteed by "witnesses" in the universe of the submodel. With this in mind, you might consider again our page 516 example to show that (L), even without $\times$ in the language, is not an example of an elementary submodel.

With a small change to the definition of an isomorphism, *embeddings* combine the notions of submodel and isomorphism. In particular, we drop the requirement that the 1:1 function ι be from the domain of one model *onto* the domain of the other.

EM For some language $\mathcal{L}$ with models L and M, there is an *ι-embedding* of L into M ($\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$) iff ι is a 1:1 function from $\mathsf{U_L}$ to $\mathsf{U_M}$ and,

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$.

(f) For a function symbol h^n, $\mathsf{M}[h^n]\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle = \iota(\mathsf{L}[h^n]\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle)$.

(r) For a relation symbol $\mathcal{R}^n$, $\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle \in \mathsf{M}(\mathcal{R}^n)$ iff $\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle \in \mathsf{L}[\mathcal{R}^n]$.

If there is some ι such that $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$, then there is an *embedding* of L into M ($\mathsf{L} \precsim \mathsf{M}$). At the extreme, if $\mathsf{L} \overset{\iota}{\cong} \mathsf{M}$ then $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$. For a 1:1 function from $\mathsf{U_L}$ onto $\mathsf{U_M}$ remains a 1:1 function from $\mathsf{U_L}$ to $\mathsf{U_M}$ and all the conditions from EM are satisfied. But the interesting thing is that for an embedding the function ι need not be onto. Again at the extreme, if $\mathsf{L} \sqsubseteq \mathsf{M}$ then $\mathsf{L} \precsim \mathsf{M}$. For the identity function $\iota(\mathsf{m}) = \mathsf{m}$ is a 1:1 function from $\mathsf{U_L}$ to $\mathsf{U_M}$ that trivially satisfies the conditions from EM. So for example, on their standard interpretations of $\mathcal{L}_{\mathrm{NT}}$, a model for the natural numbers is embedded into a model for the integers. The identity function $\iota(\mathsf{m}) = \mathsf{m}$ is a 1:1 function from $\mathbb{N}$ to the set $\mathbb{Z}$ of all integers such that $\mathsf{L} \underset{\sim}{\sqsubset} \mathsf{M}$.[5] More interestingly, suppose a language has constant a, two-place function symbol $*$, and relation symbol $<$. Then where L and M have universe $\mathbb{N}$ and,

$\mathsf{L}[a] = 0$

$\mathsf{L}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

$\mathsf{L}[*] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

(M)

$\mathsf{M}[a] = 1$

$\mathsf{M}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

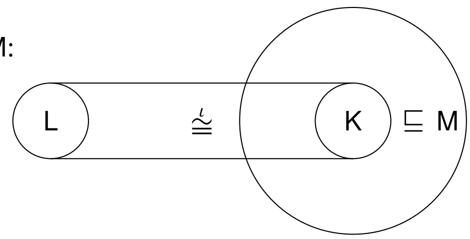
$\mathsf{M}[*] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

we have $\mathsf{L} \underset{\sim}{\sqsubset} \mathsf{M}$. Even though L and M have the same universe, given their assignments to the constant and function symbols, $\mathsf{L} \not\sqsubseteq \mathsf{M}$. Nevertheless, $\iota[\mathsf{m}] = 2^{\mathsf{m}}$ is a 1:1 function from $\mathbb{N}$ to $\mathbb{N}$ (from $\mathbb{N}$ onto a proper subset of $\mathbb{N}$) that meets the conditions from EM. So, $\mathsf{M}[a] = 1 = 2^0 = \iota(0) = \iota(\mathsf{L}[a])$. Similarly, $\langle \iota(\mathsf{a}), \iota(\mathsf{b})\rangle \in \mathsf{M}[<]$ iff $\langle 2^{\mathsf{a}}, 2^{\mathsf{b}}\rangle \in \mathsf{M}[<]$; iff $2^{\mathsf{a}} < 2^{\mathsf{b}}$; iff $\mathsf{a} < \mathsf{b}$; iff $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathsf{L}[<]$. And $\mathsf{M}[*]\langle \iota(\mathsf{a}), \iota(\mathsf{b})\rangle = \mathsf{M}[*]\langle 2^{\mathsf{a}}, 2^{\mathsf{b}}\rangle = 2^{\mathsf{a}} \times 2^{\mathsf{b}} = 2^{\mathsf{a}+\mathsf{b}} = \iota(\mathsf{a}+\mathsf{b}) = \iota(\mathsf{L}[*]\langle \mathsf{a}, \mathsf{b}\rangle)$. So $\mathsf{L} \underset{\sim}{\sqsubset} \mathsf{M}$.

In these examples, L maps to a submodel of M. And we may show that this relation holds in general, that $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$ when there is a $\mathsf{K} \sqsubseteq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$ (something like this may be suggested by the notation). The situation may be pictured as follows:

$\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$:

(N)



*T11.20. $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$ iff there is a $\mathsf{K} \sqsubseteq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$.

For a function f from r^{n} to s, let its *range* $\mathsf{ran}(\mathsf{f})$ be that subset of s onto which f maps: for a one place function f, $\mathsf{ran}(\mathsf{f}) = \{\mathsf{y} \mid \langle \mathsf{x}, \mathsf{y}\rangle \in \mathsf{f}\}$. Then when $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, L is isomorphic to a $\mathsf{K} \sqsubseteq \mathsf{M}$ that restricts the relations and functions of M to $\mathsf{ran}(\iota)$.

(i) Suppose $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$. We need a K such that $\mathsf{K} \sqsubseteq \mathsf{M}$ and $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$. Let $\mathsf{U}_{\mathsf{K}} = \mathsf{ran}(\iota)$, $\mathsf{K}[\mathcal{S}] = \mathsf{M}[\mathcal{S}]$, $\mathsf{K}[c] = \mathsf{M}[c]$, $\mathsf{K}[h^n] = \mathsf{M}[h^n] \restriction \mathsf{U}_{\mathsf{K}}$, and $\mathsf{K}[\mathcal{R}^n] = \mathsf{M}[\mathcal{R}^n] \restriction \mathsf{U}_{\mathsf{K}}$.

(a) $\mathsf{K} \sqsubseteq \mathsf{M}$. Since ι is a function from U_{L} to U_{M}, $\mathsf{U}_{\mathsf{K}} \subseteq \mathsf{U}_{\mathsf{M}}$. Then U_{K} is closed under the constants and functions of M: First, since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$; and since ι is a function to U_{K}, $\mathsf{M}[c] \in \mathsf{U}_{\mathsf{K}}$; so U_{K} is closed under the constants of M. Next, for simplicity consider a one-place function symbol h and suppose

[5]This ignores a fine point about whether members of $\mathbb{N}$ are the *same objects* as ones in $\mathbb{Z}$. I simply suppose that the natural numbers are among the integers.

$\langle \mathsf{a},\mathsf{b}\rangle \in \mathsf{M}[h^n] \restriction \mathsf{U_K}$; then $\mathsf{a} \in \mathsf{U_K}$ and we require that $\mathsf{b} \in \mathsf{U_K}$. Since $\mathsf{a} \in \mathsf{U_K}$ there is some $\mathsf{m} \in \mathsf{U_L}$ such that $\iota(\mathsf{m}) = \mathsf{a}$; so $\mathsf{M}[h^n]\langle\iota(\mathsf{m})\rangle = \mathsf{b}$; and since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, $\mathsf{M}[h^n]\langle\iota(\mathsf{m})\rangle = \iota(\mathsf{L}[h^n]\langle\mathsf{m}\rangle)$; so $\iota(\mathsf{L}[h^n]\langle\mathsf{m}\rangle) = \mathsf{b}$, and $\mathsf{b} \in \mathsf{U_K}$; so $\mathsf{U_K}$ is closed under the functions of M. Then by construction it is immediate that K meets other conditions for a submodel of M.

(b) $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$. Since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, ι is a 1:1 function from $\mathsf{U_L}$ to $\mathsf{U_M}$; so ι is a 1:1 function from $\mathsf{U_L}$ onto $\mathsf{ran}(\iota) = \mathsf{U_K}$. Now working on the conditions from IS: (s) By construction $\mathsf{K}[\mathcal{S}] = \mathsf{M}[\mathcal{S}]$; and since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$; so $\mathsf{K}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$. (c) By construction $\mathsf{K}[c] = \mathsf{M}[c]$; since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$; so $\mathsf{K}[c] = \iota(\mathsf{L}[c])$. (f) For simplicity consider a one-place function symbol h: by construction, $\mathsf{K}[h]\langle\iota(\mathsf{m})\rangle = (\mathsf{M}[h] \restriction \mathsf{U_K})\langle\iota(\mathsf{m})\rangle$; but $\iota(\mathsf{m}) \in \mathsf{U_K}$; so $(\mathsf{M}[h] \restriction \mathsf{U_K})\langle\iota(\mathsf{m})\rangle = \mathsf{M}[h]\langle\iota(\mathsf{m})\rangle$; and since $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$, $\mathsf{M}[h]\langle\iota(\mathsf{m})\rangle = \iota(\mathsf{L}[h]\langle\mathsf{m}\rangle)$; so $\mathsf{K}[h]\langle\iota(\mathsf{m})\rangle = \iota(\mathsf{L}[h]\langle\mathsf{m}\rangle)$. (r) For simplicity consider a one-place relation symbol $\mathcal{R}$: by [homework] $\iota(\mathsf{m}) \in \mathsf{K}[\mathcal{R}]$ iff $\mathsf{m} \in \mathsf{L}[\mathcal{R}]$.

(ii) Suppose there is some $\mathsf{K} \sqsubseteq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$. We need that $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$. Since $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$, ι is a 1:1 function from $\mathsf{U_L}$ onto $\mathsf{U_K}$ and since $\mathsf{K} \sqsubseteq \mathsf{M}$, $\mathsf{U_K} \subseteq \mathsf{U_M}$; so ι is a 1:1 function from $\mathsf{U_L}$ to $\mathsf{U_M}$. Now working on the conditions from EM: (s) Since $\mathsf{K} \sqsubseteq \mathsf{M}$, $\mathsf{M}[\mathcal{S}] = \mathsf{K}[\mathcal{S}]$; since $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$, $\mathsf{K}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$; so $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$. (c) Since $\mathsf{K} \sqsubseteq \mathsf{M}$, $\mathsf{M}[c] = \mathsf{K}[c]$; since $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$, $\mathsf{K}[c] = \iota(\mathsf{L}[c])$; so $\mathsf{M}[c] = \iota(\mathsf{L}[c])$. (f) For simplicity consider a one-place function symbol h: take an arbitrary $\mathsf{m} \in \mathsf{U_L}$; then $\iota(\mathsf{m}) \in \mathsf{U_K}$, and $\mathsf{M}[h]\langle\iota(\mathsf{m})\rangle = (\mathsf{M}[h] \restriction \mathsf{U_K})\langle\iota(\mathsf{m})\rangle$; since $\mathsf{K} \sqsubseteq \mathsf{M}$, $(\mathsf{M}[h] \restriction \mathsf{U_K})\langle\iota(\mathsf{m})\rangle = \mathsf{K}[h]\langle\iota(\mathsf{m})\rangle$; and since $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$, $\mathsf{K}[h]\langle\iota(\mathsf{m})\rangle = \iota(\mathsf{L}[h]\langle\mathsf{m}\rangle)$; so $\mathsf{M}[h]\langle\iota(\mathsf{m})\rangle = \iota(\mathsf{L}[h]\langle\mathsf{m}\rangle)$. (r) For simplicity consider a one-place relation symbol $\mathcal{R}$: by [homework] $\iota(\mathsf{m}) \in \mathsf{M}[\mathcal{R}]$ iff $\mathsf{m} \in \mathsf{L}[\mathcal{R}]$.

As for submodels themselves, it may be that $\mathsf{L} \underset{\sim}{\sqsubset} \mathsf{M}$ without $\mathsf{L} \equiv \mathsf{M}$. So in example (M), $\mathsf{L}[\exists x(x < a)] \neq \mathsf{T}$ but $\mathsf{M}[\exists x(x < a)] = \mathsf{T}$. But we may restrict the range of embeddings as for the restriction of submodels to elementary submodels.

EL For some language $\mathcal{L}$ with models L and M, there is an *ι-elementary embedding* of L into M ($\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$) iff $\mathsf{L} \overset{\iota}{\underset{\sim}{\sqsubset}} \mathsf{M}$ and for variable assignment d into $\mathsf{U_L}$ and h such that for all x, $\mathsf{h}(x) = \iota(\mathsf{d}[x])$, and any $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

If there is an ι such that $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$, then there is an *elementary embedding* of L into M ($\mathsf{L} \precsim \mathsf{M}$). Roughly, an elementary embedding is to an embedding, as an elementary submodel is to a submodel. Thus an elementary submodel adds a constraint on satisfaction to the conditions for a submodel; and an elementary embedding adds a constraint on satisfaction to the conditions for an embedding. To exhibit $\overset{\iota}{\precsim}$ in diagram (N) change the relation between K and M from $\sqsubseteq$ to $\preceq$. And we may obtain results for elementary embeddings like ones before.

*T11.21. $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$ iff there is a $\mathsf{K} \preceq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$.

Homework.

T11.22. If $\mathsf{L} \precsim \mathsf{M}$, then $\mathsf{L} \equiv \mathsf{M}$.

Homework.

Observe that we might have taken the notion of an embedding as fundamental and defined isomorphism and submodel in terms of it. L is ι-isomorphic to M just in case $\mathsf{L} \overset{\iota}{\sqsubsetapprox} \mathsf{M}$ and ι is a function from the universe of L onto the universe of M. L is a submodel of M just in case $\mathsf{L} \overset{\iota}{\sqsubsetapprox} \mathsf{M}$ and ι is the identity function $\iota(\mathsf{m}) = \mathsf{m}$ on a subset of the universe of M. And L is an elementary submodel of M just in case $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$ with ι the identity function on a subset of M. So *embedding* is the more general or flexible notion, and others appear as instances of it.

E11.14. (i) For P and N of example (K), explain why $\mathsf{P} \not\sqsubseteq \mathsf{N}$. (ii) Let $\mathsf{U_L} = \mathbb{P}$ and $\mathsf{U_M} = \mathbb{N}$ and the only symbol of the language be $<$ (with equality); let $\mathsf{M}[<]$ be standard, and $\mathsf{L}[<]$ be its restriction to $\mathbb{P}$; then explain why $\mathsf{L} \sqsubseteq \mathsf{M}$; explain why $\mathsf{L} \cong \mathsf{M}$; and show that $\mathsf{L} \not\preceq \mathsf{M}$. Hint: Consider the formula, $\forall y(x \neq y \rightarrow x < y)$.

E11.15. Complete the demonstration of T11.19 by completing the case for $\rightarrow$. You should set up the entire induction, but may defer parts to the text as the text defers to homework.

E11.16. Complete the demonstration of T11.20 by completing the cases for relation symbols.

E11.17. Suppose $\mathsf{L} \sqsubseteq \mathsf{M}$. (i) Let $\mathcal{P}$ be any formula without quantifiers; show that if d is an assignment into $\mathsf{U_L}$, then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$. (ii) Let $\mathcal{Q}$ be $\forall x_1 \ldots \forall x_n \mathcal{P}$ for $\mathcal{P}$ without quantifiers; show that if d is an assignment into $\mathsf{U_L}$, then if $\mathsf{M_d}[\mathcal{Q}] = \mathsf{S}$, $\mathsf{L_d}[\mathcal{Q}] = \mathsf{S}$. With T8.7 it follows that if such a $\mathcal{Q}$ is a sentence, then if $\mathsf{M}[\mathcal{Q}] = \mathsf{T}$, $\mathsf{L}[\mathcal{Q}] = \mathsf{T}$. Hint: These are arguments by induction; you will find T11.18 helpful.

*E11.18. Show T11.21 and T11.22. Hint: For the first, you will be able to focus questions by appeal to T11.20.

Basic Definitions for Model Theory

Let Σ be a set of formulas and $\mathfrak{M}$ a class of models. $|\mathfrak{M}|$ is the set of formulas true on each $\mathsf{M} \in \mathfrak{M}$; $\mathfrak{Md}(\Sigma)$ is the class of all models on which the members of Σ are true.

AX Σ is an *axiomatization* of $\mathfrak{M}$ just in case $\mathsf{M}[\Sigma] = \mathsf{T}$ iff $\mathsf{M} \in \mathfrak{M}$. $\mathfrak{M}$ is *axiomatizable* iff it is axiomatized by some Σ. $\mathfrak{M}$ is *finitely axiomatizable* iff it is axiomatized by some finite Σ.

RS Σ is *sound with respect to* $\mathfrak{M}$ just in case for any formula $\mathcal{P}$, if $\Sigma \vdash \mathcal{P}$ then $\mathcal{P} \in |\mathfrak{M}|$.

RC Σ is *complete with respect to* $\mathfrak{M}$ just in case for any formula $\mathcal{P}$, if $\mathcal{P} \in |\mathfrak{M}|$ then $\Sigma \vdash \mathcal{P}$.

SI Associate Σ with a class $\mathfrak{I}$ of *intended* models; then Σ is *$\tilde{s}$ound* iff it is sound with respect to $\mathfrak{I}$.

NC Σ is (*negation*) *$\tilde{c}$omplete* iff $\Sigma \vdash \mathcal{P}$ or $\Sigma \vdash \sim\mathcal{P}$ for every sentence $\mathcal{P}$.

CA Σ is *categorical* iff it characterizes models up to isomorphism, iff for any $\mathsf{L}, \mathsf{M} \in \mathfrak{Md}(\Sigma)$, $\mathsf{L} \cong \mathsf{M}$.

($\equiv$) EE For some language $\mathcal{L}$, models L and M are *elementarily equivalent* ($\mathsf{L} \equiv \mathsf{M}$) iff for any formula $\mathcal{P}$, $\mathsf{L}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{M}[\mathcal{P}] = \mathsf{T}$.

($\cong$) IS For some language $\mathcal{L}$ with models L and M, L is *ι-isomorphic* to M ($\mathsf{L} \stackrel{\iota}{\cong} \mathsf{M}$) iff ι is a 1:1 function from the universe of L onto the universe of M and,

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$.

(f) For a function symbol h^n, $\mathsf{M}[h^n]\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle = \iota(\mathsf{L}[h^n]\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle)$.

(r) For a relation symbol $\mathcal{R}^n$, $\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle \in \mathsf{M}(\mathcal{R}^n)$ iff $\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle \in \mathsf{L}[\mathcal{R}^n]$.

If there is some ι such that $\mathsf{L} \stackrel{\iota}{\cong} \mathsf{M}$, then L is *isomorphic* to M ($\mathsf{L} \cong \mathsf{M}$).

($\sqsubseteq$) SM For some language $\mathcal{L}$ with models L and M, L is a *submodel* of M ($\mathsf{L} \sqsubseteq \mathsf{M}$) iff,

(u) $\mathsf{U_L} \subseteq \mathsf{U_M}$, and $\mathsf{U_L}$ is closed under the constants and functions of M.

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}(c) = \mathsf{L}(c)$.

(f) For a function symbol h^n, $\mathsf{M}[h^n] \upharpoonright \mathsf{U_L} = \mathsf{L}[h^n]$.

(r) For a relation symbol $\mathcal{R}^n$, $\mathsf{M}[\mathcal{R}^n] \upharpoonright \mathsf{U_L} = \mathsf{L}[\mathcal{R}^n]$.

($\preceq$) ES For some language $\mathcal{L}$ with models L and M, L is an *elementary submodel* of M ($\mathsf{L} \preceq \mathsf{M}$) iff $\mathsf{L} \sqsubseteq \mathsf{M}$ and for any formula $\mathcal{P}$ of $\mathcal{L}$ and variable assignment d into $\mathsf{U_L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

Basic Definitions for Model Theory (cont.)

($\precapprox$) EM For some language $\mathscr{L}$ with models L and M, there is an *ι-embedding* of L into M ($\mathsf{L} \overset{\iota}{\precapprox} \mathsf{M}$) iff ι is a 1:1 function from U_L to U_M and,

(s) For a sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{L}[\mathcal{S}]$.

(c) For a constant c, $\mathsf{M}[c] = \iota(\mathsf{L}[c])$.

(f) For a function symbol h^n, $\mathsf{M}[h^n]\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle = \iota(\mathsf{L}[h^n]\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle)$.

(r) For a relation symbol $\mathcal{R}^n$, $\langle\iota(\mathsf{m_a})\ldots\iota(\mathsf{m_b})\rangle \in \mathsf{M}(\mathcal{R}^n)$ iff $\langle\mathsf{m_a}\ldots\mathsf{m_b}\rangle \in \mathsf{L}[\mathcal{R}^n]$.

If there is some ι such that $\mathsf{L} \overset{\iota}{\precapprox} \mathsf{M}$, then there is an *embedding* of L into M ($\mathsf{L} \precapprox \mathsf{M}$).

($\precsim$) EL For some language $\mathscr{L}$ with models L and M, there is an *ι-elementary embedding* of L into M ($\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$) iff $\mathsf{L} \overset{\iota}{\precapprox} \mathsf{M}$ and for variable assignment d into U_L and h such that for all x, $\mathsf{h}(x) = \iota(\mathsf{d}[x])$, and any $\mathcal{P}$ in $\mathscr{L}$, $\mathsf{M_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{L_d}[\mathcal{P}] = \mathsf{S}$.

If there is an ι such that $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$, then there is an *elementary embedding* of L into M ($\mathsf{L} \precsim \mathsf{M}$).

11.4.2 Compactness

We turn now to some applications that build upon our basic concepts. We begin with the *compactness* theorem. This apparently simple result has many interesting consequences.

The Compactness Theorem

For Σ a set formulas, compactness connects models for Σ and models for its finite subsets. Say a set Σ of formulas is *satisfiable* iff it has a model. Σ is *finitely satisfiable* iff every finite subset of it has a model.

T11.23. A set of formulas Σ is satisfiable iff it is finitely satisfiable. *Compactness*.

(i) Suppose Σ is satisfiable but not finitely satisfiable. Then there is some M such that $\mathsf{M}[\Sigma] = \mathsf{T}$; but there is a finite $\Delta \subseteq \Sigma$ such that for any L, $\mathsf{L}[\Delta] \neq \mathsf{T}$; so $\mathsf{M}[\Delta] \neq \mathsf{T}$; so there is a formula $\mathcal{P} \in \Delta$ such that $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; but since $\Delta \subseteq \Sigma$, $\mathcal{P} \in \Sigma$; so $\mathsf{M}[\Sigma] \neq \mathsf{T}$. This is impossible; reject the assumption: if Σ is satisfiable, then it is finitely satisfiable.

(ii) Suppose Σ is finitely satisfiable but not satisfiable. By T10.16, if Σ is consistent, then it has a model M. But since Σ is not satisfiable, it has no model; so it is not consistent; so there is some formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$. Consider derivations of these results, and the set Δ of premises of these derivations; since derivations are finite, Δ is finite; and since Δ includes all the premises for the derivations, $\Delta \vdash \mathcal{A}$ and $\Delta \vdash \sim\mathcal{A}$; so by s̊oundness, $\Delta \vDash \mathcal{A}$ and $\Delta \vDash \sim\mathcal{A}$. But since Σ is finitely satisfiable, there must be some model D such that $\mathsf{D}[\Delta] = \mathsf{T}$;

then by QV, $\mathsf{D}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{D}[\sim\mathcal{A}] = \mathsf{T}$; and from the latter by T8.8, $\mathsf{D}[\mathcal{A}] \neq \mathsf{T}$. This is impossible; reject the assumption: if Σ is finitely satisfiable, then it is satisfiable.

Perhaps part (i) is obvious: if there is a model for the whole of Σ, that very model is one for its parts. On its face, compactness is a semantic result that does not have anything to do with derivation systems and so the derivations to which we appeal at (ii)—and there are alternate demonstrations of compactness that do not appeal to derivations. However, given what we have already done, this demonstration is close at hand, and lets us turn directly to applications.

Infinite Domains

In Chapter 5 we learned to say 'at most' and 'at least' in simple cases. So, for example, $\forall y \forall x_1 \forall x_2(y = x_1 \vee y = x_2)$ is true iff there are at most two things. And $\exists x_1 \exists x_2(x_1 \neq x_2)$ is true iff there are at least two things. Let us generalize the method to arbitrary finite numbers. For some formulas $\mathcal{A}_1 \ldots \mathcal{A}_n$, let $\bigvee_{1 \leq i \leq n} \mathcal{A}_i$ be the disjunction of the $\mathcal{A}_i$s and $\bigwedge_{1 \leq i \leq n} \mathcal{A}_i$ be their conjunction. Then for any $n \geq 1$,

$$\mathcal{M}_n = \forall y \forall x_1 \ldots \forall x_n \bigvee_{1 \leq i \leq n} y = x_i$$

is true just in case there are at most n things. So, for example,

$$\mathcal{M}_4 = \forall y \forall x_1 \forall x_2 \forall x_3 \forall x_4(y = x_1 \vee y = x_2 \vee y = x_3 \vee y = x_4)$$

is true iff there are at most four things. With a modification to the simple indexing from above, we get 'at least'. Thus for any $n \geq 1$,

$$\mathcal{L}_n = \exists x_1 \ldots \exists x_n \left[x_1 = x_1 \;\wedge \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j \right]$$

is true iff there are at least n things. Starting with $i = 1$ include the inequality for each j greater than it up to n. Then do the same for each $i < n$. (The first conjunct in the square bracket merely guarantees that the result is a formula when $n = 1$ and there are no i, j such that $1 \leq i < j \leq n$ so that the extended conjunction includes no formulas.) Thus for example,

$$\mathcal{L}_4 = \exists x_1 \exists x_2 \exists x_3 \exists x_4[x_1 = x_1 \wedge (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4)]$$

is true iff there are at least four things. And, of course $\mathcal{M}_n \wedge \mathcal{L}_n$ is true iff there are exactly n things. Manipulating these sentences becomes impractical quickly! Nonetheless in a straightforward sense our language has the *capacity* to express these properties.

In this way, given some quantificational language $\mathcal{L}$ with equality, we can (finitely) axiomatize the class of all models whose universe has at most n elements, the class of all models whose universe has at least n elements, and the class of all models whose universe has exactly n elements. What about $\mathfrak{I}$ the class of all models with an infinite domain? This class may be axiomatized as well. Let $\Gamma = \{\mathcal{L}_n \mid n \geq 1\}$. The members of Γ are $\mathcal{L}_1, \mathcal{L}_2, \ldots$ constructed as above. So according to Γ, there is at least one thing; there are at least two things; there are at least three things; and so forth. Every member of Γ is true on an infinite domain. But on a domain with n members, $\mathcal{L}_{n+1}$ is sure to be false. So Γ is true on all and only interpretations with an infinite domain and $\mathfrak{Md}(\Gamma) = \mathfrak{I}$.

In this case, $\mathfrak{I}$ is not *finitely* axiomatized. Is there a finite axiomatization of the class of models with an infinite domain? There are, of course, finite sets true only on infinite domains. Let the language be $\mathcal{L}_{NT}$ and the members of Δ the axioms of Q; then Δ has finitely many members; and any $\mathsf{D} \in \mathfrak{Md}(\Delta)$ has an infinite domain. But the members of $\mathfrak{Md}(\Delta)$ are not *all* the models with an infinite domain. So, for example, $\forall x(x \not< \emptyset)$ is a theorem of Q (from T6.55) and so true on any member of $\mathfrak{Md}(\Delta)$. But a standard interpretation Z of $\mathcal{L}_{NT}$ on the set $\mathbb{Z}$ of all integers has an infinite domain and $\mathsf{Z}[\forall x(x \not< \emptyset)] \neq \mathsf{T}$; so $\mathsf{Z} \in \mathfrak{I}$ but $\mathsf{Z} \notin \mathfrak{Md}(\Delta)$. Still, we might wonder if there is some other set of formulas that is a finite axiomatization of $\mathfrak{I}$. We can use the compactness theorem to show that there is not.

T11.24. For $\mathfrak{I}$ the class of all models with an infinite domain, there is no finite Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{I}$.

Suppose otherwise, that for some language $\mathcal{L}$, $\mathfrak{I}$ is the class of all models with an infinite domain, and for Σ with finitely many members, $\mathfrak{Md}(\Sigma) = \mathfrak{I}$. Let $\mathcal{A}$ be the conjunction of the members of Σ and $\mathcal{A}^u$ its universal closure; with T7.6, $\mathcal{A}^u$ is true on just the same interpretations as Σ and so just on models with an infinite domain. Consider the axiomatization of $\mathfrak{I}$ from above, $\Gamma = \{\mathcal{L}_n | n \geq 1\}$. Then any finite $\Delta \subseteq \{\sim\mathcal{A}^u\} \cup \Gamma$ is satisfiable: Δ may have as members $\sim\mathcal{A}^u$ and finitely many sentences $\mathcal{L}_a \ldots \mathcal{L}_b$; let n be whichever is greater of 1 and the maximum subscript from $\mathcal{L}_a \ldots \mathcal{L}_b$; all of $\mathcal{L}_a \ldots \mathcal{L}_b$ are satisfied on a universe with n members; but by hypothesis $\mathcal{A}^u$ is satisfied on all and only interpretations with an infinite domain; so $\sim\mathcal{A}^u$ is satisfied on all and only interpretations with a finite domain; so $\sim\mathcal{A}^u$ is satisfied on a universe with n members; so all the members of Δ are satisfied on a universe with n members; so a finite $\Delta \subseteq \{\sim\mathcal{A}^u\} \cup \Gamma$ is satisfiable; and since Δ is arbitrary, by compactness $\{\sim\mathcal{A}^u\} \cup \Gamma$ is satisfiable. But this is impossible; $\sim\mathcal{A}^u$ is satisfied only on universes with a finite domain and Γ only on universes with an infinite domain; so no interpretation satisfies $\{\sim\mathcal{A}^u\} \cup \Gamma$. Reject the assumption: where $\mathfrak{I}$ is the class of all models with an infinite domain, there is no finite Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{I}$.

This first application of compactness may be less than earth-shattering. It is interesting, though, just to have seen how compactness applies to establish the universal claim about axiomatizations.

Finite Domains

Now for a language $\mathcal{L}$, let $\mathfrak{F}$ be the class of interpretations with a finite domain, and consider the question whether there is some Γ to axiomatize it. We have seen axiomatizations of models with at most n members, and of models with exactly n members. But these are not axiomatizations of the class $\mathfrak{F}$ of *all* interpretations with a finite domain. If some Σ were a finite axiomatization of the class of all models with an infinite domain, then $\sim\mathcal{A}^u$ constructed as above would axiomatize $\mathfrak{F}$; but we have just seen that there is no such axiomatization. Notice also that $\{\mathcal{M}_n \mid n \geq 1\}$ will not do: given $\mathcal{M}_1$ as an element, this set is satisfied just on interpretations with a single member! A set of formulas is true under conditions something like a big conjunction; thus our set Γ including each $\mathcal{L}_n$ says that the universe has at least 1 one member, and it has at least 2 members, and.... To say that the universe is finite, however, we would require something like a big *disjunction:* the universe has at most one member, or it has at most two members, or.... So the set of all the $\mathcal{M}_n$s is not right, and neither is any one formula of our language long enough to be the disjunction of them all (but see the box on the following page). Again, however, it is natural to wonder whether there is some other way to axiomatize $\mathfrak{F}$ the class of all interpretations with a finite domain.

That there is no axiomatization for $\mathfrak{F}$ follows as a corollary to the following theorem.

T11.25. If Σ has arbitrarily large finite models, then Σ has an infinite model.

For some language $\mathcal{L}$, suppose Σ has arbitrarily large finite models and consider again $\Gamma = \{\mathcal{L}_n \mid n \geq 1\}$. Let Δ be an arbitrary finite subset of $\Sigma \cup \Gamma$; then Δ may have as members some elements of Σ and finitely finitely many sentences $\mathcal{L}_a \ldots \mathcal{L}_b$; let n be whichever is greater of 1 and the maximum of subscripts from $\mathcal{L}_a \ldots \mathcal{L}_b$; then all of $\mathcal{L}_a \ldots \mathcal{L}_b$ are satisfied on a universe with n members; and since Σ is satisfied on arbitrarily large finite models, Σ and so each member of Σ in Δ is satisfied on a universe with n members as well; so Δ is satisfiable, and by compactness $\Sigma \cup \Gamma$ is satisfiable. But Γ is satisfied only on infinite domains; so $\Sigma \cup \Gamma$ is satisfied only on an infinite domain; and any model that satisfies $\Sigma \cup \Gamma$ satisfies Σ as well; so Σ has an infinite model.

Corollary: The class $\mathfrak{F}$ of all finite models is not axiomatizable. Suppose otherwise; that for some Σ, $\mathfrak{Md}(\Sigma) = \mathfrak{F}$; then Σ is satisfied on arbitrarily large finite models; so by the main result, Σ has an infinite model. So $\mathfrak{Md}(\Sigma) \neq \mathfrak{F}$. This is impossible; reject the assumption: there is no Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{F}$.

By our discussion of infinite and finite domains, we have answered the second and third questions posed in the introduction to section 11.4: There is an axiomatization

Extensions of Classical Logic

Difficulties about axiomatizing the class of models with a finite domain (and other cases from this section) alter in context of *infinitary* and *second-order* logics.

- Though our languages have infinitely many symbols, *formulas* are always finitely long. *Infinitary logic* permits infinitely long formulas. On this account, where Π is a (possibly infinite) set of formulas, $\bigwedge \Pi$ and $\bigvee \Pi$ are formulas, and if Ξ is a set of variables and $\mathcal{P}$ a formula, $\forall \Xi \mathcal{P}$ and $\exists \Xi \mathcal{P}$ are formulas. Intuitively, $\bigwedge \Pi$ is satisfied iff each $\mathcal{P} \in \Pi$ is satisfied, $\bigvee \Pi$ is satisfied iff some $\mathcal{P} \in \Pi$ is satisfied, $\forall \Xi \mathcal{P}$ is satisfied iff $\mathcal{P}$ is satisfied for every assignment of objects to the variables in Ξ, and $\exists \Xi \mathcal{P}$ is satisfied so long as $\mathcal{P}$ is satisfied for some assignment to the variables in Ξ. Then $\bigvee\{\mathcal{M}_n \,|\, n \geq 1\}$ is satisfied only on interpretations with a finite domain.

- Our logic is *first-order* insofar as quantifiers range just over individuals of the universe. *Second-order logic* permits quantification not only over individuals of the universe but over *relations* and *functions* as well. Then we might take advantage of a feature of infinite sets—that there is a 1:1 map from an infinite set to a proper subset of it. Thus for function variable f, $\exists f[\forall x \forall y(fx = fy \rightarrow x = y) \wedge \exists y \forall x(fx \neq y)]$ is true just on infinite domains. The first conjunct requires that f be 1:1, and the second that f not be onto. Then the negation of this sentence is true only on finite domains.

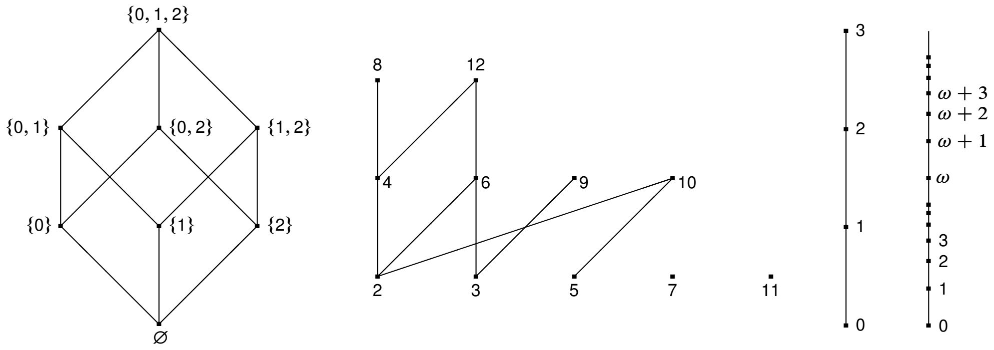
Unfortunately, both infinitary and second-order logics are $\mathring{\text{i}}$ncomplete. In this way there is a trade-off between the $\mathring{\text{c}}$ompleteness of our classical system, and the expressive power of infinitary and second-order languages.

For second-order logic see Shapiro, *Foundations Without Foundationalism*, and Manzano, *Extensions of First Order Logic*. Discussions of infinitary logic presuppose significant background in set theory, though Bell, "Infinitary Logic" and Nadel, "$\mathcal{L}_{\omega_1\omega}$ and Admissible Fragments" are a reasonable place to start.

of the class of all models with an infinite universe, and there is no axiomatization of the class of all models with a finite universe.

Orderings

Consider a language $\mathcal{L}$ with two-place relation symbol $\triangleleft$ and model M such that the interpretation of $\triangleleft$ is a relation $\triangleleft$. Then $\triangleleft$ is a *partial order* of (or on) U just in case $\triangleleft$ is transitive and irreflexive: if a $\triangleleft$ b and b $\triangleleft$ c then a $\triangleleft$ c, and it is never the case that a $\triangleleft$ a. In case $\triangleleft$ is a partial order, we say U (and derivatively the model) is *partially ordered*. Many different relations are partial orders. Thus (a), (b), (c), and (d) below each depict partial orders.



(a) (b) (c) (d)

In each case, $\mathsf{a} \lhd \mathsf{b}$ when there is a path "up" the lines from a to b. In (a) $\lhd$ is the proper subset relation on the subsets of $\{0, 1, 2\}$. (b) is a portion of a diagram that would extend infinitely vertically and to the right where $\lhd$ is the relation of proper divisibility on the natural numbers—where $\mathsf{a} \lhd \mathsf{b}$ just in case $\mathsf{a} \neq 1$ and $\mathsf{a} \neq \mathsf{b}$ and a evenly divides b. (c) applies the usual $<$ relation to $\{0, 1, 2, 3\}$. (d) depicts the natural numbers with their usual $<$ relation (in the picture, each step a quarter of the remaining distance to ω), and after all of them a copy of the natural numbers again, where every member of the first copy is less than all the members of the second (a two-place relation is a set of pairs on some domain, and nothing prevents objects and pairs so-arranged). Similarly the sets of all natural numbers, all integers, all rational numbers, and all real numbers are partially ordered by their usual $<$ relation. The class $\mathfrak{P}$ of all partial orderings is axiomatized in the natural way by,

O1 $\forall x \forall y \forall z[(x \lhd y \land y \lhd z) \to x \lhd z]$

O2 $\forall x(x \not\lhd x)$

The interpretations on which O1 and O2 are true are ones that assign to $\lhd$ some partial order $\lhd$.

A model is a *linear ordering* when it is a partial ordering and for all $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, $\mathsf{a} \lhd \mathsf{b}$ or $\mathsf{a} = \mathsf{b}$ or $\mathsf{b} \lhd \mathsf{a}$. In a linear ordering, the members of U are sorted into a "line." Examples (c) and (d) above are linear orderings, though (a) and (b) are not. Standard linear orderings are the sets of all natural numbers, all integers, all rational numbers, and all real numbers with their usual $<$ relation. Linear orderings are axiomatized by adding to O1 and O2,

O3 $\forall x \forall y[x \lhd y \lor x = y \lor y \lhd x]$

Interpretations that satisfy O1–O3 are ones that assign to $\lhd$ some linear order $\lhd$.

A model is a *well-ordering* when it is a linear ordering and every nonempty $\mathsf{S} \subseteq \mathsf{U}$ has a least member—an $\mathsf{a} \in \mathsf{S}$ such that for every $\mathsf{b} \in \mathsf{S}$ if $\mathsf{b} \neq \mathsf{a}$ then $\mathsf{a} \lhd \mathsf{b}$. Example (c) is of a well-order. The natural numbers with the usual $<$ relation are a standard example. Example (d) is a well-order too. However, although they are linear orders, the sets of all integers, all rational numbers, and all real numbers with their usual $<$ relation are not well-orders. To see this, it is enough to recognize that the whole sets, which continue infinitely in the negative direction, are subsets without a least member. But neither are the set of all rationals ≥ 0 nor the set of all reals ≥ 0 with their usual $<$ relation well-orders. In these cases the collection of all a such that $0 < \mathsf{a} \leq 1$, for example, is a subset such that every member has one less than it, and so without a least member.[6]

Notice that the well-ordering of the natural numbers is presupposed by our justification of mathematical induction at the start of Chapter 8 (page 362): Assuming some members of a series which do not "fall," we moved to the existence of a *least* member which does not fall, and from that to contradiction—and so to the conclusion that all the members fall. And our reasoning by mathematical induction has been restricted to series (well-)ordered by the natural numbers. So well-orderings are important. But we can use compactness to see that the class $\mathfrak{W}$ of all well-orderings is not axiomatizable.

T11.26. The class $\mathfrak{W}$ of all well-orderings is not axiomatizable.

Suppose otherwise, that for language $\mathscr{L}$ there is a Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{W}$; then $\mathsf{M}[\Sigma] = \mathsf{T}$ just in case its assignment to some symbol $\triangleleft$ is a well-order. Extend $\mathscr{L}$ to an $\mathscr{L}'$ by the addition of infinitely many constants $c_0, c_1, c_2, \ldots$ and let Σ' be the same as Σ except in the new language $\mathscr{L}'$. For some M and M' like M except that it makes assignments to the new constants, by T10.14, $\mathsf{M}[\Sigma] = \mathsf{T}$ iff $\mathsf{M}'[\Sigma'] = \mathsf{T}$; in particular, $\mathsf{M}'[\Sigma'] = \mathsf{T}$ iff the assignment to $\triangleleft$ is a well-order.

Let $\mathcal{A}_n = c_{n+1} \triangleleft c_n$, and set $\Gamma' = \{\mathcal{A}_n \mid 0 \leq n\}$; so, listing the members from right to left, $\Gamma' = \{\ldots c_3 \triangleleft c_2, c_2 \triangleleft c_1, c_1 \triangleleft c_0\}$. Consider an arbitrary finite $\Delta' \subseteq \Sigma' \cup \Gamma'$. Δ' may have as elements some members of Σ' together with finitely many of the sentences $c_j \triangleleft c_i$. Where m is whichever is greater of 0 and the maximum of these subscripts, consider some objects $\mathsf{c}_0 \ldots \mathsf{c}_m$ and a model D' with a universe consisting of those objects and,

$$\mathsf{D}'[c_i] = \mathsf{c}_i \text{ for } 0 \leq i \leq m, \text{ and otherwise } \mathsf{D}'[c_i] = \mathsf{c}_0$$

$$\mathsf{D}'[\triangleleft] = \{\langle \mathsf{c}_b, \mathsf{c}_a\rangle \mid \mathsf{c}_b, \mathsf{c}_a \in \mathsf{U}_\mathsf{D} \text{ and } b > a\}$$

Observe that $\mathsf{c}_b \lhd \mathsf{c}_a$ not only when $c_b \triangleleft c_a \in \Gamma'$, but for every c_b such that $b > a$. Then the assignment to $\triangleleft$ is a well-order; so Σ' and all the elements of Σ' in Δ' are satisfied on D'; additionally, each $c_j \triangleleft c_i \in \Delta'$ is satisfied on D'.

[6] In the usual ZFC set theory (Zermelo-Fraenkel set theory with the axiom of choice) it is a theorem, equivalent to the axiom of choice, that any set can be well-ordered. So the sets of all integers, all rationals, and all reals have well-orderings—only this ordering is not the usual $<$ relation. The Chapter 2 countability reference exhibits mechanisms sufficient for well-ordering of the integers and of the rationals (challenge: show how). It is less clear *how* the reals are well-ordered, though the theorem says that such an ordering must exist.

So Δ' is satisfied on D'; so Δ' is satisfiable, and by compactness, $\Sigma' \cup \Gamma'$ is satisfiable. But this is impossible: Σ' is satisfied only on well-orderings; but any well-ordering is a linear ordering; and given a linear ordering, all the members of Γ' are satisfied only when the interpretation of $\triangleleft$ has no least member, and so when the interpretation is not a well-ordering. Reject the assumption: there is no Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{W}$.

An interpretation M' of Γ' with the assignment of $\triangleleft$ to a linear order $\lhd$ need not assign c_i and c_{i+1} to *adjacent* members of the series. Nonetheless, for any m supposed to be the least member of $\mathsf{U}_{\mathsf{M}'}$, take the greatest subscript i such that $\mathsf{m} \trianglelefteq \mathsf{M}'[c_i]$; then it is not the case that $\mathsf{m} \trianglelefteq \mathsf{M}'[c_{i+1}]$; so $\mathsf{M}'[c_{i+1}] \lhd \mathsf{m}$.

This general strategy of introducing new constants is one that we saw in the demonstration of $\mathring{c}$ompleteness, and one that we shall see again.

Number Theory

Consider again the standard interpretation N for $\mathcal{L}_{NT}$ and let $\mathfrak{N}$ be the class of all models isomorphic to N. Ideally there would be some categorical Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{N}$. If $\mathfrak{Mod}(\Sigma) = \mathfrak{N}$ then by T11.9 Σ is sound with respect to $\mathfrak{N}$, and since models isomorphic to N make all the same formulas true, $\tilde{s}$ound on the intended model N. And if Σ is categorical then by the corollary to T11.16, Σ is $\tilde{c}$omplete. Unfortunately, we can show that there is no Σ to axiomatize $\mathfrak{N}$.

T11.27. For $\mathfrak{N}$ the class of models isomorphic to N, there is no Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{N}$.

Suppose otherwise, that some Σ is such that $\mathfrak{Mod}(\Sigma) = \mathfrak{N}$. Extend the language $\mathcal{L}_{NT}$ to an $\mathcal{L}'_{NT}$ by the addition of a single constant c and let Σ' be the same as Σ except in the new language $\mathcal{L}'_{NT}$. For $\mathsf{M} \in \mathfrak{N}$, let M' be like M except that it makes some assignment to c. Then by T10.14, M satisfies Σ iff M' satisfies Σ'.

Let $\overline{\mathsf{n}}$ be as in Chapter 8 (page 391) and set $\Gamma' = \{c \neq \overline{\mathsf{n}} \mid \mathsf{n} \in \mathbb{N}\}$; so $\Gamma' = \{c \neq \emptyset, c \neq S\emptyset, c \neq SS\emptyset, c \neq SSS\emptyset, \ldots\}$. Consider an arbitrary finite $\Delta' \subseteq \Sigma' \cup \Gamma'$. Δ' may have as elements some members of Σ' together with finitely many of the sentences $c \neq \overline{\mathsf{n}}$; for m whichever is larger of 0 and the greatest n such that $c \neq \overline{\mathsf{n}} \in \Delta'$, and the standard interpretation N, let N' be the same as N except that $\mathsf{N}'[c] = \mathsf{m} + 1$. All the members of Σ' in Δ' remain satisfied on N'; additionally, since c is assigned to an object other than any n such that $c \neq \overline{\mathsf{n}} \in \Delta'$, each $c \neq \overline{\mathsf{n}} \in \Delta'$ is satisfied on N'; so Δ' is satisfiable, and by compactness, $\Sigma' \cup \Gamma'$ is satisfied by some model K'.

Now for K like K' except without the assignment to c, both $\mathsf{K} \in \mathfrak{Mod}(\Sigma)$ and $\mathsf{N} \not\approx \mathsf{K}$. The first is easy: K' satisfies Σ'; so K satisfies Σ; so $\mathsf{K} \in \mathfrak{Mod}(\Sigma)$.

But N and K are not isomorphic: For any $\mathsf{a} \in \mathbb{N}$, $\mathsf{K}'[c \neq \overline{\mathsf{a}}] = \mathsf{T}$; so for arbitrary h, $\mathsf{K}'_{\mathsf{h}}[c \neq \overline{\mathsf{a}}] = \mathsf{S}$; let $\mathsf{K}'_{\mathsf{h}}[c] = \mathsf{c}$ for some $\mathsf{c} \in \mathsf{U}_{\mathsf{K}'}$, and for any $\mathsf{a} \in \mathbb{N}$, $\mathsf{K}'_{\mathsf{h}}[\overline{\mathsf{a}}] = \underset{\sim}{\mathsf{a}}$

for $\underset{\sim}{a} \in U_{K'}$; so by SF($\sim$) and SF(r), $\langle c, \underset{\sim}{a}\rangle \notin K'[=]$; so for any $a \in \mathbb{N}$ and $\overline{a}$, $c \neq \underset{\sim}{a}$ (notice that we cannot simply identify a and $\underset{\sim}{a}$, since we are not given that N and K assign the same object to $\overline{a}$).

Suppose $N \cong K$. Then for some ι, $N \overset{\iota}{\cong} K$; so for ι a 1:1 map from U_N onto U_K, there is some $a \in U_N$ such that $\iota(a) = c$. Consider a d into U_N such that $d[y] = a$, and an h into U_K such that for each x, $h(x) = \iota(d[x])$; then $h[y] = \iota(d[y]) = \iota(a) = c$. Since $N \cong K$, by T11.13 for any $\mathscr{P}$, $N_d[\mathscr{P}] = S$ iff $K_h[\mathscr{P}] = S$; so $N_d[y = \overline{a}] = S$ iff $K_h[y = \overline{a}] = S$; since $N_d[y] = a$ and $N_d[\overline{a}] = a$, by SF(r), $N_d[y = \overline{a}] = S$; so $K_h[y = \overline{a}] = S$; but $K_h[y] = c$ and $K_h[\overline{a}] = \underset{\sim}{a}$; so by SF(r), $\langle c, \underset{\sim}{a}\rangle \in K[=]$; so $c = \underset{\sim}{a}$. This is impossible; reject the assumption: $N \ncong K$, and there is no Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{N}$.

Since no Σ axiomatizes $\mathfrak{N}$, even $|\mathfrak{N}|$ does not axiomatize $\mathfrak{N}$; and no formulas are sufficient to "pin down" models up to isomorphism with N. This answers the fourth question posed in the introduction to section 11.4: There is no axiomatization of the class of all models isomorphic to N. So we have answers to all but the last.

A model of $|\mathfrak{N}|$ that is not isomorphic to N is a *nonstandard* model of arithmetic. We have shown that there are nonstandard models. It is worth pausing to think about what such models are like. Consider our model K in the case where $\Sigma = |\mathfrak{N}|$. K retains in its universe all the same objects as are in $U_{K'}$; so it retains the object c distinct from any $\underset{\sim}{a}$. But K requires much more than a single c distinct from each $\underset{\sim}{a}$ (for E7.19 we found a nonstandard model of the axioms of Q by adding just single object to the natural numbers; but, and this was the point, that was not a model for *all* the members of $|\mathfrak{N}|$). Let assignments of K to $\emptyset$, S, $+$, $\times$, and $<$ be some object $\underset{\sim}{o}$, functions s, $\oplus$, $\otimes$, and relation $\lhd$.[7] Then $\underset{\sim}{a}$, the assignment to $\overline{a}$, is $s \ldots s(\underset{\sim}{o})$ with a repetitions of s. Observe that the following are all true on N.

a. $\forall x \forall y \forall z[(x < y \land y < z) \rightarrow x < z]$

b. $\forall x(x \not< x)$

c. $\forall x \forall y(x < y \lor x = y \lor y < x)$

d. $\forall x(x \neq \emptyset \rightarrow \emptyset < x)$

e. $\forall x(x \neq \emptyset \rightarrow \exists y(Sy = x))$

f. $\forall x \forall y(x < y \rightarrow Sx \leq y)$

g. $\forall x \forall y(x < Sy \rightarrow x \leq y)$

[7] In $\mathscr{L}_{NT}$, $x < y$ abbreviates $\exists z(Sz + x = y)$. So $\lhd$ is not an assignment to any symbol of the language, but rather the set of pairs $\langle a, b\rangle$ such that $M_{d(x|a,y|b)}[\exists z(Sz + x = y] = S$—and so is fixed by the assignments to S and $+$.

h. $\forall x[(x + \emptyset) = x]$

i. $\forall x \forall y[(x + Sy) = S(x + y)]$

Since they are true on N, and K models all the formulas true on N, they are true on K (one way to see that they are true on N is to recognize that they are theorems of Q or PA). From (a), (b), (c), $\lhd$ is a linear order on $\mathsf{U_K}$. From (d), for any $\mathsf{m} \in \mathsf{U_K}$ other than $\underset{\sim}{\mathsf{0}}$, $\underset{\sim}{\mathsf{0}} \lhd \mathsf{m}$.

Let $\mathsf{m} \simeq \mathsf{n}$ just in case there is some $\underset{\sim}{\mathsf{a}}$ such that $\mathsf{m} \oplus \underset{\sim}{\mathsf{a}} = \mathsf{n}$ or $\mathsf{n} \oplus \underset{\sim}{\mathsf{a}} = \mathsf{m}$. From (h) and (i) $\mathsf{m} \simeq \mathsf{n}$ just in case you can get from one to the other by finitely many applications of s. Then $\simeq$ is reflexive, symmetric, and transitive, and so an equivalence relation. Let $[\mathsf{m}] = \{\mathsf{z} \mid \mathsf{z} \simeq \mathsf{m}\}$. Then $[\mathsf{m}]$ is an equivalence class and, as in Chapter 10, satisfies self-membership, uniqueness, and equality. Since c is distinct from each $\underset{\sim}{\mathsf{a}}$ it is not the case that applying s to any $\underset{\sim}{\mathsf{a}}$ results in c; so there are at least two such classes, $[\underset{\sim}{\mathsf{0}}]$ and $[\mathsf{c}]$. The first has members $\underset{\sim}{\mathsf{0}}, \underset{\sim}{\mathsf{1}} \ldots$. But every object has a successor; so there are objects $\mathsf{c}_{+1} = \mathsf{s(c)}$, $\mathsf{c}_{+2} = \mathsf{s}(\mathsf{c}_{+1})$, and so forth; and from (e) objects other than $\underset{\sim}{\mathsf{0}}$ are successors; so there is a c_{-1} such that $\mathsf{s}(\mathsf{c}_{-1}) = \mathsf{c}$, a c_{-2} such that $\mathsf{s}(\mathsf{c}_{-2}) = \mathsf{c}_{-1}$, and so forth. So $[\mathsf{c}]$ includes objects $\ldots, \mathsf{c}_{-2}, \mathsf{c}_{-1}, \mathsf{c}, \mathsf{c}_{+1}, \mathsf{c}_{+2}, \ldots$. Insofar as $[\underset{\sim}{\mathsf{0}}]$ "looks" like the natural numbers (discrete members with a beginning and no end) it is an $\mathbb{N}$-*chain*, and insofar as $[\mathsf{c}]$ looks like the integers (discrete members extending in both directions), it is a $\mathbb{Z}$-*chain*.

And we can see that all the members of one chain are less than all the members of another. Suppose $[\mathsf{a}] \neq [\mathsf{b}]$ and $\mathsf{a} \lhd \mathsf{b}$; let $\mathsf{x} \in [\mathsf{a}]$ and $\mathsf{y} \in [\mathsf{b}]$; we show $\mathsf{x} \lhd \mathsf{y}$. First, $\mathsf{x} \lhd \mathsf{b}$: Since $\lhd$ is a linear order, $\mathsf{x} \lhd \mathsf{a}$ or $\mathsf{x} = \mathsf{a}$ or $\mathsf{a} \lhd \mathsf{x}$. If $\mathsf{x} = \mathsf{a}$, then $\mathsf{x} \lhd \mathsf{b}$. If $\mathsf{x} \lhd \mathsf{a}$, then by transitivity, $\mathsf{x} \lhd \mathsf{b}$. For the case $\mathsf{a} \lhd \mathsf{x}$ consider first $\mathsf{x} = \mathsf{s(a)}$; from $\mathsf{a} \lhd \mathsf{b}$ and (f), $\mathsf{s(a)} \unlhd \mathsf{b}$; but $\mathsf{s(a)} \in [\mathsf{a}]$; so by uniqueness $\mathsf{s(a)} \neq \mathsf{b}$; so $\mathsf{s(a)} \lhd \mathsf{b}$; from this, $\mathsf{s(s(a))} \unlhd \mathsf{b}$, and by uniqueness, $\mathsf{s(s(a))} \lhd \mathsf{b}$; and the same for any member of the chain after a; so $\mathsf{x} \lhd \mathsf{b}$. But now, $\mathsf{y} \lhd \mathsf{b}$ or $\mathsf{y} = \mathsf{b}$ or $\mathsf{b} \lhd \mathsf{y}$. If $\mathsf{y} = \mathsf{b}$ then from $\mathsf{x} \lhd \mathsf{b}$, $\mathsf{x} \lhd \mathsf{y}$. If $\mathsf{b} \lhd \mathsf{y}$, then with $\mathsf{x} \lhd \mathsf{b}$ and transitivity $\mathsf{x} \lhd \mathsf{y}$. And for $\mathsf{y} \lhd \mathsf{b}$, consider first $\mathsf{b} = \mathsf{s(y)}$; then $\mathsf{x} \lhd \mathsf{s(y)}$; so with (g), $\mathsf{x} \unlhd \mathsf{y}$; and since $[\mathsf{x}] = [\mathsf{a}] \neq [\mathsf{b}] = [\mathsf{y}]$, by uniqueness $\mathsf{x} \lhd \mathsf{y}$; and similarly for any member of the chain prior to b; so $\mathsf{x} \lhd \mathsf{y}$. In any case then $\mathsf{x} \lhd \mathsf{y}$; so every member of $[\mathsf{a}]$ is less than all the members of $[\mathsf{b}]$. Thus given $[\underset{\sim}{\mathsf{0}}] \neq [\mathsf{c}]$ and $\underset{\sim}{\mathsf{0}} \lhd \mathsf{c}$, the situation is so far as follows:

$$
\begin{array}{lll}
\mathsf{K} & \mathbb{Z}_c: & \ldots \mathsf{c} \ldots \\
 & \mathbb{N}_0: & \underset{\sim}{\mathsf{0}}, \underset{\sim}{\mathsf{1}}, \underset{\sim}{\mathsf{2}}, \ldots
\end{array}
$$

The universe has objects in the sequences $\mathbb{N}_0$ and $\mathbb{Z}_c$. Each sequence is ordered by $\lhd$. And every member of $\mathbb{N}_0$ is less than all the members of $\mathbb{Z}_c$. It remains that $\lhd$ satisfies the conditions for a linear order. But, insofar as $\mathbb{Z}_c$ has no least member, $\lhd$ is not a well-order on $\mathsf{U_K}$.

And there is more! Say $[\mathsf{a}] \lhd [\mathsf{b}]$ just in case all the members of $[\mathsf{a}]$ are less than all the members of $[\mathsf{b}]$. Then, as we have just seen, if $[\mathsf{a}] \neq [\mathsf{b}]$ and $\mathsf{a} \lhd \mathsf{b}$, $[\mathsf{a}] \lhd [\mathsf{b}]$. Now, simply presupposing background as from (a)–(i),

(i) There can be no greatest $\mathbb{Z}$-chain. For consider a $\mathbb{Z}$-chain $[\mathsf{z}]$. Because $[\mathsf{z}]$ is a $\mathbb{Z}$ chain, $\mathsf{z} \notin [\underline{\mathsf{o}}]$; so $\mathsf{z} \neq \underline{\mathsf{o}}$; so $\mathsf{z} \lhd \mathsf{z} \oplus \mathsf{z}$. Further, since $\mathsf{z} \notin [\underline{\mathsf{o}}]$, it is not an $\underline{\mathsf{a}}$ and $\mathsf{z} \oplus \mathsf{z}$ is not of the sort $\mathsf{z} \oplus \underline{\mathsf{a}}$; so $\mathsf{z} \not\approx \mathsf{z} \oplus \mathsf{z}$, and $[\mathsf{z}] \neq [\mathsf{z} \oplus \mathsf{z}]$. So both $\mathsf{z} \lhd \mathsf{z} \oplus \mathsf{z}$ and $[\mathsf{z}] \neq [\mathsf{z} \oplus \mathsf{z}]$; so $[\mathsf{z}] \lhd [\mathsf{z} \oplus \mathsf{z}]$.

(ii) Between any two chains there is another. Suppose $[\mathsf{a}] \lhd [\mathsf{c}]$. Then $\mathsf{a} \lhd \mathsf{c}$ and there is a b, $\mathsf{a} \lhd \mathsf{b} \lhd \mathsf{c}$ such that either $\mathsf{a} \oplus \mathsf{c} = \mathsf{b} \oplus \mathsf{b}$ or $\mathsf{a} \oplus \mathsf{c} \oplus \underline{\mathsf{1}} = \mathsf{b} \oplus \mathsf{b}$. For this, observe first that $\forall u \exists b(u = b + b \vee Su = b + b)$ is a theorem of PA; intuitively then, for u whichever of $\mathsf{a} \oplus \mathsf{c}$ or $\mathsf{s}(\mathsf{a} \oplus \mathsf{c}) = \mathsf{a} \oplus \mathsf{s}(\mathsf{c})$ is even, there is a b that averages a and c or a and $\mathsf{s}(\mathsf{c})$—and so, since a and c are in different chains and separated by more than a single application of s, such that $\mathsf{a} \lhd \mathsf{b} \lhd \mathsf{c}$. To take just the first case, suppose $\mathsf{a} \oplus \mathsf{c} = \mathsf{b} \oplus \mathsf{b}$.

Suppose $[\mathsf{a}] = [\mathsf{b}]$; then $\mathsf{b} \in [\mathsf{a}]$ and there is some $\underline{\mathsf{d}}$ such that $\mathsf{b} = \mathsf{a} \oplus \underline{\mathsf{d}}$ or $\mathsf{a} = \mathsf{b} \oplus \underline{\mathsf{d}}$; since $\mathsf{a} \lhd \mathsf{b}$, not the latter; so $\mathsf{b} = \mathsf{a} \oplus \underline{\mathsf{d}}$. So $\mathsf{a} \oplus \mathsf{c} = \mathsf{b} \oplus \mathsf{b} = \mathsf{a} \oplus \underline{\mathsf{d}} \oplus \mathsf{a} \oplus \underline{\mathsf{d}}$; so $\mathsf{c} = \mathsf{a} \oplus \underline{\mathsf{d}} \oplus \underline{\mathsf{d}}$; but $\underline{\mathsf{d}} \oplus \underline{\mathsf{d}}$ is some $\underline{\mathsf{e}}$; so $\mathsf{c} \in [\mathsf{a}]$; so $[\mathsf{a}] = [\mathsf{c}]$; this is impossible; reject the assumption: $[\mathsf{a}] \neq [\mathsf{b}]$. So $\mathsf{a} \lhd \mathsf{b}$ and $[\mathsf{a}] \neq [\mathsf{b}]$; so $[\mathsf{a}] \lhd [\mathsf{b}]$.

Suppose $[\mathsf{b}] = [\mathsf{c}]$; then $\mathsf{c} \in [\mathsf{b}]$ and there is some $\underline{\mathsf{d}}$ such that $\mathsf{c} = \mathsf{b} \oplus \underline{\mathsf{d}}$ or $\mathsf{b} = \mathsf{c} \oplus \underline{\mathsf{d}}$; since $\mathsf{b} \lhd \mathsf{c}$, not the latter; so $\mathsf{c} = \mathsf{b} \oplus \underline{\mathsf{d}}$. So $\mathsf{b} \oplus \mathsf{b} = \mathsf{a} \oplus \mathsf{c} = \mathsf{a} \oplus \mathsf{b} \oplus \underline{\mathsf{d}}$; so $\mathsf{b} = \mathsf{a} \oplus \underline{\mathsf{d}}$; so $\mathsf{c} = \mathsf{b} \oplus \underline{\mathsf{d}} = \mathsf{a} \oplus \underline{\mathsf{d}} \oplus \underline{\mathsf{d}}$; but $\underline{\mathsf{d}} \oplus \underline{\mathsf{d}}$ is some $\underline{\mathsf{e}}$; so $\mathsf{c} \in [\mathsf{a}]$; so $[\mathsf{a}] = [\mathsf{c}]$; this is impossible; reject the assumption: $[\mathsf{b}] \neq [\mathsf{c}]$. So $\mathsf{b} \lhd \mathsf{c}$ and $[\mathsf{b}] \neq [\mathsf{c}]$; so $[\mathsf{b}] \lhd [\mathsf{c}]$.

So $[\mathsf{a}] \lhd [\mathsf{b}] \lhd [\mathsf{c}]$.

(iii) There is no least $\mathbb{Z}$-chain: suppose otherwise, that $[\mathsf{q}]$ is the least $\mathbb{Z}$-chain; then by (ii) there is a $\mathbb{Z}$-chain $[\mathsf{p}]$, $[\underline{\mathsf{o}}] \lhd [\mathsf{p}] \lhd [\mathsf{q}]$; this is impossible.

Thus sequence of $\mathbb{Z}$ chains is like the sequence of rational numbers—densely ordered and without endpoints. Not only do nonstandard models of arithmetic fail to be isomorphic to N, but they include (at least) objects from the infinitely many $\mathbb{Z}$-chains.[8]

Insofar as $\lhd$ is not a well-order, you might worry that there is some problem about mathematical induction. There *is* a problem reasoning in the metalanguage by induction on the order relation $\lhd$ (as in Chapter 8). If a domino falls in one chain, there is no reason to think that dominoes fall in the next; similarly we cannot contradict a supposition that some dominoes do not fall by finding a least domino that does not fall, and showing that the assumption must fail. But instances of the *induction axiom* (PA7) remain true on K. The simplest way to make this point is

[8]Interestingly, nonstandard models have mathematical applications. Notably A. Robinson, *Non-Standard Analysis* applies nonstandard models of the real numbers to find the infinitesimals of calculus without use of limits.

to observe that K is *such that* instances of the axiom are true. Up to now, we have thought of the standard model as given, and seen how instances of the axiom are true on it. But now we start with $|\mathfrak{N}|$ and show there is a model on which its members are satisfied. Since instances of the induction axiom are members of $|\mathfrak{N}|$, they are true on K. But perhaps we can say a bit more: Let $\mathsf{K}[\![\mathcal{P}(x)]\!] = \{\mathsf{a} \mid \mathsf{K}_{\mathsf{d}(x|\mathsf{a})}[\mathcal{P}(x)] = \mathsf{S}\}$; so the members of $\mathsf{K}[\![\mathcal{P}(x)]\!]$ are objects to satisfy $\mathcal{P}(x)$. Even though $\mathsf{U_K}$ has nonempty subsets without a least member, it turns out that sets defined by our formal language do—as developed in the box below, each nonempty $\mathsf{K}[\![\mathcal{P}(x)]\!]$ has a least member. But this is sufficient for reasoning by induction applied to sets so defined: Suppose $\mathsf{D} \subseteq \mathsf{U_K}$ is some $\mathsf{K}[\![\mathcal{P}(x)]\!]$; then its complement $\overline{\mathsf{D}}$ (all the members of U that are not in D) is $\mathsf{K}[\![\sim\mathcal{P}(x)]\!]$ and so such that it is either empty or has a least member. Given this, if D has the special property that $\underset{\cdot}{\mathsf{o}} \in \mathsf{D}$, and if $\mathsf{a} \in \mathsf{D}$ then $\mathsf{s(a)} \in \mathsf{D}$, it follows that $\mathsf{D} = \mathsf{U_K}$: Suppose $\mathsf{D} \neq \mathsf{U_K}$, then $\overline{\mathsf{D}}$ is nonempty; so it has a least member a; but $\underset{\cdot}{\mathsf{o}} \in \mathsf{D}$; so $\underset{\cdot}{\mathsf{o}} \notin \overline{\mathsf{D}}$; so $\underset{\cdot}{\mathsf{o}} \neq \mathsf{a}$; so a is some $\mathsf{s(m)}$; but since $\mathsf{s(m)}$ is the least member of $\overline{\mathsf{D}}$, $\mathsf{m} \in \mathsf{D}$, from which it follows that $\mathsf{s(m)} \in \mathsf{D}$; so $\mathsf{s(m)} \notin \overline{\mathsf{D}}$; this is impossible; so $\mathsf{D} = \mathsf{U_K}$. Of course, the boxed argument to show that each nonempty $\mathsf{K}[\![\mathcal{P}(x)]\!]$ has a least member relies upon the induction axiom—thus, as a *justification* for the induction axiom, this reasoning is entirely circular; it may, however, help clarify or explain *how* the induction axiom remains true on K.

From compactness there must *exist* a nonstandard model. We have described the order relation on one such model. As it turns out, functions for addition and multiplication are complex in a way that resists straightforward description (for discussion see Boolos, Burgess, and Jeffrey, *Computability and Logic*, Chapter 25). K is weird! Interestingly, its *existence* was proved considering just finite satisfiability on perfectly straightforward models of arithmetic. As we shall see in the next section, there are members of $\mathfrak{Mod}(|\mathfrak{N}|)$ weirder still.

Each nonempty $\mathsf{K}[\![\mathcal{P}(x)]\!]$ *has a least member:* Suppose otherwise, that $\mathsf{K}[\![\mathcal{P}(x)]\!]$ is nonempty but has no least element, and consider $\mathcal{Q}(y) = (\forall x \leq y)\sim\mathcal{P}(x)$.

(i) Suppose $\mathsf{K}[\mathcal{Q}(\bar{0})] \neq \mathsf{T}$; then there is some d such that $\mathsf{K_d}[\mathcal{Q}(\bar{0})] \neq \mathsf{S}$; so $\mathsf{K_d}[(\forall x \leq \bar{0})\sim\mathcal{P}(x)] \neq \mathsf{S}$; so $\mathsf{K}_{\mathsf{d}(x|\underset{\cdot}{\mathsf{o}})}[\sim\mathcal{P}(x)] \neq \mathsf{S}$, and $\mathsf{K}_{\mathsf{d}(x|\underset{\cdot}{\mathsf{o}})}[\mathcal{P}(x)] = \mathsf{S}$. But then $\underset{\cdot}{\mathsf{o}}$ is the least member of $\mathsf{K}[\![\mathcal{P}(x)]\!]$; this is impossible: $\mathsf{K}[\mathcal{Q}(\bar{0})] = \mathsf{T}$. (ii) Suppose $\mathsf{K}[\forall y(\mathcal{Q}(y) \rightarrow \mathcal{Q}(Sy))] \neq \mathsf{T}$; then there is a d and $\mathsf{m} \in \mathsf{U_K}$ such that $\mathsf{K}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(y)] = \mathsf{S}$ and $\mathsf{K}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(Sy)] \neq \mathsf{S}$; so $\mathsf{K}_{\mathsf{d}(y|\mathsf{m})}[(\forall x \leq y)\sim\mathcal{P}(x)] = \mathsf{S}$ and $\mathsf{K}_{\mathsf{d}(y|\mathsf{m})}[(\forall x \leq Sy)\sim\mathcal{P}(x)] \neq \mathsf{S}$; with the former, objects $\trianglelefteq \mathsf{m}$ fail to satisfy $\mathcal{P}(x)$; with the latter, there is an $\mathsf{a} \trianglelefteq \mathsf{s(m)}$ that does satisfy $\mathcal{P}(x)$; so $\mathsf{a} = \mathsf{s(m)}$. But then a is the least member of $\mathsf{K}[\![\mathcal{P}(x)]\!]$; this is impossible: $\mathsf{K}[\forall y(\mathcal{Q}(y) \rightarrow \mathcal{Q}(Sy))] = \mathsf{T}$.

But instances of the induction axiom are true on K; from this, with (i) and (ii), $\mathsf{K}[\forall y\mathcal{Q}(y)] = \mathsf{T}$; so $\mathsf{K}[\![\mathcal{P}(x)]\!]$ is empty. This is impossible; reject the assumption: if $\mathsf{K}[\![\mathcal{P}(x)]\!]$ is nonempty then it has a least element.

Löwenheim-Skolem

Associate the *size* of a model with the size of its domain. A countable model has a countable domain, an uncountable model an uncountable domain. Given an infinite model for some Σ, the Löwenheim-Skolem theorems tell us that Σ has models of different infinite sizes. All of this inevitably pushes us toward thinking about the infinite sets at which we said we would merely wave. Now is the time to say "hello." We shall not engage the details. However we should be able to say enough to *understand* the theorems and see something of their consequences. Along with what you have from the Chapter 2 countability and more on countability references, we require this much:

> Sets r and s are the *same size* ($\mathsf{r} \approx \mathsf{s}$) iff there is a 1:1 function from one onto the other. But not all infinite sets are the same size; $\mathsf{r} \preccurlyeq \mathsf{s}$ just in case there is a 1:1 function from r *into* s, and $\mathsf{r} \prec \mathsf{s}$ iff $\mathsf{r} \preccurlyeq \mathsf{s}$ and $\mathsf{r} \not\approx \mathsf{s}$. Then either $\mathsf{r} \preccurlyeq \mathsf{s}$ or $\mathsf{s} \prec \mathsf{r}$ (equivalent to the axiom of choice), and if both $\mathsf{r} \preccurlyeq \mathsf{s}$ and $\mathsf{s} \preccurlyeq \mathsf{r}$ then $\mathsf{r} \approx \mathsf{s}$ (the Schröder-Bernstein theorem). In particular, the set of all real numbers has more members than the set of all natural numbers. And from Cantor's theorem, for every set there is one bigger than it (see the box on page 537). The *cardinal numbers* are certain sets designated to measure the size of others—set s has cardinality α just in case $\mathsf{s} \approx \alpha$. Let $\mathsf{card}(\mathsf{s})$ be the cardinality of set s; $\mathsf{card}(\mathsf{r}) = \mathsf{card}(\mathsf{s})$ iff $\mathsf{r} \approx \mathsf{s}$, and $\mathsf{card}(\mathsf{r}) \leq \mathsf{card}(\mathsf{s})$ iff $\mathsf{r} \preccurlyeq \mathsf{s}$. If, along the lines of the example appearing on page 642, we think of a natural number n as a set with n members, then the finite cardinals are members of $\mathbb{N}$; the first infinite cardinal $\aleph_0$ is the size of the set of natural numbers; then $\aleph_1$, and so forth ($\aleph$ *aleph* is the first letter of the Hebrew alphabet).[9]
>
> The members of any set, and so of an infinite set, may be well-ordered. The *ordinal numbers* are certain well-ordered sets to measure the "length" of well-ordered sets. On a standard account the members of an ordinal are all the ordinals smaller than it (again see the example on page 642); then the finite ordinals are just the members of $\mathbb{N}$; ω is the first ordinal greater than all of them, then $\omega + 1$, $\omega + 2, \ldots$; and greater than all of them $\omega \times 2$, $\omega \times 2 + 1, \ldots$; and greater than all of them $\omega \times 3$, $\omega \times 3 + 1, \ldots$; and after continuing this way for finite multiples of ω, $\omega \times \omega = \omega^2$. Example (d) on page 528 would locate $\omega \times 2$ at the very top greater than all the members of series below, and then ω^2 greater than all members of all the series for finite multiples of ω. And the process continues to incredible lengths! Different ordinal numbers may have the same cardinality;

[9] The proposition that $\aleph_1$ is the cardinality of the set $\mathbb{R}$ of all real numbers (the continuum of points in the number line) is the *continuum hypothesis* (CH). If CH is true, then no cardinal lies between the cardinals of $\mathbb{N}$ and $\mathbb{R}$. Supposing that ZFC is consistent, P. Cohen ("The Independence of the Continuum Hypothesis") shows that CH does not follow from its axioms, and Gödel (*The Consistency of the Continuum Hypothesis*) that neither does not-CH. These are results for an intermediate course in set theory.

so for example, $\{a_0, a_1, \ldots\}$ and $\{a_0, a_1, \ldots, b_0, b_1, \ldots\}$ are like ω and $\omega \times 2$ but, as from the countability reference, $\aleph_0$ maps onto both—and, in general, if r and s are infinite sets and $\mathsf{card}(\mathsf{r}) \leq \mathsf{card}(\mathsf{s})$, then the cardinality of their union, $\mathsf{card}(\mathsf{r} \cup \mathsf{s}) = \mathsf{card}(\mathsf{s})$. It is usual to identify *initial* ordinals as ordinals whose cardinality is greater than all the ones before, and then the cardinal numbers with the initial ordinals. Because the ordinals are well-ordered it is possible to state recursive definitions and apply mathematical induction to sequences ordered by them. Reasoning is extended from what we have seen insofar as *limit ordinals* (for example ω) are not the successor of any other. Given reasoning for limit cases, however, the basic idea remains the same.

If we accept this much, we shall be in a position to make progress (for details see most any introduction to set theory as Enderton, *Elements of Set Theory*).

The Löwenheim-Skolem theorems appear in somewhat different forms. At the simplest level, they tell us about the size of models. So given an infinite model M for some formulas Σ, there are models for Σ whose size is different from M. An immediate consequence is a difficulty about our ability to "pin down" isomorphic interpretations: Given the 1:1 function ι from the universe of one onto the universe of the other, isomorphic models are the same size; so if models are not the same size, they are not isomorphic. Here is a first result of this type, accessible from what we have already done (compare E10.28):

(O) *If Σ has a model, then it has a countable model.* Suppose Σ has a model K. Then by T10.4, Σ is consistent; so by T10.16, Σ has a model M whose universe is constructed of disjoint sets of natural numbers. But $\mathsf{U_M}$ is countable, for we might map the sets in $\mathsf{U_M}$ to natural numbers by, say, their least elements. Alternatively we might set up a function ι from each set in $\mathsf{U_M}$ to its least element, to establish an isomorphic interpretation M' whose universe just *is* a set of natural numbers; then by T11.15, $\mathsf{M}'[\Sigma] = \mathsf{T}$. Either way, Σ has a countable model.

Thus, for example, if Σ has a model whose universe is the set of all real numbers, then it also has a model whose universe is a set of natural numbers. Observe that this result makes good our Chapter 4 (page 128) claim that if there is any interpretation of some formulas, then there is one whose universe is a set of integers.

The above result can be generalized by a corresponding generalization of T10.16. To reach T10.16, starting from an $\mathscr{L}'$ with constants c_n matched to natural numbers, we constructed a maximal consistent scapegoat set, and then the model whose domain is the set of disjoint sets with natural numbers as members. Suppose we relax the requirement that a language have just countably many constants. Then for some uncountable κ, the set of constants may have members c_α matched to ordinal numbers $< \kappa$—and so, on the standard account, to the members of κ. (We assume that languages have countably many symbols as usual except that they may be specified to include extra constants.) A language with uncountably many constants is a "theoretical object" to the extent that there are more symbols than can be represented by finite strings humans speak and write. All the same, we can reason about features of the theoretical object. In particular, it is possible to obtain a T10.16*, starting from an $\mathcal{L}'$ with constants c_α matched to ordinal numbers $< \kappa$, constructing a maximal consistent scapegoat set, and then a model whose domain is a set of disjoint sets whose members are ordinal numbers $< \kappa$. The argument is modified to accommodate uncountable ordinals in the specification of the "big" set, and for the demonstration that the result is a maximal consistent scapegoat set. But the basic idea is the same. Given T10.16*, we can reason very much as before:

Cantor's Theorem

The result that every set has one with more members than it underlies the discussion of this section. Though the demonstration is somewhat to the side of our main concerns, it is worth seeing how it goes.

As we have seen from the Chapter 4 set theory reference, set a is a *subset* of set b iff every member of a is a member of b. Now the *powerset* of a, $\wp(\mathsf{a})$ is the set of all the subsets of a. So the powerset of $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ is $\{\{\,\}, \{\mathsf{a}\}, \{\mathsf{b}\}, \{\mathsf{c}\}, \{\mathsf{a}, \mathsf{b}\}, \{\mathsf{a}, \mathsf{c}\}, \{\mathsf{b}, \mathsf{c}\}, \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}\}$.

T11.28. For any set s, $\wp(\mathsf{s})$ is greater than s, $\mathsf{s} \prec \wp(\mathsf{s})$ *Cantor's Theorem*.

Suppose otherwise, that $\mathsf{s} \not\prec \wp(\mathsf{s})$; then $\wp(\mathsf{s}) \preccurlyeq \mathsf{s}$, and there is a 1:1 function g from $\wp(\mathsf{s})$ into s; the converse of this function, with $\langle \mathsf{b}, \mathsf{a}\rangle$ for each $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathsf{g}$, is a 1:1 function h from (some) members of s onto all the members of $\wp(\mathsf{s})$. Since h is a function from objects in s to sets of those objects, we may ask if a given $\mathsf{x} \in \mathsf{s}$ is itself a member of $\mathsf{h}(\mathsf{x})$. Thus, with integers, if some function f has $\mathsf{f}(2) = \{2, 4, 6\}$ and $\mathsf{f}(3) = \{19, 127\}$, then 2 is a member of $\mathsf{f}(2)$ but 3 is not a member of $\mathsf{f}(3)$. Consider $\mathsf{c} = \{\mathsf{x} \in \mathsf{s} \mid \mathsf{x} \notin \mathsf{h}(\mathsf{x})\}$, the set of all elements x in s such that x is not a member of $\mathsf{h}(\mathsf{x})$; c is formed by collecting every member x in s which is not a member of the subset to which it is mapped by h. Any member of c is collected from s; so c is a subset of s and thus a member of $\wp(\mathsf{s})$. But c is designed so that it differs from every $\mathsf{h}(\mathsf{x})$ in membership of x: Consider an arbitrary set $\mathsf{h}(\mathsf{x})$; if x is a member of $\mathsf{h}(\mathsf{x})$, then by construction, x is not included in c, so $\mathsf{h}(\mathsf{x}) \neq \mathsf{c}$; if x is not a member of $\mathsf{h}(\mathsf{x})$, then by construction, x is included in c and again $\mathsf{h}(\mathsf{x}) \neq \mathsf{c}$; either way $\mathsf{h}(\mathsf{x}) \neq \mathsf{c}$. So there is no x such that $\mathsf{h}(\mathsf{x}) = \mathsf{c}$, and h does not map onto all the members of $\wp(\mathsf{s})$. This contradicts the specification of h, itself a consequence of the original assumption about s; reject the assumption: $\mathsf{s} \prec \wp(\mathsf{s})$.

T11.29. If the members of Σ are in a language $\mathcal{L}$ whose constants are matched to ordinals less than an infinite κ and Σ has a model, then Σ has a model of cardinality $\leq \kappa$. *Downward Löwenheim-Skolem.*

Suppose the members of Σ are in an $\mathcal{L}$ whose constants are matched to ordinals less than an infinite κ and Σ has a model. Then by T10.4, Σ is consistent; so by T10.16*, Σ has a model M whose universe consists of disjoint sets of ordinal numbers $< \kappa$. But then a map from sets in $\mathsf{U_M}$ to their least members is a 1:1 function from $\mathsf{U_M}$ into κ; so $\mathsf{U_M} \preccurlyeq \kappa$, and $\mathsf{card(M)} \leq \kappa$.

In the case (O) just above, κ was ω with the result that Σ in a language with countably many constants has a countable model (a model with cardinality $\leq \omega$). Now the result is generalized to arbitrary cardinal κ.

Given T10.16* we may obtain a *compactness** whose application is to languages that allow arbitrarily large infinite sets of constants. Reasoning is parallel to that for T11.23. And with compactness*, there is an "upward" Löwenheim-Skolem theorem.

T11.30. If Σ has an infinite model, then for any infinite cardinal κ, Σ has a model of cardinality $\geq \kappa$. *Upward Löwenheim-Skolem.*

Suppose Σ has an infinite model M and κ is an infinite cardinal. Extend $\mathcal{L}$ to an $\mathcal{L}'$ including some new constants c_α for $\alpha < \kappa$; and let $\Gamma' = \{c_\beta \neq c_\gamma \mid \beta, \gamma < \kappa$ and $\beta \neq \gamma\}$. Consider an arbitrary finite $\Delta' \subseteq \Sigma' \cup \Gamma'$; Δ' may have as members some elements of Σ' and finitely many sentences $c_\beta \neq c_\gamma$. Extend the infinite model M for Σ to an M' that assigns the (finitely many) new constants from Δ' to distinct members of $\mathsf{U_{M'}}$ and otherwise assigns c_α to some constant member of the domain. With T10.14, M' satisfies Σ' and so all the members of Σ' in Δ'; and since constants from Δ' are assigned to distinct members of $\mathsf{U_{M'}}$, M' satisfies sentences $c_\beta \neq c_\gamma$ from Δ' as well; so Δ' has a model. And since Δ' is arbitrary, by compactness*, $\Sigma' \cup \Gamma'$ has a model K'. Since K' is a model for $\Sigma' \cup \Gamma'$, it is a model for Σ'; and with T10.14 again, K a model for Σ. But K' satisfies $c_\beta \neq c_\gamma$ for all $\beta, \gamma < \kappa$ such that $\beta \neq \gamma$; so $\mathsf{K}'[c_\beta] \neq \mathsf{K}'[c_\gamma]$; so ι such that $\iota(\alpha) = \mathsf{K}'[c_\alpha]$ maps the ordinal numbers $< \kappa$ to distinct members of the universe, and so is a 1:1 function from κ into $\mathsf{U_{K'}}$; so $\kappa \leq \mathsf{card(K')}$; but $\mathsf{U_K} = \mathsf{U_{K'}}$; so $\kappa \leq \mathsf{card(K)}$.

Just as the universe for our nonstandard model of arithmetic includes more nonstandard members than a single element assigned to the extra constant c, so we cannot be sure that K does not include more than members assigned to extra constants. But given that K' satisfies the inequalities from Γ' we can be sure that K' and so K include *at least* as many objects as there are extra constants, and so that the cardinality of K is at least as great as the cardinality of the set of extra constants.

And T11.29 and T11.30 combine to a "full" version of the Löwenheim-Skolem theorem:

T11.31. If the members of Σ are in a language whose constants are matched to ordinals less than an infinite θ and Σ has an infinite model, then for any infinite cardinal $\kappa \geq \theta$, Σ has a model of cardinality κ. *Full Löwenheim-Skolem.*

Suppose the memebers of Σ are in a language whose constants are matched to ordinals less than an infinite θ, and Σ has an infinite model. Let κ be an infinite cardinal $\geq \theta$. Extend $\mathscr{L}$ to an $\mathscr{L}'$ including some new constants c_α for $\alpha < \kappa$; and let $\Gamma' = \{c_\beta \neq c_\gamma \mid \beta, \gamma < \kappa \text{ and } \beta \neq \gamma\}$ and $\Phi' = \Sigma' \cup \Gamma'$. Since Σ has an infinite model, the upward Löwenheim-Skolem theorem tells us that Σ has a model K of cardinality $\geq \kappa$; so there are at least as many members of $\mathsf{U_K}$ as there are constants c_α; extend K to a K' that assigns distinct members of $\mathsf{U_{K'}}$ to the constants c_α; then K' is a model for Φ'. So Φ' has a model; further, since $\kappa \geq \theta$ the cardinality of the set of constants in $\mathscr{L}'$ is κ, and all the constants of $\mathscr{L}'$ match to ordinals less than κ; so by the downward Löwenheim-Skolem theorem Φ' has a model M' such that $\mathsf{card}(\mathsf{M}') \leq \kappa$. But given the inequalities satisfied by M', an ι such that $\iota(\alpha) = \mathsf{M}'[c_\alpha]$ is a 1:1 function from κ into $\mathsf{U_{M'}}$; so $\kappa \leq \mathsf{card}(\mathsf{M}')$. So $\mathsf{card}(\mathsf{M}') = \kappa$; but $\mathsf{U_M} = \mathsf{U_{M'}}$; so $\mathsf{card}(\mathsf{M}) = \kappa$. And since M' is a model for $\Phi' = \Sigma' \cup \Gamma'$, M' is a model for Σ', and M a model for Σ.

From these results, there is no infinite α such that the class of models with cardinality α is axiomatizable. Suppose otherwise, that for some language $\mathscr{L}$, $\mathfrak{M}_\alpha$ is the class of all models with cardinality α, and $\mathfrak{Md}(\Sigma) = \mathfrak{M}_\alpha$; then for some $\mathsf{M} \in \mathfrak{M}_\alpha$, $\mathsf{M}[\Sigma] = \mathsf{T}$; so Σ has an infinite model; so Σ has models of cardinality other than than α; so $\mathfrak{Md}(\Sigma) \neq \mathfrak{M}_\alpha$. Similarly, if M is a model of infinite cardinality α and $\mathfrak{M}$ is the class of models isomorphic to it, there is no Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{M}$. Suppose otherwise; then $\mathsf{M}[\Sigma] = \mathsf{T}$; so Σ has an infinite model; so Σ has models of cardinality other than α, and so models not isomorphic to M; so $\mathfrak{Md}(\Sigma) \neq \mathfrak{M}$.

It is worth observing that the Löwenheim-Skolem theorems extend to forms that specify a certain content for the models of different size. In particular, models may be such that one is an elementary submodel of the other:

(i) Suppose the members of Σ are in a language whose constants are matched to ordinals less than an infinite θ, and Σ has a model M of infinite cardinality $\gamma \geq \theta$. Then for any β, $\theta \leq \beta \leq \gamma$ there is an L of cardinality β such that $\mathsf{L} \preceq \mathsf{M}$.

(ii) Suppose the members of Σ are in a language whose constants are matched to ordinals less than an infinite θ, and Σ has model M of infinite cardinality $\gamma \geq \theta$. Then for any $\beta \geq \gamma$, there is a model L of cardinality β such that $\mathsf{M} \preceq \mathsf{L}$.

These are interesting (and not the only results of the sort). But demonstrations would unduly stress our background from set theory. Since we do not require the additional results, we rest content with what we have.

Skolem's paradox. At first glance, results from the Löwenheim-Skolem theorems may seem strange: If there are more real numbers than natural numbers, how is it that a *theory of real numbers* can be true on a universe of the natural numbers? And if there are more real numbers than natural numbers, how is it that a *theory of natural numbers* can be true on the reals? Similarly, and more dramatically, there is a formula $\mathcal{U}(x)$ in the language of ZFC set theory that is true of just uncountable sets; and $\exists x\mathcal{U}(x)$ is a theorem of ZFC. But if ZFC is consistent then it has a model; and ZFC is expressed in an ordinary countable language; so by the downward Löwenheim-Skolem theorem it has a countable model M; and since $\exists x\mathcal{U}(x)$ is a theorem of ZFC, it is true on M; so there is some $\mathsf{m} \in \mathsf{U_M}$ such that $\mathsf{M}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{U}(x)] = \mathsf{S}$—but, clearly, there are not enough objects in $\mathsf{U_M}$ for any member of it to have uncountably many elements. Technically, we already understand the response: No Σ is sufficient to pin down the cardinality of an infinite model; so the axioms of ZFC (or any other theory in first-order language) are not sufficient to pin down the cardinality of an infinite universe. On their intended interpretations, functions and relations assigned to $+$, $\times$, $\in$, apply as we expect. But, as for the case of number theory, nonstandard interpretations reinterpret the vocabulary so as to model sentences in alternative ways. Thus $\exists x\mathcal{U}(x)$ may be true on a model, even though the model is without an uncountable universe.

E11.19. In showing that between any two chains there is another ((ii) on page 533), there were cases for $\mathsf{b} \oplus \mathsf{b} = \mathsf{a} \oplus \mathsf{c}$ and $\mathsf{b} \oplus \mathsf{b} = \mathsf{a} \oplus \mathsf{c} \oplus \underset{\sim}{1}$. Work the second case. Hint: From $\mathsf{a} \lhd \mathsf{b}$ and $\mathsf{b} = \mathsf{a} \oplus \underset{\sim}{\mathsf{d}}$, it follows that $\underset{\sim}{\mathsf{d}} \neq \underset{\sim}{\mathsf{o}}$ (else $\mathsf{a} \not\lhd \mathsf{b}$); so there is some $\underset{\sim}{\mathsf{e}}$ such that $\underset{\sim}{\mathsf{d}} = \underset{\sim}{\mathsf{e}} \oplus \underset{\sim}{1}$.

E11.20. Use compactness to show that if $\Sigma \vDash \mathcal{P}$, then for some finite $\Delta \subseteq \Sigma$, $\Delta \vDash \mathcal{P}$. Hint: If $\Sigma \cup \{\sim\mathcal{P}^u\}$ is unsatisfiable, then by compactness some finite $\Delta \subseteq \Sigma \cup \{\sim\mathcal{P}^u\}$ is unsatisfiable.

*E11.21. Use the compactness theorem to show that if $\mathfrak{K} = \mathfrak{Mod}(\Sigma)$ is axiomatized by some finite Φ then $\mathfrak{K}$ is axiomatized also by a finite subset of Σ. Hint: Where $\mathcal{A}$ is the conjunction of the members of Φ, and $\mathcal{A}^u$ its universal closure, $\Gamma = \Sigma \cup \{\sim\mathcal{A}^u\}$ is not satisfiable, and by compactness must have an unsatisfiable finite subset.

E11.22. Let $\mathfrak{K}$ be any class of models and $\mathfrak{M}$ the class of all models that are not members of $\mathfrak{K}$ (so $\mathfrak{K} \cup \mathfrak{M}$ is the class of all models and $\mathfrak{K} \cap \mathfrak{M} = \varnothing$). (a) Show $\mathfrak{K}$ is finitely axiomatizable iff both $\mathfrak{K}$ and $\mathfrak{M}$ are axiomatizable. (b) Use this result with our demonstration that there is an axiomatization of models with an infinite domain and T11.24 to provide another demonstration that the class of all finite models is not axiomatizable. Hint for (a) right to left: If $\mathfrak{K}$ and $\mathfrak{M}$ are axiomatizable, there are some Γ and Σ such that $\mathfrak{K} = \mathfrak{Mod}(\Gamma)$ and $\mathfrak{M} = \mathfrak{Mod}(\Sigma)$; consider an application of compactness to $\Gamma \cup \Sigma$.

*E11.23. In 1977 Appel and Haken, "Solution of the Four-Color Map Problem" solved a longstanding problem, proving that every planar map can be colored with four colors without adjacent regions having the same color. Such a map is understood to be finite. Supposing their result, use the compactness theorem to show that even an infinite map can be colored with four colors.[10]

Suppose an infinite M with language $\mathscr{L}$ assigns a two-place irreflexive and symmetric relation $\shortparallel$ to symbol $\shortmid$ (so $\forall x(x \nmid x)$ and $\forall x \forall y(x \shortmid y \to y \shortmid x)$); let extensions M_+ and $\mathscr{L}_+$ be like M and $\mathscr{L}$ but with one-place relations $\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3, \mathsf{C}_4$ assigned to symbols C_1, C_2, C_3, C_4. We think of members of the universe as regions, $\mathsf{m} \shortparallel \mathsf{n}$ when m is adjacent to (shares a border with) n, and $\mathsf{m} \in \mathsf{C}_i$ when m has color C_i. Let,

$$\Phi_+ = \begin{cases} \forall x(C_1x \vee C_2x \vee C_3x \vee C_4x), \\ \forall x[(C_1x \to \sim(C_2x \vee C_3x \vee C_4x)) \wedge (C_2x \to \sim(C_1x \vee C_3x \vee C_4x)) \wedge \\ \qquad (C_3x \to \sim(C_1x \vee C_2x \vee C_4x)) \wedge (C_4x \to \sim(C_1x \vee C_2x \vee C_3x))], \\ \forall x \forall y[x \shortmid y \to (\sim(C_1x \wedge C_1y) \wedge \sim(C_2x \wedge C_2y) \wedge \sim(C_3x \wedge C_3y) \wedge \sim(C_4x \wedge C_4y))] \end{cases}$$

Intuitively: Every region has a color; no region has more than one color; and adjacent regions do not have the same color. By the four-color theorem, for any finite $\mathsf{L} \sqsubseteq \mathsf{M}$ there is an L_+ such that $\mathsf{L}_+[\Phi_+] = \mathsf{T}$. The task is to show that there exists an M_+ such that $\mathsf{M}_+[\Phi_+] = \mathsf{T}$, and so that M is four-colorable.

Hints: This breaks into two interesting parts: (i) Extend $\mathscr{L}_+$ to an $\mathscr{L}'_+$ by the addition of a constant $\bar{\mathsf{a}}$ for each $\mathsf{a} \in \mathsf{U_M}$; let $\Sigma'_+ = \Phi_+ \cup \{\bar{\mathsf{m}} \shortmid \bar{\mathsf{n}} \mid \mathsf{m}, \mathsf{n} \in \mathsf{U_M}$ and $\mathsf{m} \shortparallel \mathsf{n}\} \cup \{\bar{\mathsf{m}} \neq \bar{\mathsf{n}} \mid \mathsf{m}, \mathsf{n} \in \mathsf{U_M}$ and $\mathsf{m} \neq \mathsf{n}\}$. For finite $\Delta'_+ \subseteq \Sigma'_+$, let the members of $\mathsf{U_H}$ be objects to which constants in Δ'_+ are assigned and $\mathsf{H}[\shortmid]$ be the restriction of $\mathsf{M}[\shortmid]$ to $\mathsf{U_H}$; then $\mathsf{H} \sqsubseteq \mathsf{M}$ and by the four-color theorem there is a H_+ extending H such that $\mathsf{H}_+[\Phi_+] = \mathsf{T}$. From H_+ you will be able to find a H'_+ to satisfy Δ'_+ and so a J'_+ to satisfy Σ'. (ii) While J'_+ satisfies Φ_+, it is not yet the M_+ we want insofar as its universe may be other than the universe of M—but we can manipulate J'_+ to obtain the desired interpretation. Consider J_+ like J'_+ but without assignments to extra constants, and then $\mathsf{K}_+ \sqsubseteq \mathsf{J}_+$ restricting the universe of K_+ to $\{\mathsf{J}'_+[\bar{\mathsf{a}}] \mid \mathsf{a} \in \mathsf{U_M}\}$, and finally the $\mathsf{L}_+ \cong \mathsf{K}_+$ that maps $\mathsf{J}'_+[\bar{\mathsf{a}}]$ to a; you will be able to show that $\mathsf{L}_+[\Phi_+]$ remains true, and that L_+ is an M_+—so that M is four-colorable. You should find E11.17 to be helpful.

[10]Interestingly, Appel and Haken employ a computer to verify cases (more cases than can be verified by hand). This inspires debate about the nature of *proof*.

E11.24. On a finite universe, it is always possible to extend a partial order $\lhd$ to a linear order $\LHD$—so where the relations are sets of pairs, $\lhd \subseteq \LHD$ and $\LHD$ is a linear order: Intuitively, for U with n members, *choose* an $\mathsf{m} \in \mathsf{U}$ such that no member of U is less than it (a *minimal* element of U), and let $\mathsf{a}_1 = \mathsf{m}$; then choose a minimal $\mathsf{n} \in \mathsf{U} - \{\mathsf{a}_1\}$—from among objects in U but not in $\{\mathsf{a}_1\}$—and let $\mathsf{a}_2 = \mathsf{n}$; then choose a minimal $\mathsf{o} \in \mathsf{U} - \{\mathsf{a}_1, \mathsf{a}_2\}$ and let $\mathsf{a}_3 = \mathsf{o}$; continue in this way until all the members of U are included in $\mathsf{a}_1 \dots \mathsf{a}_n$. Then the relation $\mathsf{a}_i \LHD \mathsf{a}_j$ iff $1 \leq i < j \leq n$ has $\mathsf{a}_1 \LHD \mathsf{a}_2 \LHD \mathsf{a}_3 \LHD \cdots \LHD \mathsf{a}_n$ and extends $\lhd$ to a linear order.

Use this result with compactness* to show that it is possible to do the same on an infinite universe. That is, suppose language $\mathscr{L}$ has just a single two-place relation symbol $\triangleleft$ and infinite model M for language $\mathscr{L}$ assigns a partial order $\lhd$ to the symbol $\triangleleft$. Show that M can be extended to an M_+ that assigns to $\triangleleft$ a linear order $\LHD$. Observe that this is not obvious: at least, on an infinite universe there might be no minimal element of a partial order so that intuitive reasoning for the finite sets is inapplicable.

Hints: This works very much like the previous exercise. Let Λ be the set whose members are the axioms O1, O2, and O3 for a liner order. Extend $\mathscr{L}$ to an $\mathscr{L}'$ by the addition of a constant $\overline{\mathsf{a}}$ for each $\mathsf{a} \in \mathsf{U_M}$; let $\Sigma' = \Lambda \cup \{\overline{\mathsf{m}} \triangleleft \overline{\mathsf{n}} \mid \mathsf{m}, \mathsf{n} \in \mathsf{U_M}$ and $\mathsf{m} \lhd \mathsf{n}\} \cup \{\overline{\mathsf{m}} \neq \overline{\mathsf{n}} \mid \mathsf{m}, \mathsf{n} \in \mathsf{U_M}$ and $\mathsf{m} \neq \mathsf{n}\}$. You should be able to find a J' to satisfy Σ', and manipulate this interpretation to a linear order M_+ extending M. Again, you will find E11.17 useful.

11.4.3 C̃ompleteness

We have seen from the corollary to T11.16 that a categorical Σ is c̃omplete. But from the Löwenheim-Skolem theorems, no Σ with an infinite model is categorical. This may seem a problem to the extent that we desire c̃omplete theories. So far as the Löwenheim-Skolem theorems go, however, "space" for c̃ompleteness remains: If M is finite, nothing from the Löwenheim-Skolem theorems blocks an axiomatization of the class of all models isomorphic to it. Further, by T11.15 isomorphism implies elementary equivalence; and from T11.16, a Σ whose models are elementarily equivalent is c̃omplete. But elementary equivalence does not require isomorphism—this is a moral of our discussion of number theory and Löwenheim-Skolem; so models might be elementarily equivalent without being isomorphic—and so Σ c̃omplete without being categorical. In this section we see that, in some cases at least, c̃omplete theories do occupy this space.[11]

[11] Especially with respect to arithmetic and results of Part IV, questions of c̃ompleteness are motivated in the introductions to Part IV and to Chapter 12. You might want to look over that material now.

Finite Models

Suppose D is a finite model and let $\mathfrak{D}$ be the class of all models isomorphic to it. Then we may show that there is a Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{D}$. Given the finite model D, we construct Σ, and show that members of $\mathfrak{Mod}(\Sigma)$ are isomorphic—and so that Σ is categorical. First, then, given D, we set out to construct Σ. We do this by finding a sequence of formulas $\mathcal{C}^e_0, \mathcal{C}^e_1, \ldots$ that, taken together, are an axiomatization of $\mathfrak{D}$.

First, since D is finite, U_D is some $\{\mathsf{m}_1, \mathsf{m}_2, \ldots, \mathsf{m}_\mathsf{n}\}$. For some enumeration of variables $x_1, x_2, \ldots$ consider an assignment d such that $\mathsf{d}[x_1] = \mathsf{m}_1$, and $\mathsf{d}[x_2] = \mathsf{m}_2$, and ... and $\mathsf{d}[x_n] = \mathsf{m}_\mathsf{n}$. Then, drawing upon our discussion of 'at least' and 'at most', let $\mathcal{C}_0$ be the open formula,

$$\bigwedge_{1 \le i < j \le n} x_i \neq x_j \;\wedge\; \forall v \bigvee_{1 \le i \le n} v = x_i$$

So in the case of a four-member universe $\mathcal{C}_0$ is,

$$(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4) \wedge \forall v(v = x_1 \vee v = x_2 \vee v = x_3 \vee v = x_4)$$

By analogy with $\exists x \exists y(x \neq y \wedge \forall v(v = x \vee v = y))$ for 'there are exactly two' the existential closure of this expression, $\exists x_1 \exists x_2 \ldots \exists x_n \mathcal{C}_0$ is true just when there are exactly n things.

Now consider an enumeration, $\mathcal{A}_1, \mathcal{A}_2, \ldots$ of those atomic formulas in $\mathcal{L}$ whose only variables are $x_1 \ldots x_n$—so a member of the enumeration is either a sentence letter $\mathcal{S}$, or an atomic $\mathcal{R}t_a \ldots t_b$ such that the only variables in $t_a \ldots t_b$ are from $x_1 \ldots x_n$. With $\mathcal{C}_0$ and d as above, set $\mathcal{C}_i = \mathcal{C}_{i-1} \wedge \mathcal{A}_i$ if $\mathsf{D}_\mathsf{d}[\mathcal{A}_i] = \mathsf{S}$, and otherwise $\mathcal{C}_i = \mathcal{C}_{i-1} \wedge \sim\!\mathcal{A}_i$. It is easy to see that for any i, $\mathsf{D}_\mathsf{d}[\mathcal{C}_i] = \mathsf{S}$. The argument is by induction on the sequence $\mathcal{C}_0, \mathcal{C}_1, \ldots$.

Basis: For any a and b such that $1 \le a < b \le n$, since d assigns to x_a and x_b distinct members of U_D, $\mathsf{D}_\mathsf{d}[x_a \neq x_b] = \mathsf{S}$; so by repeated applications of SF$'$($\wedge$), $\mathsf{D}_\mathsf{d}[\bigwedge_{1 \le i < j \le n} x_i \neq x_j] = \mathsf{S}$. And since each member of U_D is assigned to some variable in $x_1 \ldots x_n$, for any $\mathsf{m} \in \mathsf{U}_\mathsf{D}$, there is some a, $1 \le a \le n$ such that $\mathsf{D}_{\mathsf{d}(v|\mathsf{m})}[v = x_a] = \mathsf{S}$; then by repeated applications of SF$'$($\vee$), $\mathsf{D}_{\mathsf{d}(v|\mathsf{m})}[\bigvee_{1 \le i \le n} v = x_i] = \mathsf{S}$; and since this is so for every $\mathsf{m} \in \mathsf{U}_\mathsf{D}$, by SF($\forall$), $\mathsf{D}_\mathsf{d}[\forall v \bigvee_{1 \le i \le n} v = x_i] = \mathsf{S}$. So by SF$'$($\wedge$), $\mathsf{D}_\mathsf{d}[\mathcal{C}_0] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, $\mathsf{D}_\mathsf{d}[\mathcal{C}_i] = \mathsf{S}$.

Show: $\mathsf{D}_\mathsf{d}[\mathcal{C}_k] = \mathsf{S}$.

$\mathcal{C}_k$ is of the form $\mathcal{C}_{k-1} \wedge \mathcal{A}_k$ or $\mathcal{C}_{k-1} \wedge \sim\!\mathcal{A}_k$. In the first case, by assumption, $\mathsf{D}_\mathsf{d}[\mathcal{C}_{k-1}] = \mathsf{S}$, and by construction, $\mathsf{D}_\mathsf{d}[\mathcal{A}_k] = \mathsf{S}$; so by SF$'$($\wedge$), $\mathsf{D}_\mathsf{d}[\mathcal{C}_{k-1} \wedge \mathcal{A}_k] = \mathsf{S}$; which is to say, $\mathsf{D}_\mathsf{d}[\mathcal{C}_k] = \mathsf{S}$. In the second case, again $\mathsf{D}_\mathsf{d}[\mathcal{C}_{k-1}] = \mathsf{S}$; and by construction, $\mathsf{D}_\mathsf{d}[\mathcal{A}_k] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{D}_\mathsf{d}[\sim\!\mathcal{A}_k] = \mathsf{S}$; so by SF$'$($\wedge$), $\mathsf{D}_\mathsf{d}[\mathcal{C}_{k-1} \wedge \sim\!\mathcal{A}_k] = \mathsf{S}$; which is to say, $\mathsf{D}_\mathsf{d}[\mathcal{C}_k] = \mathsf{S}$.

Indct: For any i, $\mathsf{D}_\mathsf{d}[\mathcal{C}_i] = \mathsf{S}$.

This paves the way to construct Σ, and show that it is true on D. For any $\mathcal{C}_i$, let its *existential closure* $\mathcal{C}_i^e = \exists x_1 \dots \exists x_n \mathcal{C}_i$; and set $\Sigma = \{\mathcal{C}_i^e \mid i \geq 0\}$—so $\Sigma = \{\mathcal{C}_0^e, \mathcal{C}_1^e, \dots\}$. Then it is easy to see that each $\mathcal{C}_i^e$, and so Σ itself, is true on D:

> Suppose otherwise, that not every $\mathcal{C}_i^e$ is true on D; then for some i, $\mathsf{D}[\exists x_1 \dots \exists x_n \mathcal{C}_i] \neq \mathsf{T}$; so by TI, there is some assignment d' such that $\mathsf{D}_{\mathsf{d}'}[\exists x_1 \dots \exists x_n \mathcal{C}_i] \neq \mathsf{S}$; so, since there are no free variables, by T8.5, $\mathsf{D}_{\mathsf{d}}[\exists x_1 \dots \exists x_n \mathcal{C}_i] \neq \mathsf{S}$; then repeatedly removing an x_a-quantifier by SF$'(\exists)$ leaves the formula unsatisfied for every assignment to x_a and so unsatisfied on d itself, so that $\mathsf{D}_{\mathsf{d}}[\mathcal{C}_i] \neq \mathsf{S}$; but as have just seen, this is impossible; reject the assumption: every $\mathcal{C}_i^e$ is true on D, and so $\mathsf{D}[\Sigma] = \mathsf{T}$.

Now consider an arbitrary $\mathsf{H} \in \mathfrak{Mod}(\Sigma)$; we set out to show that $\mathsf{D} \cong \mathsf{H}$. Since $\mathsf{H} \in \mathfrak{Mod}(\Sigma)$, $\mathsf{H}[\Sigma] = \mathsf{T}$; so $\mathsf{H}[\mathcal{C}_0^e] = \mathsf{T}$, that is $\mathsf{H}[\exists x_1 \dots \exists x_n \mathcal{C}_0] = \mathsf{T}$; and, as we have already noted, this can be the case iff there are exactly n members of $\mathsf{U_H}$. We set out to find an assignment h such that for any i, $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$. Given d and this h, our idea is to let ι consist of pairs $\langle \mathsf{d}[x], \mathsf{h}[x] \rangle$, and with this to show $\mathsf{D} \overset{\iota}{\cong} \mathsf{H}$.

For some assignment k into $\mathsf{U_H}$, let h range over assignments that differ from k at most in assignments to $x_1 \dots x_n$. Set $\Omega_i = \{\mathsf{h} \mid \mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}\}$, and $\Omega = \bigcap_{i \geq 0} \Omega_i$. So each Ω_i is the set of all assignments h on which the open formula $\mathcal{C}_i$ is satisfied, and Ω collects assignments that are common to them all.

(i) No Ω_i is empty. Since $\mathsf{H}[\Sigma] = \mathsf{T}$, $\mathsf{H}[\exists x_1 \dots \exists x_n \mathcal{C}_i] = \mathsf{T}$; so by TI, that formula is satisfied on any assignment; in particular for the assignment k into $\mathsf{U_H}$, $\mathsf{H_k}[\exists x_1 \dots \exists x_n \mathcal{C}_i] = \mathsf{S}$; so by repeated applications of SF$'(\exists)$, there is some h such that $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$. When the quantifiers come off, the result is some assignment that differs at most in assignments to $x_1 \dots x_n$ and so some assignment h.

(ii) For any $j \geq i$, $\Omega_j \subseteq \Omega_i$. Intuitively, since $\mathcal{C}_j$ adds conjuncts to $\mathcal{C}_i$, $\mathcal{C}_j$ is satisfied by fewer assignments than $\mathcal{C}_i$—so that Ω_j is reduced relative to Ω_i. Suppose otherwise, that $j \geq i$ but $\Omega_j \not\subseteq \Omega_i$; then there is some h such that $\mathsf{h} \in \Omega_j$ but $\mathsf{h} \notin \Omega_i$; so by construction, $\mathsf{H_h}[\mathcal{C}_j] = \mathsf{S}$ but $\mathsf{H_h}[\mathcal{C}_i] \neq \mathsf{S}$; if $j = i$ this is impossible; so suppose $j > i$; then $\mathcal{C}_j$ is of the sort, $\mathcal{C}_i \wedge \mathcal{B}_{i+1} \wedge \mathcal{B}_{i+2} \wedge \dots \wedge \mathcal{B}_j$ where $\mathcal{B}_{i+1} \dots \mathcal{B}_j$ are either atomics or negated atomics; so by repeated application of SF$'(\wedge)$, $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$. This is impossible; reject the assumption: $\Omega_j \subseteq \Omega_i$.

(iii) There are at most finitely many assignments of the sort h: Since any h differs from k at most in assignments to $x_1 \dots x_n$, and there are just n members of $\mathsf{U_H}$, there are n^n assignments of the sort h; so there are finitely many assignments h.

From these results it follows that Ω is non-empty:

Suppose otherwise, that no h is a member of each Ω_i; then for any h, there is some Ω_i such that $\mathsf{h} \notin \Omega_i$. For each h consider the least a such that $\mathsf{h} \notin \Omega_a$ and let A be the set of these subscripts; then for any h there is some $a \in A$ such that $\mathsf{h} \notin \Omega_a$. Since by (iii) there are finitely many assignments, A has finitely many members; let b be the maximum of the members in A and consider Ω_b; by (i) there is some $\mathsf{h} \in \Omega_b$; but for every $a \in A$, $b \geq a$ so by (ii) $\Omega_b \subseteq \Omega_a$; so $\mathsf{h} \in \Omega_a$; so there is no $a \in A$ such that $\mathsf{h} \notin \Omega_a$. This is impossible; reject the assumption: Ω is not empty.

So we have what we wanted: assignments in Ω are such that for any i, $\mathsf{H_h}[\mathscr{C}_i] = \mathsf{S}$; and since Ω is non-empty there must exist an assignment h such that for any i, $\mathsf{H_h}[\mathscr{C}_i] = \mathsf{S}$. And we are ready for the result at which we have been aiming.

*T11.32. If D is a finite model and $\mathfrak{D}$ is the class of all models isomorphic to it, then there is a categorical Σ such that $\mathfrak{Mod}(\Sigma) = \mathfrak{D}$.

Suppose D is a finite model and $\mathfrak{D}$ is the class of all models isomorphic to it. Let Σ be as above and suppose $\mathsf{H} \in \mathfrak{Mod}(\Sigma)$. Then, as above, there are assignments d and h such that for each i, $\mathsf{D_d}[\mathscr{C}_i] = \mathsf{S}$ and $\mathsf{H_h}[\mathscr{C}_i] = \mathsf{S}$. For $1 \leq i \leq n$, let ι have members $\langle \mathsf{d}[x_i], \mathsf{h}[x_i]\rangle$—so that $\mathsf{h}[x_i] = \iota(\mathsf{d}[x_i])$. Since $\mathsf{H_h}[\mathscr{C}_0] = \mathsf{S}$, h assigns each x_i a different member of $\mathsf{U_H}$ and each member of $\mathsf{U_H}$ is assigned to some x_i; so ι is 1:1 and onto $\mathsf{U_H}$, as it should be. We now set out to show that the other conditions for isomorphism are met.

(s) If $\mathscr{S}$ is a sentence letter, then $\mathscr{S}$ is some $\mathscr{A}_i$. (i) Suppose $\mathsf{D}[\mathscr{S}] = \mathsf{T}$; then $\mathsf{D}[\mathscr{A}_i] = \mathsf{T}$; so by SF(s), $\mathsf{D_d}[\mathscr{A}_i] = \mathsf{S}$; and by construction, $\mathscr{A}_i$ is a conjunct of $\mathscr{C}_i$. But $\mathsf{H_h}[\mathscr{C}_i] = \mathsf{S}$; so by SF$'$($\wedge$), $\mathsf{H_h}[\mathscr{A}_i] = \mathsf{S}$; so by SF(s), $\mathsf{H}[\mathscr{A}_i] = \mathsf{T}$; so $\mathsf{H}[\mathscr{S}] = \mathsf{T}$. (ii) Suppose $\mathsf{D}[\mathscr{S}] \neq \mathsf{T}$; then $\mathsf{D}[\mathscr{A}_i] \neq \mathsf{T}$; so by SF(s), $\mathsf{D_d}[\mathscr{A}_i] \neq \mathsf{S}$; and by construction, $\sim\mathscr{A}_i$ is a conjunct of $\mathscr{C}_i$. But $\mathsf{H_h}[\mathscr{C}_i] = \mathsf{S}$; so by SF$'$($\wedge$), $\mathsf{H_h}[\sim\mathscr{A}_i] = \mathsf{S}$; so by SF($\sim$), $\mathsf{H_h}[\mathscr{A}_i] \neq \mathsf{S}$; so by SF(s), $\mathsf{H}[\mathscr{A}_i] \neq \mathsf{T}$; so $\mathsf{H}[\mathscr{S}] \neq \mathsf{T}$. So $\mathsf{D}[\mathscr{S}] = \mathsf{H}[\mathscr{S}]$.

(c) If c is a constant, we require $\mathsf{H}[c] = \iota(\mathsf{D}[c])$. For some $\mathsf{m_i}$, $\mathsf{D}[c] = \mathsf{m_i}$; and since $\mathsf{d}[x_i] = \mathsf{m_i}$, $\mathsf{h}[x_i] = \iota(\mathsf{d}[x_i]) = \iota(\mathsf{m_i})$.

By TA(c), $\mathsf{D_d}[c] = \mathsf{D}[c] = \mathsf{m_i}$; and by TA(v), $\mathsf{D_d}[x_i] = \mathsf{d}[x_i] = \mathsf{m_i}$; so $\mathsf{D_d}[c] = \mathsf{D_d}[x_i]$; so $\langle \mathsf{D_d}[c], \mathsf{D_d}[x_i]\rangle \in \mathsf{D}[=]$; so by SF(r), $\mathsf{D_d}[c = x_i] = \mathsf{S}$; so $c = x_i$ is a conjunct in some $\mathscr{C}_n$. But $\mathsf{H_h}[\mathscr{C}_n] = \mathsf{S}$; so by SF$'$($\wedge$), $\mathsf{H_h}[c = x_i] = \mathsf{S}$; so by SF(r), $\langle \mathsf{H_h}[c], \mathsf{H_h}[x_i]\rangle \in \mathsf{H}[=]$; so $\mathsf{H_h}[c] = \mathsf{H_h}[x_i]$; but by TA(c), $\mathsf{H_h}[c] = \mathsf{H}[c]$, and by TA(v), $\mathsf{H_h}[x_i] = \mathsf{h}[x_i]$; so $\mathsf{H}[c] = \mathsf{h}[x_i]$; so $\mathsf{H}[c] = \iota(\mathsf{m_i}) = \iota(\mathsf{D}[c])$.

(f) For simplicity, consider a one-place function symbol h. For any $\mathsf{m_a}$, we require $\mathsf{H}[h]\langle \iota(\mathsf{m_a})\rangle = \iota(\mathsf{D}[h]\langle \mathsf{m_a}\rangle)$. For some $\mathsf{m_b}$, (*) $\mathsf{D}[h]\langle \mathsf{m_a}\rangle = \mathsf{m_b}$. And since $\mathsf{d}[x_a] = \mathsf{m_a}$ and $\mathsf{d}[x_b] = \mathsf{m_b}$, (**) both $\mathsf{h}[x_a] = \iota(\mathsf{d}[x_a]) = \iota(\mathsf{m_a})$ and $\mathsf{h}[x_b] = \iota(\mathsf{d}[x_b]) = \iota(\mathsf{m_b})$.

With TA(v), $\mathsf{D_d}[x_a] = \mathsf{d}[x_a] = \mathsf{m_a}$ and $\mathsf{D_d}[x_b] = \mathsf{d}[x_b] = \mathsf{m_b}$; from this with with TA(f) and (∗), $\mathsf{D_d}[\hbar x_a] = \mathsf{D}[\hbar]\langle \mathsf{D_d}[x_a]\rangle = \mathsf{D}[\hbar]\langle \mathsf{m_a}\rangle = \mathsf{m_b} = \mathsf{D_d}[x_b]$; so $\mathsf{D_d}[\hbar x_a] = \mathsf{D_d}[x_b]$; so $\langle \mathsf{D_d}[\hbar x_a], \mathsf{D_d}[x_b]\rangle \in \mathsf{D}[=]$; so by SF(r), $\mathsf{D_d}[\hbar x_a = x_b] = \mathsf{S}$; so $\hbar x_a = x_b$ is a conjunct of some $\mathscr{C}_n$. But $\mathsf{H_h}[\mathscr{C}_n] = \mathsf{S}$; so by SF′(∧), $\mathsf{H_h}[\hbar x_a = x_b] = \mathsf{S}$; so by SF(r), $\langle \mathsf{H_h}[\hbar x_a], \mathsf{H_h}[x_b]\rangle \in \mathsf{H}[=]$; so $\mathsf{H_h}[\hbar x_a] = \mathsf{H_h}[x_b]$; and by TA(f), $\mathsf{H_h}[\hbar x_a] = \mathsf{H}[\hbar]\langle \mathsf{H_h}[x_a]\rangle$; so $\mathsf{H}[\hbar]\langle \mathsf{H_h}[x_a]\rangle = \mathsf{H_h}[x_b]$. But by TA(v) with (∗∗), $\mathsf{H_h}[x_a] = \mathsf{h}[x_a] = \iota(\mathsf{m_a})$; and $\mathsf{H_h}[x_b] = \mathsf{h}[x_b] = \iota[\mathsf{m_b}]$; so $\mathsf{H}[\hbar]\langle\iota(\mathsf{m_a})\rangle = \iota(\mathsf{m_b})$; so with (∗), $\mathsf{H}[\hbar]\langle\iota(\mathsf{m_a})\rangle = \iota(\mathsf{D}[\hbar]\langle \mathsf{m_a}\rangle)$.

(r) For simplicity consider a one-place relation symbol $\mathscr{R}$. For any $\mathsf{m_a}$, we require $\iota(\mathsf{m_a}) \in \mathsf{H}(\mathscr{R})$ iff $\mathsf{m_a} \in \mathsf{D}[\mathscr{R}]$. (i) Suppose $\mathsf{m_a} \in \mathsf{D}[\mathscr{R}]$; by [homework], $\iota(\mathsf{m_a}) \in \mathsf{H}[\mathscr{R}^n]$. (ii) Suppose $\mathsf{m_a} \notin \mathsf{D}[\mathscr{R}]$; by [homework] $\iota(\mathsf{m_a}) \notin \mathsf{H}[\mathscr{R}]$.

This is an interesting result! Since every $\mathsf{D}, \mathsf{H} \in \mathfrak{Md}(\Sigma)$ is such that $\mathsf{D} \cong \mathsf{H}$, Σ is categorical; so by the corollary to T11.16, Σ is $\tilde{\text{c}}$omplete. Many structures, including some from abstract algebra, have a finite domain—although most of the structures we shall care about do not. Even so, we have a first case where $\tilde{\text{c}}$ompleteness is possible.

*E11.25. Complete the demonstration of T11.32 by completing the case for relation symbols.

E11.26. Consider a language with no function symbols or constants and just relation symbols A^1, B^2 (and =). Let $\mathsf{U_D} = \{1, 2\}$, $\mathsf{D}[A] = \{1\}$, and $\mathsf{D}[B] = \{\langle 1, 1\rangle, \langle 1, 2\rangle\}$. (i) By the method of this section, find Σ to axiomatize $\mathfrak{D}$, the class of all models isomorphic to D. (ii) As a sample for $\mathring{\text{c}}$ompleteness, show that $\Sigma \vdash \forall x(Ax \to \exists yBxy)$. Hint: Σ should have as members some $\mathscr{C}^e_0 \ldots \mathscr{C}^e_{10}$.

Quantifier Elimination

Consider a sentential language with just two sentence letters A and B. Suppose $\Sigma = \{A, \sim B\}$. On a truth table, there is just one row were the members of Σ are both true, and on that row, any $\mathscr{P}$ in the language is either T or F, so that one of $\mathscr{P}$ or $\sim\mathscr{P}$ is T.

(P)

A	B	A	$\sim B$	/	$\mathscr{P}$	$\sim\mathscr{P}$
T	T	***T***	***F***		–	–
T	F	***T***	***T***		***T/F***	***F/T***
F	T	***F***	***F***		–	–
F	F	***F***	***T***		–	–

So for any $\mathscr{P}$, either $\Sigma \vDash \mathscr{P}$ or $\Sigma \vDash \sim\mathscr{P}$. But by $\mathring{\text{c}}$ompleteness, if $\Gamma \vDash \mathscr{P}$ then $\Gamma \vdash \mathscr{P}$; so for any $\mathscr{P}$, either $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$, and Σ is $\tilde{\text{c}}$omplete. Alternatively, from $\Sigma \vdash A$ and $\Sigma \vdash \sim B$, one might reason by induction on the number of operators in $\mathcal{P}$ that $\Sigma \vdash \mathcal{P}$ or $\Sigma \vdash \sim\mathcal{P}$. Either way Σ is c̃omplete. Of course, this case is not very interesting!

Still, if we could show that things are "like this" for some more interesting Σ, then we could show that Σ is c̃omplete. That is the strategy of quantifier elimination: Say Σ *admits quantifier elimination* just in case for each formula $\mathcal{P}$ in its language $\mathcal{L}$, there is some quantifier-free $\mathcal{Q}$ with the same free variables as $\mathcal{P}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$. Observe that such a $\mathcal{P}$ and $\mathcal{Q}$ need not be equivalent in the sense that they take the same value for any M and d—rather, when Σ is true then they are satisfied or not together. All the same, suppose Σ admits quantifier elimination; then for any sentence $\mathcal{P}$, there is a quantifier-free sentence $\mathcal{Q}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$, and so by c̊ompleteness, such that $\Sigma \vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. Suppose further that $\Sigma \vdash \mathcal{A}$ or $\Sigma \vdash \sim\mathcal{A}$ for atomic sentences of its language; then by reasoning as above, $\Sigma \vdash \mathcal{Q}$ or $\Sigma \vdash \sim\mathcal{Q}$ for quantifier-free sentences of $\mathcal{L}$. From these together, by $\leftrightarrow$E or NB, $\Sigma \vdash \mathcal{P}$ or $\Sigma \vdash \sim\mathcal{P}$, and Σ is c̃omplete. It is not always the case that when Σ is c̃omplete, it may be shown to be c̃omplete by quantifier elimination (and quantifier elimination is not the only approach). Still, the method applies for some cases that we shall care about. We have seen how a Σ might prove $\mathcal{A}$ or $\sim\mathcal{A}$ for atomic sentences of its language (think about Q). The trick, then, is to see how (in the world) an interesting Σ including quantified expressions admits quantifier elimination. We make a start with the following theorem:

T11.33. If every $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ with $\mathcal{A}_1 \ldots \mathcal{A}_n$ atomic or negated atomic has a quantifier-free $\mathcal{Q}$ with the same free variables such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$, then Σ admits quantifier elimination.

Suppose every $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ with $\mathcal{A}_1 \ldots \mathcal{A}_n$ atomic or negated atomic has a quantifier-free $\mathcal{Q}$ with the same free variables such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$. By induction on the number of operator symbols,

Basis: Suppose $\mathcal{P}$ has no operator symbols; then $\mathcal{P}$ is an atomic; let $\mathcal{Q}$ be the same formula. Then $\mathcal{Q}$ is quantifier-free has the same free variables as $\mathcal{P}$. But $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{P}$; so $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then there is a quantifier-free $\mathcal{Q}$ with the same free variables as $\mathcal{P}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

Show: If $\mathcal{P}$ has k operator symbols, then there is a quantifier-free $\mathcal{Q}$ with the same free variables as $\mathcal{P}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$, or $\exists x\mathcal{A}$ for $\mathcal{A}$, $\mathcal{B}$ with $< k$ operator symbols (treating $\forall x\mathcal{A}$ as equivalent to $\sim\exists x\sim\mathcal{A}$).

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$. By assumption there is some quantifier-free $\mathcal{B}$ with the same free variables as $\mathcal{A}$ such that $\Sigma \vDash \mathcal{A} \leftrightarrow \mathcal{B}$; let $\mathcal{Q}$ be $\sim\mathcal{B}$. It remains that $\sim\mathcal{B}$ is quantifier-free and has the same free variables as $\sim\mathcal{A}$. Further, it is easy to see that $\Sigma \vDash \sim\mathcal{A} \leftrightarrow \sim\mathcal{B}$; so $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

($\to$) $\mathcal{P}$ is $\mathcal{A} \to \mathcal{B}$. Homework.

($\exists$) $\mathcal{P}$ is $\exists x \mathcal{A}$. By assumption there is some quantifier-free $\mathcal{B}$ with the same free variables as $\mathcal{A}$ such that $\Sigma \vDash \mathcal{A} \leftrightarrow \mathcal{B}$. Then it is easy to see that $\Sigma \vDash \exists x \mathcal{A} \leftrightarrow \exists x \mathcal{B}$—although, of course, $\exists x \mathcal{B}$ is not yet quantifier-free. By reasoning from T8.1 as applied to SF (rather than ST), for any quantifier-free $\mathcal{B}$ there is a $\mathcal{B}_{\mathrm{N}}$ in normal form with the same free variables such that $\mathsf{I}_{\mathsf{d}}[\mathcal{B}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}}[\mathcal{B}_{\mathrm{N}}] = \mathsf{S}$. And as developed in E8.22 for each $\mathcal{B}_{\mathrm{N}}$ there is a formula $\mathcal{B}_{\mathrm{D}}$ in *disjunctive normal form*,

$$\mathcal{B}_{\mathrm{D}} = (\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_d) \vee (\mathcal{E}_1 \wedge \ldots \wedge \mathcal{E}_e) \vee \ldots \vee (\mathcal{F}_1 \wedge \ldots \wedge \mathcal{F}_f)$$

with each $\mathcal{D}_i$ and $\mathcal{E}_i$ and ... and $\mathcal{F}_i$ atomic or negated atomic, where free variables the same, and $\mathsf{I}_{\mathsf{d}}[\mathcal{B}_{\mathrm{N}}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}}[\mathcal{B}_{\mathrm{D}}] = \mathsf{S}$. From these together, for any quantifier-free $\mathcal{B}$, there is a $\mathcal{B}_{\mathrm{D}}$ in disjunctive normal form with the same free variables such that $\mathsf{I}_{\mathsf{d}}[\mathcal{B}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}}[\mathcal{B}_{\mathrm{D}}] = \mathsf{S}$, and given that I and d are arbitrary such that $\vDash \mathcal{B} \leftrightarrow \mathcal{B}_{\mathrm{D}}$.[12] Then with T9.10,

$$\Sigma \vDash \exists x \mathcal{A} \leftrightarrow \exists x[(\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_d) \vee (\mathcal{E}_1 \wedge \ldots \wedge \mathcal{E}_e) \vee \ldots \vee (\mathcal{F}_1 \wedge \ldots \wedge \mathcal{F}_f)]$$

and with (a semantic version of) QD,

$$\Sigma \vDash \exists x \mathcal{A} \leftrightarrow [\exists x(\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_d) \vee \exists x(\mathcal{E}_1 \wedge \ldots \wedge \mathcal{E}_e) \vee \ldots \vee \exists x(\mathcal{F}_1 \wedge \ldots \wedge \mathcal{F}_f)]$$

But by the assumption to the theorem, there is some quantifier-free $\mathcal{Q}_d$ with the same free variables as $\exists x(\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_d)$ such that $\Sigma \vDash \exists x(\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_d) \leftrightarrow \mathcal{Q}_d$ and ... and there is a quantifier-free $\mathcal{Q}_f$ with the same free variables as $\exists x(\mathcal{F}_1 \wedge \ldots \wedge \mathcal{F}_f)$ such that $\Sigma \vDash \exists x(\mathcal{F}_1 \wedge \ldots \wedge \mathcal{F}_f) \leftrightarrow \mathcal{Q}_f$. So by T9.10 again, $\Sigma \vDash \exists x \mathcal{A} \leftrightarrow (\mathcal{Q}_d \vee \mathcal{Q}_e \vee \ldots \vee \mathcal{Q}_f)$; so $\mathcal{Q} = \mathcal{Q}_d \vee \mathcal{Q}_e \vee \ldots \vee \mathcal{Q}_f$ is quantifier-free and has the same free variables as $\exists x \mathcal{A}$; and $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

If $\mathcal{P}$ has k operator symbols, then there is a quantifier-free $\mathcal{Q}$ with the same free variables as $\mathcal{P}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

Indct: For every $\mathcal{P}$ there is a quantifier-free $\mathcal{Q}$ with the same free variables as $\mathcal{P}$ such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$.

So the project of showing that Σ admits quantifier elimination reduces to the project of showing that each $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ with $\mathcal{A}_1 \ldots \mathcal{A}_n$ atomic or negated atomic has a quantifier-free $\mathcal{Q}$ with the same free variables such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$. We turn now to showing that some theories in fact admit quantifier elimination.

[12]It is also possible to appeal to reasoning from T11.1: On any I and d, a quantifier-free $\mathcal{B}$ and its atomics are satisfied or not, and by the method of the theorem there is a $\mathcal{B}_{\mathrm{D}}$ in disjunctive normal form satisfied under the same conditions.

E11.27. Consider the sentential language whose only sentence letters are A and B, where $\Sigma \vdash A$ and $\Sigma \vdash \sim B$. Produce the argument by induction on the number of operator symbols to show that $\Sigma \vdash \mathscr{P}$ or $\Sigma \vdash \sim\mathscr{P}$.

E11.28. Provide a demonstration of the $(\rightarrow)$ case for T11.33 in which you work through all the semantic details.

Theory S. Consider a language like $\mathscr{L}_{\text{NT}}$ whose only function symbol is S. Then terms are of the sort $S^n t$ where S^n indicates n instances of S and t is a variable or $\emptyset$; atomic formulas are $S^m s = S^n t$ again where s and t are the constant $\emptyset$ or a variable. Let S be a set whose members are as follows:

(S1) $Sx \neq \emptyset$

(S2) $Sx = Sy \rightarrow x = y$

(S3) $x \neq \emptyset \rightarrow \exists y(x = Sy)$

(S4) $S^n x \neq x$ for any $n \geq 1$

As from E7.17, S1 and S2 require that the universe have infinitely many members. And this theory is s̃ound on a standard interpretation with universe $\mathbb{N}$, 0 assigned to $\emptyset$, and the interpretation of S the successor function. Observe that there are infinitely many axioms corresponding to instances of S4. From these axioms we have as theorems,

(Sa) If $m = n$ then $\mathsf{S} \vdash S^m x = S^n x$.

(Sb) If $m \neq n$ then $\mathsf{S} \vdash S^m x \neq S^n x$.

(Sc) $\mathsf{S} \vdash S^a s = t \leftrightarrow S^{d+a} s = S^d t$.

(Sd) $\mathsf{S} \vdash \exists x(S^n x = t) \leftrightarrow (t = t \wedge t \neq S^0\emptyset \wedge t \neq S^1\emptyset \wedge \ldots \wedge t \neq S^{n-1}\emptyset)$ – x not in t.

The left side of Sd requires that $t \geq n$ and so "big enough" that n successors of some x may be equal to it. Intuitively, in our limited vocabulary, the right side requires the same (and with the first conjunct the expression remains defined when n is zero). For hints see the associated exercise, E11.29. By s̊oundness, these theorems are true on models of S.

To show that S admits quantifier elimination, consider $\mathscr{P} = \exists x(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_n)$, where $\mathscr{A}_1 \ldots \mathscr{A}_n$ are $S^m s = S^n t$ or $S^m s \neq S^n t$ and s and t are the constant $\emptyset$ or a variable. We require a quantifier-free $\mathscr{Q}$ with the same free variables as $\mathscr{P}$ such that $\mathsf{S} \vDash \mathscr{P} \leftrightarrow \mathscr{Q}$. For this, consider an arbitrary M such that $\mathsf{M}[\mathsf{S}] = \mathsf{T}$; then to obtain $\mathsf{M}[\mathscr{P} \leftrightarrow \mathscr{Q}] = \mathsf{T}$, we require that for arbitrary d, $\mathsf{M_d}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathscr{Q}] = \mathsf{S}$.

As a preliminary, consider $\exists x(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_i \wedge \ldots \wedge \mathscr{A}_n)$ with $\mathsf{S} \vdash \mathscr{A}_i \leftrightarrow \mathscr{B}$. By s̊oundness $\mathsf{S} \vDash \mathscr{A}_i \leftrightarrow \mathscr{B}$; so $\mathsf{M}[\mathscr{A}_i \leftrightarrow \mathscr{B}] = \mathsf{T}$ and $\mathsf{M_d}[\mathscr{A}_i \leftrightarrow \mathscr{B}] = \mathsf{S}$; so from T9.10, $\mathsf{M_d}[\exists x(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_i \wedge \ldots \wedge \mathscr{A}_n)] = \mathsf{S}$ iff $\mathsf{M_d}[\exists x(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{B} \wedge \ldots \wedge \mathscr{A}_n)] = \mathsf{S}$. In such cases, I typically cite just the original theorem, $\mathsf{S} \vdash \mathcal{A}_i \leftrightarrow \mathcal{B}$, and move directly to the result.

For $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$, first suppose x does not appear in some $\mathcal{A}_i$; then,

(Q)
$$\mathsf{M_d}[\exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_i \wedge \ldots \wedge \mathcal{A}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\mathcal{A}_i \wedge \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n)] = \mathsf{S}$$

$\mathcal{A}_i$ moves to the front by standard quantifier placement rules (QP). And free variables remain the same. So if we reduce the second conjunct to quantifier-free form, we will have reduced the whole.

Concentrating then on the second conjunct, consider $\exists x(\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_n)$ where x is a component of each $\mathcal{B}_i$. Suppose $\mathcal{B}_i$ is of the sort $S^m x = S^n x$. First let $m = n$; then,

(R)
$$\mathsf{M_d}[\exists x(\mathcal{B}_1 \wedge \ldots \wedge S^m x = S^n x \wedge \ldots \wedge \mathcal{B}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(\mathcal{B}_1 \wedge \ldots \wedge \emptyset = \emptyset \wedge \ldots \wedge \mathcal{B}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\emptyset = \emptyset \wedge \exists x(\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_{i-1} \wedge \mathcal{B}_{i+1} \wedge \ldots \wedge \mathcal{B}_n)] = \mathsf{S}$$

$S^m x = S^n x$ is replaced by $\emptyset = \emptyset$ and that variable-free conjunct moved to the front. For the first move: Since $m = n$, by Sa we get $\mathsf{S} \vdash S^m x = S^n x$; and again by Sa, $\mathsf{S} \vdash \emptyset = \emptyset$; so $\mathsf{S} \vdash S^m x = S^n x \leftrightarrow \emptyset = \emptyset$. Then the second move is by quantifier-placement rules. And similarly using Sb to replace $S^m x = S^n x$ with $\emptyset \neq \emptyset$ when $m \neq n$. For a negated atomic $\sim(S^m x = S^n x)$, replace $S^m x = S^n x$, and move the negation of it outside. Again, free variables remain the same.

Concentrating again on the second conjunct, consider $\exists x(\mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_n)$ where each $\mathcal{C}_i$ is $S^n x = t_i$ or $S^n x \neq t_i$ and x does not appear in t_i. Suppose first that each $\mathcal{C}_i$ is of the sort $S^n x \neq t_i$; then,

(S)
$$\mathsf{M_d}[\exists x(S^a x \neq t_1 \wedge \ldots \wedge S^b x \neq t_n) = \mathsf{S}$$
iff
$$\mathsf{M_d}[t_1 = t_1 \wedge \ldots \wedge t_n = t_n] = \mathsf{S}$$

The quantifier is dropped and each $S^n x \neq t_i$ is replaced by $t_i = t_i$. Then free variables remain the same. And consider the objects $\mathsf{m_1} \ldots \mathsf{m_n}$ assigned to $t_1 \ldots t_n$; on an infinite domain there is sure to be an object different from each $\mathsf{m_i}$ and so an object to satisfy the existential quantification; and since there is such an object, the upper existential quantification is satisfied on $\mathsf{M_d}$; and trivially the lower formula is satisfied as well; so S entails the biconditional between the two.

Suppose then that some $\mathcal{C}_i$ is $S^a x = t_i$ where x does not appear in t_i. Then beginning with that equality as the first conjunct,

(T)
$$\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^b x = t_2 \wedge \ldots \wedge S^c x = t_i \wedge S^d x \neq t_{i+1} \wedge \ldots \wedge S^e x \neq t_j))] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^{b+a} x = S^a t_2 \wedge \ldots \wedge S^{c+a} x = S^a t_i \wedge S^{d+a} x \neq S^a t_{i+1} \wedge \ldots \wedge S^{e+a} x \neq S^a t_j))] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^b t_1 = S^a t_2 \wedge \ldots \wedge S^c t_1 = S^a t_i \wedge S^d t_1 \neq S^a t_{i+1} \wedge \ldots \wedge S^e t_1 \neq S^a t_j))] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(S^a x = t_1) \wedge (S^b t_1 = S^a t_2 \wedge \ldots \wedge S^c t_1 = S^a t_i \wedge S^d t_1 \neq S^a t_{i+1} \wedge \ldots \wedge S^e t_1 \neq S^a t_j)] = \mathsf{S}$$

Superscripts on terms not in the first conjunct are increased by the superscript from the first; then t_1 is substituted for $S^a x$ in conjuncts other than the first; thus x appears just in the first term, and the quantifier is restricted just to it. The equivalences are by Sc; then application of the equality from the first conjunct to the other members; and finally by quantifier placement. Again, free variables remain the same.

Finally, concentrating on the first conjunct,

(U)
$$\begin{gathered}\mathsf{M_d}[\exists x(S^a x = t_1)] = \mathsf{S} \\ \text{iff} \\ \mathsf{M_d}[t_1 = t_1 \wedge t_1 \neq S^0\emptyset \wedge t_1 \neq S^1\emptyset \wedge \ldots \wedge t_1 \neq S^{a-1}\emptyset] = \mathsf{S}.\end{gathered}$$

the quantification is reduced to quantifier-free form by Sd. Again, free variables remain the same.

Thus the original $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ is reduced to quantifier-free $\mathcal{Q}$ with the same free variables such that S $\vDash \mathcal{P} \leftrightarrow \mathcal{Q}$. So by T11.33, S admits quantifier elimination. And now it is easy to see that S is $\tilde{\text{c}}$omplete.

T11.34. S is $\tilde{\text{c}}$omplete.

> Since S admits quantifier elimination, for any $\mathcal{P}$ there is a quantifier-free $\mathcal{Q}$ with the same free variables such that S $\vDash \mathcal{P} \leftrightarrow \mathcal{Q}$; so by $\mathring{\text{c}}$ompleteness S $\vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. Suppose $\mathcal{P}$ is a sentence; then $\mathcal{Q}$ is a sentence. Atomic sentences in the language of S are of the sort $S^m\emptyset = S^n\emptyset$; so by Sa and Sb, for any atomic sentence $\mathcal{A}$, S $\vdash \mathcal{A}$ or S $\vdash \sim\mathcal{A}$; so by a simple induction on number of operator symbols, for any quantifier-free sentence $\mathcal{Q}$, S $\vdash \mathcal{Q}$ or S $\vdash \sim\mathcal{Q}$; so by $\leftrightarrow$E or NB, S $\vdash \mathcal{P}$ or S $\vdash \sim\mathcal{P}$; so S is $\tilde{\text{c}}$omplete.

S is a particularly simple theory with a particularly simple language. Still, its universe is infinite. And it is of considerable interest to have established $\tilde{\text{c}}$ompleteness for a theory to which the Löwenheim-Skolem theorems apply.

For an example, consider the sentence $\mathcal{P} = \forall y \exists x \sim(Sx = y \rightarrow SSx \neq Sy)$ and the box on the following page. Begin replacing $\forall y\mathcal{P}$ with $\sim\exists y\sim\mathcal{P}$ to obtain $\sim\exists y\sim\exists x\sim(Sx = y \rightarrow SSx \neq Sy)$. Then given the usual tree as on the left of (V), construct a parallel tree as on the right replacing each existential quantification by its quantifier-free form—as worked out in the sequences immediately following. You should be able to follow each step. Observe that we might have collapsed the second-to-last step on the right-hand side of (V) to $\emptyset = \emptyset$ once we identified $\emptyset = \emptyset$ as a disjunct, and similarly there may be natural simplifications in other cases.

Thus we obtain a quantifier-free $\mathcal{Q} = \sim(\emptyset \neq \emptyset \vee \emptyset = \emptyset \vee \emptyset \neq \emptyset)$ such that S $\vDash \mathcal{P} \leftrightarrow \mathcal{Q}$, and by $\mathring{\text{c}}$ompleteness S $\vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. From S $\vdash \emptyset = \emptyset$ with $\vee$I and DN, S $\vdash \sim\mathcal{Q}$, and so by NB, S $\vdash \sim\mathcal{P}$. Thus our method tells us not only that *there is* a proof of $\mathcal{P}$ or $\sim\mathcal{P}$ for each $\mathcal{P}$, but constitutes a method to *decide* which of $\mathcal{P}$ or $\sim\mathcal{P}$ is proved. Of course, in this case, we might have derived $\sim\mathcal{P}$ in fewer than ten lines with S1 (try it). So our approach is not particularly efficient. Still, it is of considerable interest to have found a *general* method to decide whether S $\vdash \mathcal{P}$ or S $\vdash \sim\mathcal{P}$.

Quantifier Elimination in Theory S:

$$\mathsf{S} \vDash \forall y \exists x {\sim}(Sx = y \rightarrow SSx \neq Sy) \leftrightarrow {\sim}(\emptyset \neq \emptyset \vee \emptyset = \emptyset \vee \emptyset \neq \emptyset)$$

Begin replacing the universal quantifier to obtain,

$${\sim}\exists y {\sim}\exists x {\sim}(Sx = y \rightarrow SSx \neq Sy)$$

Then,

$$\text{(V)} \qquad \begin{array}{cc}
Sx = y \quad SSx = Sy & Sx = y \quad SSx = Sy \\
SSx \neq Sy & SSx \neq Sy \\
Sx = y \rightarrow SSx \neq Sy & Sx = y \rightarrow SSx \neq Sy \\
{\sim}(Sx = y \rightarrow SSx \neq Sy) & {\sim}(Sx = y \rightarrow SSx \neq Sy) \\
\exists x {\sim}(Sx = y \rightarrow SSx \neq Sy) & y = y \wedge y \neq \emptyset \wedge SSy = SSy \\
{\sim}\exists x {\sim}(Sx = y \rightarrow SSx \neq Sy) & {\sim}(y = y \wedge y \neq \emptyset \wedge SSy = SSy) \\
\exists y {\sim}\exists x {\sim}(Sx = y \rightarrow SSx \neq Sy) & \emptyset \neq \emptyset \vee \emptyset = \emptyset \vee \emptyset \neq \emptyset \\
{\sim}\exists y {\sim}\exists x {\sim}(Sx = y \rightarrow SSx \neq Sy) & {\sim}(\emptyset \neq \emptyset \vee \emptyset = \emptyset \vee \emptyset \neq \emptyset)
\end{array}$$

For the x-quantifier,

$$\begin{array}{cl}
\mathsf{M_d}[\exists x {\sim}(Sx = y \rightarrow S^2 x \neq Sy)] = \mathsf{S} & \\
\text{iff} & \\
\mathsf{M_d}[\exists x (Sx = y \wedge S^2 x = Sy)] = \mathsf{S} & \text{disjunctive normal form} \\
\text{iff} & \\
\mathsf{M_d}[\exists x (Sx = y \wedge S^2 Sx = SSy)] = \mathsf{S} & \text{(T)} \\
\text{iff} & \\
\mathsf{M_d}[\exists x (Sx = y \wedge S^2 y = SSy)] = \mathsf{S} & \text{(T)} \\
\text{iff} & \\
\mathsf{M_d}[\exists x (Sx = y) \wedge SSy = SSy] = \mathsf{S} & \text{(T)} \\
\text{iff} & \\
\mathsf{M_d}[(y = y \wedge y \neq \emptyset) \wedge SSy = SSy] = \mathsf{S} & \text{(U)}
\end{array}$$

For the y-quantifier,

$$\begin{array}{cl}
\mathsf{M_d}[\exists y {\sim}(y = y \wedge y \neq \emptyset \wedge SSy = SSy)] = \mathsf{S} & \\
\text{iff} & \\
\mathsf{M_d}[\exists y (y \neq y \vee y = \emptyset \vee SSy \neq SSy)] = \mathsf{S} & \text{disjunctive normal form} \\
\text{iff} & \\
\mathsf{M_d}[\exists y (y \neq y) \vee \exists y (y = \emptyset) \vee \exists y (SSy \neq SSy)] = \mathsf{S} & \text{QD} \\
\text{iff} & \\
\mathsf{M_d}[\emptyset \neq \emptyset \vee \emptyset = \emptyset \vee \emptyset \neq \emptyset] = \mathsf{S} & \text{(R), (U), (R)}
\end{array}$$

E11.29. Demonstrate theorems Sa–Sd.

Hints: (Sb): without loss of generality, suppose $n > m$; then there is some $d > 0$ such that $n = m + d$; then use S4. (Sc): Left to right: use =I and =E. Right to left: suppose $\mathrm{S} \vdash S^a s \neq t$; by induction on the value of n, $\mathrm{S} \vdash S^n S^a s \neq S^n t$ (with S2). (Sd): Left to right: Begin showing that for any n and $m < n$, $\mathrm{S} \vdash S^n j \neq S^m \emptyset$; this gives you $\mathrm{S} \vdash S^n j \neq S^0 \emptyset \wedge \ldots \wedge S^n j \neq S^{n-1}\emptyset$; with this the derivation is easy. Right to left: By induction on the value of n; put cases for both $n = 0$ and $n = 1$ in the basis; then assume for $1 \leq i < k$.

*E11.30. Supposing that the only conjuncts are $S^a x = t_1 \wedge S^{c+a} x = S^a t_i \wedge S^{e+a} x \neq S^a t_j$, provide a detailed semantic argument for the equivalence in (T) that is justified by “application of the equality from the first conjunct to the other members.” You may take it that $\mathsf{M_h}[S^a q] = \mathsf{a} + \mathsf{M_h}[q]$.

E11.31. Let $\mathcal{P} = \forall y \exists x (Sx = SSy \wedge SSx = SSSy)$. (i) Use our method to find a quantifier-free $\mathcal{Q}$ such that $\mathrm{S} \vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. (ii) Use this result to decide whether $\mathrm{S} \vdash \mathcal{P}$ or $\mathrm{S} \vdash \sim\mathcal{P}$.

Theory L. Given its very simple language, S is a very simple theory. Let us turn to a case that increases complexity a bit. Consider a language like $\mathcal{L}_{NT}^{<}$ whose only function symbol is S. Then terms are of the sort $S^n t$ where t is a variable or $\emptyset$; and atomic formulas are $S^m s = S^n t$ or $S^m s < S^n t$ again where s and t are a variable or $\emptyset$. Let L be a set whose members are as follows:

(L1) $x \neq \emptyset \rightarrow \exists y(x = Sy)$ (and so S3)

(L2) $x < Sy \leftrightarrow (x = y \vee x < y)$

(L3) $x \not< \emptyset$

(L4) $x < y \vee x = y \vee y < x$

(L5) $x < y \rightarrow y \not< x$

(L6) $x < y \rightarrow (y < z \rightarrow x < z)$

So L has finitely many axioms. Again this theory is s̃ound on a standard interpretation with universe $\mathbb{N}$, $\mathsf{0}$ assigned to $\emptyset$, interpretation of S the successor function, and interpretation of $<$ the less-than relation. From these axioms we have as theorems,

(La) If $n \geq 1$ then $\mathrm{L} \vdash x < S^n x$

(Lb) $\mathrm{L} \vdash x \not< x$

(Lc) $\mathrm{L} \vdash x \not< y \leftrightarrow (y = x \vee y < x)$

(Ld) $\mathrm{L} \vdash x \neq y \leftrightarrow (x < y \vee y < x)$

(Le) L $\vdash x < y \leftrightarrow Sx < Sy$

(Lf) If $m < n$ then L $\vdash S^m x < S^n x$

(Lg) If $m \not< n$ then L $\vdash S^m x \not< S^n x$

(Lh) L $\vdash S^a s < t \leftrightarrow S^{d+a} s < S^d t$

(Li) L $\vdash t < S^a s \leftrightarrow S^d t < S^{d+a} s$

(Lj) L $\vdash Sx \neq \emptyset$ (and so S1)

(Lk) L $\vdash Sx = Sy \rightarrow x = y$ (and so S2)

(Ll) L $\vdash S^n x \neq x$ for $n \geq 1$ (and so S4)

Given that we have each of S1–S4, we retain theorems from S. And with L4–L6 any model of L is a linear order.

Now to show that L admits quantifier elimination, consider $\exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$, where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are $S^m s = S^n t$, $S^m s \neq S^n t$, $S^m s < S^n t$, or $S^m s \not< S^n t$, and s and t are either the constant $\emptyset$ or a variable.

First we can eliminate negations. Thus where $\mathcal{A}_i$ is $S^m s \neq S^n t$,

(W)

$$\mathsf{M_d}[\exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge S^m s \neq S^n t \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge (S^m s < S^n t \vee S^n t < S^m s) \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x((\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge S^m s < S^n t \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n) \vee (\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge S^n t < S^m s \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n))] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge S^m s < S^n t \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n) \vee \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_{i-1} \wedge S^n t < S^m s \wedge \mathcal{A}_{i+1} \wedge \ldots \wedge \mathcal{A}_n)] = \mathsf{S}$$

The negated equality is replaced by Ld, then distribution, and the quantifier is pushed in by QD. And similarly for a negated inequality, beginning with Lc to replace $S^m s \not< S^n t$ by $S^n t = S^m s \vee S^n t < S^m s$. So if we can reduce the disjuncts of the resultant expression to quantifier-free form, we will have reduced the whole.

Consider then $\exists x(\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_n)$ where each $\mathcal{B}_i$ is of the sort $S^m s = S^n t$ or $S^m s < S^n t$. Suppose x does not appear in some $\mathcal{B}_i$; then reasoning as before,

(X)

$$\mathsf{M_d}[\exists x(\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_i \wedge \ldots \wedge \mathcal{B}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\mathcal{B}_i \wedge \exists x(\mathcal{B}_1 \wedge \ldots \wedge \mathcal{B}_{i-1} \wedge \mathcal{B}_{i+1} \wedge \ldots \wedge \mathcal{B}_n)] = \mathsf{S}$$

by quantifier placement rules.

Concentrating on the second conjunct, consider $\exists x(\mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_n)$ where x is a component of each $\mathcal{C}_i$; suppose $\mathcal{C}_i$ is of the sort $S^m x = S^n x$ where $m = n$. Then reasoning as before,

(Y)

$$\mathsf{M_d}[\exists x(\mathcal{C}_1 \wedge \ldots \wedge S^m x = S^n x \wedge \ldots \wedge \mathcal{C}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\exists x(\mathcal{C}_1 \wedge \ldots \wedge \emptyset = \emptyset \wedge \ldots \wedge \mathcal{C}_n)] = \mathsf{S}$$
iff
$$\mathsf{M_d}[\emptyset = \emptyset \wedge \exists x(\mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_{i-1} \wedge \mathcal{C}_{i+1} \wedge \ldots \wedge \mathcal{C}_n)] = \mathsf{S}$$

by Sa and then quantifier-placement rules. And similarly by Sb replacing $S^m x = S^n x$ with $\emptyset \neq \emptyset$ when $m \neq n$; by Lf replacing $S^m x < S^n x$ with $\emptyset = \emptyset$ when $m < n$; and by Lg replacing $S^m x < S^n x$ with $\emptyset \neq \emptyset$ when $m \not< n$.

Concentrating then on the second conjunct, consider $\exists x(\mathcal{D}_1 \wedge \ldots \wedge \mathcal{D}_n)$ where each $\mathcal{D}_i$ is $S^n x = t_i$, $S^n x < t_i$, or $t_i < S^n x$ and x does not appear in t_i. Suppose first that some $\mathcal{D}_i$ is $S^a x = t_i$; then reasoning as before,

(Z)
$$\begin{gathered}
\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^b x = t_i \wedge \ldots \wedge t_j < S^c x \wedge \ldots \wedge S^d x < t_k \wedge \ldots))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^{b+a} x = S^a t_i \wedge \ldots \wedge S^a t_j < S^{c+a} x \wedge \ldots \wedge S^{d+a} x < S^a t_k \wedge \ldots))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[\exists x(S^a x = t_1 \wedge (S^b t_1 = S^a t_i \wedge \ldots \wedge S^a t_j < S^c t_1 \wedge \ldots \wedge S^d t_1 < S^a t_k \wedge \ldots))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[\exists x(S^a x = t_1) \wedge (S^b t_1 = S^a t_i \wedge \ldots \wedge S^a t_j < S^c t_1 \wedge \ldots \wedge S^d t_1 < S^a t_k \wedge \ldots)] = \mathsf{S}
\end{gathered}$$

Superscripts on terms not in the first conjunct are increased by the superscript from the first; then t_1 is substituted for $S^a x$ in conjuncts other than the first; and the quantifier is restricted just to it. These are by Sc/Lh/Li; then application of the equality from the first conjunct to the other members; and finally by quantifier placement. Then $\exists x(S^a x = t_1)$ reduces to quantifier-free form just as in (U).

So suppose each $\mathcal{D}_i$ is $t_i < S^n x$ or $S^n x < t_i$ and consider $\exists x((t_1 < S^a x \wedge \ldots \wedge t_i < S^b x) \wedge (S^c x < t_j \wedge \ldots \wedge S^d x < t_k))$. Intuitively, the left conjunct sets lower bounds for x and the right upper. Thus where $S^n x$ simply adds n to x, from the left conjunct, $t_1 - a < x$ and ... and $t_i - b < x$, and from the right, $x < t_j - c$ and ... and $x < t_k - d$ (notice that these latter conditions cannot be met if $t_j \leq c$ or ... or $t_k \leq d$).

Suppose that the main right conjunct is empty. Then,

(AA)
$$\begin{gathered}
\mathsf{M_d}[\exists x(t_1 < S^a x \wedge \ldots \wedge t_i < S^b x)] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[t_1 = t_1 \wedge \ldots \wedge t_j = t_j] = \mathsf{S}
\end{gathered}$$

The quantifier is dropped and each conjunct is replaced by the corresponding identity. Consider the objects $\mathsf{m}_1 \ldots \mathsf{m}_\mathsf{i}$ assigned to $t_1 \ldots t_i$. On an unending linear order, there is sure to be an object greater than each of them, and so an object to satisfy the existential quantification.

Now suppose the left main conjunct is empty. Then,

(AB)
$$\begin{gathered}
\mathsf{M_d}[\exists x(S^c x < t_j \wedge \ldots \wedge S^d x < t_k)] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[S^c \emptyset < t_j \wedge \ldots \wedge S^d \emptyset < t_k] = \mathsf{S}
\end{gathered}$$

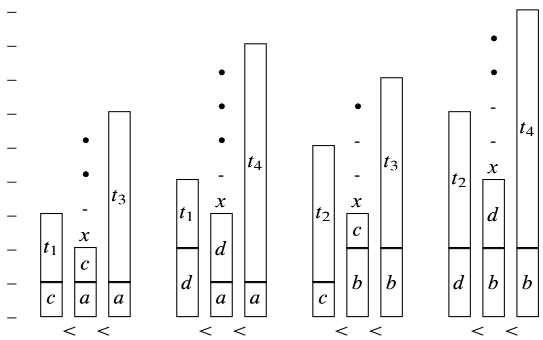
The quantifier is dropped and each x replaced by $\emptyset$. If there is some object under the upper bounds, then $\mathsf{0}$ is under the upper bounds; and if $\mathsf{0}$ is under the upper bounds, then some object is under the upper bounds.

So suppose the $\mathcal{D}_i$ include both members $t_i < S^n x$ and $S^n x < t_i$ and consider $\exists x((t_1 < S^a x \wedge \ldots \wedge t_i < S^b x) \wedge (S^c x < t_j \wedge \ldots \wedge S^d x < t_k))$. For simplicity take a case with just two atomics of each type. Then,

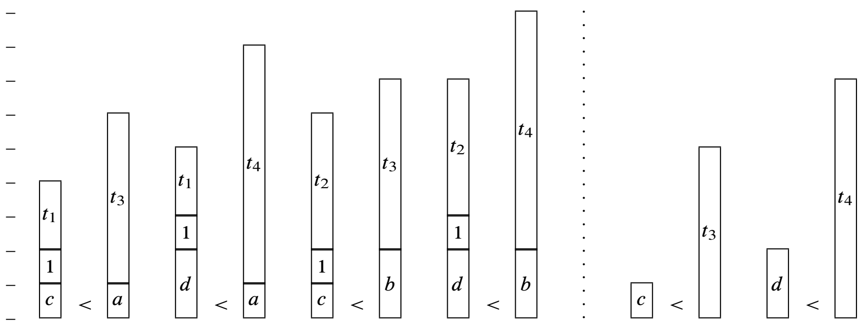
$$\begin{gathered}
\mathsf{M_d}[\exists x((t_1 < S^a x \wedge t_2 < S^b x) \wedge (S^c x < t_3 \wedge S^d x < t_4))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[\exists x((t_1 < S^a x \wedge S^c x < t_3) \wedge (t_1 < S^a x \wedge S^d x < t_4) \wedge \\
(t_2 < S^b x \wedge S^c x < t_3) \wedge (t_2 < S^b x \wedge S^d x < t_4))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[\exists x((S^c t_1 < S^{a+c} x \wedge S^{a+c} x < S^a t_3) \wedge (S^d t_1 < S^{a+d} x \wedge S^{a+d} x < S^a t_4) \wedge \\
(S^c t_2 < S^{b+c} x \wedge S^{b+c} x < S^b t_3) \wedge (S^d t_2 < S^{b+d} x \wedge S^{b+d} x < S^b t_4))] = \mathsf{S} \\
\text{iff} \\
\mathsf{M_d}[(S^{c+1} t_1 < S^a t_3 \wedge S^{d+1} t_1 < S^a t_4 \wedge S^{c+1} t_2 < S^b t_3 \wedge S^{d+1} t_2 < S^b t_4) \wedge (S^c \emptyset < t_3 \wedge S^d \emptyset < t_4)] = \mathsf{S}
\end{gathered} \tag{AC}$$

Atomics from the left conjunct are conjoined with each of the ones from the right; superscripts are adjusted so that "middle" terms with the variable x have the same superscript; then the quantifier is dropped and, for the main left conjunct, middle terms are eliminated and the superscript of the first term increased by one; the right conjunct is like the right conjunct at the first stage, with x replaced by $\emptyset$. These are first by Idem with Com and Assoc; then by Lh and Li; then the last stage is the most interesting; it is perhaps best understood visually:

(AD)



(AE)



Boxes represent values of terms. This example sets $a = 1$, $b = 2$, $c = 1$, $d = 2$, $t_1 = 2$, $t_2 = 4$, $t_3 = 5$, and $t_4 = 7$. Then (AD) represents the third row of (AC) and (AE) the last. From (AD), in each group of three, the sum of the leftmost boxes is less than the sum of center boxes plus some value of x, which is in turn less than the sum of the rightmost boxes. Dots indicate values of x in this range. The left-hand part of (AE) relates outer columns from a group of three—requiring that the right be greater than the left plus one. The right-hand part of (AE) relates columns from the center to ones on the right, subtracting off common boxes.

In each case, the conditions require "space" for some x over the lower bounds and beneath the upper. In (AD) the left two groups require space for values of x over the first lower bound and under both upper bounds; the right two groups require space for values of x over the second lower bound but under both upper bounds.[13] From (AE), the left-hand pairs require space between the upper and lower bounds—adding one to the leftmost columns to leave space for a center above the left and below the right; and the right-hand pairs that the center leave room for some x to occupy those spaces. If there is an x to satisfy the conditions from (AD), then clearly there are the spaces from (AE). And if there are the spaces from (AE) then there is room for an x to satisfy the conditions: put abstractly, consider some lower bounds q, r and upper bounds s, t; suppose there is space for i, j, k, l such that $q < i < s$, $q < j < t$, $r < k < s$, and $r < l < t$; without loss of generality, suppose $q \geq r$; then $m = q + 1$, say, satisfies each of the conditions, and there is an x over each of the lower bounds and beneath the upper.

Thus the original $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ is reduced to quantifier-free $\mathcal{Q}$ with the same free variables such that $\mathrm{L} \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$. So by T11.33, L admits quantifier elimination. And now it is easy to see that L is $\tilde{\text{c}}$omplete.

*T11.35. L is $\tilde{\text{c}}$omplete.

Homework.

By similar methods, it is possible to show that there is a $\tilde{\text{s}}$ound and $\tilde{\text{c}}$omplete theory Pr (*Presburger Arithmetic*) for the standard interpretation of a language like $\mathcal{L}_{NT}$ but with just S and $+$; and there is a $\tilde{\text{s}}$ound and $\tilde{\text{c}}$omplete theory of *real closed fields* (RCF) for the standard interpretation of a language with constants $\emptyset$ and 1, function symbols $+$, $-$, $\times$, and relation symbols $=$ and $<$ on a universe of the real numbers.[14] Given these theories, one might reasonably hope for a $\tilde{\text{s}}$ound and $\tilde{\text{c}}$omplete theory for the standard interpretation of $\mathcal{L}_{NT}$. Unfortunately, this hope is not to be met. As we see in the next part, there is no $\tilde{\text{s}}$ound and $\tilde{\text{c}}$omplete theory for the arithmetic of $\mathcal{L}_{NT}$ including S, $+$, and $\times$.

E11.32. Demonstrate theorems La–Ll.

Hints: (La): by induction on the value of n; you will be able to use L2. (Lb): from L5. (Lc): from left to right with L4; from right to left with L5 and Lb. (Le): from left to right apply Lc with NB and L2 with NB to reach $y \not< Sx$ then you can apply Lc and L2 again; this first subderivation is "reversible." (Lf): suppose $m < n$; then there is some $d \geq 1$ such that $m + d = n$; by induction on the value

[13]In a group of three, the lower bound subtracts from the leftmost t the non-common center; the upper bound subtracts from the rightmost t the non-common center. Then, to take just the first case, from $t_1 + c < a + c + x < t_3 + a$, it follows that $t_1 - a < x < t_3 - c$—which is to say that x is greater than the lower bound but less than the upper.

[14]See, for example, Chapter 3 of Marker, *Model Theory*. These results add considerable complication.

of m, L $\vdash S^m x < S^{m+d} x$. (Lh): By induction on the value of d. (Lj): use L3 and La. (Lk): under the assumption for $\rightarrow$I, you will be able to use Lb, Le, and then L4.

E11.33. Write out the stages from (AC) but starting from $\exists x((t_1 < S^a x \wedge t_2 < S^b x) \wedge (S^c x < t_3 \wedge S^d x < t_4 \wedge S^e x < t_5))$.

*E11.34. Complete the demonstration of T11.35 to show that L is c̃omplete.

E11.35. Let $\mathcal{P} = \forall x \forall z[(\exists y(Sx < y \wedge y < z) \rightarrow SSx < z]$. (i) Use our method to find a quantifier-free $\mathcal{Q}$ such that L $\vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. (ii) Use this result to decide whether S $\vdash \mathcal{P}$ or S $\vdash \sim\mathcal{P}$.

E11.36. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Expressive completeness, and how our languages have it.

b. Unique readability, and how our languages have it.

c. Independence and how *ADs* has it.

d. The relations between relative soundness, s̊oundness, and s̃oundness, and between relative completeness, c̊ompleteness, and c̃ompleteness.

e. The significance of the (full) Löwenheim-Skolem theorem for theory c̃ompleteness.

f. Quantifier elimination, and how it can be utilized to show c̃ompleteness.

Final Theorems of Chapter 11

T11.17 If $\mathsf{L} \preceq \mathsf{M}$ then $\mathsf{L} \equiv \mathsf{M}$.

T11.18 Suppose $\mathsf{L} \sqsubseteq \mathsf{M}$ and d is a variable assignment into $\mathsf{U_L}$. Then for any term t, $\mathsf{M_d}[t] = \mathsf{L_d}[t]$.

T11.19 Suppose that $\mathsf{L} \sqsubseteq \mathsf{M}$ and that for any formula $\mathcal{P}$ and every variable assignment d into $\mathsf{U_L}$ such that $\mathsf{M_d}[\exists x\mathcal{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U_L}$ such that $\mathsf{M}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{L} \preceq \mathsf{M}$.

T11.20 $\mathsf{L} \overset{\iota}{\sqsubseteq} \mathsf{M}$ iff there is a $\mathsf{K} \sqsubseteq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$.

T11.21 $\mathsf{L} \overset{\iota}{\precsim} \mathsf{M}$ iff there is a $\mathsf{K} \preceq \mathsf{M}$ such that $\mathsf{L} \overset{\iota}{\cong} \mathsf{K}$.

T11.22 If $\mathsf{L} \precsim \mathsf{M}$, then $\mathsf{L} \equiv \mathsf{M}$.

T11.23 A set of formulas Σ is satisfiable iff it is finitely satisfiable. *Compactness.*

T11.24 For $\mathfrak{I}$ the class of all models with an infinite domain, there is no finite Γ such that $\mathfrak{Md}(\Gamma) = \mathfrak{I}$.

T11.25 If Γ has arbitrarily large finite models, then Γ has an infinite model.

Corollary: The class $\mathfrak{M}$ of all finite models is not axiomatizable.

T11.26 The class $\mathfrak{W}$ of all well-orderings is not axiomatizable.

T11.27 For $\mathfrak{N}$ the class of models isomorphic to N, there is no Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{N}$.

T11.28 For any set s, $\wp(\mathsf{s})$ is greater than s, $\mathsf{s} \prec \wp(\mathsf{s})$ *Cantor's Theorem.*

T11.29 If the members of Σ are in a language $\mathcal{L}$ whose constants are matched to ordinals less than an infinite κ and Σ has a model, then Σ has a model of cardinality $\leq \kappa$. *Downward Löwenheim-Skolem.*

T11.30 If Σ has an infinite model, then for any infinite cardinal κ, Σ has a model of cardinality $\geq \kappa$. *Upward Löwenheim-Skolem.*

T11.31 If the members of Σ are in a language whose constants are matched to ordinals less than an infinite θ and Σ has an infinite model, then for any infinite cardinal $\kappa \geq \theta$, Σ has a model of cardinality κ. *Full Löwenheim-Skolem.*

T11.32 If D is a finite model and $\mathfrak{D}$ is the class of all models isomorphic to it, then there is a categorical Σ such that $\mathfrak{Md}(\Sigma) = \mathfrak{D}$.

T11.33 If every $\mathcal{P} = \exists x(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n)$ with $\mathcal{A}_1 \ldots \mathcal{A}_n$ atomic or negated atomic has a quantifier-free $\mathcal{Q}$ with the same free variables such that $\Sigma \vDash \mathcal{P} \leftrightarrow \mathcal{Q}$, then Σ admits quantifier elimination.

T11.34 S is $\tilde{\text{c}}$omplete.

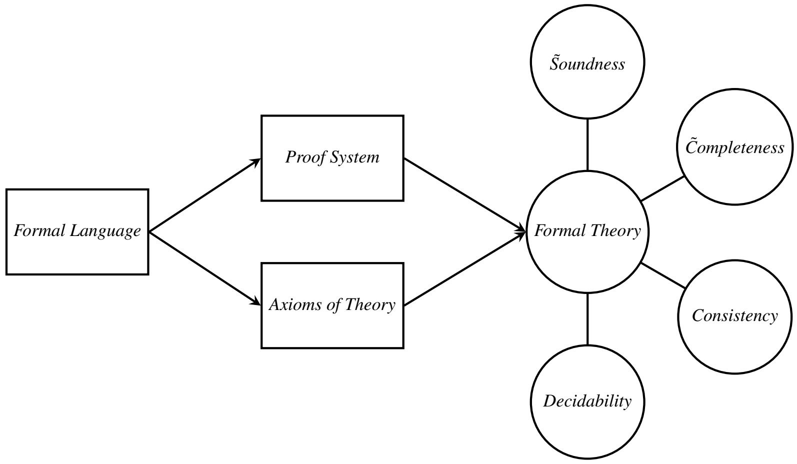
T11.35 L is $\tilde{\text{c}}$omplete.

Part IV

Logic and Arithmetic: Ĩncompleteness and Computability

Introductory

A formal *theory* consists of a formal language with some proof system and theory axioms. Q and PA are example theories. We have had a good bit to say about languages and proof systems. In this part we encounter a cluster of issues associated with theories.



The *theorems* of a theory are all the formulas proved by its axioms. A theory is *s̃ound* when all its theorems are true on an intended model; (negation) *c̃omplete* when it proves one of $\mathscr{P}$ or $\sim\mathscr{P}$ for every sentence $\mathscr{P}$; *consistent* when there is no $\mathscr{P}$ such that it proves both $\mathscr{P}$ and $\sim\mathscr{P}$; and *decidable* when there is an "effective" method to decide whether any given formula is a theorem.

In Chapter 10 we showed that exactly the same arguments are semantically valid as are provable. So,

(A) $$\Gamma \vdash \mathscr{P} \quad \text{iff} \quad \Gamma \vDash \mathscr{P}$$

Thus our derivation systems are both s̊ound and c̊omplete. In Chapter 11 we encountered limitations about the ability to axiomatize certain models, but also exhibited some simple consistent, s̃ound, c̃omplete, and decidable theories. These results for s̊oundness and c̊ompleteness, and then for consistency, s̃oundness, c̃ompleteness, and decidability are the good news. In this part, we encounter a series of limiting results—with particular application to arithmetic and computing.

As for the simple theories of Chapter 11, it is natural to think that mathematics more generally is characterized by proofs and derivations. Thus one might anticipate that there would be some system of premises Δ such that for any $\mathscr{P}$ in $\mathscr{L}_{NT}$ we would have,

(B) $$\Delta \vdash \mathscr{P} \quad \text{iff} \quad \mathsf{N}[\mathscr{P}] = \mathsf{T}$$

where N is the standard interpretation of number theory. Such a theory would be consistent, s̃ound, c̃omplete and, as we shall see, supposing it is "nicely specified," decidable.[1] Note the difference between (A) and (B). In (A) derivations are matched to entailments; in (B) derivations (and so entailments) are matched to truths on an interpretation. Perhaps inspired by suspicions about the existence or nature of numbers, one might expect that derivations would even entirely replace the notion of mathematical truth. And Q or PA may already seem to be deductive systems as in (B). But we shall see that there can be no such deductive system: Consider any nicely specified theory at least as strong as Q; from Gödel's first ĩncompleteness theorem, if that theory is consistent then it is ĩncomplete. But then any such theory must omit some truths of arithmetic from among its theorems.[2]

Suppose there is no one-to-one map between truths of arithmetic and consequences of our theories. Rather, we propose a theory R(eal) whose consequences are unproblematically true, and another theory I(deal) whose consequences outrun those of R and whose literal truth is therefore somehow suspect. Perhaps R is sufficient only for something like basic arithmetic, whereas I seems to quantify over all members of a far-flung infinite domain. Even though not itself a vehicle for truth, theory I may be useful under certain circumstances. Suppose,

(a) For any $\mathscr{P}$ in the scope of R, if $\mathscr{P}$ is not true, then $R \vdash \sim\mathscr{P}$

(b) I *extends* R: If $R \vdash \mathscr{P}$ then $I \vdash \mathscr{P}$

(c) I is *consistent*: There is no $\mathscr{P}$ such that $I \vdash \mathscr{P}$ and $I \vdash \sim\mathscr{P}$

Then theory I may be treated as a tool for achieving results in the scope of R: Suppose $\mathscr{P}$ is a result in the scope of R, and $I \vdash \mathscr{P}$; then by consistency, $I \nvdash \sim\mathscr{P}$; and because I extends R, $R \nvdash \sim\mathscr{P}$; so by (a), $\mathscr{P}$ is true. This is (a sketch of) the famous 'Hilbert program' for mathematics, which aims to make sense of infinitary mathematics based not on the truth but rather the consistency of theory I (this project is developed in a number of places including Hilbert, "On the Infinite").

Suppose the language of R permits universal generalizations as $\forall x \forall y(x \times y = y \times x)$. In Chapter 12 we shall show that Q proves particular results sufficient to

[1] In the following we identify the "nicely specified" formal theories with the (precisely defined) *recursively axiomatized* formal theories.

[2] Gödel's groundbreaking paper is "On the Formally Undecidable Propositions of *Principia Mathematica* and Related Systems."

establish the negation of such generalizations whenever the generalizations are false. In this way a theory R might be sufficient to prove $\sim\mathscr{P}$ whenever a $\mathscr{P}$ in its scope is false, and so to satisfy (a). And just as PA extends Q, there will be many extensions of R as in (b). And because consistency is a syntactical result about proof systems, not itself about far-flung mathematical structures, one might have hoped for proofs of consistency from real, rather than ideal, theories—and so for proof of (c). So there is an intuitive plausibility to Hilbert's proposal. But Gödel's second ĩncompleteness theorem tells us that derivation systems extending PA cannot prove even their own consistency. So a weaker "real" theory will not be able to prove the consistency of PA and its extensions. This seems to remove a demonstration of (c) and so to doom the Hilbert strategy.[3]

Even though no one derivation system has as consequences every mathematical truth, derivations remain useful, and mathematicians continue to do proofs! Given that we care about them, there is a question about the automation of proofs. A property or relation is *effectively decidable* iff there is an algorithm or program that for any given case, decides in a finite number of steps whether the property or relation applies. Abstracting from the limitations of particular computing devices, we shall identify a class of relations which are decidable. A corollary to reasoning for Gödel's first ĩncompleteness theorem is that *being provable* in theories like Q and PA, as well as *validity* in systems like *ND* and *AD* are not among the decidable relations. Thus there are important limits on what computing devices can do.

Chapter 12 lays down background required for chapters that follow. It begins with a discussion of *recursive functions*, and concludes with a few essential results, including a demonstration of the ĩncompleteness of arithmetic. Chapters 13 and 14 deepen and extend those results in different ways. Chapter 13 shows ĩncompleteness again by the construction of a sentence such that neither it nor its negation is provable, and then turns to demonstration of the second ĩncompleteness theorem. Chapter 14 shows once more that there must exist a sentence such that neither it nor its negation is provable, but this time in association with an account of computability. Chapter 12 is required for either Chapter 13 or Chapter 14; but those chapters may be taken in either order.

[3]We are familiar with the Pythagorean Theorem according to which the hypotenuse and sides of a right triangle are such that $a^2 = b^2 + c^2$. In the 1600s Pierre de Fermat famously proposed that there are no integers a, b, c such that $a^n = b^n + c^n$ for $n > 2$; so, for example, there are no a, b, c such that $a^3 = b^3 + c^3$. In 1995 Andrew Wiles proved that this is so. But Wiles's proof requires some fantastically abstract (and difficult) mathematics. Even if Wiles's abstract theory (I) is not *true* Hilbert could still accept the demonstration of Fermat's (real) theorem so long as I is shown to be *consistent*. Gödel's result seems to block this strategy. There are alternative conceptions of the Hilbert program. And, of course, one might simply accept Wiles's proof on the ground that his advanced mathematics is *s̃ound* and so its consequences true. But these are topics in philosophy of mathematics, not logic. See, for example, Shapiro, *Thinking About Mathematics* for an introduction to options in the philosophy of mathematics including Hilbert's program. Our limiting results may very well stimulate interest in that field!

Chapter 12

Recursive Functions and Q

We have said that a formal *theory* consists of a language, with some axioms and proof system. The *theorems* of a theory are all the formulas proved by its axioms. Q and PA are example theories. A theory is *sound with respect to* a class of models iff its theorems are true on every member of the class, and *s̃ound* iff it is sound with respect to a class of intended models. From T11.9, a theory is s̃ound iff its intended models are among the ones on which its axioms are true. A theory is *complete with respect to* a class of models iff its theorems include every formula true on all the members of that class. From T11.11, a theory is complete with respect to some models if they include the models on which its axioms are true. A theory T is (*negation*) *c̃omplete* iff for any sentence $\mathcal{P}$ either $T \vdash \mathcal{P}$ or $T \vdash \sim\mathcal{P}$; and models are *elementarily equivalent* iff they make all the same formulas true. Then from T11.16, a theory is c̃omplete iff it is complete with respect to some class of models whose members are elementarily equivalent, and in particular iff models on which its axioms are true are elementarily equivalent. A theory whose proof system is s̊ound and c̊omplete is sound and complete with respect to the class of all models on which its axioms are true. But s̊oundness and c̊ompleteness do not by themselves yield s̃oundness and c̃ompleteness. S̃oundness requires also that intended models are among models on which the axioms are true, and c̃ompleteness that models on which the axioms are true are elementarily equivalent.

Let us pause to consider why s̃oundness and c̃ompleteness matter: Consider a theory and some intended interpretation. Say we want to characterize by means of the theory all and only sentences that are true on the interpretation. If the theory is not s̃ound, then some theorem $\mathcal{P}$ is not true on the interpretation; but then with Gen its universal closure $\mathcal{P}^u$ is a theorem, and by by T7.6, is not true. So if a theory is not s̃ound, its theorems include sentences that are not true. And from E8.29, as soon as a language $\mathcal{L}$ has an interpretation I, for any sentence $\mathcal{P}$ in $\mathcal{L}$, either $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$. So if theorems are to include all the sentences that are true on some interpretation, the theory must have among its consequences $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$. Put the other way around, if a theory is not such that for any $\mathcal{P}$, either $T \vdash \mathcal{P}$

or $T \vdash \sim\mathscr{P}$ (if it is ĩncomplete), then it is sure to omit some sentences true on the interpretation. To the extent that we desire a characterization of all and only sentences true on an interpretation, for arithmetic or whatever, a s̃ound and c̃omplete theory is a desirable theory.

Demonstrating s̃oundness is a matter of showing that axioms are true on some intended interpretation(s). As exhibited by controversies about the axioms of set theory, this can be both difficult and controversial (see, for example, Feferman, "Does Mathematics Need New Axioms?"). But some cases are clear enough: In particular, by reasoning from E7.19, the axioms of Q are true on its intended model N; and so with T11.9, Q is s̃ound.

So we focus on c̃ompleteness. C̃ompleteness is sometimes, but not always attainable. In section 11.4.3 we saw that there are c̃omplete theories whose theorems include all the sentences true on a finite model. Similarly there is a c̃omplete theory S for the standard interpretation of a language like $\mathcal{L}_{NT}$ but without $+$ and $\times$ (and so with $\emptyset$, S, and $=$), and a c̃omplete theory L for the standard interpretation of a language like $\mathcal{L}^{<}_{NT}$ without $+$ and $\times$ (and so with $\emptyset$, S, $<$, and $=$). By similar methods, it is possible to show that there is a c̃omplete theory Pr (*Presburger Arithmetic*) for the standard interpretation of a language like $\mathcal{L}_{NT}$ but without $\times$; and there is a c̃omplete theory of real closed fields (RCF) for the standard interpretation of a language with constants $\emptyset$ and 1, function symbols $+$, $-$, $\times$, and relation symbols $=$ and $<$ on a universe of the real numbers. From the existence of these c̃omplete theories, one might reasonably hope for a c̃omplete theory for the standard interpretation of $\mathcal{L}_{NT}$. But this hope is not to be realized. There is no nicely specified s̃ound and c̃omplete theory for the arithmetic of $\mathcal{L}_{NT}$ which includes $\emptyset$, S, $+$, $\times$, and $=$.

It turns out that theories are something like superheros: In the ordinary case, a c̃omplete, and so a "happy" life is at least within reach. However, as theories acquire certain powers, they take on a "fatal flaw" just because of their powers—where this flaw makes c̃ompleteness unattainable. On its face, theory Q does not appear particularly heroic. We have seen already from T10.5 and E7.19 that Q $\nvdash$ $\forall x \forall y(x \times y = y \times x)$ and Q $\nvdash$ $\sim\forall x \forall y(x \times y = y \times x)$. So Q is negation ĩncomplete. PA which does prove $x \times y = y \times x$ along with other standard results in arithmetic might seem a more likely candidate for heroism. But Q already includes features sufficient to generate the fatal flaw—and a theory, like PA, which includes all the powers of Q must have the flaw as well. Our task in this chapter and the beginning of the next is to identify that flaw.[1]

As it happens, a system with the powers of Q including $\emptyset$, S, $+$, and $\times$ can express and capture all the *recursive* functions, where this power is essential to having the fatal flaw. Thus in this chapter we focus on the recursive functions (and recursive

[1]Interestingly, although RCF has many powers, it lacks certain of the powers of Q and so does not possess the flaw. This is because no formula in the language of RCF is true just of the natural numbers—and it is therefore not possible in this language to make *general* claims according to which the natural numbers have this property or that.

relations built upon them), associate them with powers of our formal systems, and show how these powers result in the flaw. We begin in section 12.1 saying what recursive functions are; then in 12.2 and 12.3 we show that $\mathscr{L}_{NT}$ *expresses* and Q *captures* the recursive functions; 12.4 assigns numbers to formulas and sequences of formulas and extends the range of recursive functions and relations to include a relation that identifies proofs. Finally, from these results, 12.5 concludes with some applications, including the ĩncompleteness of arithmetic.

12.1 Recursive Functions

In chapters 3 and 6 for Q and PA we had axioms of the sort,

a. $x + \emptyset = x$
b. $x + Sy = S(x + y)$

and

c. $x \times \emptyset = \emptyset$
d. $x \times Sy = (x \times y) + x$

These enable us to derive $x + y$ and $x \times y$ for arbitrary values of x and y. Thus, by (a) $\overline{2} + \overline{0} = \overline{2}$; so by (b) $\overline{2} + \overline{1} = \overline{3}$; and by (b) again, $\overline{2} + \overline{2} = \overline{4}$; and so forth. From the values at any one stage, we are in a position to calculate values at the next. And similarly for multiplication. From pages 304–305 of Chapter 6 all this should be familiar.

While axioms thus supply effective means for calculating the values of these functions, the functions themselves might be similarly *identified* or *specified*. So, given a successor function $\mathsf{suc}(\mathsf{x})$, we may identify the functions $\mathsf{plus}(\mathsf{x}, \mathsf{y})$:

a. $\mathsf{plus}(\mathsf{x}, 0) = \mathsf{x}$
b. $\mathsf{plus}(\mathsf{x}, \mathsf{suc}(\mathsf{y})) = \mathsf{suc}(\mathsf{plus}(\mathsf{x}, \mathsf{y}))$

and $\mathsf{times}(\mathsf{x}, \mathsf{y})$:

c. $\mathsf{times}(\mathsf{x}, 0) = 0$
d. $\mathsf{times}(\mathsf{x}, \mathsf{suc}(\mathsf{y})) = \mathsf{plus}(\mathsf{times}(\mathsf{x}, \mathsf{y}), \mathsf{x})$

For ease of reading, let us typically revert to the more ordinary notation S, $\boldsymbol{+}$, and $\boldsymbol{\times}$ for these functions, though we stick with the sans serif font (emphasized for $\boldsymbol{+}$ and $\boldsymbol{\times}$). We have been thinking of functions as certain complex sets. Thus the plus function is a set with elements $\{\ldots, \langle\langle 2, 0\rangle, 2\rangle, \langle\langle 2, 1\rangle, 3\rangle, \langle\langle 2, 2\rangle, 4\rangle, \ldots\}$. Our specification picks out this set. From (a), $\mathsf{plus}(\mathsf{x}, \mathsf{y})$ has $\langle\langle 2, 0\rangle, 2\rangle$ as a member; given this, from (b), $\langle\langle 2, 1\rangle, 3\rangle$ is a member; and so forth. So the two clauses work together to specify the plus function. And similarly for times. In each case, the first clause gives the value for $\mathsf{y} = 0$, and the second for Sy given the value for y.

But these are not the only sets which may be specified this way. Thus the standard factorial $\mathsf{fact}(\mathsf{y})$:

e. $\mathsf{fact}(0) = 1$
f. $\mathsf{fact}(Sy) = \mathsf{fact}(y) \times Sy$

Zero factorial is one. And the factorial of Sy multiplies $1 \times 2 \times \cdots \times y$ by Sy. Again, we will often revert to the more typical $x!$ notation. Similarly $\mathsf{power}(x, y)$:

g. $\mathsf{power}(x, 0) = 1$
h. $\mathsf{power}(x, Sy) = \mathsf{power}(x, y) \times x$

Any number to the power of zero is one ($x^0 = 1$). And then x^{Sy} multiplies $x^y = x \times x \times \cdots \times x$ (y times) by another x. Again, we revert to the more standard notation.

We shall be interested in a class of functions, the *recursive* functions, which may be specified (in part) by this strategy. To make progress, we turn to a general account in five stages.

12.1.1 Initial Functions

Our examples have simply taken $\mathsf{suc}(x)$ as given. Similarly, we shall require a stock of *initial functions*. Recursive functions operate on and take values in the natural numbers. Thus we continue to allow variables and constants as x and 0, whose values are natural numbers. Then there are initial functions of three different types.

(i) $\mathsf{zero}()$ is the very simple function which "operates" on nothing to return the value 0. It may be strange to think of a function without inputs, however it will streamline things to come if we do. A one-place function has members of the sort $\langle x, v\rangle$ and so is really a kind of restricted two-place relation; and, generally, we can see an n-place function as a restricted $(n+1)$-place relation. Then a 0-place function is a 1-place relation, whose restriction requires that it has a single member—in this case, $\langle 0\rangle$. And it is hard to imagine a simpler function—$\mathsf{zero}()$ is the constant zero-place function that always returns the number 0.[2]

(ii) For any $j \geq k \geq 1$, there is an *identity* function $\mathsf{idnt}^j_k(x_1 \ldots x_j)$. Again, the identity functions are very simple. Each idnt^j_k has j places and returns the value from the k^{th} place; that is, for a k such that $1 \leq k \leq j$, $\mathsf{idnt}^j_k = \{\langle\langle x_1 \ldots x_j\rangle, x_k\rangle \mid x_1 \ldots x_j \in \mathbb{N}\}$. So $\mathsf{idnt}^3_2 = \{\ldots, \langle\langle 1, 2, 3\rangle, 2\rangle, \ldots, \langle\langle 4, 5, 6\rangle, 5\rangle, \ldots\}$ and $\mathsf{idnt}^3_2(4, 5, 6) = 5$. In the simplest case, $\mathsf{idnt}^1_1(x) = x$.

(iii) Finally, we shall continue to include $\mathsf{suc}(x)$ among the initial functions. $\mathsf{suc}(x)$ returns the number following x on the usual ordering of the natural numbers; that is, $\mathsf{suc}(x) = \{\langle x, x+1\rangle \mid x \in \mathbb{N}\}$. So $\mathsf{suc}(x) = \{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \ldots\}$ and $\mathsf{suc}(1) = 2$.

[2]This generalizes the usual account on which a function is a set of pairs (as from the Chapter 4 set theory reference). For a zero-place function the restriction $\mathcal{A}m_1 \ldots \mathcal{A}m_n\mathcal{A}a\mathcal{A}b[(\langle m_1 \ldots m_n, a\rangle \in f \wedge \langle m_1 \ldots m_n, b\rangle \in f) \Rightarrow a = b]$ reduces to $\mathcal{A}a\mathcal{A}b[(\langle a\rangle \in f \wedge \langle b\rangle \in f) \Rightarrow a = b]$.

These are very simple building blocks. However we shall be able to use them to produce functions of amazing complexity!

12.1.2 Composition

In our examples, we have let one function be *composed* from others—as when we consider $\mathsf{suc}(\mathsf{plus}(x, y))$ or the like. Say $\vec{x}$, $\vec{y}$, and $\vec{z}$ represent possibly-empty sequences of variables $x_1 \ldots x_a$, $y_1 \ldots y_b$, and $z_1 \ldots z_c$ (by an expression familiar from geometry, we sometimes refer to such a sequence as a *vector*).

CM Let $g(\vec{y})$ and $h(\vec{x}, w, \vec{z})$ be any functions. Then $f(\vec{x}, \vec{y}, \vec{z})$ is defined by *composition* from $g(\vec{y})$ and $h(\vec{x}, w, \vec{z})$ iff $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$.

So $h(\vec{x}, w, \vec{z})$ gets its value in the w-place from $g(\vec{y})$. Here is a simple example: $f(y, z) = \mathsf{times}(\mathsf{suc}(y), z)$ results by composition from substitution of $\mathsf{suc}(y)$ into $\mathsf{times}(w, z)$; so $\mathsf{times}(w, z)$ gets its value in the w-place from $\mathsf{suc}(y)$. The result is a set with members, $\{\ldots, \langle\langle 2, 0\rangle, 0\rangle, \langle\langle 2, 1\rangle, 3\rangle, \langle\langle 2, 2\rangle, 6\rangle, \ldots\}$. When the input to $f(y, z)$ is, say, $\langle 2, 1\rangle$, $\mathsf{suc}(y)$ receives the input 2 and supplies 3 to the first place of the $\mathsf{times}(w, z)$ function; then from $\mathsf{times}(w, z)$ the result is the product of 3 and 1 which is 3. And similarly in other cases. In contrast, $\mathsf{suc}(y \times z)$ has members $\{\ldots, \langle\langle 2, 0\rangle, 1\rangle$, $\langle\langle 2, 1\rangle, 3\rangle, \langle\langle 2, 2\rangle, 5\rangle, \ldots\}$. You should see how this works.

Here are a couple of cases we shall have occasion to use just below: First, for any $n \geq 0$ and (possibly empty) $\vec{x} = x_1 \ldots x_n$,

$$\mathsf{zero}^n(x_1 \ldots x_n) = \mathsf{idnt}^{n+1}_{n+1}(x_1 \ldots x_n, \mathsf{zero}())$$

So $\mathsf{zero}^n(x_1 \ldots x_n)$, has free variables $x_1 \ldots x_n$ but always returns 0. Observe that $\mathsf{zero}^0() = \mathsf{idnt}^1_1(\mathsf{zero}()) = \mathsf{zero}() = 0$. In the ordinary case, we drop the superscript n and take the number of places from context. Second,

$$\widehat{n} = \overbrace{\mathsf{suc}(\mathsf{suc}(\ldots \mathsf{suc}(\mathsf{zero}())\ldots))}^{\text{n instances of suc}}$$

$\widehat{n}$ is a zero-place function that returns the number n. So we have $\widehat{1} = \mathsf{suc}(\mathsf{zero}()) = 1$ and $\widehat{0} = \mathsf{zero}() = 0$.

12.1.3 Recursion

For each of our examples, $\mathsf{plus}(x, y)$, $\mathsf{times}(x, y)$, $\mathsf{fact}(y)$, and $\mathsf{power}(x, y)$, the value of the function is set for $y = 0$ and then for $\mathsf{suc}(y)$ given its value for y. These illustrate the method of recursion. Put generally, where $\vec{x}$ is a possibly empty sequence $x_1 \ldots x_n$,

RD Given some functions $g(\vec{x})$ and $h(\vec{x}, y, u)$, $f(\vec{x}, y)$ is defined by *recursion* when,

$$f(\vec{x}, 0) = g(\vec{x})$$
$$f(\vec{x}, Sy) = h(\vec{x}, y, f(\vec{x}, y))$$

So there are functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. Then $\mathsf{f}(\vec{\mathsf{x}}, 0)$ gets its value from $\mathsf{g}(\vec{\mathsf{x}})$; and $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{Sy})$ gets its value from h with the value of $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ substituted for u. Thus as in the examples above, we fix the value of $\mathsf{f}(\vec{\mathsf{x}}, 0)$ and then set the value at any other stage depending on the stage before.

We adopt the general scheme so that we can operate on recursive functions in a consistent way. However the general scheme includes flexibility that is not always required. So, for example, in the cases of plus, times, and power, $\vec{\mathsf{x}}$ reduces to a simple variable x, and for fact it disappears entirely. And, as we shall see, the functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ need not depend on each of their variables $\vec{\mathsf{x}}$ and $\vec{\mathsf{x}}$, y, and u. However, by clever use of our initial functions, it is possible to see each of our sample functions on this pattern. Thus for $\mathsf{plus}(\mathsf{x}, \mathsf{y})$, set $\mathsf{gplus}(\mathsf{x}) = \mathsf{idnt}^1_1(\mathsf{x})$ and $\mathsf{hplus}(\mathsf{x}, \mathsf{y}, \mathsf{u}) = \mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{u}))$. Then by RD, $\mathsf{plus}(\mathsf{x}, 0)$ is set to $\mathsf{gplus}(\mathsf{x})$ and $\mathsf{plus}(\mathsf{x}, \mathsf{Sy})$ to $\mathsf{hplus}(\mathsf{x}, \mathsf{y}, \mathsf{plus}(\mathsf{x}, \mathsf{y}))$.

a' $\mathsf{plus}(\mathsf{x}, 0) = \mathsf{idnt}^1_1(\mathsf{x})$
b' $\mathsf{plus}(\mathsf{x}, \mathsf{Sy}) = \mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{plus}(\mathsf{x}, \mathsf{y})))$

And these work as they should: $\mathsf{idnt}^1_1(\mathsf{x}) = \mathsf{x}$ and $\mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{plus}(\mathsf{x}, \mathsf{y})))$ is equivalent to $\mathsf{suc}(\mathsf{plus}(\mathsf{x}, \mathsf{y}))$. So we recover the conditions (a) and (b) from above. Similarly, for $\mathsf{times}(\mathsf{x}, \mathsf{y})$ let $\mathsf{gtimes}(\mathsf{x}) = \mathsf{zero}(\mathsf{x})$ and $\mathsf{htimes}(\mathsf{x}, \mathsf{y}, \mathsf{u}) = \mathsf{plus}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{u}), \mathsf{x})$. Then,

c' $\mathsf{times}(\mathsf{x}, 0) = \mathsf{zero}(\mathsf{x})$
d' $\mathsf{times}(\mathsf{x}, \mathsf{Sy}) = \mathsf{plus}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{times}(\mathsf{x}, \mathsf{y})), \mathsf{x})$

So $\mathsf{times}(\mathsf{x}, 0) = 0$ and $\mathsf{times}(\mathsf{x}, \mathsf{Sy}) = \mathsf{plus}(\mathsf{times}(\mathsf{x}, \mathsf{y}), \mathsf{x})$, and all is well. Observe that we would obtain the same result with $\mathsf{htimes}(\mathsf{x}, \mathsf{y}, \mathsf{u}) = \mathsf{plus}(\mathsf{u}, \mathsf{idnt}^3_1(\mathsf{x}, \mathsf{y}, \mathsf{u}))$ or perhaps, $\mathsf{plus}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{u}), \mathsf{idnt}^3_1(\mathsf{x}, \mathsf{y}, \mathsf{u}))$.

By the identity functions we standardize the specification of recursive $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ so that g is always a function of $\vec{\mathsf{x}}$, and h always a function of $\vec{\mathsf{x}}$, y, and u. This will matter when it comes to reasoning generally about recursive functions. However, as for multiplication, there may be different ways to produce a function with the desired characteristics. We might require that variables appear immediately *only* in identity functions applied to $\vec{\mathsf{x}}$, and then $\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u}$. However, this would be needlessly tedious. So we won't worry about differences so long as specifications are in the standard form.

Recall that $\vec{\mathsf{x}}$ is a possibly empty sequence of variables. For plus and times it has just a single member. In the case of $\mathsf{fact}(\mathsf{y})$, there are no places to the $\vec{\mathsf{x}}$ vector. Then gfact is reduced to a zero-place function, and hfact to a function of y and u. For $\mathsf{fact}(\mathsf{y})$, set $\mathsf{gfact}() = \widehat{1}$ and $\mathsf{hfact}(\mathsf{y}, \mathsf{u}) = \mathsf{times}(\mathsf{u}, \mathsf{suc}(\mathsf{y}))$. This time identity functions do not appear at all: All the variables of $\mathsf{gfact}()$ and $\mathsf{hfact}(\mathsf{y}, \mathsf{u})$ appear in a natural way, and identity functions are not required. It is left as an exercise to show that gfact and hfact identify the same function as constraints (e), (f), and then to find $\mathsf{gpower}(\mathsf{x})$ and $\mathsf{hpower}(\mathsf{x}, \mathsf{y}, \mathsf{u})$.

The Recursion Theorem

One may wonder whether our specification $\mathsf{f}(\mathsf{x},\mathsf{y})$ by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ results in a unique function. However it is possible to show that it does.

RT Suppose $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ are total functions on $\mathbb{N}$; then there exists a unique function $\mathsf{f}(\vec{\mathsf{x}},\mathsf{y})$ such that,

(r) For any $\vec{\mathsf{x}}$ and $\mathsf{y} \in \mathbb{N}$,

a. $\mathsf{f}(\vec{\mathsf{x}},0) = \mathsf{g}(\vec{\mathsf{x}})$

b. $\mathsf{f}(\vec{\mathsf{x}},\mathsf{suc}(\mathsf{y})) = \mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{f}(\vec{\mathsf{x}},\mathsf{y}))$

We identify this function as a union of functions which may be constructed by means of g and h. The *domain* of a total function from r^{n} to s is always r^{n}; for a partial function, the domain of the function is that subset of r^{n} whose members are matched by the function to members of s (for background see the Chapter 4 set theory reference). Say a (maybe partial) function $\mathsf{s}(\vec{\mathsf{x}},\mathsf{y})$ is *acceptable* iff,

i. If $\langle\vec{\mathsf{x}},0\rangle \in \mathsf{dom}(\mathsf{s})$, then $\mathsf{s}(\vec{\mathsf{x}},0) = \mathsf{g}(\vec{\mathsf{x}})$

ii. If $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{n})\rangle \in \mathsf{dom}(\mathsf{s})$, then $\langle\vec{\mathsf{x}},\mathsf{n}\rangle \in \mathsf{dom}(\mathsf{s})$ and $\mathsf{s}(\vec{\mathsf{x}},\mathsf{suc}(\mathsf{n})) = \mathsf{h}(\vec{\mathsf{x}},\mathsf{n},\mathsf{s}(\vec{\mathsf{x}},\mathsf{n}))$

If $\mathsf{g}(\vec{\mathsf{x}}) = \mathsf{a}$ and $\mathsf{h}(\vec{\mathsf{x}},0,\mathsf{a}) = \mathsf{b}$, then a partial function $\mathsf{s} = \{\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{a}\rangle,\langle\langle\vec{\mathsf{x}},1\rangle,\mathsf{b}\rangle\}$, that sets $\mathsf{s}(\vec{\mathsf{x}},0)$ to $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{s}(\vec{\mathsf{x}},1)$ to $\mathsf{h}(\vec{\mathsf{x}},0,\mathsf{s}(\vec{\mathsf{x}},0))$, would satisfy (i) and (ii). A function which satisfies (r) is acceptable, though not every function which is acceptable satisfies (r); we show that exactly one acceptable function satisfies (r). Let F be the collection of all acceptable functions, and f be $\bigcup\mathsf{F}$. Thus $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{f}$ iff $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle$ is a member of some acceptable s. We sketch reasoning to show that f has the right features.

I. For any acceptable s and s', if $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{b}\rangle \in \mathsf{s}'$, then $\mathsf{a} = \mathsf{b}$. By induction on n: Suppose $\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{b}\rangle \in \mathsf{s}'$; then by (i), $\mathsf{a} = \mathsf{b} = \mathsf{g}(\vec{\mathsf{x}})$. Assume that if $\langle\langle\vec{\mathsf{x}},\mathsf{k}\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{k}\rangle,\mathsf{b}\rangle \in \mathsf{s}'$ then $\mathsf{a} = \mathsf{b}$. Show that if $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{c}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{d}\rangle \in \mathsf{s}'$ then $\mathsf{c} = \mathsf{d}$. Suppose $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{c}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{d}\rangle \in \mathsf{s}'$. Then by (ii), $\langle\vec{\mathsf{x}},\mathsf{k}\rangle$ is in the domain of both s and s', with $\mathsf{c} = \mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{s}(\vec{\mathsf{x}},\mathsf{k}))$ and $\mathsf{d} = \mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{s}'(\vec{\mathsf{x}},\mathsf{k}))$. But by assumption $\mathsf{s}(\vec{\mathsf{x}},\mathsf{k}) = \mathsf{s}'(\vec{\mathsf{x}},\mathsf{k})$; so $\mathsf{c} = \mathsf{d}$.

II. $\mathsf{dom}(\mathsf{f})$ includes every $\langle\vec{\mathsf{x}},\mathsf{n}\rangle$. By induction on n: For any $\vec{\mathsf{x}}$, $\{\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{g}(\vec{\mathsf{x}})\rangle\}$ is itself an acceptable function. Assume that for any $\vec{\mathsf{x}}$, $\langle\vec{\mathsf{x}},\mathsf{k}\rangle \in \mathsf{dom}(\mathsf{f})$. Show that for any $\vec{\mathsf{x}}$, $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle \in \mathsf{dom}(\mathsf{f})$. Suppose otherwise, and consider a function, $\mathsf{s} = \mathsf{f} \cup \{\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{f}(\vec{\mathsf{x}},\mathsf{k}))\rangle\}$. But we may show that s so defined is an acceptable function; and since s is acceptable, it is a subset of f; so $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle \in \mathsf{dom}(\mathsf{f})$. Reject the assumption.

III. Now by (I), if $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{f}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{b}\rangle \in \mathsf{f}$, then $\mathsf{a} = \mathsf{b}$; so f is a function; and by (II) the domain of f includes every $\langle\vec{\mathsf{x}},\mathsf{n}\rangle$; by construction it is easy to see that f is itself acceptable. From these, f satisfies (r). Suppose some f' also satisfies (r); then f' is acceptable; so by construction, f' is a subset of f; but since f' satisfies (r), its domain includes every $\langle\vec{\mathsf{x}},\mathsf{n}\rangle$; so $\mathsf{f}' = \mathsf{f}$. So (r) is uniquely satisfied.

Intuitively, the acceptable functions are defined on initial segments of the natural numbers, and the union of them all is a function defined on the entire domain.

*We employ *weak* induction from the Chapter 8 (page 368) induction schemes reference. Enderton, *Elements of Set Theory*, and Drake and Singh, *Intermediate Set Theory*, include nice discussions of this result.

12.1.4 Regular Minimization

So far, the method of our examples is easily matched to the capacities of computing devices. To find the value of a function defined by recursion, begin by finding the value for `y` = `0`, and then calculate other values, from one stage to the next. But this is just what computing devices do well. So, for example, in the syntax of the Ruby language,[3] given some functions `g(x)` and `h(x,y,u)`,

(A)

```
1. def recfunc(a,b)
2.   f = g(a)
3.   for y in 0..b-1
4.     f = h(a,y,f)
5.   end
6.   return f
7. end
```

Here `f` tracks the value of the function for different values of `y`. First, the program uses `g(a)` to set the value of `f` for input `(a, 0)`. Lines (3)–(5) are a loop which begins at `y` = `0` using `h` with `a`, `y`, and `f` to find the value of `f` for (`a`, `1`); the program then increments `y` to `1` and uses `h` with `a` and current values of `y` and `f` to find `f` for (`a`, `Sy`); this continues until it uses `h` with `a`, `y` = `b - 1`, and `f` to find the value of the function for (`a`, `b`). And it returns the final result. This strategy of moving from one value to the next corresponds to the way we moved from one value to the next for addition and multiplication in Chapter 6. Observe that the calculation of `recfunc(a,b)` requires exactly `b` iterations before it completes.

But there is a different repetitive mechanism available for computing devices—where this mechanism does not begin with a fixed number of iterations. Suppose we have some function `g(a,b)` with values `g(a,0)`, `g(a,1)`, `g(a,2)`, ... where for each value of `a` there are at least some values of `b` such that `g(a,b)` = `0`. For any value of `a`, suppose we want the least `b` such that `g(a,b)` = `0`. Then we might reason as follows:

(B)

```
1. def minfunc(a)
2.   y = 0
3.   until g(a,y) == 0
4.     y = y + 1
5.   end
6.   return y
7. end
```

This program begins with `y = 0`. Then lines (3)–(5) are a loop which tests tests to see if `g(a,y)` = `0`; if it is not, it increments the value of `y` and tries again. Once it finds a `y` such that `g(a,y)` = `0`, it exits the loop and returns the value of `y`. Supposing that for any `a` there are some values of `b` such that `g(a,b)` = `0`, then, this program returns a value of `y` for any input value `a`.

---

[3] Ruby is convenient insofar as it is interpreted and so easy to run, and available at no cost on multiple platforms (see http://www.ruby-lang.org/en/downloads/). We depend only on very basic features familiar from most any exposure to computing. See E12.3 and related exercises.

But, as before, we might reason similarly to *specify* functions so calculated. For this, recall from the Chapter 4 set theory reference that a function is *total* iff it is defined on all members of its domain. Say a function $\mathsf{g}(\vec{x}, y)$ is *regular* iff it is total and for all values of $\vec{x}$ there is at least one y such that $\mathsf{g}(\vec{x}, y) = 0$. Then,

RM If $\mathsf{g}(\vec{x}, y)$ is a regular function, the function $\mathsf{f}(\vec{x}) = \mu y[\mathsf{g}(\vec{x}, y) \mathbf{=} \hat{0}]$ which for each $\vec{x}$ takes as its value the least y such that $\mathsf{g}(\vec{x}, y) = 0$ is defined by *regular minimization* from $\mathsf{g}(\vec{x}, y)$.

For a simple example, consider a function $\mathsf{g}(x, y)$ which takes nonempty subsets of $\mathbb{N}$ for x and members of $\mathbb{N}$ for y; suppose $\mathsf{g}(x, y) = 0$ if $y \in x$ and otherwise $\mathsf{g}(x, y) = 1$. So, for example,

$$\begin{array}{cccccc} \mathsf{g}(\{2,4,6\},0) & \mathsf{g}(\{2,4,6\},1) & \mathsf{g}(\{2,4,6\},2) & \mathsf{g}(\{2,4,6\},3) & \mathsf{g}(\{2,4,6\},4) & \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \dots \\ 1 & 1 & 0 & 1 & 0 & \end{array}$$

This function is regular so long as sets in the x-place are nonempty; if x is empty then $\mathsf{g}(x, y)$ returns 1 for each value of y and the function is not regular. Supposing, then, that the sets are nonempty $\mathsf{f}(x) = \mu y[\mathsf{g}(x, y) \mathbf{=} \hat{0}]$ is always the least element of x. Notice that a loop which checks whether numbers up to a fixed n are in the set will not do—for the least element could always be larger than that. But, so long as the sets have members, our open-ended search is sure to return a result. In our simple case, $\mu y[\mathsf{g}(\{2,4,6\}, y) \mathbf{=} \hat{0}] = 2$.

12.1.5 Final Definition

Finally, our sample functions are *cumulative*. Thus $\mathsf{plus}(x, y)$ depends on $\mathsf{suc}(x)$, $\mathsf{times}(x, y)$ on $\mathsf{plus}(x, y)$, and so forth. We are thus led to our final account.

RF A function f_k is *recursive* iff there is a series of functions $\mathsf{f}_0, \mathsf{f}_1, \dots, \mathsf{f}_k$ such that for any $i \leq k$,

(i) f_i is an initial function $\mathsf{zero}()$, $\mathsf{idnt}^j_k(x_1 \dots x_j)$, or $\mathsf{suc}(x)$.

(c) There are $a, b < i$ such that $\mathsf{f}_i(\vec{x}, \vec{y}, \vec{z})$ results by composition from $\mathsf{f}_a(\vec{y})$ and $\mathsf{f}_b(\vec{x}, w, \vec{z})$.

(r) There are $a, b < i$ such that $\mathsf{f}_i(\vec{x}, y)$ results by recursion from $\mathsf{f}_a(\vec{x})$ and $\mathsf{f}_b(\vec{x}, y, u)$.

(m) There is some $a < i$ such that $\mathsf{f}_i(\vec{x})$ results by regular minimization from $\mathsf{f}_a(\vec{x}, y)$.

If there is a series of functions $\mathsf{f}_0, \mathsf{f}_1, \dots, \mathsf{f}_k$ such that for any $i \leq k$, just (i), (c), or (r), then (PR) f_k is *primitive recursive*.

So any recursive function results from a series of functions each of which satisfies one of these conditions. And such a series demonstrates that its members are recursive. For a simple example, plus is primitive recursive.

(C)

1.	$\mathsf{idnt}^1_1(\mathsf{x})$	initial function
2.	$\mathsf{idnt}^3_3(\mathsf{x},\mathsf{y},\mathsf{u})$	initial function
3.	$\mathsf{suc}(\mathsf{w})$	initial function
4.	$\mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x},\mathsf{y},\mathsf{u}))$	2,3 composition
5.	$\mathsf{plus}(\mathsf{x},\mathsf{y})$	1,4 recursion

And we might recast this list into a tree like ones from Part I, starting with initial functions on the top, and others built from ones above. From the list by itself, one might reasonably wonder whether $\mathsf{plus}(\mathsf{x},\mathsf{y})$, so defined, is the addition function we know and love. What follows, given primitive recursive functions $\mathsf{idnt}^1_1(\mathsf{x})$ and $\mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x},\mathsf{y},\mathsf{u}))$ is that a primitive recursive function results by recursion from them. It turns out that this is the addition function. It is left as an exercise to exhibit $\mathsf{times}(\mathsf{x},\mathsf{y})$ as primitive recursive as well.

*E12.1. (i) Show that the proposed gfact and $\mathsf{hfact}(\mathsf{y},\mathsf{u})$ result in conditions (e) and (f). Then (ii) produce a defininition for $\mathsf{power}(\mathsf{x},\mathsf{y})$ by finding functions $\mathsf{gpower}(\mathsf{x})$, and $\mathsf{hpower}(\mathsf{x},\mathsf{y},\mathsf{u})$ and then show that they have the same result as conditions (g) and (h).

E12.2. Generate a sequence of functions sufficient to show that $\mathsf{times}(\mathsf{x},\mathsf{y})$ is primitive recursive.

E12.3. Find the textbook website, `https://tonyroyphilosophy.net/symbolic-logic/`, and install some convenient version of Ruby on your computing platform (see the file "Running Ruby" for help). Open `recursive1.rb` and extend the sequence of functions there to include `fact(x)` and `power(x,y)`. Calculate some values of these functions and print the results, along with your program (do not worry if these latter functions run slowly for even moderate values of `x` and `y`). This assignment does not require any particular computing expertise—especially, there should be no appeal to functions except from earlier in the chain. This exercise exhibits the recursive functions in action and suggests a point, to be developed in Chapter 14, that recursive functions are *computable*.

12.2 Expressing Recursive Functions

Having identified the recursive functions, we turn now to the first of two powers to be associated with theory ĩncompleteness. In this case, it is an *expressive* power. Recall that a theory is *s̃ound* iff it is sound with respect to a class of intended models; let us consider theories whose single intended model is N. In section 12.5.2 (and again in 13.1.2 and 14.2.3) we show that if such a theory is s̃ound and its interpreted language *expresses* all the recursive functions, it must be negation ĩncomplete. In this section then, as a basis for that argument, we show that $\mathscr{L}_{NT}$, on its standard interpretation, expresses the recursive functions.

12.2.1 Definition and First Results

For a language $\mathscr{L}$ and interpretation I, suppose that for each $m \in U$, there is a variable-free term $\overline{m}$ such that $I(\overline{m}) = m$—so for any variable assignment d, $I_d[\overline{m}] = m$ (see definition AI in Chapter 8). The simplest way for this to happen is if for each $m \in U$ there is a unique constant to which m is assigned; then for any m, $\overline{m}$ is just that constant. But the standard interpretation for number theory N also has the special feature that each member of U is assigned to a variable-free term. On this interpretation the same object may be assigned to different variable-free terms (as $SS\emptyset$ and $S\emptyset + S\emptyset$ are each assigned 2). Given this, as in section 8.4, we simply choose to let $\overline{n}$ be $S \ldots S\emptyset$ with n repetitions of the successor operator. So $\overline{0}$ abbreviates the term $\emptyset$, $\overline{1}$ the term $S\emptyset$, and so forth.

With such variable-free terms, we shall say that a formula $\mathcal{R}(x)$ *expresses* a relation $R(x)$ on interpretation I, just in case if $m \in R$ then $I[\mathcal{R}(\overline{m})] = T$ and if $m \notin R$ then $I[\sim\mathcal{R}(\overline{m})] = T$. So the formula is true when the individual is a member of the relation and false when it is not. To express a relation on an interpretation, a formula must "say" which individuals fall under the relation. Expressing a relation is closely related to translation. A formula $\mathcal{R}(x)$ expresses a relation $R(x)$ when every sentence $\mathcal{R}(\overline{m})$ is a good translation of the sentence $m \in R$ on the single intended interpretation I (compare section 5.1). Thus, generalizing,

EXr For any language $\mathscr{L}$, interpretation I, and objects $m_1 \ldots m_n \in U$, relation $R(x_1 \ldots x_n)$ is *expressed* by formula $\mathcal{R}(x_1 \ldots x_n)$ iff,

(i) if $\langle m_1 \ldots m_n\rangle \in R$ then $I[\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = T$

(ii) if $\langle m_1 \ldots m_n\rangle \notin R$ then $I[\sim\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = T$

Say $R(x_1 \ldots x_n)$ is expressed by $\mathcal{R}(x_1 \ldots x_n)$. By (i) if $\langle m_1 \ldots m_n\rangle \in R$ then $I[\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = T$. And from (ii) if $\langle m_1 \ldots m_n\rangle \notin R$ then $I[\sim\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = T$; then since $\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)$ is a sentence, with T8.8, $I[\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] \neq T$. So $\langle m_1 \ldots m_n\rangle \in R$ iff $I[\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = T$—with T8.7, iff $I_d[\mathcal{R}(\overline{m}_1 \ldots \overline{m}_n)] = S$ for some assignment d.

Similarly, a one-place function $f(x)$ is a kind of two-place relation. Thus to express a function $f(x)$, we require a formula $\mathcal{F}(x, v)$ where if $\langle m, a\rangle \in f$, then $I[\mathcal{F}(\overline{m}, \overline{a})] = T$. It would be natural to go on to require that if $\langle m, a\rangle \notin f$ then $I[\sim\mathcal{F}(\overline{m}, \overline{a})] = T$. However this is not necessary once we build in another feature of functions—that they have a *unique* output for each input value. Thus we shall require, EXf For any language $\mathscr{L}$, interpretation I, and objects $\mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}, \mathsf{a} \in \mathsf{U}$, function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n})$ is *expressed* by formula $\mathscr{F}(x_1 \ldots x_n, v)$ iff,

if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$ then,

(i) $\mathsf{I}[\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_\mathsf{n}, \overline{\mathsf{a}})] = \mathsf{T}$

(ii) $\mathsf{I}[\forall z(\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_\mathsf{n}, z) \to \overline{\mathsf{a}} = z)] = \mathsf{T}$

When $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$, from (i) $\mathscr{F}$ is true for a; and from (ii) any z for which it is true is identical to a.[4]

Let us illustrate these definitions with some first applications. First, on any interpretation with the required variable-free terms, the formula $x = y$ expresses the equality relation $\textsc{eq}(\mathsf{x}, \mathsf{y})$. For if $\langle \mathsf{m}, \mathsf{n}\rangle \in \textsc{eq}$ then $\mathsf{I}[\overline{\mathsf{m}}] = \mathsf{I}[\overline{\mathsf{n}}]$ so that $\mathsf{I}[\overline{\mathsf{m}} = \overline{\mathsf{n}}] = \mathsf{T}$; and if $\langle \mathsf{m}, \mathsf{n}\rangle \notin \textsc{eq}$ then $\mathsf{I}[\overline{\mathsf{m}}] \neq \mathsf{I}[\overline{\mathsf{n}}]$ so that $\mathsf{I}[\overline{\mathsf{m}} \neq \overline{\mathsf{n}}] = \mathsf{T}$. This works because $\mathsf{I}[=]$ just is the equality relation $\textsc{eq}$. Turning to some functions, on the standard interpretation N for number theory, $\mathsf{zero}()$ is expressed by the formula $\overline{0} = v$. For if $\langle \mathsf{a}\rangle \in \mathsf{zero}()$ then a is just 0 so that $\mathsf{N}[\overline{0} = \overline{\mathsf{a}}] = \mathsf{T}$ and $\mathsf{N}[\forall z(\overline{0} = z \to \overline{\mathsf{a}} = z)] = \mathsf{T}$; so both EXf(i) and EXf(ii) are satisfied. Similarly, on the standard interpretation N for number theory, $\mathsf{suc}(\mathsf{x})$ is expressed by $Sx = v$, $\mathsf{plus}(\mathsf{x}, \mathsf{y})$ by $x + y = v$, and $\mathsf{times}(\mathsf{x}, \mathsf{y})$ by $x \times y = v$. Taking just the addition case, suppose $\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{plus}$; then $\mathsf{N}[\overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{a}}] = \mathsf{T}$. And because addition is a function, $\mathsf{N}[\forall z((\overline{\mathsf{m}} + \overline{\mathsf{n}} = z) \to \overline{\mathsf{a}} = z)] = \mathsf{T}$. Again, this works because $\mathsf{N}[+]$ just is the plus function. And similarly in the other cases. Put more generally,

T12.1. For an interpretation on which members of the universe are assigned to the required variable-free terms: (a) If $\textsc{r}$ is a relation, and $\mathsf{I}[\mathscr{R}] = \textsc{r}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n})$, then $\textsc{r}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n})$ is expressed by $\mathscr{R}x_1 \ldots x_n$. And (b) if h is a function and $\mathsf{I}[\mathscr{h}] = \mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n})$ then $\mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n})$ is expressed by $\mathscr{h}x_1 \ldots x_n = v$.

It is possible to argue semantically for these claims. However, as for translation, we take the project of demonstrating expression to be one of *providing* or supplying relevant formulas. So, having supplied formulas to express the basic functions and relations, we are done.

Notice that the case for $\mathsf{zero}()$ generalizes to other 0-place functions, so that $\hat{\mathsf{n}}$ is expressed by $\overline{\mathsf{n}} = v$. For if $\langle \mathsf{a}\rangle \in \hat{\mathsf{n}}$, then a just is n and we have both $\mathsf{N}[\overline{\mathsf{n}} = \overline{\mathsf{a}}] = \mathsf{T}$ and $\mathsf{N}[\forall z(\overline{\mathsf{n}} = z \to \overline{\mathsf{a}} = z)] = \mathsf{T}$.

Also, as we have suggested, EXf(ii) yields a condition like EXr(ii). Recall that a function is *total* just in case it has an output for any input.

[4]There is a problem of terminology for 'expression'. Different texts offer somewhat different definitions and employ somewhat different vocabulary. The best advice is to pay close attention to details in any particular work.

T12.2. If total function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is expressed by formula $\mathscr{F}(x_1 \dots x_n, y)$, then if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$.

For simplicity, consider just a one-place total function $\mathsf{f}(\mathsf{x})$. Suppose $\mathsf{f}(\mathsf{x})$ is expressed by $\mathscr{F}(x, y)$ and $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$. Then since f is total, there is some $\mathsf{b} \neq \mathsf{a}$ such that $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{f}$.

Suppose $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{T}$; then by TI, for some d, $\mathsf{I}_\mathsf{d}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$; let h be a particular assignment of this sort; so $\mathsf{I}_\mathsf{h}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{I}_\mathsf{h}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{S}$. But since $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{f}$ by EXf(ii), $\mathsf{I}[\forall z(\mathscr{F}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{b}} = z)] = \mathsf{T}$; so by TI, for any d, $\mathsf{I}_\mathsf{d}[\forall z(\mathscr{F}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{b}} = z)] = \mathsf{S}$; so $\mathsf{I}_\mathsf{h}[\forall z(\mathscr{F}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{b}} = z)] = \mathsf{S}$; so by SF($\forall$), $\mathsf{I}_{\mathsf{h}(z|\mathsf{a})}[\mathscr{F}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{b}} = z] = \mathsf{S}$; so since $\mathsf{I}_\mathsf{h}[\overline{\mathsf{a}}] = \mathsf{a}$, by T10.2, $\mathsf{I}_\mathsf{h}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \to \overline{\mathsf{b}} = \overline{\mathsf{a}}] = \mathsf{S}$; so by SF($\to$), $\mathsf{I}_\mathsf{h}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$ or $\mathsf{I}_\mathsf{h}[\overline{\mathsf{b}} = \overline{\mathsf{a}}] = \mathsf{S}$; so $\mathsf{I}_\mathsf{h}[\overline{\mathsf{b}} = \overline{\mathsf{a}}] = \mathsf{S}$; but $\mathsf{I}_\mathsf{h}[\overline{\mathsf{a}}] = \mathsf{a}$ and $\mathsf{I}_\mathsf{h}[\overline{\mathsf{b}}] = \mathsf{b}$; so by SF(r), $\langle \mathsf{b}, \mathsf{a}\rangle \in \mathsf{I}[=]$; so $\mathsf{b} = \mathsf{a}$. This is impossible; reject the assumption: If $\mathsf{f}(\mathsf{x})$ is expressed by $\mathscr{F}(x, y)$ and $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$, then $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$.

Suppose $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is expressed by formula $\mathscr{F}(x_1 \dots x_n, y)$. From the definition, if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \in \mathsf{f}$ then $\mathsf{I}[\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$. And by this theorem with T8.8, if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$ then $\mathsf{I}[\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] \neq \mathsf{T}$. So $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \in \mathsf{f}$ iff $\mathsf{I}[\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$—with T8.7, iff $\mathsf{I}_\mathsf{d}[\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{S}$ for some assignment d. For the most part, I appeal to definitions EXr and EXf, along with the resultant biconditionals, just 'by expression'.

E12.4. Provide semantic reasoning to demonstrate the first part of T12.1. So assume $\mathsf{I}[\mathscr{R}(x_1 \dots x_n)] = \mathsf{R}(\mathsf{x}_1 \dots \mathsf{x}_n)$. Then show (i) if $\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle \in \mathsf{R}$ then $\mathsf{I}[\mathscr{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)] = \mathsf{T}$; and (ii) if $\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle \notin \mathsf{R}$ then $\mathsf{I}[\sim\mathscr{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)] = \mathsf{T}$.

12.2.2 Core Result

So far, on interpretation N, we have been able to express the relation eq, and the functions, $\mathsf{zero}()$, suc, plus, and times. But our aim is to show that, on the standard interpretation N of $\mathscr{L}_{NT}$, given some $\vec{\mathsf{x}} = \mathsf{x}_1 \dots \mathsf{x}_n$ and $\vec{x} = x_1 \dots x_n$, every recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$.

However it is not obvious that this can be done. At least some functions must remain inexpressible in any language that has a countable vocabulary, and so in $\mathscr{L}_{NT}$. We shall see a concrete example later in the chapter. For now, consider a diagonal argument: By reasoning as from T10.7 there is an enumeration of all the formulas in a countable language. Isolate just formulas $\mathscr{P}_0, \mathscr{P}_1, \mathscr{P}_2, \dots$ that express functions of one variable, and consider the functions $\mathsf{f}_0(\mathsf{x}), \mathsf{f}_1(\mathsf{x}), \mathsf{f}_2(\mathsf{x}), \dots$ so expressed. These are all the expressible functions of one variable. Consider a grid with the functions listed down the left-hand column, and their values for each natural number from left to right:

	0	1	2	$\dots$
$\mathsf{f}_0(\mathsf{x})$	$\mathbf{f_0(0)}$	$\mathsf{f}_0(1)$	$\mathsf{f}_0(2)$	
$\mathsf{f}_1(\mathsf{x})$	$\mathsf{f}_1(0)$	$\mathbf{f_1(1)}$	$\mathsf{f}_1(2)$	
$\mathsf{f}_2(\mathsf{x})$	$\mathsf{f}_2(0)$	$\mathsf{f}_2(1)$	$\mathbf{f_2(2)}$	
$\vdots$				

Moving along the main diagonal, consider a function $\mathsf{f_d(x)}$ such that for any n, $\mathsf{f_d(n)} = \mathsf{f_n(n)}+1$. So $\mathsf{f_d(x)}$ is $\{\langle 0, \mathsf{f}_0(0)+1\rangle, \langle 1, \mathsf{f}_1(1)+1\rangle, \langle 2, \mathsf{f}_2(2)+1\rangle, \dots\}$. But this function $\mathsf{f_d(x)}$ cannot be any of the expressible functions. It differs from $\mathsf{f_0(x)}$ insofar as $\mathsf{f_d}(0) \neq \mathsf{f}_0(0)$; it differs from $\mathsf{f_1(x)}$ insofar as $\mathsf{f_d}(1) \neq \mathsf{f}_1(1)$; and so forth. So $\mathsf{f_d(x)}$ is an inexpressible function. Though it has a unique output for every input value, there is no formula to express it. We have already seen that the recursive $\mathsf{plus(x, y)}$ and $\mathsf{times(x, y)}$ are expressible in $\mathscr{L}_{NT}$. But there is no obvious mechanism in $\mathscr{L}_{NT}$ to express, say, $\mathsf{fact(x)}$. Given that not all functions are expressible, it is a significant matter, then, to see that all the recursive functions are expressible with interpretation N in $\mathscr{L}_{NT}$. It is to this task that we now turn.

We begin with some preliminary theorems to set up the main result. These are not hard, but need to be wrapped up before we can attack the main problem. For interpretation N, they work like derived clauses to SF for inequalities and bounded quantifiers.

T12.3. On the standard interpretation N for $\mathscr{L}_{NT}$, (i) $\mathsf{N_d}[\mathscr{s} \leq \mathscr{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathscr{s}] \leq \mathsf{N_d}[\mathscr{t}]$, and (ii) $\mathsf{N_d}[\mathscr{s} < \mathscr{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathscr{s}] < \mathsf{N_d}[\mathscr{t}]$.

(i) By abv $\mathsf{N_d}[\mathscr{s} \leq \mathsf{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\exists u(u + \mathscr{s} = \mathscr{t})] = \mathsf{S}$ where u does not appear in $\mathscr{s}$ or $\mathscr{t}$; by SF′(∃), iff there is some $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u + \mathscr{s} = \mathscr{t}] = \mathsf{S}$. But $\mathsf{d}(u|\mathsf{m})[u] = \mathsf{m}$; so by TA(v), $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u] = \mathsf{m}$; let $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[\mathscr{s}] = \mathsf{a}$ and $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[\mathscr{t}] = \mathsf{b}$; then by TA(f), $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u + \mathscr{s}] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{a}\rangle = \mathsf{m} + \mathsf{a}$. So by SF(r), $\mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[u + \mathscr{s} = \mathscr{t}] = \mathsf{S}$ iff $\langle \mathsf{m}+\mathsf{a}, \mathsf{b}\rangle \in \mathsf{N}[=]$; iff $\mathsf{m}+\mathsf{a} = \mathsf{b}$. But since u is not in $\mathscr{s}$ or $\mathscr{t}$, d and $\mathsf{d}(u|\mathsf{m})$ make the same assignments to variables in $\mathscr{s}$ and $\mathscr{t}$; so by T8.4, $\mathsf{N_d}[\mathscr{s}] = \mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[\mathscr{s}]$ and $\mathsf{N_d}[\mathscr{t}] = \mathsf{N}_{\mathsf{d}(u|\mathsf{m})}[\mathscr{t}]$. So $\mathsf{m}+\mathsf{a} = \mathsf{b}$ iff $\mathsf{m} + \mathsf{N_d}[\mathscr{s}] = \mathsf{N_d}[\mathscr{t}]$; and there exists such an m just in case $\mathsf{N_d}[\mathscr{s}] \leq \mathsf{N_d}[\mathscr{t}]$. So $\mathsf{N_d}[\mathscr{s} \leq \mathsf{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathscr{s}] \leq \mathsf{N_d}[\mathscr{t}]$.

(ii) Homework.

As an immediate corollary, $\mathsf{N_d}[\mathscr{s} \leq \mathsf{t}] \neq \mathsf{S}$ just in case $\mathsf{N_d}[\mathscr{s}] > \mathsf{N_d}[\mathscr{t}]$; and $\mathsf{N_d}[\mathscr{s} < \mathsf{t}] \neq \mathsf{S}$ just in case $\mathsf{N_d}[\mathscr{s}] \geq \mathsf{N_d}[\mathscr{t}]$. Observe that, as distinguished by context and (slight) typographical difference, '≤' is a symbol of the object language, where '≤' is used to convey the relation—and similarly in other cases.

T12.4. On the standard interpretation N for $\mathscr{L}_{NT}$,

(a) $\mathsf{N_d}[(\forall x \leq \mathscr{t})\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{o} \leq \mathsf{N_d}[\mathscr{t}]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$. And $\mathsf{N_d}[(\forall x < \mathscr{t})\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{o} < \mathsf{N_d}[\mathscr{t}]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$.

(b) $\mathsf{N_d}[(\exists x \leq t)\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{o} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$. And $\mathsf{N_d}[(\exists x < t)\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{o} < \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$.

These are straightforward with T7.7 and T12.3. The case for $(\forall x \leq t)\mathcal{P}$ is worked as an example. Recall that x does not appear in t.

(i) Suppose $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$ but for some $\mathsf{o} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] \neq \mathsf{S}$. Let m be a particular individual of this sort; then $\mathsf{m} \leq \mathsf{N_d}[t]$ and $\mathsf{N_{d(x|m)}}[\mathcal{P}] \neq \mathsf{S}$. Since $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$, by T7.7a, for all $\mathsf{o} \in \mathsf{U}$, $\mathsf{N_{d(x|o)}}[x \leq t] \neq \mathsf{S}$ or $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$; so $\mathsf{N_{d(x|m)}}[x \leq t] \neq \mathsf{S}$ or $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$; so $\mathsf{N_{d(x|m)}}[x \leq t] \neq \mathsf{S}$; so with T12.3 $\mathsf{N_{d(x|m)}}[x] \not\leq \mathsf{N_{d(x|m)}}[t]$; but $\mathsf{N_{d(x|m)}}[x] = \mathsf{m}$; and since x does not appear in t, d and $\mathsf{d}(x|\mathsf{m})$ agree on assignments to variables in t, so by T8.4 $\mathsf{N_{d(x|m)}}[t] = \mathsf{N_d}[t]$; so $\mathsf{m} \not\leq \mathsf{N_d}[t]$. This is impossible; reject the assumption: if $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$ then for all $\mathsf{o} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$.

(ii) Suppose that for every $\mathsf{o} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$ but $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] \neq \mathsf{S}$. From the latter, by T7.7a, for some $\mathsf{m} \in \mathsf{U}$, $\mathsf{N_{d(x|m)}}[x \leq t] = \mathsf{S}$ but $\mathsf{N_{d(x|m)}}[\mathcal{P}] \neq \mathsf{S}$. With the first of these, by T12.3 $\mathsf{N_{d(x|m)}}[x] \leq \mathsf{N_{d(x|m)}}[t]$; but $\mathsf{N_{d(x|m)}}[x] = \mathsf{m}$; and since x does not appear in t, d and $\mathsf{d}(x|\mathsf{m})$ agree on assignments to variables in t, so by T8.4 $\mathsf{N_{d(x|m)}}[t] = \mathsf{N_d}[t]$; so $\mathsf{m} \leq \mathsf{N_d}[t]$; but every $\mathsf{o} \leq \mathsf{N_d}[t]$ is such that $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$, so $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$. This is impossible; reject the assumption: if for every $\mathsf{o} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$ then $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$.

Now we are ready for the main result. From definition RF recursive functions arise in a sequence indexed by the natural numbers. This puts us in a position to reason about the sequence by mathematical induction. Thus our main argument is an induction on the sequence of recursive functions. We show that initial functions are expressed, and then functions from composition, recursion, and regular minimization. For one key case, we defer discussion into the next section.

T12.5. On the standard interpretation N of $\mathcal{L}_{NT}$, each recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathcal{F}(\vec{x}, v)$.

For any recursive function $\mathsf{f_a}$ there is a sequence of functions $\mathsf{f_0}, \mathsf{f_1}, \ldots, \mathsf{f_a}$ such that each member is an initial function or arises from previous members by composition, recursion, or regular minimization. By induction on functions in this sequence:

Basis: $\mathsf{f_0}$ is an initial function $\mathsf{zero}()$, $\mathsf{suc}(\mathsf{x})$, or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x_1} \ldots \mathsf{x_j})$.

(z) $\mathsf{f_0}$ is $\mathsf{zero}()$. Then as described on page 575, $\mathsf{f_0}$ is expressed by $\mathcal{F}(v) = \overline{0} = v$.

(s) $\mathsf{f_0}$ is $\mathsf{suc}(\mathsf{x})$. Then by T12.1, $\mathsf{f_0}$ is expressed by $\mathcal{F}(x, v) = Sx = v$.

(i) $\mathsf{f_0}$ is $\mathsf{idnt}^j_k(\mathsf{x_1 \ldots x_j})$. Then $\mathsf{f_0}$ is expressed by

$$\mathscr{F}(x_1 \ldots x_j, v) = (x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v$$

Suppose $\langle\langle \mathsf{m_1 \ldots m_j}\rangle, \mathsf{a}\rangle \in \mathsf{idnt}^j_k$. Then since $\mathsf{a} = \mathsf{m_k}$, $\mathsf{N}[(\overline{\mathsf{m}}_1 = \overline{\mathsf{m}}_1 \wedge \ldots \wedge \overline{\mathsf{m}}_j = \overline{\mathsf{m}}_j) \wedge \overline{\mathsf{m}}_k = \overline{\mathsf{a}}] = \mathsf{T}$. And any $z = \mathsf{m_k}$ is equal to a—so that $\mathsf{N}[\forall z((\overline{\mathsf{m}}_1 = \overline{\mathsf{m}}_1 \wedge \ldots \wedge \overline{\mathsf{m}}_j = \overline{\mathsf{m}}_j \wedge \overline{\mathsf{m}}_k = z) \to \overline{\mathsf{a}} = z)] = \mathsf{T}$.[5]

Assp: For any i, $0 \leq i < k$, $\mathsf{f_i}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{F}(\vec{x}, v)$.

Show: $\mathsf{f_k}(\mathsf{x})$ is expressed by some $\mathscr{F}(\vec{x}, v)$.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion, or regular minimization. If it is an initial function then reason as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ is expressed by some $\mathscr{G}(\vec{y}, w)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ by $\mathscr{H}(\vec{x}, w, \vec{z}, v)$. Given this, the composition $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is expressed by $\mathscr{F}(\vec{x}, \vec{y}, \vec{z}, v) = \exists w[\mathscr{G}(\vec{y}, w) \wedge \mathscr{H}(\vec{x}, w, \vec{z}, v)]$.

For simplicity, consider a case where $\vec{\mathsf{x}}$ and $\vec{\mathsf{z}}$ drop out and $\vec{\mathsf{y}}$ is a single variable y; so $\mathscr{F}(y, v) = \exists w[\mathscr{G}(y, w) \wedge \mathscr{H}(w, v)]$. Suppose $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$; then by composition there is some b such that $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{g}$ and $\langle \mathsf{b}, \mathsf{a}\rangle \in \mathsf{h}$. Because $\mathscr{G}$ and $\mathscr{H}$ express g and h, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{b}})] = \mathsf{T}$ and $\mathsf{N}[\mathscr{H}(\overline{\mathsf{b}}, \overline{\mathsf{a}})] = \mathsf{T}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{b}}) \wedge \mathscr{H}(\overline{\mathsf{b}}, \overline{\mathsf{a}})] = \mathsf{T}$, and $\mathsf{N}[\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, \overline{\mathsf{a}}))] = \mathsf{T}$. Further, by expression, $\mathsf{N}[\forall w(\mathscr{G}(\overline{\mathsf{m}}, w) \to \overline{\mathsf{b}} = w)] = \mathsf{T}$ and $\mathsf{N}[\forall z(\mathscr{H}(\overline{\mathsf{b}}, z) \to \overline{\mathsf{a}} = z)] = \mathsf{T}$; so that for a given m, there is just one $w = \mathsf{b}$ and so one $z = \mathsf{a}$ to satisfy $\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)$; and $\mathsf{N}[\forall z(\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)) \to \overline{\mathsf{a}} = z)] = \mathsf{T}$.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ is expressed by $\mathscr{H}(\vec{x}, y, u, v)$. And the expression of $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ in terms of $\mathscr{G}$ and $\mathscr{H}$ utilizes Gödel's β-function, as developed in the next section.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed by some $\mathscr{G}(\vec{x}, y, z)$. Then $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed by $\mathscr{F}(\vec{x}, v) = \mathscr{G}(\vec{x}, v, \overline{0}) \wedge (\forall y < v)\exists z(\mathscr{G}(\vec{x}, y, z) \wedge \overline{0} \neq z)$.[6]

Suppose $\vec{x}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f}$; then $\langle\langle \mathsf{m}, \mathsf{a}\rangle, 0\rangle \in \mathsf{g}$ and for any $\mathsf{n} < \mathsf{a}$, $\langle\langle \mathsf{m}, \mathsf{n}\rangle, 0\rangle \notin \mathsf{g}$.

(a) Since $\langle\langle \mathsf{m}, \mathsf{a}\rangle, 0\rangle \in \mathsf{g}$, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, \overline{0})] = \mathsf{T}$. And since for $\mathsf{n} < \mathsf{a}$, $\langle\langle \mathsf{m}, \mathsf{n}\rangle, 0\rangle$ is not in (total function) g, for $\mathsf{n} < \mathsf{a}$, there is some $\mathsf{b} \neq 0$ such

[5] Perhaps it has already occurred to you that $\mathsf{idnt}^3_2(\mathsf{x}, \mathsf{y}, \mathsf{z})$, say, is expressed by the somewhat cleaner $x = x \wedge y = v \wedge z = z$. This illustrates the point that different formulas may express the same function. Our formulation permits an "economy" of reasoning we shall find convenient (see also note 6).

[6] We might appeal to the somewhat cleaner $\mathscr{G}(\vec{x}, v, \overline{0}) \wedge (\forall y < v) \sim \mathscr{G}(\vec{x}, y, \overline{0})$. However, our formulation smooths results down the line—especially, with the negation applied to the atomic equality, $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed by a Σ_1 formula (as discussed below, page 589).

that $\langle\langle \mathsf{m},\mathsf{n}\rangle,\mathsf{b}\rangle \in \mathsf{g}$; so for $\mathsf{n} < \mathsf{a}$, there is a b such that $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{\mathsf{b}})] = \mathsf{T}$ and $\mathsf{N}[\overline{0} \neq \overline{\mathsf{b}}] = \mathsf{T}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{\mathsf{b}}) \wedge \overline{0} \neq \overline{\mathsf{b}}] = \mathsf{T}$; so with T10.2, $\mathsf{N}_{\mathsf{d}(z|\mathsf{b})}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},z) \wedge \overline{0} \neq z] = \mathsf{S}$; so $\mathsf{N}_{\mathsf{d}}[\exists z(\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},z) \wedge \overline{0} \neq z)] = \mathsf{S}$; so with T10.2 again, $\mathsf{N}_{\mathsf{d}(y|\mathsf{n})}[\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] = \mathsf{S}$; and since this is so for any $\mathsf{n} < \mathsf{a}$, with T12.4, $\mathsf{N}_{\mathsf{d}}[(\forall y < \overline{\mathsf{a}})\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] = \mathsf{S}$; and since there are no free variables, $\mathsf{N}[(\forall y < \overline{\mathsf{a}})\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] = \mathsf{T}$. So with T8.8, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{0}) \wedge (\forall y < \overline{\mathsf{a}})\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] = \mathsf{T}$; so $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{a}})] = \mathsf{T}$.

(b) We begin showing that for any n, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}}) \to \overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$. For any n, $\mathsf{n} < \mathsf{a}$ or $\mathsf{a} = \mathsf{n}$ or $\mathsf{a} < \mathsf{n}$. (i) Suppose $\mathsf{a} = \mathsf{n}$; then $\mathsf{N}[\overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$; so $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}}) \to \overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$. (ii) Suppose $\mathsf{n} < \mathsf{a}$; then $\langle\langle \mathsf{m},\mathsf{n}\rangle,0\rangle \notin \mathsf{g}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{0})] \neq \mathsf{T}$; and since $\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{0})$ is a conjunct of $\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}})$, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}})] \neq \mathsf{T}$; so $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}}) \to \overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$. (iii) Suppose $\mathsf{a} < \mathsf{n}$. First, for any b, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{\mathsf{b}}) \wedge \overline{0} \neq \overline{\mathsf{b}}] \neq \mathsf{T}$: For any b, $0 = \mathsf{b}$ or $0 \neq \mathsf{b}$. Say $0 = \mathsf{b}$; then $\mathsf{N}[\overline{0} \neq \overline{\mathsf{b}}] \neq \mathsf{T}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{\mathsf{b}}) \wedge \overline{0} \neq \overline{\mathsf{b}}] \neq \mathsf{T}$. Say $0 \neq \mathsf{b}$; then since $\langle\langle \mathsf{m},\mathsf{a}\rangle,0\rangle$ is in (function) g, $\langle\langle \mathsf{m},\mathsf{a}\rangle,\mathsf{b}\rangle \notin \mathsf{g}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{\mathsf{b}})] \neq \mathsf{T}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{\mathsf{b}}) \wedge \overline{0} \neq \overline{\mathsf{b}}] \neq \mathsf{T}$. So for any b, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},\overline{\mathsf{b}}) \wedge \overline{0} \neq \overline{\mathsf{b}}] \neq \mathsf{T}$; so with T10.2, $\mathsf{N}_{\mathsf{d}(z|\mathsf{b})}[\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},z) \wedge \overline{0} \neq z] \neq \mathsf{S}$; and since this is so for any b, $\mathsf{N}_{\mathsf{d}}[\exists z(\mathscr{G}(\overline{\mathsf{m}},\overline{\mathsf{a}},z) \wedge \overline{0} \neq z)] \neq \mathsf{S}$; so with T10.2, $\mathsf{N}_{\mathsf{d}(y|\mathsf{a})}[\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] \neq \mathsf{S}$; and since $\mathsf{a} < \mathsf{n}$, with T12.4, $\mathsf{N}_{\mathsf{d}}[(\forall y < \overline{\mathsf{n}})\exists z(\mathscr{G}(\overline{\mathsf{m}},y,z) \wedge \overline{0} \neq z)] \neq \mathsf{S}$; and since this is a conjunct of $\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}})$, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}})] \neq \mathsf{T}$; so $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}}) \to \overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$. From (i), (ii), and (iii), for any n, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}}) \to \overline{\mathsf{a}} = \overline{\mathsf{n}}] = \mathsf{T}$; so with T10.2, $\mathsf{N}_{\mathsf{d}(u|\mathsf{n})}[\mathscr{F}(\overline{\mathsf{m}},u) \to \overline{\mathsf{a}} = u] = \mathsf{S}$; so $\mathsf{N}_{\mathsf{d}}[\forall u(\mathscr{F}(\overline{\mathsf{m}},u) \to \overline{\mathsf{a}} = u)] = \mathsf{S}$; and since there are no free variables, $\mathsf{N}[\forall u(\mathscr{F}(\overline{\mathsf{m}},u) \to \overline{\mathsf{a}} = u)] = \mathsf{T}$.

Indct: Any recursive $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{F}(\vec{x},v)$.

Some of the reasoning is merely sketched, however the general idea should be clear.[7] There might be formulas other than the stated $\mathscr{F}(\vec{x},v)$ to express a recursive $\mathsf{f}(\vec{\mathsf{x}})$: We have seen examples already in notes; and similarly, if $\mathscr{F}(\vec{x},v)$ expresses $\mathsf{f}(\vec{\mathsf{x}})$, then so does $\mathscr{F}(\vec{x},v) \wedge \mathscr{A}$ for any tautology $\mathscr{A}$. We shall see an important alternative to $\mathscr{F}(\vec{x},v)$ in the following. Let us say that the $\mathscr{F}(\vec{x},v)$ here-described is the *original* formula by which $\mathsf{f}(\vec{\mathsf{x}})$ is expressed. Of course, it remains to fill out the case for the recursion clause. This is the task of the next section.

E12.5. Show (ii) of T12.3 and the case for $(\exists x \leq t)\mathscr{P}$ of T12.4. These should be straightforward, given parts worked in the text.

---

[7]Note that we have dropped the (obvious) sub-conclusion for the "show" step of the induction—that merely repeats the initial "show" line.

*E12.6. From T12.5 there is some formula to express any recursive function: the argument by induction works by showing how to *construct* a formula for each recursive function. Following the method of our induction, write down formulas to express the following recursive functions:

a. $\mathsf{zero(x)}$

b. $\mathsf{suc(zero(x))}$

Hint: As setup for the compositions, give each function a different output variable, where the output to one is the input to the next.

*E12.7. For the (c) clause to T12.5, in the case where $\vec{\mathsf{x}}$ and $\vec{\mathsf{z}}$ drop out and $\vec{\mathsf{y}}$ reduces to a single variable y, fill out semantic reasoning to demonstrate that proposed (original) formula satisfies the uniqueness condition for expression—that $\mathsf{N}[\forall z(\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)) \rightarrow \overline{\mathsf{a}} = z)] = \mathsf{T}$. In places you may find that T10.2 will smooth the result.

12.2.3 The β-Function

Suppose a function $\mathsf{f(m, n)}$ is defined by recursion and $\mathsf{f(m, n) = a}$. Then for the given value of m, there is a sequence $\mathsf{k_0, k_1, \ldots, k_n}$ with $\mathsf{k_n = a}$, such that $\mathsf{k_0}$ takes some initial value, and each of the other members is specially related to the one before. Thus, in the simple case of $\mathsf{times(m, n)}$, if $\mathsf{m = 2}$ then $\mathsf{k_0 = 0}$, and each $\mathsf{k_i}$ adds two to the one before. So corresponding to $\mathsf{2 \times 5 = 10}$ is the sequence,

$$\mathsf{0 \quad 2 \quad 4 \quad 6 \quad 8 \quad 10}$$

whose first member is set by $\mathsf{gtimes(2)}$, where subsequent members result from the one before by $\mathsf{times(2, Sy) = htimes(2, y, times(2, y))}$, whose last member is $\mathsf{10}$. And, generalizing, we shall be in a position to express functions defined by recursion if we can express the existence of *sequences* of integers so defined. We shall be able to say $\mathsf{f(m, n) = a}$ if we can say "there is a sequence whose first member is $\mathsf{g(m)}$, with members related one to another by $\mathsf{f(m, Sy) = h(m, y, f(m, y))}$, whose $\mathsf{n^{th}}$ member (counting from zero) is a." This is a mouthful. And $\mathscr{L}_{NT}$ is not obviously equipped to do it. In particular, $\mathscr{L}_{NT}$ has straightforward mechanisms for asserting the existence of natural numbers—but on its face, it is not clear how to assert the existence of the arbitrary sequences which result from the recursion clause.

But Gödel shows a way out. We have already seen an instance of the general strategy we shall require in our discussion of Gödel numbering from Chapter 10. In that case, we took a sequence of integers (keyed to vocabulary), $g_0, g_1, \ldots, g_n$ and collected them into a single Gödel number $G = 2^{g_0} \times 3^{g_1} \times \cdots \times p_n^{g_n}$ where $2, 3, \ldots, p_n$ are the first n primes. By the fundamental theorem of arithmetic, any number has a unique prime factorization, so the original sequence is recovered from G by factoring to find the power of 2, the power of 3, and so forth. So the single integer G represents the original sequence. And $\mathscr{L}_{NT}$ has no problem expressing the existence of a single integer! Unfortunately, however, this particular way out is unavailable to us insofar as it involves exponentiation, and the resources of $\mathscr{L}_{NT}$ so-far include only S, $+$, and $\times$.[8]

All the same, within the resources of $\mathscr{L}_{NT}$, with the Chinese remainder theorem, there must be *pairs* of integers sufficient to represent any finite sequence.[9] Consider the *remainder* function $\mathsf{rem(x, y)}$ which returns the remainder after x is divided by y. The *remainder* of x divided by y equals z just in case $\mathsf{z} < \mathsf{y}$ and for some w, $\mathsf{x = (y \times w) + z}$. Then let,

$$\beta(\mathsf{p, q, i}) = \mathsf{rem[p, S(q \times S(i))]}$$

So for some fixed values of p and q the β function yields different remainders for different values of i. With the Chinese remainder theorem, for any sequence $\mathsf{k_0}$, $\mathsf{k_1, \ldots, k_n}$ there are some p and q such that for $\mathsf{i \leq n}$, $\beta(\mathsf{p, q, i}) = \mathsf{k_i}$. So p and q together code the sequence—given p and q to code the sequence, the β-function returns each member $\mathsf{k_i}$ as a function of i. Intuitively, when we divide p by $\mathsf{S(q \times S(i))}$ for $\mathsf{i \leq n}$, the result is a series of $n + 1$ remainders. The theorem tells us that *any* series $\mathsf{k_0, k_1, \ldots, k_n}$ may be so represented (see the beta function reference on the following page).

Here is a simple example. Suppose $\mathsf{k_0}$, $\mathsf{k_1}$, and $\mathsf{k_2}$ are $\mathsf{5, 2, 3}$. We require p and q such that $\beta(\mathsf{p, q, 0}) = \mathsf{5}$, $\beta(\mathsf{p, q, 1}) = \mathsf{2}$, and $\beta(\mathsf{p, q, 2}) = \mathsf{3}$. The last subscript n in our series $\mathsf{k_0}$, $\mathsf{k_1}$, $\mathsf{k_2}$ equals $\mathsf{2}$. As developed in the beta function reference, let s be the maximum of $\mathsf{n, 5, 2, 3}$, and then set $\mathsf{q = s!}$ In this case, $\mathsf{s = 5}$ and so $\mathsf{s! = 120}$. So $\beta(\mathsf{p, q, i}) = \mathsf{rem[p, S(120 \times S(i))]}$. So as i ranges between $\mathsf{0}$ and $\mathsf{n = 2}$, we are looking at,

$$\mathsf{rem(p, 121)} \qquad \mathsf{rem(p, 241)} \qquad \mathsf{rem(p, 361)}$$

But $\mathsf{121}$, $\mathsf{241}$, and $\mathsf{361}$ so constructed must have no common factor other than $\mathsf{1}$; and under this condition as p varies between $\mathsf{0}$ and $\mathsf{121 \times 241 \times 361 - 1 = 10527120}$ the remainders must take on every possible sequence of remainder values. But the remainders will be values up to $\mathsf{120}$, $\mathsf{240}$, and $\mathsf{360}$, which is to say, $\mathsf{q = s!}$ is large enough that our simple sequence must therefore appear among the sequences of remainders. In this case, $\mathsf{p = 261728}$ gives $\mathsf{rem(p, 121) = 5}$, $\mathsf{rem(p, 241) = 2}$, and $\mathsf{rem(p, 361) = 3}$. There may be easier ways to generate this sequence. But there is

[8]Some treatments begin with a language including exponentiation precisely in order to smooth the exposition at this stage. But our results are all the more interesting insofar as even the relatively weak $\mathscr{L}_{NT}$ retains powers sufficient for the fatal flaw.

[9]The remainder theorem generalizes a problem posed in a third-century treatise, *Sun Zi Suanjing*. Given finitely many remainder/denominator pairs whose denominators have no primes in common, the remainder theorem finds a unique numerator to yield the remainder for each denominator.

Arithmetic for the Beta Function

Say $\mathsf{rem}(c, d)$ is the remainder of c/d. For a sequence, $d_0, d_1, \ldots, d_n$, let $|D|$ be the product $d_0 \times d_1 \times \cdots \times d_n$. We say $d_0, d_1, \ldots, d_n$ are *relatively prime* if no two members have a common factor other than 1. Then,

I. For any relatively prime sequence $d_0, d_1, \ldots, d_n$, the sequences of remainders $\langle \mathsf{rem}(c, d_0), \mathsf{rem}(c, d_1), \ldots, \mathsf{rem}(c, d_n) \rangle$ as c runs from 0 to $|D| - 1$ are all different from each other.

   Suppose otherwise. Then there are c_1 and c_2, $0 \leq c_1 < c_2 < |D|$ such that $\langle \mathsf{rem}(c_1, d_0), \ldots, \mathsf{rem}(c_1, d_n) \rangle$ is the same as $\langle \mathsf{rem}(c_2, d_0), \ldots, \mathsf{rem}(c_2, d_n) \rangle$. So for each d_i, $\mathsf{rem}(c_1, d_i) = \mathsf{rem}(c_2, d_i)$, and there are some a, b and r such that $c_1 = ad_i + r$ and $c_2 = bd_i + r$; then since the remainders are equal, $c_2 - c_1 = bd_i - ad_i$; so each d_i divides $c_2 - c_1$ evenly. So each d_i collects a distinct set of prime factors of $c_2 - c_1$; and since $c_2 - c_1$ is divided by any product of its primes, $c_2 - c_1$ is divided by $|D|$. So $|D| \leq c_2 - c_1$. But $0 \leq c_1 < c_2 < |D|$ so $c_2 - c_1 < |D|$. Reject the assumption: The sequences of remainders as c runs from 0 to $|D| - 1$ are distinct.

II. For any relatively prime sequence $d_0, d_1, \ldots, d_n$, the sequences of remainders $\langle \mathsf{rem}(c, d_0), \mathsf{rem}(c, d_1), \ldots, \mathsf{rem}(c, d_n) \rangle$ as c runs from 0 to $|D| - 1$ are all the possible sequences of remainders.

   There are d_i possible remainders a number might have when divided by d_i, $(0, 1, \ldots, d_i - 1)$. But if $\mathsf{rem}(c, d_0)$ takes d_0 possible values, $\mathsf{rem}(c, d_1)$ may take its d_1 values for each value of $\mathsf{rem}(c, d_0)$; and so forth. So the there are $|D|$ possible sequences of remainders. But as c runs from 0 to $|D| - 1$, by (I), there are $|D|$ different sequences. So there are all the possible sequences.

III. Let s be the maximum of $n, k_0, k_1, \ldots, k_n$. Then for $0 \leq i \leq n$, the numbers $d_i = s!(i + 1) + 1$—that is, $S(s! \times S(i))$—are greater than any k_j and relatively prime.

   Since s is the maximum of $n, k_0, k_1, \ldots, k_n$, the first is obvious. To see that the d_i are relatively prime, suppose otherwise. Then for some a, b, $0 \leq a < b \leq n$, $s!(a + 1) + 1$ and $s!(b + 1) + 1$ have a common factor p; so, absorbing the inner addition, for some $j = a + 1$ and $k = b + 1$, $0 < j < k \leq n + 1$, $s!j + 1$ and $s!k + 1$ have a common factor p. Any number up to s leaves remainder 1 when dividing $s!j + 1$; so $p > s$. But since p divides $s!j + 1$ and $s!k + 1$ it divides their difference, $s!(k - j)$; if p divides $s!$, then it does not evenly divide $s!j + 1$; so p does not divide $s!$; so p divides $k - j$. But $0 < j < k \leq n + 1$; so $k - j \leq n$; so $p \leq n$; and since by the construction $n \leq s$, we get $p \leq s$. Reject the assumption: the d_i are relatively prime.

IV. For any $k_0, k_1, \ldots, k_n$, there is a pair of numbers p, q such that for $i \leq n$, $\beta(p, q, i) = k_i$.

   With s as above, set $q = s!$, and let $\beta(p, q, i) = \mathsf{rem}[p, S(q \times S(i))]$. By (III), for $0 \leq i \leq n$ the numbers $S(q \times S(i))$ are relatively prime. So by (II), there are all the possible sequences of remainders as p ranges from 0 to $|D| - 1$. And since by (III) each d_i is greater than any k_j, the sequence $k_0, k_1, \ldots, k_n$ is among the possible sequences of remainders. So there is some p such that the k_i are $\mathsf{rem}[p, S(q \times S(i))]$.

no shortage of integers (!) so there are no worries about using large ones, and by this method Gödel gives a perfectly general way to represent the arbitrary finite sequence.

And we can express the β-function with the resources of $\mathscr{L}_{NT}$. Thus, for $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i})$,

$$\mathscr{B}(p, q, i, v) = (\exists w \leq p)[p = (S(q \times Si) \times w) + v \wedge v < S(q \times Si)]$$

So v is the remainder after p is divided by $S(q \times Si)$. And for appropriate choice of p and q, the variable v takes on the values k_0 through k_n as i runs through the values $\mathsf{0}$ to n.

Now return to our claim that when a function defined by recursion $\mathsf{f}(\mathsf{m}, \mathsf{n}) = \mathsf{a}$ there is a sequence $\mathsf{k}_0, \mathsf{k}_1, \ldots, \mathsf{k}_\mathsf{n}$ with $\mathsf{k}_\mathsf{n} = \mathsf{a}$ such that k_0 takes some initial value, and each of the other members is recursively related to the one before. More officially, $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y}) = \mathsf{z}$ just in case there is a sequence $\mathsf{k}_0, \mathsf{k}_1, \ldots, \mathsf{k}_\mathsf{y}$ with,

(i) $\mathsf{k}_0 = \mathsf{g}(\vec{\mathsf{x}})$

(ii) if $\mathsf{i} < \mathsf{y}$, then $\mathsf{k}_{\mathsf{Si}} = \mathsf{h}(\vec{\mathsf{x}}, \mathsf{i}, \mathsf{k}_\mathsf{i})$

(iii) $\mathsf{k}_\mathsf{y} = \mathsf{z}$

Put in terms of the β-function, this requires $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y}) = \mathsf{z}$ just in case there are some p, q such that,

(i) $\beta(\mathsf{p}, \mathsf{q}, \mathsf{0}) = \mathsf{g}(\vec{\mathsf{x}})$

(ii) if $\mathsf{i} < \mathsf{y}$, then $\beta(\mathsf{p}, \mathsf{q}, \mathsf{Si}) = \mathsf{h}(\vec{\mathsf{x}}, \mathsf{i}, \beta(\mathsf{p}, \mathsf{q}, \mathsf{i}))$

(iii) $\beta(\mathsf{p}, \mathsf{q}, \mathsf{y}) = \mathsf{z}$

By assumption, $\mathsf{g}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by some $\mathscr{H}(\vec{x}, y, u, v)$. So we can express the combination of these conditions as follows: $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed by $\mathscr{F}(\vec{x}, y, z) =$

$$\exists p \exists q \{\exists v[\mathscr{B}(p, q, \emptyset, v) \wedge \mathscr{G}(\vec{x}, v)] \wedge$$

$$(\forall i < y)\exists u \exists v[\mathscr{B}(p, q, i, u) \wedge \mathscr{B}(p, q, Si, v) \wedge \mathscr{H}(\vec{x}, i, u, v)] \wedge$$

$$\mathscr{B}(p, q, y, z)\}$$

So $\mathscr{G}$ is satisfied by the first member; then for any $i < y$, $\mathscr{H}$ is satisfied by the i^{th} member and its successor; and the y^{th} member of the series is z. Observe that we do not find particular values for p and q—rather it is enough that there *is* a sequence whose members meet the recursive conditions.

In the case of factorial, $\mathsf{gfact}() = \hat{\mathsf{1}}$ is expressed by $\mathscr{G}(v) = (\overline{1} = v)$ and $\mathsf{hfact}(\mathsf{y}, \mathsf{u}) = \mathsf{times}(\mathsf{u}, \mathsf{suc}(\mathsf{y}))$ by $\mathscr{H}(y, u, v) = (u \times Sy = v)$. Given this, by our method, the factorial function is expressed by $\mathscr{F}(y, z) =$

$$\exists p \exists q \{\exists v[\mathscr{B}(p,q,\emptyset,v) \wedge \overline{1} = v] \wedge$$

$$(\forall i < y)\exists u \exists v[\mathscr{B}(p,q,i,u) \wedge \mathscr{B}(p,q,Si,v) \wedge u \times Si = v] \wedge$$

$$\mathscr{B}(p,q,y,z)\}$$

This expression is long—particularly if expanded to unabbreviate $\mathscr{B}$.[10] But the result is what we want: if $\langle \mathsf{n}, \mathsf{a}\rangle \in \mathsf{fac}$, then $\mathsf{N}[\mathscr{F}(\overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$ and $\mathsf{N}[\forall w(\mathscr{F}(\overline{\mathsf{n}}, w) \rightarrow \overline{\mathsf{a}} = w)] = \mathsf{T}$. Thus we express functions defined by recursion, and complete the demonstration of T12.5.

So far, our discussion *exhibits* a formula to express the recursion clause and, ideally, explains how it works by the β-function. This is sufficient for the rest of the text—and you may decide to skip directly to exercises. Even so, we can demonstrate more explicitly that if $\langle \mathsf{m}, \mathsf{n}, \mathsf{a}\rangle \in \mathsf{f}$, then (i) $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$, and (ii) $\mathsf{N}[\forall w(\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \rightarrow \overline{\mathsf{a}} = w)] = \mathsf{T}$. Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{n}, \mathsf{a}\rangle \in \mathsf{f}$. Then there are $\mathsf{k}_0 \dots \mathsf{k}_\mathsf{n}$ such that $\mathsf{g}(\mathsf{m}) = \mathsf{k}_0$; $\mathsf{k}_\mathsf{n} = \mathsf{a}$; and there are p, q such that for $0 \le \mathsf{i} < \mathsf{n}$, $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i}) = \mathsf{k}_\mathsf{i}$, $\beta(\mathsf{p}, \mathsf{q}, \mathsf{Si}) = \mathsf{k}_{\mathsf{Si}}$, and $\mathsf{h}(\mathsf{m}, \mathsf{i}, \mathsf{k}_\mathsf{i}) = \mathsf{k}_{\mathsf{Si}}$. To manage long formulas let,

$$\mathscr{P}(p,q,x) = \exists v[\mathscr{B}(p,q,\emptyset,v) \wedge \mathscr{G}(x,v)]$$

$$\mathscr{Q}(p,q,x,y) = (\forall i < y)\exists u \exists v[\mathscr{B}(p,q,i,u) \wedge \mathscr{B}(p,q,Si,v) \wedge \mathscr{H}(x,i,u,v)]$$

Then $\mathscr{F}(x,y,z) = \exists p \exists q[\mathscr{P}(p,q,x) \wedge \mathscr{Q}(p,q,x,y) \wedge \mathscr{B}(p,q,y,z)]$. Here is an outline of reasoning for (i):

> Consider an arbitrary variable assignment d. With expression for $\mathscr{G}$ and $\mathscr{B}$, $\mathsf{N}_\mathsf{d}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{k}}_0)] = \mathsf{S}$ and $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{0}, \overline{\mathsf{k}}_0)] = \mathsf{S}$; so $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{0}, \overline{\mathsf{k}}_0) \wedge \mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{k}}_0)] = \mathsf{S}$; so generalizing, $\mathsf{N}_\mathsf{d}[\mathscr{P}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{m}})] = \mathsf{S}$. Suppose $\mathsf{i} < \mathsf{n}$; then with expression for $\mathscr{B}$ and $\mathscr{H}$, $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, \overline{\mathsf{k}}_\mathsf{i})] = \mathsf{S}$, and $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, S\overline{\mathsf{i}}, \overline{\mathsf{k}}_{\mathsf{Si}})] = \mathsf{S}$, and $\mathsf{N}_\mathsf{d}[\mathscr{H}(\overline{\mathsf{m}}, \overline{\mathsf{i}}, \overline{\mathsf{k}}_\mathsf{i}, \overline{\mathsf{k}}_{\mathsf{Si}})] = \mathsf{S}$; so $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, \overline{\mathsf{k}}_\mathsf{i}) \wedge \mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, S\overline{\mathsf{i}}, \overline{\mathsf{k}}_{\mathsf{Si}}) \wedge \mathscr{H}(\overline{\mathsf{m}}, \overline{\mathsf{i}}, \overline{\mathsf{k}}_\mathsf{i}, \overline{\mathsf{k}}_{\mathsf{Si}})] = \mathsf{S}$; so $\mathsf{N}_\mathsf{d}[\exists u \exists v(\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, u) \wedge \mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, S\overline{\mathsf{i}}, v) \wedge \mathscr{H}(\overline{\mathsf{m}}, \overline{\mathsf{i}}, u, v))] = \mathsf{S}$; since this is so for all $\mathsf{i} < \mathsf{n}$, with T12.4, $\mathsf{N}_\mathsf{d}[\mathscr{Q}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{m}}, \overline{\mathsf{n}})] = \mathsf{S}$. By expression for $\mathscr{B}$, $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{n}}, \overline{\mathsf{k}}_\mathsf{n})] = \mathsf{S}$; so $\mathsf{N}_\mathsf{d}[\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{S}$. So $\mathsf{N}_\mathsf{d}[\mathscr{P}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{m}}) \wedge \mathscr{Q}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{m}}, \overline{\mathsf{n}}) \wedge \mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{S}$; so generalizing, $\mathsf{N}_\mathsf{d}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{S}$; and since d is arbitrary, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$.

Expression yields truth at the level of the parts, and we reason from truth at the level of the parts to truth for the whole. To see (ii) that $\mathsf{N}[\forall w(\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \rightarrow \overline{\mathsf{a}} = w)] = \mathsf{T}$, we shall be able to show that uniqueness at the level of the parts yields uniqueness for the whole. In this case, the result "propagates" from one stage to the next, and we reason by induction on the value of n. For an outline of the argument, see the box on the next page.

[10]Even more, although this $\mathscr{G}$ and $\mathscr{H}$ express gfact and hfact, they are not the same as the more complex formulas that would result from composition and such by the method of T12.5.

T12.5(r.ii)

To show $N[\forall w(\mathscr{F}(\overline{m},\overline{n},w) \to \overline{a} = w)] = T$, it will be convenient to lapse into induction scheme III from the Chapter 8 induction schemes reference—starting with $n = \overline{0}$, making the assumption for a single member of the series n, and showing that it holds for the next.

Basis: Suppose $N[\forall w(\mathscr{F}(\overline{m},\overline{0},w) \to \overline{k}_0 = w)] \neq T$; then there is some assignment h and object b such that $N_h[\mathscr{F}(\overline{m},\overline{0},\overline{b}) \to \overline{k}_0 = \overline{b}] \neq S$; so $N_h[\mathscr{F}(\overline{m},\overline{0},\overline{b})] = S$ and $N_h[\overline{k}_0 = \overline{b}] \neq S$. With the latter, $k_0 \neq b$. With the former, there are objects p and q such that $N_h[\mathscr{P}(\overline{p},\overline{q},\overline{m}) \wedge \mathscr{Q}(\overline{p},\overline{q},\overline{m},\overline{0}) \wedge \mathscr{B}(\overline{p},\overline{q},\overline{0},\overline{b})] = S$; so ***(a)*** $N_h[\mathscr{P}(\overline{p},\overline{q},\overline{m})] = S$ and ***(b)*** $N_h[\mathscr{B}(\overline{p},\overline{q},\overline{0},\overline{b})] = S$. With (a) there is some v such that $N_h[\mathscr{B}(\overline{p},\overline{q},\overline{0},\overline{v}) \wedge \mathscr{G}(\overline{m},\overline{v})] = S$; so ***(c)*** $N_h[\mathscr{B}(\overline{p},\overline{q},\overline{0},\overline{v})] = S$ and ***(d)*** $N_h[\mathscr{G}(\overline{m},\overline{v})] = S$. By uniqueness, $N[\forall z(\mathscr{G}(\overline{m},z) \to \overline{k}_0 = z)] = T$; so with (d), $k_0 = v$. But by (c) and expression, $\langle p,q,0,v\rangle \in \beta$; so by uniqueness, $N[\forall z(\mathscr{B}(\overline{p},\overline{q},\overline{0},z) \to \overline{v} = z)] = T$; so with (b), $v = b$; so $k_0 = b$. This is impossible; reject the assumption: $N[\forall w(\mathscr{F}(\overline{m},\overline{0},w) \to \overline{k}_0 = w)] = T$.

Assp: $N[\forall w(\mathscr{F}(\overline{m},\overline{n},w) \to \overline{k}_n = w)] = T$.

Show: $N[\forall w(\mathscr{F}(\overline{m},S\overline{n},w) \to \overline{k}_{Sn} = w)] = T$. Suppose otherwise; then there is some assignment h and object b such that $N_h[\mathscr{F}(\overline{m},S\overline{n},\overline{b}) \to \overline{k}_{Sn} = \overline{b}] \neq S$; so $N_h[\mathscr{F}(\overline{m},S\overline{n},\overline{b})] = S$ and $N_h[\overline{k}_{Sn} = \overline{b}] \neq S$. With the latter, $k_{Sn} \neq b$. With the former, there are objects p and q such that $N_h[\mathscr{P}(\overline{p},\overline{q},\overline{m}) \wedge \mathscr{Q}(\overline{p},\overline{q},\overline{m},S\overline{n}) \wedge \mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{b})] = S$; so ***(a)*** $N_h[\mathscr{P}(\overline{p},\overline{q},\overline{m})] = S$ and ***(b)*** $N_h[\mathscr{Q}(\overline{p},\overline{q},\overline{m},S\overline{n})] = S$ and ***(c)*** $N_h[\mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{b})] = S$. From (b), $N_h[(\forall i < S\overline{n})\exists u\exists v(\mathscr{B}(\overline{p},\overline{q},i,u) \wedge \mathscr{B}(\overline{p},\overline{q},Si,v) \wedge \mathscr{H}(\overline{m},i,u,v))] = S$; so with T12.4, ***(d)*** for all $i < Sn$, $N_h[\exists u\exists v(\mathscr{B}(\overline{p},\overline{q},\overline{i},u) \wedge \mathscr{B}(\overline{p},\overline{q},S\overline{i},v) \wedge \mathscr{H}(\overline{m},\overline{i},u,v))] = S$; but $n < Sn$, so $N_h[\exists u\exists v(\mathscr{B}(\overline{p},\overline{q},\overline{n},u) \wedge \mathscr{B}(\overline{p},\overline{q},S\overline{n},v) \wedge \mathscr{H}(\overline{m},\overline{n},u,v))] = S$; so for some u and v, $N_h[\mathscr{B}(\overline{p},\overline{q},\overline{n},\overline{u}) \wedge \mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{v}) \wedge \mathscr{H}(\overline{m},\overline{n},\overline{u},\overline{v})] = S$; so ***(e)*** $N_h[\mathscr{B}(\overline{p},\overline{q},\overline{n},\overline{u})] = S$ and ***(f)*** $N_h[\mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{v})] = S$ and ***(g)*** $N_h[\mathscr{H}(\overline{m},\overline{n},\overline{u},\overline{v})] = S$; since any $i < n$ is less than Sn, by (d), for all $i < n$, $N_h[\exists u\exists v(\mathscr{B}(\overline{p},\overline{q},\overline{i},u) \wedge \mathscr{B}(\overline{p},\overline{q},S\overline{i},v) \wedge \mathscr{H}(\overline{m},\overline{i},u,v))] = S$, so with T12.4 ***(h)*** $N_h[\mathscr{Q}(\overline{p},\overline{q},\overline{m},\overline{n})] = S$; so with (a,h,e) $N_h[\mathscr{P}(\overline{p},\overline{q},\overline{m}) \wedge \mathscr{Q}(\overline{p},\overline{q},\overline{m},\overline{n}) \wedge \mathscr{B}(\overline{p},\overline{q},\overline{n},\overline{u})] = S$; generalizing, $N_h[\mathscr{F}(\overline{m},\overline{n},\overline{u})] = S$; so with the assumption, $k_n = u$; so with (g), $N_h[\mathscr{H}(\overline{m},\overline{n},\overline{k}_n,\overline{v})] = S$; but by uniqueness, $N[\forall z(\mathscr{H}(\overline{m},\overline{n},\overline{k}_n,z) \to \overline{k}_{Sn} = z)] = T$; so $k_{Sn} = v$; so with (f), $N_h[\mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{k}_{Sn})] = S$; so by expression $\langle p,q,Sn,k_{Sn}\rangle \in \beta$; so by uniqueness, $N[\forall z(\mathscr{B}(\overline{p},\overline{q},S\overline{n},z) \to \overline{k}_{Sn} = z)] = T$; and since from (c) $N_h[\mathscr{B}(\overline{p},\overline{q},S\overline{n},\overline{b})] = S$, $k_{Sn} = b$. This is impossible.

Indct: For any n, $N[\forall w(\mathscr{F}(\overline{m},\overline{n},w) \to \overline{k}_n = w)] = T$.

So $N[\forall w(\mathscr{F}(\overline{m},\overline{n},w) \to \overline{a} = w)] = T$.

E12.8. Suppose k_0, k_1, k_2, k_3 are 3, 2, 0, 1. Find values of p and q so that $\beta(i) = k_i$. Use your values of p and q to calculate $\beta(p, q, 0)$, $\beta(p, q, 1)$, $\beta(p, q, 2)$, and $\beta(p, q, 3)$. From an introduction to modular arithmetic (such as would be part of a course in number theory), it is possible to calculate p directly from the remainder/denominator pairs. Otherwise you may employ some programmable device to search for the value of p. In Ruby, a routine along the following lines, with numerical values for the remainders `r0...r3` and deominators `d0...d3` should suffice.

```
r = [r0, r1, r2, r3]
d = [d0, d1, d2, d3]
p = 0
puts "p = #{p}"
until p%d[0]==r[0] and p%d[1]==r[1] and p%d[2]==r[2] and p%d[3]==r[3]
  p = p + 1
  puts "p = #{p}"
end
puts "rem(#{p},#{d[0]}) = #{p % d[0]}, rem(#{p},#{d[1]}) = #{p % d[1]}"
puts "rem(#{p},#{d[2]}) = #{p % d[2]}, rem(#{p},#{d[3]}) = #{p % d[3]}"
```

In Ruby `x % y` returns the remainder of `x` divided by `y`. So, for this routine, you insert the remainder and denominator values, and then search (by brute force) for the value of p to return the remainders.

E12.9. Produce a formula to show that $\mathscr{L}_{NT}$ expresses the plus function by the initial functions with the beta function. You need not reduce the beta form to its primitive expression!

E12.10. Say a function f_k is *simple* iff there is a series of functions $f_0, f_1, \ldots, f_k$ such that for any $i \leq k$,

(b) $f_0(x, y)$ is $\mathrm{plus}(x, y)$

(r) There are $a, b < i$ such that $f_i(x, y)$ is $\mathrm{plus}(f_a(x, y), f_b(x, y))$

Show that on the standard interpretation N of $\mathscr{L}_{NT}$ each simple $f(x, y)$ is expressed by some formula $\mathcal{F}(x, y, v)$. You may appeal to T10.2 as appropriate—and your reasoning may have the "quick" character of T12.5. Hint: (r) works by a sort of "double" composition.

12.3 Capturing Recursive Functions

The second of the powers to be associated with theory ĩncompleteness has to do with the theory's *proof* system. In section 12.5.2 (and again in 13.1.2 and 14.2.3) we show that if a theory is consistent and *captures* recursive functions, then it is negation ĩncomplete. Thus we shall have separate paths to ĩncompleteness: drawing upon the previous section, one through expression and s̃oundness; and drawing upon this section, another through capture and consistency. In this section, then, as a ground for that second argument, we show that Q, and so any theory that includes Q, captures the recursive functions.

12.3.1 Definition and First Results

Where expression requires that if objects stand in a given relation, then a corresponding formula be true, capture requires that when objects stand in a relation, a corresponding formula be *provable* in the theory. In this section we define capture and then set up the argument that Q captures the recursive functions by extending, to a wider range of sentences, our T8.18 result that Q decides atomic sentences of $\mathcal{L}_{NT}$.[11]

CP For any language $\mathcal{L}$, interpretation I, objects $\mathsf{m}_1 \dots \mathsf{m}_n, \mathsf{a} \in \mathsf{U}$ and theory T,

(r) Relation $\mathsf{R}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is *captured* by formula $\mathcal{R}(x_1 \dots x_n)$ in T just in case,

(i) if $\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle \in \mathsf{R}$ then $T \vdash \mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)$

(ii) if $\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle \notin \mathsf{R}$ then $T \vdash \sim\mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)$

(f) Function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is *captured* by formula $\mathcal{F}(x_1 \dots x_n, y)$ in T just in case,

if $\langle \langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a} \rangle \in \mathsf{f}$ then,

(i) $T \vdash \mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$

(ii) $T \vdash \forall z(\mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, z) \to \overline{\mathsf{a}} = z)$

Again, let us illustrate with some applications. First, in any T at least as strong as Q, $x = y$ captures $\mathsf{EQ}(\mathsf{x}, \mathsf{y})$. For if $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{EQ}$, then $\mathsf{m} = \mathsf{n}$ and by T8.14, $T \vdash \overline{\mathsf{m}} = \overline{\mathsf{n}}$; and if $\langle \mathsf{m}, \mathsf{n} \rangle \notin \mathsf{EQ}$ then $\mathsf{m} \neq \mathsf{n}$ and by T8.16, $T \vdash \overline{\mathsf{m}} \neq \overline{\mathsf{n}}$. Turning to a simple function, $\hat{\mathsf{n}}$ is captured by $\overline{\mathsf{n}} = v$—for if $\langle \mathsf{a} \rangle \in \hat{\mathsf{n}}$, then $\mathsf{n} = \mathsf{a}$ and we have both $T \vdash \overline{\mathsf{n}} = \overline{\mathsf{a}}$ and $T \vdash \forall z(\overline{\mathsf{n}} = z \to \overline{\mathsf{a}} = z)$. Similarly, in a theory at least as strong as Q, $\mathsf{plus}(\mathsf{x}, \mathsf{y})$ is captured by $x + y = v$, for if $\mathsf{m} + \mathsf{n} = \mathsf{a}$, then by T8.12, $T \vdash \overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{a}}$; and given $T \vdash \overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{a}}$, it is easy to see that $T \vdash \forall z(\overline{\mathsf{m}} + \overline{\mathsf{n}} = z \to \overline{\mathsf{a}} = z)$.

In addition, we can show that for a theory at least as strong as Q, condition (f.ii) yields a result like (r.ii).

T12.6. If T includes Q and total function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is captured by formula $\mathcal{F}(x_1 \dots x_n, y)$, then if $\langle \langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a} \rangle \notin \mathsf{f}$, $T \vdash \sim\mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

Suppose a total $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is captured by $\mathcal{F}(x_1 \dots x_n, y)$ and $\langle \langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a} \rangle \notin \mathsf{f}$. Then since f is total, there is some $\mathsf{b} \neq \mathsf{a}$ such that $\langle \langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{b} \rangle \in \mathsf{f}$; so by (f.ii),

[11] Again there is a problem of terminology for 'capture'. Different texts offer somewhat different definitions and employ somewhat different vocabulary. See note 4 on page 575.

$T \vdash \forall z(\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, z) \to \overline{\mathsf{b}} = z)$; so instantiating to $\overline{\mathsf{a}}$, $T \vdash \mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}}) \to \overline{\mathsf{b}} = \overline{\mathsf{a}}$. But since $\mathsf{b} \neq \mathsf{a}$ and T includes Q, by T8.16, $T \vdash \overline{\mathsf{b}} \neq \overline{\mathsf{a}}$; so by MT, $T \vdash \sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

Our aim is to show that recursive functions are captured in Q. In Chapter 8, we showed that Q correctly decides atomic sentences of $\mathscr{L}_{NT}$—and supporting results played a role in examples just above. As a preliminary to showing that Q captures all the recursive functions, we extend the result from Chapter 8 to show that Q correctly decides a broadened range of sentences, the Δ_0 sentences.

To set up this result, we identify some important subclasses of formulas in $\mathscr{L}_{NT}$: the Δ_0, Σ_1, and Π_1 formulas.

Δ_0
- (a) If $\mathscr{P}$ is of the form $s = t$, $s < t$, or $s \leq t$ for terms s and t, then $\mathscr{P}$ is a Δ_0 formula.
- (b) If $\mathscr{P}$ and $\mathscr{Q}$ are Δ_0 formulas, then so are $\sim\mathscr{P}$ and $(\mathscr{P} \to \mathscr{Q})$.
- (c) If $\mathscr{P}$ is a Δ_0 formula, then so are $(\forall x \leq t)\mathscr{P}$ and $(\forall x < t)\mathscr{P}$.
- (CL) Any Δ_0 formula may be formed by repeated application of these rules.

Recall that, for the bounded quantifiers, variable x does not appear in the bound t. We allow the usual abbreviations and so $\wedge$, $\vee$, $\leftrightarrow$, and bounded existentials. Δ_0 formulas take equalities and inequalities as basic—after that, all is as usual except that quantifiers are bounded. Because of the restriction to bounded quantifiers, a Δ_0 sentence is true or false by some (potentially complex) fact about a finite collection of numbers. Then Σ_1 and Π_1 formulas take Δ_0 formulas as basic elements.

Σ_1
- (a) If $\mathscr{P}$ is a Δ_0 formula, then $\mathscr{P}$ is a Σ_1 formula.
- (b) If $\mathscr{P}$ is a Σ_1 formula, so is $\exists x\mathscr{P}$.
- (c) If $\mathscr{P}$ and $\mathscr{Q}$ are Σ_1 formulas, then so are $(\mathscr{P} \wedge \mathscr{Q})$ and $(\mathscr{P} \vee \mathscr{Q})$.
- (d) If $\mathscr{P}$ is a Σ_1 formula, then so are $(\exists x \leq t)\mathscr{P}$ and $(\exists x < t)\mathscr{P}$.
- (e) If $\mathscr{P}$ is a Σ_1 formula, then so are $(\forall x \leq t)\mathscr{P}$ and $(\forall x < t)\mathscr{P}$.
- (CL) Any Σ_1 formula may be formed by repeated application of these rules.

Given the Δ_0 formulas, operators are $\wedge$, $\vee$, bounded quantifiers, and the unbounded existential. From the sigma-1 and pi-1 reference on page 591, Σ_1 formulas have simplified equivalents of the sort $\exists\vec{v}\mathscr{Q}$, where $\mathscr{Q}$ is Δ_0 and $\exists\vec{v}$ is a block of zero or more unbounded existential quantifiers. This helps to exhibit what Σ_1 sentences *say:* a Σ_1 sentence is true when some assignment d satisfies the Δ_0 condition.

Π_1
- (a) If $\mathscr{P}$ is a Δ_0 formula, then $\mathscr{P}$ is a Π_1 formula.
- (b) If $\mathscr{P}$ is a Π_1 formula, so is $\forall x\mathscr{P}$.

(c) If $\mathcal{P}$ and $\mathcal{Q}$ are Π_1 formulas, then so are $(\mathcal{P} \wedge \mathcal{Q})$ and $(\mathcal{P} \vee \mathcal{Q})$.

(d) If $\mathcal{P}$ is a Π_1 formula, then so are $(\exists x \leq t)\mathcal{P}$ and $(\exists x < t)\mathcal{P}$.

(e) If $\mathcal{P}$ is a Π_1 formula, then so are $(\forall x \leq t)\mathcal{P}$ and $(\forall x < t)\mathcal{P}$.

(CL) Any Π_1 formula may be formed by repeated application of these rules.

Given the Δ_0 formulas, operators are $\wedge$, $\vee$, bounded quantifiers, and the unbounded universal. From the sigma-1 and pi-1 reference, Π_1 formulas have simplified equivalents of the sort $\forall\vec{v}\mathcal{Q}$, where $\mathcal{Q}$ is Δ_0 and $\forall\vec{v}$ is a block of zero or more unbounded universal quantifiers. Again, this helps exhibit what Π_1 formulas say: a Π_1 sentence is true when every assignment d satisfies the Δ_0 condition.

Given their simplified equivalents, the negation of a Σ_1 formula is equivalent to some $\sim\exists\vec{v}\mathcal{Q}$, and so by reasoning as for QN, to a Π_1 formula. And similarly, the negation of a Π_1 formula is equivalent to a Σ_1 formula. Thus, so long as they are not themselves equivalent to Δ_0 formulas, the negation of a Σ_1 formula is not Σ_1, and the negation of a Π_1 formula is not Π_1.

We show that Q correctly decides Δ_0 sentences: If $\mathcal{P}$ is Δ_0 and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathcal{P}$. Further, Q *proves* true Σ_1 sentences: If $\mathcal{P}$ is Σ_1 and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathrm{Q} \vdash \mathcal{P}$. As we have just seen, when $\mathcal{P}$ is Σ_1, $\sim\mathcal{P}$ might not be Σ_1. So, though we show Q proves true Σ_1 sentences, we will not have shown that Q proves $\sim\mathcal{P}$ when $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ and so not have shown that Q decides all Σ_1 sentences. First, then, Q correctly decides Δ_0 sentences of $\mathcal{L}_{NT}$. And then Q proves true Σ_1 sentences.

T12.7. For any Δ_0 sentence $\mathcal{P}$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathcal{P}$.

Suppose $\mathcal{P}$ is a Δ_0 sentence. Treating equalities and inequalities as atomic, the argument is by induction on the number of operators in $\mathcal{P}$.

Basis: If $\mathcal{P}$ is atomic it is $t = s$, $t \leq s$, or $t < s$. So by T8.18, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathcal{P}$.

Assp: For any $i, 0 \leq i < k$, if a Δ_0 sentence $\mathcal{P}$ has i operator symbols, then if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathcal{P}$.

Show: If a Δ_0 sentence $\mathcal{P}$ has k operator symbols, then if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathcal{P}$.

If a Δ_0 sentence $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, $(\forall x \leq t)\mathcal{A}$, or $(\forall x < t)\mathcal{A}$ where $\mathcal{A}$, $\mathcal{B}$ have $< k$ operator symbols and x does not appear in t.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$. Since $\mathcal{P}$ is a Δ_0 sentence, $\mathcal{A}$ is a Δ_0 sentence. (i) Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[\sim\mathcal{A}] = \mathsf{T}$; so by T8.8, $\mathsf{N}[\mathcal{A}] \neq \mathsf{T}$; so by assumption, $\mathrm{Q} \vdash \sim\mathcal{A}$; so $\mathrm{Q} \vdash \mathcal{P}$. (ii) Suppose $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$; then $\mathsf{N}[\sim\mathcal{A}] \neq \mathsf{T}$; so by T8.8, $\mathsf{N}[\mathcal{A}] = \mathsf{T}$; so by assumption $\mathrm{Q} \vdash \mathcal{A}$; so by DN, $\mathrm{Q} \vdash \sim\sim\mathcal{A}$; so $\mathrm{Q} \vdash \sim\mathcal{P}$.

Sigma-1 and Pi-1 Formulas

The Δ_0, Σ_1, and Π_1 formulas are the first stages of a more general measure of complexity for formulas of $\mathscr{L}_{NT}$ (the *arithmetical hierarchy*). Given Δ_0 formulas, *strict* versions of Σ_1 and Π_1 formulas are typically introduced as follows:

Σ_1 A formula $\mathcal{P}$ is *strictly* Σ_1 iff there are zero or more existential quantifiers such that $\mathcal{P} = \exists x_1 \exists x_2 \ldots \exists x_n \mathcal{Q}$ for Δ_0 formula $\mathcal{Q}$.

Π_1 A formula $\mathcal{P}$ is *strictly* Π_1 iff there are zero or more universal quantifiers such that $\mathcal{P} = \forall x_1 \forall x_2 \ldots \forall x_n \mathcal{Q}$ for Δ_0 formula $\mathcal{Q}$.

From (a) and (b) of the original definitions, each strictly Σ_1 formula is Σ_1, and each strictly Π_1 formula is Π_1. But further, where $\mathcal{P}$ and $\mathcal{Q}$ are *equivalent* just in case on any d, $\mathsf{N_d}[\mathcal{P}] = \mathsf{N_d}[\mathcal{Q}]$, each Σ_1 formula is equivalent to a strictly Σ_1 formula, and each Π_1 formula to a strictly Π_1 formula.

To show that each Σ_1 formula $\mathcal{P}$ is equivalent to a strictly Σ_1 formula, reasoning is by induction on the number of operators in $\mathcal{P}$. With standard quantifier placement rules, reasoning is straightforward except when $\mathcal{P}$ is $(\forall x \leq t)\mathcal{A}$ or $(\forall x < t)\mathcal{A}$. For the first, by assumption $\mathcal{A}$ is equivalent to some $\exists\vec{v}\mathcal{A}'$ for Δ_0 formula $\mathcal{A}'$, and $\mathcal{P}$ to $(\forall x \leq t)\exists\vec{v}\mathcal{A}'$. This time, standard quantifier placement rules do not suffice.

For a simplified case, consider $(\forall x \leq y)\exists v\mathcal{A}'(x, v)$; this requires that for each $\mathsf{x} \leq \mathsf{y}$ there is at least one v to satisfy $\mathcal{A}'(x, v)$; for each $\mathsf{x} \leq \mathsf{y}$ consider the least such v, and let a be the greatest member of this collection. Then $(\forall x \leq y)(\exists v \leq \overline{\mathsf{a}})\mathcal{A}'(x, v)$ is equivalent to the original expression—for given an $\mathsf{x} \leq \mathsf{y}$, if there is a v to satisfy $\mathcal{A}'$, then there is a $\mathsf{v} \leq \mathsf{a}$ to satisfy $\mathcal{A}'$, and if there is a $\mathsf{v} \leq \mathsf{a}$ to satisfy $\mathcal{A}'$, then there is some v to satisfy $\mathcal{A}'$. If the original expression is satisfied, there is some such a; and if it is not satisfied, then for some $\mathsf{x} < \mathsf{y}$ there is no v that satisfies $\mathcal{A}'(x, v)$—and so no a under which such a v could appear. And therefore, $\exists z(\forall x \leq y)(\exists v \leq z)\mathcal{A}'(x, v)$ is satisfied iff the original expression is satisfied. Thus the existential quantifier comes past the bounded universal, leaving behind a bounded existential "shadow."

In general, it is not proper to drag an existential quantifier out past a universal quantifier; however, it is legitimate to drag an existential past a *bounded* universal, leaving behind a bounded existential shadow. Observe that, corresponding to this semantic equivalence, PA $\vdash (\forall x \leq y)\exists v\mathcal{F}xv \leftrightarrow \exists z(\forall x \leq y)(\exists v \leq z)\mathcal{F}xv$; for this see T13.11ag.

Reasoning by induction is similar to show that each Π_1 formula is equivalent to a strictly Π_1 formula. For the case $(\exists x \leq y)\forall v\mathcal{A}'(x, v)$, begin with $\mathsf{N_d}[(\forall x \leq y)\exists v{\sim}\mathcal{A}'(x, v)] = \mathsf{N_d}[\exists z(\forall x \leq y)(\exists v \leq z){\sim}\mathcal{A}'(x, v)]$ from above. Then consider the negation of both sides. By considerations parallel to QN and then DN, $\mathsf{N_d}[(\exists x \leq y)\forall v\mathcal{A}'(x, v)] = \mathsf{N_d}[\forall z(\exists x \leq y)(\forall v \leq z)\mathcal{A}'(x, v)]$. So the universal comes past the bounded existential leaving behind a bounded universal shadow.

($\rightarrow$) $\mathcal{P}$ is $\mathcal{A} \rightarrow \mathcal{B}$. Since $\mathcal{P}$ is a Δ_0 sentence, $\mathcal{A}$ and $\mathcal{B}$ are Δ_0 sentences. (i) Suppose $\mathsf{N}[\mathcal{A} \rightarrow \mathcal{B}] = \mathsf{T}$; then by T8.8, $\mathsf{N}[\mathcal{A}] \neq \mathsf{T}$ or $\mathsf{N}[\mathcal{B}] = \mathsf{T}$; so by assumption, $\mathsf{Q} \vdash \sim\mathcal{A}$ or $\mathsf{Q} \vdash \mathcal{B}$; in either case, by $\vee$I, $\mathsf{Q} \vdash \sim\mathcal{A} \vee \mathcal{B}$; so by Impl, $\mathsf{Q} \vdash \mathcal{A} \rightarrow \mathcal{B}$. Part (ii) is homework.

($\forall \leq$) $\mathcal{P}$ is $(\forall x \leq t)\mathcal{A}(x)$. Since $\mathcal{P}$ is a Δ_0 sentence, $\mathcal{A}$ is a Δ_0 formula whose only free variable is x. In addition, since x does not appear in t, t must be variable-free; so $\mathsf{N_d}[t] = \mathsf{N}[t]$ and where $\mathsf{N}[t] = \mathsf{n}$, by T8.17, $\mathsf{Q} \vdash t = \overline{\mathsf{n}}$; so by =E, $\mathsf{Q} \vdash \mathcal{P}$ just in case $\mathsf{Q} \vdash (\forall x \leq \overline{\mathsf{n}})\mathcal{A}(x)$.

(i) Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{T}$; so by TI, for any d, $\mathsf{N_d}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{S}$; so by T12.4, for any $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; so where $\mathsf{N_d}[t] = \mathsf{N}[t] = \mathsf{n}$, for any $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; but $\mathsf{N_d}[\overline{\mathsf{m}}] = \mathsf{m}$, so with T10.2, $\mathsf{N_d}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{S}$; so since d is arbitrary, $\mathsf{N}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{T}$; so $\mathsf{N}[\mathcal{A}(\emptyset)] = \mathsf{T}$ and $\mathsf{N}[\mathcal{A}(\overline{1})] = \mathsf{T}$ and ... and $\mathsf{N}[\mathcal{A}(\overline{\mathsf{n}})] = \mathsf{T}$; so by assumption, $\mathsf{Q} \vdash \mathcal{A}(\emptyset)$ and $\mathsf{Q} \vdash \mathcal{A}(\overline{1})$ and ... and $\mathsf{Q} \vdash \mathcal{A}(\overline{\mathsf{n}})$; so by T8.25, $\mathsf{Q} \vdash (\forall x \leq \overline{\mathsf{n}})\mathcal{A}(x)$; so with our preliminary result, $\mathsf{Q} \vdash \mathcal{P}$.

(ii) Suppose $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$; then $\mathsf{N}[(\forall x \leq t)\mathcal{A}(x)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{N_d}[(\forall x \leq t)\mathcal{A}(x)] \neq \mathsf{S}$; so by T12.4, for some $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] \neq \mathsf{S}$; so where $\mathsf{N_d}[t] = \mathsf{N}[t] = \mathsf{n}$, for some $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] \neq \mathsf{S}$; but $\mathsf{N_d}[\overline{\mathsf{m}}] = \mathsf{m}$, so with T10.2, $\mathsf{N_d}[\mathcal{A}(\overline{\mathsf{m}})] \neq \mathsf{S}$; so by TI, $\mathsf{N}[\mathcal{A}(\overline{\mathsf{m}})] \neq \mathsf{T}$; so by assumption, $\mathsf{Q} \vdash \sim\mathcal{A}(\overline{\mathsf{m}})$; so by T8.24, $\mathsf{Q} \vdash (\exists x \leq \overline{\mathsf{n}})\sim\mathcal{A}(x)$; so by RQN, $\mathsf{Q} \vdash \sim(\forall x \leq \overline{\mathsf{n}})\mathcal{A}(x)$; so with our preliminary result, $\mathsf{Q} \vdash \sim\mathcal{P}$.

($\forall <$) Homework.

Indct: For any Δ_0 sentence $\mathcal{P}$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathsf{Q} \vdash \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathsf{Q} \vdash \sim\mathcal{P}$.

T12.8. For any Σ_1 sentence $\mathcal{P}$ if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash \mathcal{P}$.

Suppose $\mathcal{P}$ is a Σ_1 sentence. Treating Δ_0 formulas as atomic, the argument is by induction on the number of operator symbols in $\mathcal{P}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a Δ_0 sentence. Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then by T12.7, $\mathsf{Q} \vdash \mathcal{P}$.

Assp: For any i, $0 \leq i < k$, if a Σ_1 sentence $\mathcal{P}$ has i operator symbols and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash \mathcal{P}$.

Show: If a Σ_1 sentence $\mathcal{P}$ has k operator symbols and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash \mathcal{P}$.

If a Σ_1 sentence $\mathcal{P}$ has k operator symbols, then it is of the form $\exists x\mathcal{A}$, $\mathcal{A} \wedge \mathcal{B}$, $\mathcal{A} \vee \mathcal{B}$, $(\exists x \leq t)\mathcal{A}$, $(\exists x < t)\mathcal{A}$, $(\forall x \leq t)\mathcal{A}$, or $(\forall x < t)\mathcal{A}$ where $\mathcal{A}$, $\mathcal{B}$ have $< k$ operator symbols and x does not appear in t.

As a preliminary to the bounded quantifier cases, consider say $\mathcal{P} = (\exists x \leq t)\mathcal{A}(x)$; since $\mathcal{P}$ is a sentence and x does not appear in t, t is variable-free; so $\mathsf{N_d}[t] = \mathsf{N}[t]$ and where $\mathsf{N}[t] = \mathsf{n}$, by T8.17, $\mathsf{Q} \vdash t = \overline{\mathsf{n}}$; so by

=E, Q $\vdash \mathcal{P}$ just in case Q $\vdash (\exists x \leq \overline{\mathsf{n}})\mathcal{A}(x)$. And similarly for the other bounded quantifiers.

($\exists$) $\mathcal{P}$ is $\exists x\mathcal{A}(x)$. Since $\mathcal{P}$ is a Σ_1 sentence, $\mathcal{A}$ is a Σ_1 formula whose only free variable is x. Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[\exists x\mathcal{A}(x)] = \mathsf{T}$; so for any d, $\mathsf{N_d}[\exists x\mathcal{A}(x)] = \mathsf{S}$; so there is some m such that $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; but $\mathsf{N_d}[\overline{\mathsf{m}}] = \mathsf{m}$, so with T10.2, $\mathsf{N_d}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{S}$; so since d is arbitrary, $\mathsf{N}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{T}$; so by assumption Q $\vdash \mathcal{A}(\overline{\mathsf{m}})$; so by $\exists$I, Q $\vdash \exists x\mathcal{A}(x)$, which is to say Q $\vdash \mathcal{P}$.

($\wedge$) $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$. Since $\mathcal{P}$ is a Σ_1 sentence, $\mathcal{A}$ and $\mathcal{B}$ are Σ_1 sentences. Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[\mathcal{A} \wedge \mathcal{B}] = \mathsf{T}$; so with T8.8, $\mathsf{N}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{N}[\mathcal{B}] = \mathsf{T}$, and by assumption, Q $\vdash \mathcal{A}$ and Q $\vdash \mathcal{B}$; so with $\wedge$I, Q $\vdash \mathcal{A} \wedge \mathcal{B}$, which is to say Q $\vdash \mathcal{P}$. And similarly for $\mathcal{A} \vee \mathcal{B}$.

($\exists \leq$) $\mathcal{P}$ is $(\exists x \leq t)\mathcal{A}(x)$. Since $\mathcal{P}$ is a Σ_1 sentence, $\mathcal{A}$ is a Σ_1 formula whose only free variable is x. Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[(\exists x \leq t)\mathcal{A}(x)] = \mathsf{T}$; so for any d, $\mathsf{N_d}[(\exists x \leq t)\mathcal{A}(x)] = \mathsf{S}$; so with T12.4, for some $\mathsf{m} \leq \mathsf{N_d}[t] = \mathsf{N}[t] = \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; but $\mathsf{N_d}[\overline{\mathsf{m}}] = \mathsf{m}$, so with T10.2, $\mathsf{N_d}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{S}$; so since d is arbitrary, $\mathsf{N}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{T}$; so by assumption, Q $\vdash \mathcal{A}(\overline{\mathsf{m}})$; so by T8.24, Q $\vdash (\exists x \leq \overline{\mathsf{n}})\mathcal{A}(x)$. So by our preliminary result, Q $\vdash \mathcal{P}$. And similarly for $(\exists x < t)\mathcal{A}(x)$.

($\forall \leq$) $\mathcal{P}$ is $(\forall x \leq t)\mathcal{A}(x)$. Since $\mathcal{P}$ is a Σ_1 sentence, $\mathcal{A}$ is a Σ_1 formula whose only free variable is x. Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{T}$; so for any d, $\mathsf{N_d}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{S}$; so by T12.4, for any $\mathsf{m} \leq \mathsf{N_d}[t] = \mathsf{N}[t] = \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; but $\mathsf{N_d}[\overline{\mathsf{m}}] = \mathsf{m}$, so with T10.2, $\mathsf{N_d}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{S}$; so since d is arbitrary, $\mathsf{N}[\mathcal{A}(\overline{\mathsf{m}})] = \mathsf{T}$; so by assumption, Q $\vdash \mathcal{A}(\overline{\mathsf{m}})$; so Q $\vdash \mathcal{A}(\overline{0})$ and ... and Q $\vdash \mathcal{A}(\overline{\mathsf{n}})$; so by T8.25, Q $\vdash (\forall x \leq \overline{\mathsf{n}})\mathcal{A}(x)$. So by our preliminary result, Q $\vdash \mathcal{P}$. And similarly for $(\forall x < t)\mathcal{A}(x)$.

Indct: For any Σ_1 sentence $\mathcal{P}$ if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then Q $\vdash \mathcal{P}$.

These theorems complete what we set out to show in this subsection. And the results should seem intuitive: Q proves results about particular numbers, $1+1 = 2$ and the like. But Δ_0 sentences assert (potentially complex) particular facts about numbers—and so Q proves any true Δ_0 sentence. Similarly, a Σ_1 sentence is true *because* of some particular fact about numbers; since Q proves that particular fact, it is sufficient to prove the Σ_1 sentence.

*E12.11. Provide an argument to demonstrate (i) that each Σ_1 formula is equivalent to a strictly Σ_1 formula, and (ii) that each Π_1 formula to a strictly Π_1 formula. You may appeal to semantic versions of *ND+* quantifier rules along with reasoning from the sigma-1 and pi-1 reference as appropriate.

*E12.12. (i) Complete the demonstration of T12.7 by finishing the remaining cases. You should set up the entire argument, but may appeal to the text for parts already completed, as the text appeals to homework. (ii) Show directly the case for ($\exists\leq$).

E12.13. Provide an argument to fill in cases marked "and similarly" for T12.8. You should set up the entire argument, but may refer to the text for parts worked there.

E12.14. Reconsider the Hilbert strategy from the introduction to Part IV, and consider theory Q as it applies to Π_1 sentences. Show that Q satisfies condition (a) so that if a Π_1 sentence is not true, then $Q \vdash \sim\mathcal{P}$. Hint: As applied to *strictly* Π_1 sentences, this results immediately from T12.8 with (semantical and syntactical versions of) QN. But Q is not sufficient to demonstrate the biconditional between an arbitrary Π_1 formula and its strict equivalent—and the assignment is to produce an argument parallel to that for T12.8.

12.3.2 Basic Result

We now set out to show that Q captures all the recursive functions. We shall get our result in two forms. First a straightforward basic version. This version gets a result slightly weaker than we would like. However it is easily strengthened to the final form.

First the basic version. Here is the sense in which our result is weaker than we might like: Rather than Q, let us suppose we are in a system Q_s, *strengthened* Q, which has as a(n axiom or) theorem *uniqueness of remainder* as follows:

$$\forall y[((\exists w \leq m)[m = Sn \times w + \bar{\mathsf{a}} \wedge \bar{\mathsf{a}} < Sn] \wedge (\exists w \leq m)[m = Sn \times w + y \wedge y < Sn]) \rightarrow \bar{\mathsf{a}} = y]$$

If $\bar{\mathsf{a}}$ is the remainder of m/Sn and y is the remainder of m/Sn then $\bar{\mathsf{a}} = y$. Q does not itself prove this result.[12] One way to obtain Q_s is simply to add uniqueness of remainder to the axioms of Q; also, as we shall see (on page 661), PA has uniqueness of remainder as a theorem. If the free variables m and n from uniqueness of remainder are instantiated to p and $q \times Si$ respectively, there immediately follows a parallel uniqueness result for the original formula $\mathcal{B}(p,q,i,v) = (\exists w \leq p)[p = (S(q \times Si) \times w) + v \wedge v < S(q \times Si)]$ that expresses the β-function,

$$Q_s \vdash \forall y[(\mathcal{B}(p,q,i,\bar{\mathsf{a}}) \wedge \mathcal{B}(p,q,i,y)) \rightarrow \bar{\mathsf{a}} = y]$$

[12] As we have seen, Q is good at proving particular results as $\bar{1} \times \bar{2} = \bar{2} \times \bar{1}$. And given that for any m, n, $\mathsf{N}[\bar{\mathsf{m}} \times \bar{\mathsf{n}} = \bar{\mathsf{n}} \times \bar{\mathsf{m}}] = \mathsf{T}$, from T8.18 it follows that for any m and n, $Q \vdash \bar{\mathsf{m}} \times \bar{\mathsf{n}} = \bar{\mathsf{n}} \times \bar{\mathsf{m}}$—where arguments by mathematical induction, and so the quantification, are in the metalanguage. In contrast, $PA \vdash \forall m \forall n(m \times n = n \times m)$—with its induction axiom and so the quantification in the theory (T6.66). Uniqueness of remainder is a result of this latter sort not provable in Q.

Further, if $\langle\langle \mathsf{p}, \mathsf{q}, \mathsf{i}\rangle, \mathsf{a}\rangle \in \beta$ then since $\mathcal{B}$ expresses the β-function, $\mathsf{N}[\mathcal{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, \overline{\mathsf{a}})] = \mathsf{T}$; and since $\mathcal{B}$ is Δ_0, by T12.7, $\mathrm{Q} \vdash \mathcal{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, \overline{\mathsf{a}})$. From this, with uniqueness, it is immediate that $\mathrm{Q_s} \vdash \forall y[\mathcal{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, \overline{\mathsf{i}}, y) \to \overline{\mathsf{a}} = y]$. So $\mathcal{B}$ *captures* β in $\mathrm{Q_s}$.

Simple inspection reveals that original formulas from T12.5 to express recursive functions are all Σ_1: Formulas to express the initial functions and β-function are Δ_0 and so Σ_1; after that, if $\mathcal{G}$ and $\mathcal{H}$ are Σ_1 then original formulas built from them are Σ_1. Given this, since Q proves true Σ_1 formulas, Q (and so $\mathrm{Q_s}$) proves any true original formula. With this, we are positioned to offer a straightforward argument for capture of the recursive functions in $\mathrm{Q_s}$. Again our main argument is an induction on the sequence of recursive functions. We show that $\mathrm{Q_s}$ captures the initial functions, and then that it captures functions from composition, recursion, and regular minimization.

*T12.9. On the standard interpretation N for $\mathcal{L}_{NT}$, any recursive function is captured in $\mathrm{Q_s}$ by the original formula by which it is expressed.

By induction on the sequence of recursive functions,

Basis: f_0 is an initial function $\mathsf{suc(x)}$, $\mathsf{zero()}$, or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \dots \mathsf{x}_\mathsf{j})$.

(s) The original formula $\mathcal{F}(x, v)$ by which $\mathsf{suc(x)}$ is expressed is $Sx = v$. Suppose $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{suc}$.

(i) Since $Sx = v$ expresses $\mathsf{suc(x)}$, $\mathsf{N}[S\overline{\mathsf{m}} = \overline{\mathsf{a}}] = \mathsf{T}$; so, since it is Δ_0, by T12.7, $\mathrm{Q} \vdash S\overline{\mathsf{m}} = \overline{\mathsf{a}}$; so $\mathrm{Q_s} \vdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Reason as follows:

1.	$S\overline{\mathsf{m}} = \overline{\mathsf{a}}$	from (i)
2.	$S\overline{\mathsf{m}} = j$	A (g, →I)
3.	$\overline{\mathsf{a}} = j$	2,1 =E
4.	$S\overline{\mathsf{m}} = j \to \overline{\mathsf{a}} = j$	2-3 →I
5.	$\forall z(S\overline{\mathsf{m}} = z \to \overline{\mathsf{a}} = z)$	4 ∀I

So $\mathrm{Q_s} \vdash \forall z[\mathcal{F}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{a}} = z]$.

(oth) It is left as homework to show that $\mathsf{zero()}$ is captured by $\overline{0} = v$ and $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \dots \mathsf{x}_\mathsf{j})$ by $(x_1 = x_1 \wedge \dots \wedge x_j = x_j) \wedge x_k = v$.

Assp: For any i, $0 \leq i < k$, $\mathsf{f_i}(\vec{\mathsf{x}})$ is captured in $\mathrm{Q_s}$ by the original formula by which it is expressed.

Show: $\mathsf{f_k}(\vec{\mathsf{x}})$ is captured in $\mathrm{Q_s}$ by the original formula by which it is expressed.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion, or regular minimization. If it is an initial function, then reason as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ is captured by some $\mathcal{G}(\vec{y}, w)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ by $\mathcal{H}(\vec{x}, w, \vec{z}, v)$; the original formula $\mathcal{F}(\vec{x}, \vec{y}, \vec{z}, v)$ by which the composition $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is expressed is $\exists w[\mathcal{G}(\vec{y}, w) \wedge \mathcal{H}(\vec{x}, w, \vec{z}, v)]$. For simplicity, consider a case where $\vec{\mathsf{x}}$ and $\vec{\mathsf{z}}$ drop out and $\vec{\mathsf{y}}$ is a single variable y. Suppose $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$; then by composition there is some b such that $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{g}$ and $\langle \mathsf{b}, \mathsf{a}\rangle \in \mathsf{h}$.

(i) Since $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$, and $\mathcal{F}(y, v)$ expresses $\mathsf{f_k}$, $\mathsf{N}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; so, since $\mathcal{F}(y, v)$ is Σ_1, by T12.8, $\mathrm{Q_s} \vdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Since $\mathcal{G}(y, w)$ captures $\mathsf{g(y)}$ and $\mathcal{H}(w, v)$ captures $\mathsf{h(w)}$, by assumption $\mathrm{Q_s} \vdash \forall z(\mathcal{G}(\overline{\mathsf{m}}, z) \rightarrow \overline{\mathsf{b}} = z)$ and $\mathrm{Q_s} \vdash \forall z(\mathcal{H}(\overline{\mathsf{b}}, z) \rightarrow \overline{\mathsf{a}} = z)$. It is then a simple derivation for you to show that $\mathrm{Q_s} \vdash \forall z(\exists w[\mathcal{G}(\overline{\mathsf{m}}, w) \wedge \mathcal{H}(w, z)] \rightarrow \overline{\mathsf{a}} = z)$.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathcal{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\mathcal{H}(\vec{x}, y, u, v)$; the original formula $\mathcal{F}(\vec{x}, y, z)$ by which $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed is,

$$\exists p \exists q\{\exists v[\mathcal{B}(p, q, \emptyset, v) \wedge \mathcal{G}(\vec{x}, v)] \wedge$$
$$(\forall i < y)\exists u \exists v[\mathcal{B}(p, q, i, u) \wedge \mathcal{B}(p, q, Si, v) \wedge \mathcal{H}(\vec{x}, i, u, v)] \wedge \mathcal{B}(p, q, y, z)\}$$

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f_k}$. (i) Then since $\mathcal{F}(x, y, z)$ expresses $\mathsf{f_k}$, $\mathsf{N}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$; so, since $\mathcal{F}(x, y, z)$ is Σ_1, by T12.8, $\mathrm{Q_s} \vdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})$. And (ii) by T12.10 just below, $\mathrm{Q_s} \vdash \forall w[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \rightarrow \overline{\mathsf{a}} = w]$.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ is captured by some $\mathcal{G}(\vec{x}, y, z)$; the original formula $\mathcal{F}(\vec{x}, v)$ by which $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed is $\mathcal{G}(\vec{x}, v, \overline{0}) \wedge (\forall y < v)\exists z(\mathcal{G}(\vec{x}, y, z) \wedge \overline{0} \neq z)$. Suppose $\vec{x}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$.

(i) Since $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$ and $\mathcal{F}(x, v)$ expresses $\mathsf{f_k}$, $\mathsf{N}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; so since $\mathcal{F}(x, v)$ is Σ_1, by T12.8, $\mathrm{Q_s} \vdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Since $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$, we have $\langle\langle \mathsf{m}, \mathsf{a}\rangle, 0\rangle \in \mathsf{g}$, and for $\mathsf{n} < \mathsf{a}$, $\langle\langle \mathsf{m}, \mathsf{n}\rangle, 0\rangle \notin \mathsf{g}$. With the first of these, $\mathrm{Q_s} \vdash \forall z[\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, z) \rightarrow \overline{0} = z]$. From the second with T12.6, for $\mathsf{n} < \mathsf{a}$, $\mathrm{Q_s} \vdash \sim\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{0})$; so by T8.25, $\mathrm{Q_s} \vdash (\forall z < \overline{\mathsf{a}})\sim\mathcal{G}(\overline{\mathsf{m}}, z, \overline{0})$. Then by the derivation in the box on the next page $\mathrm{Q_s} \vdash \forall u[\mathcal{F}(\overline{\mathsf{m}}, u) \rightarrow \overline{\mathsf{a}} = u]$.

Indct: Any recursive $\mathsf{f}(\vec{\mathsf{x}})$ is captured in $\mathrm{Q_s}$ by the original formula that expresses it.

For this theorem, each part (i) simply relies on the ability of $\mathrm{Q_s}$ to prove particular truths, and so the original Δ_0 and Σ_1 sentences that express recursive functions. The uniqueness clauses are not Σ_1, so we have to show them directly. The uniqueness result for recursion remains outstanding, and is addressed in the theorem immediately following:

*T12.10. Suppose $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ results by recursion from functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ where $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathcal{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\mathcal{H}(\vec{x}, y, u, v)$. Then for the original expression $\mathcal{F}(\vec{x}, y, z)$, if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_b, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$, $\mathrm{Q}_s \vdash \forall w[\mathcal{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_b, \overline{\mathsf{n}}, w) \rightarrow \overline{\mathsf{a}} = w]$.

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{n}, \mathsf{a}\rangle \in \mathsf{f}$. Then there are $\mathsf{k}_0 \ldots \mathsf{k}_n$ such that $\mathsf{g}(\mathsf{m}) = \mathsf{k}_0$; $\mathsf{k}_n = \mathsf{a}$; and there are p, q such that for $0 \le \mathsf{i} < \mathsf{n}$, $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i}) = \mathsf{k}_i$, $\beta(\mathsf{p}, \mathsf{q}, \mathsf{Si}) = \mathsf{k}_{Si}$, and $\mathsf{h}(\mathsf{m}, \mathsf{i}, \mathsf{k}_i) = \mathsf{k}_{Si}$. The argument by induction on the value of n is structurally parallel to semantic reasoning from page 586. Again to manage long formulas let,

$$\mathcal{P}(p, q, x) = \exists v[\mathcal{B}(p, q, \emptyset, v) \wedge \mathcal{G}(x, v)]$$

$$\mathcal{Q}(p, q, x, y) = (\forall i < y)\exists u \exists v[\mathcal{B}(p, q, i, u) \wedge \mathcal{B}(p, q, Si, v) \wedge \mathcal{H}(x, i, u, v)]$$

Then $\mathcal{F}(x, y, z) = \exists p \exists q[\mathcal{P}(p, q, x) \wedge \mathcal{Q}(p, q, x, y) \wedge \mathcal{B}(p, q, y, z)]$. And it will be convenient to lapse into induction scheme III from the Chapter 8 induction

T12.9(m)

1.	$\forall z[\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, z) \rightarrow \overline{0} = z]$	capture
2.	$(\forall z < \overline{\mathsf{a}})\sim\mathcal{G}(\overline{\mathsf{m}}, z, \overline{0})$	capture
3.	$\mathcal{F}(\overline{\mathsf{m}}, j)$	A (g, $\rightarrow$I)
4.	$\mathcal{G}(\overline{\mathsf{m}}, j, \overline{0}) \wedge (\forall y < j)\exists z(\mathcal{G}(\overline{\mathsf{m}}, y, z) \wedge \overline{0} \neq z)$	3 abv
5.	$\mathcal{G}(\overline{\mathsf{m}}, j, \overline{0})$	4 $\wedge$E
6.	$(\forall y < j)\exists z(\mathcal{G}(\overline{\mathsf{m}}, y, z) \wedge \overline{0} \neq z)$	4 $\wedge$E
7.	$j < \overline{\mathsf{a}} \vee \overline{\mathsf{a}} = j \vee \overline{\mathsf{a}} < j$	T8.23
8.	$j < \overline{\mathsf{a}}$	A (c, $\sim$I)
9.	$\sim\mathcal{G}(\overline{\mathsf{m}}, j, \overline{0})$	2,8 ($\forall$E)
10.	$\bot$	5,9 $\bot$I
11.	$j \not< \overline{\mathsf{a}}$	8-10 $\sim$I
12.	$\overline{\mathsf{a}} < j$	A (c, $\sim$I)
13.	$\exists z(\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, z) \wedge \overline{0} \neq z)$	6,12 ($\forall$E)
14.	$\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, k) \wedge \overline{0} \neq k$	A (c, 13$\exists$E)
15.	$\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, k)$	14 $\wedge$E
16.	$\mathcal{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, k) \rightarrow \overline{0} = k$	1 $\forall$E
17.	$\overline{0} = k$	16,15 $\rightarrow$E
18.	$\overline{0} \neq k$	14 $\wedge$E
19.	$\bot$	17,18 $\bot$I
20.	$\bot$	13,14-19 $\exists$E
21.	$\overline{\mathsf{a}} \not< j$	12-20 $\sim$I
22.	$\overline{\mathsf{a}} = j$	7,11,21 DS
23.	$\mathcal{F}(\overline{\mathsf{m}}, j) \rightarrow \overline{\mathsf{a}} = j$	3-22 $\rightarrow$I
24.	$\forall u[\mathcal{F}(\overline{\mathsf{m}}, u) \rightarrow \overline{\mathsf{a}} = u]$	23 $\forall$I

schemes reference—starting with $\mathsf{n} = \overline{0}$, making the assumption for a single member of the series n, and showing that it holds for the next.

Basis: Suppose $\mathsf{n} = 0$. From capture, $\mathrm{Q_s} \vdash \forall z[\mathscr{G}(\overline{\mathsf{m}}, z) \to \overline{\mathsf{k}}_0 = z]$. By uniqueness of remainder (and generalizing on p and q), $\mathrm{Q_s} \vdash \forall p \forall q \forall y[(\mathscr{B}(p, q, \emptyset, \overline{\mathsf{k}}_0) \wedge \mathscr{B}(p, q, \emptyset, y)) \to \overline{\mathsf{k}}_0 = y]$. Then it is easy to show $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}, \emptyset, w) \to \overline{\mathsf{k}}_0 = w]$.

Assp: $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to \overline{\mathsf{k}}_\mathsf{n} = w]$

Show: $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}, S\overline{\mathsf{n}}, w) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = w]$

From capture, $\mathrm{Q_s} \vdash \forall w[\mathscr{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{k}}_\mathsf{n}, w) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = w]$. By uniqueness of remainder, $\mathrm{Q_s} \vdash \forall p \forall q \forall y[(\mathscr{B}(p, q, S\overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{Sn}}) \wedge \mathscr{B}(p, q, S\overline{\mathsf{n}}, y)) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = y]$. See the derivation on the following page.

Indct: For any n, $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to \overline{\mathsf{k}}_\mathsf{n} = w]$.

Both the basis and show clauses require generalized uniqueness for $\mathscr{B}$: this is because uniqueness is being applied inside assumptions for ∃E, where p and q are arbitrary variables, rather than numerals $\overline{\mathsf{p}}$ and $\overline{\mathsf{q}}$—as would appear in a uniqueness condition for capture as $\forall y(\mathscr{B}(\overline{\mathsf{p}}, \overline{\mathsf{q}}, S\overline{\mathsf{n}}, y) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = y)$. So $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to \overline{\mathsf{a}} = w]$. So we satisfy the (r) case for T12.9. So the theorem is proved. And we have shown that $\mathrm{Q_s}$ has the resources to capture any recursive function.

This theorem has a number of attractive features: We show that recursive functions are captured by the original formulas that express them. A byproduct is that recursive functions are captured by Σ_1 formulas. The argument is a straightforward induction on the sequence of recursive functions, of a type we have seen before. On this basis, we might proceed directly to showing that consistent theories extending $\mathrm{Q_s}$ are ĩncomplete. But we have not obtained the more interesting result that recursive functions are captured in theory Q—and so that Q also has powers sufficient for the fatal flaw. It is that to which we now turn.

*E12.15. Complete the demonstration of T12.9 by completing the remaining cases, including the basis and part (ii) of the case for composition.

*E12.16. Produce a derivation to show the basis of T12.10.

E12.17. Return to the simple functions from E12.10. Show that on the standard interpretation N of $\mathscr{L}_{NT}$ each simple function $\mathsf{f}(\vec{\mathsf{x}})$ is captured in $\mathrm{Q_s}$ by the formula used to express it. Restrict appeal to external theorems just to your result from E12.10 and T8.18 as appropriate.

T12.10 (show)

1.	$\forall w[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to \overline{\mathsf{k}}_{\mathsf{n}} = w]$	by assumption
2.	$\forall w[\mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{n}}, w) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = w]$	capture
3.	$\forall p \forall q \forall y[(\mathcal{B}(p, q, S\overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{Sn}}) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, y)) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = y]$	uniqueness
4.	$\mathcal{F}(\overline{\mathsf{m}}, S\overline{\mathsf{n}}, j)$	A (g, $\to$I)
5.	$\exists p \exists q[\mathcal{P}(p, q, \overline{\mathsf{m}}) \wedge \mathcal{Q}(p, q, \overline{\mathsf{m}}, S\overline{\mathsf{n}}) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, j)]$	4 abv
6.	$\exists q[\mathcal{P}(p, q, \overline{\mathsf{m}}) \wedge \mathcal{Q}(p, q, \overline{\mathsf{m}}, S\overline{\mathsf{n}}) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, j)]$	A (g, 5$\exists$E)
7.	$\mathcal{P}(p, q, \overline{\mathsf{m}}) \wedge \mathcal{Q}(p, q, \overline{\mathsf{m}}, S\overline{\mathsf{n}}) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, j)$	A (g, 6$\exists$E)
8.	$\exists v[\mathcal{B}(p, q, \emptyset, v) \wedge \mathcal{G}(\overline{\mathsf{m}}, v)]$	7 $\wedge$E ($\mathcal{P}$)
9.	$(\forall i < S\overline{\mathsf{n}})\exists u \exists v[\mathcal{B}(p, q, i, u) \wedge \mathcal{B}(p, q, Si, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, i, u, v)]$	7 $\wedge$E ($\mathcal{Q}$)
10.	$\mathcal{B}(p, q, S\overline{\mathsf{n}}, j)$	7 $\wedge$E
11.	$\overline{\mathsf{n}} < S\overline{\mathsf{n}}$	T8.14
12.	$\exists u \exists v[\mathcal{B}(p, q, \overline{\mathsf{n}}, u) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u, v)]$	9,11 ($\forall$E)
13.	$\exists v[\mathcal{B}(p, q, \overline{\mathsf{n}}, u) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u, v)]$	A (g, 12$\exists$E)
14.	$\mathcal{B}(p, q, \overline{\mathsf{n}}, u) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u, v)$	A (g, 13$\exists$E)
15.	$\mathcal{B}(p, q, \overline{\mathsf{n}}, u)$	14 $\wedge$E
16.	$a < \overline{\mathsf{n}}$	A (g ($\forall$I))
17.	$\emptyset \neq \emptyset \vee a = \overline{\mathsf{0}} \vee \ldots \vee a = \overline{\mathsf{n}-1}$	16 T8.20
18.	$\emptyset \neq \emptyset \vee a = \overline{\mathsf{0}} \vee \ldots \vee a = \overline{\mathsf{n}-1} \vee a = \overline{\mathsf{Sn}-1}$	17 $\vee$I
19.	$a < S\overline{\mathsf{n}}$	18 T8.21
20.	$\exists u \exists v[\mathcal{B}(p, q, a, u) \wedge \mathcal{B}(p, q, Sa, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, a, u, v)]$	9,19 ($\forall$E)
21.	$(\forall i < \overline{\mathsf{n}})\exists u \exists v[\mathcal{B}(p, q, i, u) \wedge \mathcal{B}(p, q, Si, v) \wedge \mathcal{H}(\overline{\mathsf{m}}, i, u, v)]$	16-20 ($\forall$I)
22.	$\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u)$	8,21,15 with $\exists$I
23.	$\overline{\mathsf{k}}_{\mathsf{n}} = u$	1,22 with $\forall$E
24.	$\mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u, v)$	14 $\wedge$E
25.	$\mathcal{H}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{n}}, v)$	24,23 =E
26.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = v$	2,25 with $\forall$E
27.	$\mathcal{B}(p, q, S\overline{\mathsf{n}}, v)$	14 $\wedge$E
28.	$\mathcal{B}(p, q, S\overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{Sn}})$	27,26 =E
29.	$\mathcal{B}(p, q, S\overline{\mathsf{n}}, \overline{\mathsf{k}}_{\mathsf{Sn}}) \wedge \mathcal{B}(p, q, S\overline{\mathsf{n}}, j)$	28,10 $\wedge$I
30.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = j$	3,29 with $\forall$E
31.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = j$	13,14-30 $\exists$E
32.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = j$	12,13-31 $\exists$E
33.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = j$	6,7-32 $\exists$E
34.	$\overline{\mathsf{k}}_{\mathsf{Sn}} = j$	5,6-33 $\exists$E
35.	$\mathcal{F}(\overline{\mathsf{m}}, S\overline{\mathsf{n}}, j) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = j$	4-34 $\to$I
36.	$\forall w[\mathcal{F}(\overline{\mathsf{m}}, S\overline{\mathsf{n}}, w) \to \overline{\mathsf{k}}_{\mathsf{Sn}} = w]$	35 $\forall$I

Once we show $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, u)$ at (22), the inductive assumption lets us "pin" u onto $\overline{\mathsf{k}}_{\mathsf{n}}$. Then uniqueness conditions for $\mathcal{H}$ and $\mathcal{B}$ allow us to move to unique outputs for $\mathcal{H}$ and $\mathcal{B}$ and so for $\mathcal{F}$.

12.3.3 The Result Strengthened

T12.9 shows that the recursive functions are captured in $\mathrm{Q_s}$ by their Σ_1 original expressers. However, as we have suggested, the argument is easily strengthened to show that the recursive functions are captured in Q. To do so, we give up capture by the original expressers, though we retain the result that the recursive functions are captured by Σ_1 formulas.

In the previous section, we appealed to uniqueness of remainder for the β-function. In $\mathrm{Q_s}$, the original formula $\mathscr{B}$ captures the β-function, and has the strengthened uniqueness result important for T12.10. But we can simulate these effects in Q by modifying the formula $\mathscr{B}$. Then there are results for capture, uniqueness, and expression. For the first, recall that $\mathscr{B}$ is Δ_0 and expresses the (total) β-function.

T12.11. If a total function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by a Δ_0 formula $\mathscr{F}(\vec{x}, v)$, then $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow v = z]$ is Δ_0 and captures f in Q.

Suppose a total $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by a Δ_0 formula $\mathscr{F}(\vec{x}, v)$. Since $\mathscr{F}(\vec{x}, v)$ is Δ_0, $\mathscr{F}'(\vec{x}, v)$ is Δ_0. Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f}$. As preliminaries: (a) By expression $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; and since $\mathscr{F}$ is Δ_0, by T12.7, $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$. (b) Consider $\mathsf{n} \neq \mathsf{a}$; then $\langle \mathsf{m}, \mathsf{n}\rangle \notin \mathsf{f}$; so by expression $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})] \neq \mathsf{T}$; so by T12.7, $\mathrm{Q} \vdash \sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

(i) From (a), $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$. Now suppose $\mathsf{p} \leq \mathsf{a}$; then $\mathsf{p} = \mathsf{a}$ or $\mathsf{p} < \mathsf{a}$; in the first case, $\vdash \overline{\mathsf{a}} = \overline{\mathsf{p}}$, so $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{p}}$; and for $\mathsf{p} < \mathsf{a}$, from (b), $\mathrm{Q} \vdash \sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{p}})$; so $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{p}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{p}}$; so for any $\mathsf{p} \leq \mathsf{a}$, $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{p}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{p}}$; so by T8.25, $\mathrm{Q} \vdash (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow \overline{\mathsf{a}} = z)$. So with $\wedge$I, $\mathrm{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \wedge (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow \overline{\mathsf{a}} = z)$; which is to say, $\mathrm{Q} \vdash \mathscr{F}'(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Hint: You need to show,

$$\mathrm{Q} \vdash \forall w([\mathscr{F}(\overline{\mathsf{m}}, w) \wedge (\forall z \leq w)(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow w = z)] \rightarrow \overline{\mathsf{a}} = w)$$

Take as premises $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \wedge (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow \overline{\mathsf{a}} = z)$ from (i), along with $\forall x(x \leq \overline{\mathsf{a}} \vee \overline{\mathsf{a}} \leq x)$ from T8.23.

So $\mathscr{F}'$ captures f. And so, since our original formula $\mathscr{B}$ is Δ_0 and expresses the β-function, $\mathscr{B}'$ captures β in Q.

Intuitively, the second conjunct of $\mathscr{F}'$ requires that whenever z is less than v, $\mathscr{F}(\vec{x}, z)$ is unsatisfied, and so that $\mathscr{F}'$ is satisfied by the *least* v that satisfies $\mathscr{F}$. Then at most one v satisfies $\mathscr{F}'$ and it is not surprising that formulas of the sort $\mathscr{F}'$ yield a uniqueness result.

T12.12. For $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow v = z]$ and for any a, $\mathrm{Q} \vdash \forall\vec{x}\forall y[(\mathscr{F}'(\vec{x}, \overline{\mathsf{a}}) \wedge \mathscr{F}'(\vec{x}, y)) \rightarrow \overline{\mathsf{a}} = y]$.

Supposing $\vec{x}$ reduces to a single variable reason as in the box on the next page.

Finally, insofar as $\mathscr{F}'(\vec{x}, v)$ is built on an $\mathscr{F}(\vec{x}, v)$ that expresses $\mathsf{f}(\vec{\mathsf{x}})$, $\mathscr{F}'(\vec{x}, v)$ continues to express $\mathsf{f}(\vec{\mathsf{x}})$. Perhaps this is obvious given what $\mathscr{F}'$ says. However, we can argue for the result directly.

T12.13. If $\mathscr{F}(\vec{x}, v)$ expresses a total $\mathsf{f}(\vec{\mathsf{x}})$, then $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow v = z]$ expresses $\mathsf{f}(\vec{\mathsf{x}})$.

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and total $\mathsf{f}(\mathsf{x})$ is expressed by $\mathscr{F}(x, v)$. Suppose $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f}$. (a) By expression, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$. (b) Suppose $\mathsf{n} \neq \mathsf{a}$; then $\langle \mathsf{m}, \mathsf{n} \rangle \notin \mathsf{f}$; so with T12.2, $\mathsf{N}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})] = \mathsf{T}$.

(i) Suppose $\mathsf{N}[\mathscr{F}'(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{T}$. By [homework] this is impossible. You will need applications of T12.4 and T10.2; observe that for $\mathsf{n} \leq \mathsf{a}$ either $\mathsf{n} = \mathsf{a}$ or $\mathsf{n} < \mathsf{a}$ (so that $\mathsf{n} \neq \mathsf{a}$).

(ii) Suppose $\mathsf{N}[\forall w(\mathscr{F}'(\overline{\mathsf{m}}, w) \rightarrow \overline{\mathsf{a}} = w)] \neq \mathsf{T}$—that is that $\mathsf{N}[\forall w([\mathscr{F}(\overline{\mathsf{m}}, w) \wedge (\forall z \leq w)(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow w = z)] \rightarrow \overline{\mathsf{a}} = w)] \neq \mathsf{T}$. By [homework] this is impossible.

From T12.13, $\mathscr{B}'$ expresses the β-function. From T12.11, $\mathscr{B}'$ captures the β-function in Q. And from T12.12, $\mathsf{Q} \vdash \forall y[(\mathscr{B}'(p, q, i, \overline{\mathsf{a}}) \wedge \mathscr{B}'(p, q, i, y)) \rightarrow \overline{\mathsf{a}} = y]$. This is what we had before except applied to $\mathscr{B}'$ and Q, rather than to $\mathscr{B}$ and $\mathsf{Q_s}$. And

T12.12

1.	$\forall x(x \leq \overline{\mathsf{a}} \vee \overline{\mathsf{a}} \leq x)$	T8.23
2.	$\mathscr{F}'(j, \overline{\mathsf{a}}) \wedge \mathscr{F}'(j, k)$	A (g, →I)
3.	$\mathscr{F}(j, \overline{\mathsf{a}}) \wedge (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(j, z) \rightarrow \overline{\mathsf{a}} = z)$	2 ∧E ($\mathscr{F}'(j, \overline{\mathsf{a}})$)
4.	$\mathscr{F}(j, k) \wedge (\forall z \leq k)(\mathscr{F}(j, z) \rightarrow k = z)$	2 ∧E ($\mathscr{F}'(j, k)$)
5.	$k \leq \overline{\mathsf{a}} \vee \overline{\mathsf{a}} \leq k$	1 ∀E
6.	$k \leq \overline{\mathsf{a}}$	A (g, 5∨E)
7.	$(\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(j, z) \rightarrow \overline{\mathsf{a}} = z)$	3 ∧E
8.	$\mathscr{F}(j, k) \rightarrow \overline{\mathsf{a}} = k$	7,6 (∀E)
9.	$\mathscr{F}(j, k)$	4 ∧E
10.	$\overline{\mathsf{a}} = k$	8,9 →E
11.	$\overline{\mathsf{a}} \leq k$	A (g, 5∨E)
	$\vdots$	
12.	$\overline{\mathsf{a}} = k$	similarly
13.	$\overline{\mathsf{a}} = k$	5,6-10,11-12 ∨E
14.	$(\mathscr{F}'(j, \overline{\mathsf{a}}) \wedge \mathscr{F}'(j, k)) \rightarrow \overline{\mathsf{a}} = k$	2-13 →I
15.	$\forall y[(\mathscr{F}'(j, \overline{\mathsf{a}}) \wedge \mathscr{F}'(j, y)) \rightarrow \overline{\mathsf{a}} = y]$	14 ∀I
16.	$\forall x \forall y[(\mathscr{F}'(x, \overline{\mathsf{a}}) \wedge \mathscr{F}'(x, y)) \rightarrow \overline{\mathsf{a}} = y]$	15 ∀I

Reasoning for the second subderivation is similar to the first.

now we are in a position to recover the main result, except that the recursive functions are captured in Q rather than Q_s.

*T12.14. Any recursive function is captured by a Σ_1 formula in Q.

The β-function is total and expressed by a Δ_0 formula $\mathscr{B}(p, q, i, v)$; so by T12.13 and T12.11 there is a Δ_0 formula $\mathscr{B}'(p, q, i, v)$ that expresses and captures it in Q. For any $\mathsf{f}(\vec{\mathsf{x}})$ originally expressed by $\mathscr{F}(\vec{x}, v)$, let $\mathscr{F}^{\dagger}$ be like $\mathscr{F}$ except that instances of $\mathscr{B}$ are replaced by $\mathscr{B}'$. Since $\mathscr{B}'$ is Δ_0, $\mathscr{F}^{\dagger}$ remains Σ_1.

The argument is now a matter of showing that demonstrations of T12.5, T12.9, and T12.10 go through with application to these formulas and in Q. But the argument is nearly trivial: everything is the same as before with formulas of the sort $\mathscr{F}^{\dagger}$ replacing $\mathscr{F}$ and T12.12 for uniqueness of remainder.

Be clear that expressions of the sort $\mathscr{F}^{\dagger}$ might appear all along in the show part of T12.5, T12.9, and T12.10. Expressions from the basis do not involve $\mathscr{B}'$. It is included by recursion; after that, composition and regular minimization might be applied to expressions of any sort, and so to ones which involve $\mathscr{B}'$ as well.

As before, formulas other than $\mathscr{F}^{\dagger}(\vec{x}, v)$ might capture the recursive functions—for example, if $\mathscr{F}^{\dagger}(\vec{x}, v)$ captures $\mathsf{f}(\vec{\mathsf{x}})$, then so does $\mathscr{F}^{\dagger}(\vec{x}, v) \wedge \mathscr{A}$ for any theorem $\mathscr{A}$. Let us say that $\mathscr{F}^{\dagger}(\vec{x}, v)$ is the *canonical* formula that captures $\mathsf{f}(\vec{\mathsf{x}})$ in Q. Of course, the canonical formula which captures $\mathsf{f}(\vec{\mathsf{x}})$ need not be the same as the corresponding original formula. Because the β-function is captured by a Δ_0 formula we do, however, retain the result that every recursive function is captured in Q by some Σ_1 formula. For the rest of this chapter, unless otherwise noted, when we assert the existence of a formula to express or capture some recursive function, we shall have in mind the *canonical* formula.

E12.18. Provide an argument to demonstrate (ii) of T12.11.

E12.19. Complete the demonstration of T12.13.

*E12.20. Work carefully through the demonstration of T12.14 by setting up revised arguments T12.$5^{\dagger}$, T12.$9^{\dagger}$, and T12.$10^{\dagger}$. As feasible, you may simply explain how parts differ from the originals.

12.4 More Recursive Functions

Now that we have seen what the recursive functions are, and the powers of our logical systems to express and capture recursive functions, we turn to extending their range. In fact, in this section, we shall generate a series of functions that are *primitive* recursive.

So far, in addition to the initial functions, we have seen that plus, times, fact, and power are primitive recursive. As we increase the range of (primitive) recursive functions, it immediately follows that our logical systems have the power to express and capture all the same functions. And, again, these powers are at the base of our demonstrations of ĩncompleteness to come.

12.4.1 Preliminary Functions

We begin with some simple primitive recursive functions that will serve as a foundation for things to follow.

Predecessor with cutoff. Set the predecessor of zero to zero itself, and for any other value to the one before. Since $\mathsf{pred(y)}$ is a one-place function, $\mathsf{gpred()}$ is a 0-place function, in this case, $\mathsf{gpred()} = \widehat{0}$. And $\mathsf{hpred(y, u)} = \mathsf{idnt}^2_1\mathsf{(y, u)}$. So, as we expect for $\mathsf{pred(y)}$,

$$\begin{aligned}&\mathsf{pred(0)} = 0\\&\mathsf{pred(suc(y))} = \mathsf{y}\end{aligned}$$

So predecessor is a primitive recursive function.

Subtraction with cutoff. When $\mathsf{x} < \mathsf{y}$, $\mathsf{subc(x, y)} = 0$. Otherwise $\mathsf{subc(x, y)} = \mathsf{x} - \mathsf{y}$. For $\mathsf{subc(x, y)}$, set $\mathsf{gsubc(x)} = \mathsf{idnt}^1_1\mathsf{(x)}$. And $\mathsf{hsubc(x, y, u)} = \mathsf{pred(idnt}^3_3\mathsf{(x, y, u))}$. So,

$$\begin{aligned}&\mathsf{subc(x, 0)} = \mathsf{x}\\&\mathsf{subc(x, suc(y))} = \mathsf{pred(subc(x, y))}\end{aligned}$$

So as y increases by one, the difference decreases by one. Informally, indicate $\mathsf{subc(x, y)} = (\mathsf{x} \dot{-} \mathsf{y})$.

Absolute value. $\mathsf{absval(x - y)} = (\mathsf{x} \dot{-} \mathsf{y}) + (\mathsf{y} \dot{-} \mathsf{x})$. We find the absolute value of the difference between x and y by doing the subtraction with cutoff both ways. One direction yields zero. The other yields the value we want. So the sum comes out to the absolute value. This is a function with two input values (only separated by '-' rather than comma to remind us of the nature of the function). This function results entirely by composition, without a recursion clause. Informally, we indicate absolute value in the usual way, $\mathsf{absval(x - y)} = |\mathsf{x - y}|$.

Sign. The function $\mathsf{sg(y)}$ is zero when y is zero and otherwise one. For $\mathsf{sg(y)}$, set $\mathsf{gsg()} = \widehat{0}$. And $\mathsf{hsg(y, u)} = \mathsf{suc(zero(y, u))}$. So,

$$\begin{aligned}&\mathsf{sg(0)} = 0\\&\mathsf{sg(suc(y))} = \mathsf{suc(0)} = 1\end{aligned}$$

So the sign of any successor is just the successor of zero, which is one.

Converse sign. The function $\mathsf{csg}(\mathsf{y})$ is one when y is zero and otherwise zero. So it inverts sg. For $\mathsf{csg}(\mathsf{y})$, set $\mathsf{gcsg}() = \hat{1}$. And $\mathsf{hcsg}(\mathsf{y}, \mathsf{u}) = \mathsf{zero}(\mathsf{y}, \mathsf{u})$. So,

$$\mathsf{csg}(0) = 1$$
$$\mathsf{csg}(\mathsf{suc}(\mathsf{y})) = 0$$

So the converse sign of any successor is just zero. Informally, we indicate the converse sign with a bar, $\overline{\mathsf{sg}}(\mathsf{y})$.

E12.21. Consider again your file `recursive1.rb` from E12.3. Extend your sequence of functions to include `pred(x)`, `subc(x,y)`, `absval(x,y)`, `sg(x)`, and `csg(x)`. Calculate some values of these functions and print the results, along with your program. Again, there should be no appeal to functions except from earlier in the chain.

12.4.2 Characteristic Functions

We shall be able to extend our results for the expression and capture of recursive functions to the expression and capture of (recursive) relations by the notion of a *characteristic function*. The characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$ of a relation R takes the value 0 when $\vec{\mathsf{x}} \in \mathsf{R}$ and 1 when $\vec{\mathsf{x}} \notin \mathsf{R}$.

CF For any function $\mathsf{p}(\vec{\mathsf{x}})$, $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}}))$ is the *characteristic* function of that relation R such that $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}})) = 0$ iff $\vec{\mathsf{x}} \in \mathsf{R}$.

So the characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$ for relation R takes the value 0 if $\mathsf{R}(\vec{\mathsf{x}})$ is true, and 1 if $\mathsf{R}(\vec{\mathsf{x}})$ is not true.[13] A (*primitive*) *recursive* property or relation is one that has a (primitive) recursive characteristic function. If the outputs of the function p are already just 0 and 1 so that $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}})) = \mathsf{p}(\vec{\mathsf{x}})$, we generally omit sg from our specifications.

These definitions immediately result in corollaries to T12.5 and T12.14. Recall that expression (capture) for a relation requires that if $\mathsf{m} \in \mathsf{R}$, then $\mathcal{R}(\overline{\mathsf{m}})$ is true (proved), and if $\mathsf{m} \notin \mathsf{R}$, then $\sim\mathcal{R}(\overline{\mathsf{m}})$ is true (proved).

T12.5 (corollary). On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ is expressed by some formula $\mathcal{R}(\vec{x})$.

Suppose $\mathsf{R}(\vec{\mathsf{x}})$ is a recursive relation; then it has a recursive and so total characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$; so by T12.5 there is some formula $\mathcal{R}(\vec{x}, y)$ that expresses $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$. So in the case where $\vec{x}$ reduces to a single variable, if $\mathsf{m} \in \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \in \mathsf{ch}_{\mathsf{R}}$; and by expression, $\mathsf{I}[\mathcal{R}(\overline{\mathsf{m}}, \overline{0})] = \mathsf{T}$; and if $\mathsf{m} \notin \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \notin \mathsf{ch}_{\mathsf{R}}$; and since the function is total, by T12.2, $\mathsf{I}[\sim\mathcal{R}(\overline{\mathsf{m}}, \overline{0})] = \mathsf{T}$. So, generally, $\mathcal{R}(\vec{x}, \overline{0})$ expresses $\mathsf{R}(\vec{\mathsf{x}})$.

[13] It is perhaps more common to reverse the values of zero and one for the characteristic function. However, the choice is arbitrary, and this choice is technically convenient.

T12.14 (corollary). Any recursive relation is captured by a Σ_1 formula in Q.

> Suppose $\mathsf{R}(\vec{\mathsf{x}})$ is a recursive relation; then it has a recursive and so total characteristic function $\mathsf{ch_R}(\vec{\mathsf{x}})$; so by T12.14 there is some Σ_1 formula $\mathscr{R}(\vec{x}, y)$ that captures $\mathsf{ch_R}(\vec{\mathsf{x}})$ in Q. So in the case where $\vec{x}$ reduces to a single variable, if $\mathsf{m} \in \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \in \mathsf{ch_R}$; and by capture $\mathrm{Q} \vdash \mathscr{R}(\overline{\mathsf{m}}, \overline{0})$; and if $\mathsf{m} \notin \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \notin \mathsf{ch_R}$; and since the function is total, by capture with T12.6, $\mathrm{Q} \vdash \sim\mathscr{R}(\overline{\mathsf{m}}, \overline{0})$. So, generally $\mathscr{R}(\vec{x}, \overline{0})$ captures $\mathsf{R}(\vec{\mathsf{x}})$ in Q.

So our results for the expression and capture of recursive functions extend directly to the expression and capture of recursive relations: A recursive relation has a recursive characteristic function; as such, the function is expressed and captured; so, as we have just seen, the corresponding relation is expressed and captured.

Equality. $\mathsf{EQ}(\mathsf{x}, \mathsf{y})$, typically rendered $\mathsf{x} = \mathsf{y}$, is a recursive relation with characteristic function $\mathsf{ch_{EQ}}(\mathsf{x}, \mathsf{y}) = \mathsf{sg}|\mathsf{x} - \mathsf{y}|$. When x is equal to y, the absolute value of the difference is zero so the value of sg is zero. But when x is other than y, the absolute value of the difference is other than zero, so the value of sg is one. And, generally, if functions $\mathsf{s}(\vec{\mathsf{x}})$ and $\mathsf{t}(\vec{\mathsf{y}})$ are recursive, by composition $\mathsf{sg}|\mathsf{s}(\vec{\mathsf{x}}) - \mathsf{t}(\vec{\mathsf{y}})|$ is a recursive function and so $\mathsf{s}(\vec{\mathsf{x}}) = \mathsf{t}(\vec{\mathsf{y}})$ a recursive relation.

A couple of observations: First, be clear that EQ is the standard relation we all know and love. The trick is to show that it is recursive. We are not *given* that EQ is a recursive relation—so we demonstrate that it is, by showing that it has a recursive characteristic function. Second, one might think that we could express $\mathsf{f}(\vec{\mathsf{x}}) = \mathsf{g}(\vec{\mathsf{y}})$ by some relatively simple expression that would compose expressions for the functions with equality as, $\exists u \exists v[\mathscr{F}(\vec{x}, u) \wedge \mathscr{G}(\vec{y}, v) \wedge u = v]$. This would be fine. However we have offered a general account which, as is often the case for these things, need not be the most efficient. Where $\mathsf{sg}|\mathsf{f}(\vec{\mathsf{x}}) - \mathsf{g}(\vec{\mathsf{y}})|$ is expressed and captured by some $Sgabs(\vec{x}, \vec{y}, v)$ our approach, which works by modification of the characteristic function, generates the relatively complex, $Eq(\vec{x}, \vec{y}) = Sgabs(\vec{x}, \vec{y}, \emptyset)$.

Inequality. The relation $\mathsf{LEQ}(\mathsf{x}, \mathsf{y})$ has characteristic function $\mathsf{sg}(\mathsf{x} \dot{-} \mathsf{y})$. When $\mathsf{x} \leq \mathsf{y}$, $\mathsf{x} \dot{-} \mathsf{y} = 0$; so $\mathsf{sg} = 0$; Otherwise the value is 1. The relation $\mathsf{LESS}(\mathsf{x}, \mathsf{y})$ has characteristic function $\mathsf{sg}(\mathsf{suc}(\mathsf{x}) \dot{-} \mathsf{y})$. When $\mathsf{x} < \mathsf{y}$, $\mathsf{suc}(\mathsf{x}) \dot{-} \mathsf{y} = 0$; so $\mathsf{sg} = 0$. Otherwise the value is 1. These are typically represented $\mathsf{x} \leq \mathsf{y}$ and $\mathsf{x} < \mathsf{y}$.

With equality and inequality, we have atomic recursive relations. And we set out to exhibit ones that are more complex in the usual way.

Truth functions. Suppose $\mathsf{P}(\vec{\mathsf{x}})$ and $\mathsf{Q}(\vec{\mathsf{y}})$ are recursive relations. Then $\mathsf{NEG}(\mathsf{P}(\vec{\mathsf{x}}))$ and $\mathsf{DSJ}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{y}}))$ are recursive relations. Suppose $\mathsf{ch_P}(\vec{\mathsf{x}})$ and $\mathsf{ch_Q}(\vec{\mathsf{y}})$ are the characteristic functions of $\mathsf{P}(\vec{\mathsf{x}})$ and $\mathsf{Q}(\vec{\mathsf{y}})$.

First Results of Chapter 12

T12.1 For an interpretation on which members of the universe are assigned to the required variable-free terms: (a) If R is a relation, and $\mathsf{I}[\mathscr{R}] = \mathsf{R}(\mathsf{x}_1 \dots \mathsf{x}_n)$, then $\mathsf{R}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is expressed by $\mathscr{R}x_1 \dots x_n$. And (b) if h is a function and $\mathsf{I}[\hbar] = \mathsf{h}(\mathsf{x}_1 \dots \mathsf{x}_n)$ then $\mathsf{h}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is expressed by $\hbar x_1 \dots x_n = v$.

T12.2 If total function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is expressed by formula $\mathscr{F}(x_1 \dots x_n, y)$, then if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$.

T12.3 On the standard interpretation N for $\mathscr{L}_{NT}$, (i) $\mathsf{N_d}[\mathscr{s} \le \mathsf{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathscr{s}] \le \mathsf{N_d}[\mathscr{t}]$, and (ii) $\mathsf{N_d}[\mathscr{s} < \mathsf{t}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathscr{s}] < \mathsf{N_d}[\mathscr{t}]$.

T12.4 On the standard interpretation N for $\mathscr{L}_{NT}$,

(a) $\mathsf{N_d}[(\forall x \le t)\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{o} \le \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathscr{P}] = \mathsf{S}$. And $\mathsf{N_d}[(\forall x < t)\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{o} < \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathscr{P}] = \mathsf{S}$.

(b) $\mathsf{N_d}[(\exists x \le t)\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{o} \le \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathscr{P}] = \mathsf{S}$. And $\mathsf{N_d}[(\exists x < t)\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{o} < \mathsf{N_d}[t]$, $\mathsf{N_{d(x|o)}}[\mathscr{P}] = \mathsf{S}$.

T12.5 On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$.

Corollary: On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{R}(\vec{x})$.

T12.6 If T includes Q and total function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is captured by formula $\mathscr{F}(x_1 \dots x_n, y)$, then if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, $T \vdash \sim\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

T12.7 For any Δ_0 sentence $\mathscr{P}$, if $\mathsf{N}[\mathscr{P}] = \mathsf{T}$ then $\mathrm{Q} \vdash \mathscr{P}$, and if $\mathsf{N}[\mathscr{P}] \ne \mathsf{T}$ then $\mathrm{Q} \vdash \sim\mathscr{P}$.

T12.8 For any Σ_1 sentence $\mathscr{P}$ if $\mathsf{N}[\mathscr{P}] = \mathsf{T}$, then $\mathrm{Q} \vdash \mathscr{P}$.

T12.9 On the standard interpretation N for $\mathscr{L}_{NT}$, any recursive function is captured in $\mathrm{Q_s}$ by the original formula by which it is expressed.

T12.10 Suppose $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ results by recursion from functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ where $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathscr{G}(\vec{x}, z)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\mathscr{H}(\vec{x}, y, u, z)$. Then for the original expression $\mathscr{F}(\vec{x}, y, z)$, if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_b, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$, $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_b, \overline{\mathsf{n}}, w) \to \overline{\mathsf{a}} = w]$.

T12.11 If a total function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by a Δ_0 formula $\mathscr{F}(\vec{x}, v)$, then $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \le v)[\mathscr{F}(\vec{x}, z) \to v = z]$ is Δ_0 and captures f in Q.

T12.12 For $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \le v)[\mathscr{F}(\vec{x}, z) \to v = z]$ and for any n, $\mathrm{Q} \vdash \forall\vec{x}\forall y[(\mathscr{F}'(\vec{x}, \overline{\mathsf{n}}) \wedge \mathscr{F}'(\vec{x}, y)) \to \overline{\mathsf{n}} = y]$.

T12.13 If $\mathscr{F}(\vec{x}, v)$ expresses a total $\mathsf{f}(\vec{\mathsf{x}})$, then $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \le v)[\mathscr{F}(\vec{x}, z) \to v = z]$ expresses $\mathsf{f}(\vec{\mathsf{x}})$.

T12.14 Any recursive function is captured by a Σ_1 formula in Q.

Corollary: Any recursive relation is captured by a Σ_1 formula in Q.

$\text{NEG}(\text{P}(\vec{x}))$ (typically $\sim\text{P}(\vec{x})$) has characteristic function $\overline{\text{sg}}(\text{ch}_{\text{P}}(\vec{x}))$. When $\text{P}(\vec{x})$ does not obtain, the characteristic function of $\text{P}(\vec{x})$ takes value one, so the converse sign goes to zero. And when $\text{P}(\vec{x})$ does obtain, its characteristic function is zero, so the converse sign is one—which is as it should be.

$\text{DSJ}(\text{P}(\vec{x}), \text{Q}(\vec{y}))$ (typically $\text{P}(\vec{x}) \vee \text{Q}(\vec{y})$) has characteristic function $\text{ch}_{\text{P}}(\vec{x}) \times \text{ch}_{\text{Q}}(\vec{y})$. When one of $\text{P}(\vec{x})$ or $\text{Q}(\vec{y})$ is true, the disjunction is true; but in this case, at least one characteristic function, and so the product of functions, equals zero. If neither $\text{P}(\vec{x})$ nor $\text{Q}(\vec{y})$ is true, the disjunction is not true; in this case, both characteristic functions, and so the product of functions, take the value one.

Other truth functions are definable in terms of negation and disjunction. So, for example, $\text{IMP}(\text{P}(\vec{x}), \text{Q}(\vec{y}))$ that is, $\text{P}(\vec{x}) \rightarrow \text{Q}(\vec{y})$ is $\sim\text{P}(\vec{x}) \vee \text{Q}(\vec{y})$; and $\text{CNJ}(\text{P}(\vec{x}), \text{Q}(\vec{y}))$ that is, $\text{P}(\vec{x}) \wedge \text{Q}(\vec{y})$ is $\sim(\sim\text{P}(\vec{x}) \vee \sim\text{Q}(\vec{y}))$.

Bounded quantifiers. Consider a relation $(\exists y \leq z)\text{P}(\vec{x}, y)$ which obtains when there is a y less than or equal to z such that $\text{P}(\vec{x}, y)$. As usual, y is distinct from the bound z. Let v be a variable not in $\vec{x}$ and not y (and so other than z if z is in $\vec{x}$). Consider a function $\text{eleq}(\vec{x}, v)$—intuitively this will be the characteristic function of $(\exists y \leq v)\text{P}(\vec{x}, y)$. So $\text{eleq}(\vec{x}, v)$ lets the bound vary independently of the variables in P, and we shall be able to define the function by recursion as v ranges from 0 to z. For $\text{eleq}(\vec{x}, v)$ set,

$$\text{geleq}(\vec{x}) = \text{ch}_{\text{P}}(\vec{x}, \hat{0})$$
$$\text{heleq}(\vec{x}, v, u) = u \times \text{ch}_{\text{P}}(\vec{x}, Sv)$$

In the simple case where $\vec{x}$ drops out, $\text{eleq}(0) = \text{ch}_{\text{P}}(\hat{0})$. And $\text{eleq}(Sv) = \text{eleq}(v) \times \text{ch}_{\text{P}}(Sv)$. So,

$$\text{eleq}(v) = \text{ch}_{\text{P}}(0) \times \text{ch}_{\text{P}}(1) \times \cdots \times \text{ch}_{\text{P}}(v)$$

Think of these as grouped to the left. So the result has $\text{eleq}(n) = 1$ unless and until one of the members is zero, and then stays zero. Thus $\text{eleq}(n)$ goes to zero just in case $\text{P}(v)$ is true for some value between 0 and n. So set the characteristic function of the bounded quantification $(\exists y \leq z)\text{P}(\vec{x}, y)$ to $\text{eleq}(\vec{x}, z)$—the characteristic function for the bounded quantifier runs the eleq function up to the bound z.

For $(\forall y \leq z)\text{P}(\vec{x}, y)$ it is simplest just to take $\sim(\exists y \leq z)\sim\text{P}(\vec{x}, y)$. Similarly, $(\exists y < z)\text{P}(\vec{x}, y)$ is $(\exists y \leq z)(y \neq z \wedge \text{P}(\vec{x}, y))$; and for $(\forall y < z)\text{P}(\vec{x}, y)$ we can take $\sim(\exists y < z)\sim\text{P}(\vec{x}, y)$. So we are done by previous results.

Bounded minimization. As we have seen, $f(\vec{x}) = \mu y[g(\vec{x}, y) = \hat{0}]$ defined by regular minimization returns the least y such that $g(\vec{x}, y) = 0$. Observe that the minimization operation is applied to a recursive *relation* in the square brackets—and that finding the least y such that $g(\vec{x}, y) = 0$ is finding the least y such that the relation obtains. But for an arbitrary recursive $\text{P}(\vec{x}, y)$, $\text{P}(\vec{x}, y)$ iff $\text{ch}_{\text{P}}(\vec{x}, y) = \hat{0}$. So we often encounter $f(\vec{x}) = \mu y[\text{ch}_{\text{P}}(\vec{x}, y) = \hat{0}]$ in the equivalent form, $f(\vec{x}) = \mu y[\text{P}(\vec{x}, y)]$. Of course, for regular minimization, it remains that $\mathsf{ch_P}(\vec{x}, y)$ has to be regular—so that for any $\vec{x}$, there is some y for which $\mathsf{P}(\vec{x}, y)$ obtains.

But we can bypass the regularity requirement by a primitive recursive *bounded* minimization. For this, let $(\mu y \leq z)\mathsf{P}(\vec{x}, y)$ be the least $y \leq z$ such that $\mathsf{P}(\vec{x}, y)$ if one exists, and otherwise z. If $\mathsf{P}(\vec{x}, y)$ is a recursive relation, $(\mu y \leq z)\mathsf{P}(\vec{x}, y)$ is a recursive function. Again, let v be a variable not in $\vec{x}$ and not y. First, take $\mathsf{eleq}(\vec{x}, v)$ as in the bounded quantifier case; so $\mathsf{eleq}(\vec{x}, v)$ goes to 0 when P is true for some $j \leq v$. Then, second, introduce a function $\mathsf{mleq}(\vec{x}, v)$—intuitively this is to be $(\mu y \leq v)\mathsf{P}(\vec{x}, y)$. So $\mathsf{mleq}(\vec{x}, v)$ lets the bound vary independently of the variables in P—and the desired minimization results when the bound reaches z. For $\mathsf{mleq}(\vec{x}, v)$ set,

$$\mathsf{gmleq}(\vec{x}) = \mathsf{zero}(\vec{x})$$
$$\mathsf{hmleq}(\vec{x}, v, u) = u + \mathsf{eleq}(\vec{x}, v)$$

So in the simple case where $\vec{x}$ drops out gmleq becomes a zero-place function and $\mathsf{mleq}(0) = \mathsf{zero}() = 0$—for the least $y \leq 0$ that satisfies any $\mathsf{P}(y)$ can only be 0. And then $\mathsf{mleq}(Sv) = \mathsf{mleq}(v) + \mathsf{eleq}(v)$. The result is,

$$\mathsf{mleq}(Sn) = 0 + \mathsf{eleq}(0) + \cdots + \mathsf{eleq}(n)$$

where eleq is 1 until it hits a member that is P and then goes to 0 and stays there. Set the first member to the side. Then since this series starts with $v = 0$ and ends with $v = n$ it has Sn members. So if all the values are 1 it evaluates to Sn. If there is some first a such that $\mathsf{eleq}(a)$ is zero, then all the members prior to it are 1 and the sum is a. So take the sum up to the limit z and set $(\mu y \leq z)\mathsf{P}(\vec{x}, y) = \mathsf{mleq}(\vec{x}, z)$. Observe that $(\mu y \leq z)\mathsf{P}(\vec{x}, y) = z$ does not require that $\mathsf{P}(\vec{x}, z)$—only that no $a < z$ is such that $\mathsf{P}(\vec{x}, a)$.

Selection by cases. Selection by cases introduces a function which responds to different classes of objects in different ways. So, for some reason, we might require a function which squares even numbers and cubes odd so that, say, $f(2) = 4$ and $f(3) = 27$. For this, suppose $f_0(\vec{x}) \ldots f_k(\vec{x})$ are recursive functions and $\mathsf{C}_0(\vec{x}) \ldots \mathsf{C}_k(\vec{x})$ are mutually exclusive recursive relations. Then $f(\vec{x})$ defined as follows is recursive.

$$f(\vec{x}) = \begin{cases} f_0(\vec{x}) \text{ if } \mathsf{C}_0(\vec{x}) \\ f_1(\vec{x}) \text{ if } \mathsf{C}_1(\vec{x}) \\ \vdots \\ f_k(\vec{x}) \text{ if } \mathsf{C}_k(\vec{x}) \\ \text{and otherwise } \hat{a} \end{cases}$$

Observe that, $f(\vec{x}) =$

$$[\overline{sg}(ch_{\mathsf{C}_0}(\vec{x})) \times f_0(\vec{x})] + [\overline{sg}(ch_{\mathsf{C}_1}(\vec{x})) \times f_1(\vec{x})] + \cdots + [\overline{sg}(ch_{\mathsf{C}_k}(\vec{x})) \times f_k(\vec{x})] +$$
$$[ch_{\mathsf{C}_0}(\vec{x}) \times ch_{\mathsf{C}_1}(\vec{x}) \times \cdots \times ch_{\mathsf{C}_k}(\vec{x}) \times \hat{a}]$$

works as we want. Each term of the upper sum is 0 unless its c_i is met, in which case $\overline{sg}(ch_{c_i}(\vec{x}))$ is 1 and the term goes to $f_i(\vec{x})$—then, supposing the conditions are mutually exclusive, the upper sum itself is just $f_i(\vec{x})$. The lower product is 0 unless no condition c_i is met, in which case it goes to a. So $f(\vec{x})$ is a composition of recursive functions, and itself recursive.

We turn now to some applications that will be particularly useful for things to come. In many ways, the project is like a cool translation exercise—pitched at the level of functions and relations.

Factor. Let $\textsc{fctr}(m, n)$ be the relation that obtains between m and n when $m + 1$ evenly divides n (typically, $m \mid n$). Division is by $m+1$ to avoid worries about division by zero.[14] Then $m \mid n$ is recursive. This relation is defined as follows:

$$(\exists y \leq n)(Sm \times y = n)$$

Observe that this makes (the predecessor of) both 1 and n factors of n, and any number a factor of zero. Since each part is recursive, the whole is recursive. The argument is from the parts to the whole: $Sm \times y = n$ has a recursive characteristic function; so the bounded quantification has a recursive characteristic function; so the factor relation is recursive.

Prime number. Say $\textsc{prime}(n)$ is true just when n is a prime number. This property is defined as follows:

$$\hat{1} < n \wedge (\forall j < n)[j \mid n \rightarrow (Sj = \hat{1} \vee Sj = n)]$$

So n is greater than 1 and the successor of any number that divides it is either 1 or n itself.

Prime sequence. Say the primes are $p_0, p_1, \ldots$. Let the value of the function $pi(n)$ (usually $\pi(n)$) be p_n. Then $\pi(n)$ is defined by recursion as follows:

$$gpi() = \hat{2}$$
$$hpi(y, u) = idnt^2_2[y, (\mu z \leq S\,fact(u))(u < z \wedge \textsc{prime}(z))]$$

So the first prime, $\pi(0) = 2$. And $\pi(Sn) = (\mu z \leq S\,fact(\pi(n)))(\pi(n) < z \wedge \textsc{prime}(z))$. So at any stage, the next prime is the least prime which is greater than $\pi(n)$. This depends on the point that the next prime after $\pi(n)$ is less than or equal to $S\,fact(\pi(n))$. Let $p(n) = \pi(0) \times \pi(1) \times \cdots \times \pi(n)$. By a standard argument (see G2 in the arithmetic for Gödel numbering reference), $p(n) + 1$ is not divisible by any of the primes up to

[14]In fact, this is a (minor) complication at this stage. It is usual to let $m \mid n$ when m evenly divides n. Still, it will be helpful down the road to have excluded division by zero. Compare page 661 note 10, and comment (k) on page 670.

$\pi(\mathsf{n})$; so either $\mathsf{p(n)}+1$ is itself prime, or there is some prime greater than $\pi(\mathsf{n})$ but less than $\mathsf{p(n)}+1$. But since $\mathsf{fact}(\pi(\mathsf{n}))$ is a product including all the primes up to $\pi(\mathsf{n})$, $\mathsf{p(n)} \leq \mathsf{fact}(\pi(\mathsf{n}))$; so either $\mathsf{fact}(\pi(\mathsf{n})) \mathbf{+} \hat{1} = \mathsf{S\,fact}(\pi(\mathsf{n}))$ is prime or there is a prime greater than $\pi(\mathsf{n})$ but less than $\mathsf{S\,fact}(\pi(\mathsf{n}))$—and the next prime is sure to appear in the specified range.

Prime exponent. Let $\mathsf{exp(n,i)}$ be the (possibly 0) exponent of $\pi(\mathsf{i})$ in the unique prime factorization of n. Then $\mathsf{exp(n,i)}$ is recursive. This function may be defined as follows:

$$(\mu \mathsf{x} \leq \mathsf{n})[\mathsf{pred}(\pi(\mathsf{i})^{\mathsf{Sx}}) \nmid \mathsf{n}]$$

If Sx is the least power of $\pi(\mathsf{i})$ that does not divide into n, then x is the greatest power that does—and so the exponent. Observe that no exponent in the prime factorization of n is greater than n itself—for any $\mathsf{p} \geq 2$, $\mathsf{p}^{\mathsf{n}} > \mathsf{n}$—so p^{n} does not divide n, and the bound is safe.

Prime length. Say a prime $\pi(\mathsf{a})$ is *included* in the factorization of n just in case there is some $\mathsf{b} \geq \mathsf{a}$ and $\mathsf{e} > 0$ such that (the predecessor of) $\pi(\mathsf{b})^{\mathsf{e}}$ is a factor of n. So we think of a prime factorization as,

$$\pi(0)^{\mathsf{e}_0} \times \pi(1)^{\mathsf{e}_1} \times \cdots \times \pi(\mathsf{b})^{\mathsf{e}_\mathsf{b}}$$

where $\mathsf{e}_\mathsf{b} > 0$, but exponents for prior members of the series may be zero or not. Then $\mathsf{len(n)}$ is the number of primes included in the prime factorization of n; so $\mathsf{len}(0) = \mathsf{len}(1) = 0$ and otherwise, since the series of primes begins with $\pi(0)$, $\mathsf{len(n)} = \mathsf{b}+1$. For this set,

$$\mathsf{len(n)} = (\mu \mathsf{y} \leq \mathsf{n})(\forall \mathsf{z} : \mathsf{y} \leq \mathsf{z} \leq \mathsf{n})\mathsf{exp(n,z)} = \hat{0}$$

Officially: $(\mu \mathsf{y} \leq \mathsf{n})(\forall \mathsf{z} \leq \mathsf{n})[\mathsf{y} \leq \mathsf{z} \to \mathsf{exp(n,z)} = \hat{0}]$. So we find the least y such that none of the primes between $\pi(\mathsf{y})$ and $\pi(\mathsf{n})$ are part of the factorization of n; but then all of the primes prior to it are members of the factorization so that y numbers the length of the factorization. This depends on its being the case that $\mathsf{n} < \pi(\mathsf{n})$ so that primes greater than or equal $\pi(\mathsf{n})$ are never included in the factorization of n.

E12.22. Returning to your file `recursive1.rb` from E12.3 and E12.21, extend the sequence of functions to include the characteristic function for $\textsc{fctr}(\mathsf{m},\mathsf{n})$. You will need to begin with `cheq(a,b)` for the characteristic function of $\mathsf{a} = \mathsf{b}$ and then the characteristic function of $\mathsf{Sm} \times \mathsf{y} = \mathsf{n}$. Then you will require $\mathsf{eleqf(m,n,v)}$ that is the characteristic function of $(\exists \mathsf{y} \leq \mathsf{v})(\mathsf{Sm} \times \mathsf{y} = \mathsf{n})$, and finally `chfctr(m,n)`. Calculate some values of these functions and print the results, along with your program.

E12.23. Continue in your file `recursive1.rb` to build the characteristic function for $\text{PRIME}(n)$. You will have to build gradually to this result. You will need `chless(a,b)` and then `chneg(a)`, `chdsj(a,b)`, `chimp(a,b)`, and `chand(a,b)` for the relevant truth functions. With these in hand, you can build a function `chpm(n,j)` corresponding to $j \neq n \wedge \sim(j \mid n \rightarrow (Sj = \hat{1} \vee Sj = n))$ to which the bounded existential quantifier will apply; and with that, you can obtain $\mathsf{eleqp}(n, j, v)$ and so the characteristic function of the bounded existential. Then, finally, build `chprime(n)`. Calculate some values of these functions and print the results, along with your program.

E12.24. Continue in your file `recursive1.rb` to generate $\mathsf{lcm}(m, n)$ the least common multiple of Sm and Sn—that is, $(\mu y \leq Sm \times Sn)[\hat{0} < y \wedge m \mid y \wedge n \mid y]$. For this you will need the characteristic function of $\hat{0} < y \wedge m \mid y \wedge n \mid y$; and then $\mathsf{eleql}(m, n, v)$ corresponding to $(\exists y \leq v)[\hat{0} < y \wedge m \mid y \wedge n \mid y]$. Then you will be able to find the function $\mathsf{mleql}(m, n, v)$ corresponding to $(\mu y \leq v)[\hat{0} < y \wedge m \mid y \wedge n \mid y]$ and finally the lcm. Calculate some values of these functions and print the results, along with your program.

*E12.25. Provide definitions for the recursive functions $\mathsf{rm}(m, n)$ and $\mathsf{qt}(m, n)$ for the remainder and quotient of $m/n + 1$. Notice that this rm is a *total* function and so distinct from rem as described in section 12.2.3. Hint: You can use rm in your account of qt.

*E12.26. Functions $f_1(\vec{x}, y)$ and $f_2(\vec{x}, y)$ are defined by *simultaneous* (mutual) recursion just in case,

(a) $f_1(\vec{x}, 0) = g_1(\vec{x})$

(b) $f_2(\vec{x}, 0) = g_2(\vec{x})$

(c) $f_1(\vec{x}, Sy) = h_1(\vec{x}, y, f_1(\vec{x}, y), f_2(\vec{x}, y))$

(d) $f_2(\vec{x}, Sy) = h_2(\vec{x}, y, f_1(\vec{x}, y), f_2(\vec{x}, y))$

Show that f_1 and f_2 so defined are recursive. Hint: Let $F(\vec{x}, y) = \pi(0)^{f_1(\vec{x},y)} \times \pi(1)^{f_2(\vec{x},y)}$; then find $G(\vec{x})$ in terms of g_1 and g_2, and $H(\vec{x}, y, u)$ in terms of h_1 and h_2 so that $F(\vec{x}, 0) = G(\vec{x})$ and $F(\vec{x}, Sy) = H(\vec{x}, y, F(\vec{x}, y))$. So $F(\vec{x}, y)$ is recursive. Then $f_1(\vec{x}, y) = \exp(F(\vec{x}, y), \hat{0})$ and $f_2(\vec{x}, y) = \exp(F(\vec{x}, y), \hat{1})$; so f_1 and f_2 are recursive. You will need to show that this specification satisfies conditions (a)–(d).

12.4.3 Arithmetization

Our aim in this section and the next is to assign numbers to expressions and sequences of expressions in $\mathscr{L}_{NT}$ and build a (primitive) recursive relation $\text{PRFQ}(\mathsf{m}, \mathsf{n})$ which is true just in case m numbers a sequence of expressions that is a proof in Robinson Arithmetic of the expression numbered by n. This requires a number of steps. In this section, we develop at least the notion of a *sentential* proof which should be sufficient for the general idea. The next section continues with details for quantifiers and theory Q.

Gödel numbers. We begin with a strategy familiar from 10.2.2 and 10.3.2 (to which you may find it helpful to refer), now adapted to $\mathscr{L}_{NT}$. The idea is to assign numbers to symbols and expressions of $\mathscr{L}_{NT}$. Then we shall be able to operate on the associated numbers by means of ordinary numerical functions. Insofar as the variable symbols in any quantificational language are countable, they are capable of being sorted into series, $x_0, x_1, \ldots$. Supposing that this is done, begin by assigning to each symbol $\mathscr{s}$ in $\mathscr{L}_{NT}$ an integer $g[\mathscr{s}]$ called its *Gödel number*.

a.	$g[(] = 3$	f.	$g[\forall] = 13$
b.	$g[)] = 5$	g.	$g[\emptyset] = 15$
c.	$g[\sim] = 7$	h.	$g[S] = 17$
d.	$g[\rightarrow] = 9$	i.	$g[+] = 19$
e.	$g[=] = 11$	j.	$g[\times] = 21$

k. $g[x_i] = 23 + 2i$

So, for example, $g[x_5] = 23 + 2 \times 5 = 33$. Clearly each symbol gets a unique Gödel number, and Gödel numbers for individual symbols are odd positive integers.

Now we are in a position to assign a Gödel number to each expression as follows: Where $\mathscr{s}_0, \mathscr{s}_1, \ldots, \mathscr{s}_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\mathscr{s}_0]} \times 3^{g[\mathscr{s}_1]} \times 5^{g[\mathscr{s}_2]} \times \cdots \times p_n{}^{g[\mathscr{s}_n]}$$

where $2, 3, 5, \ldots, p_n$ are the first n prime numbers. So, for example, $g[x_0 \times x_5] = 2^{23} \times 3^{21} \times 5^{33}$. This is a big integer. But it is an integer, and different expressions get different Gödel numbers. Given a Gödel number, we can find the corresponding expression by finding its prime factorization; then if there are twenty-three 2s in the factorization, the first symbol is x_0; if there are twenty-one 3s, the second symbol is $\times$; and so forth. Notice that numbers for individual symbols are odd, where numbers for expressions are even.

Now consider a sequence of expressions, $\mathcal{Q}_0, \mathcal{Q}_1, \ldots, \mathcal{Q}_n$ (as in an axiomatic derivation). These expressions have Gödel numbers $g_0, g_1, \ldots, g_n$. Then,

$$2^{g_0} \times 3^{g_1} \times 5^{g_2} \times \cdots \times p_n{}^{g_n}$$

is the *super* Gödel number for the sequence $\mathcal{Q}_0, \mathcal{Q}_1, \ldots, \mathcal{Q}_n$. Again, given a super Gödel number, we can find the corresponding expressions by finding its prime factorization; then, if there are g_0 2s, we can proceed to the prime factorization of g_0 to discover the symbols of the first expression; and so forth. Observe that super Gödel numbers are even, but are distinct from Gödel numbers for expressions, insofar as the exponent of 2 in the factorization of any expression is odd (the first element of any expression is a symbol and so has an odd number); and the exponent of 2 in the factorization of any super Gödel number is even (the first element of a sequence is an expression and so has an even number).[15]

Recall that $\mathsf{exp}(\mathsf{n}, \mathsf{i})$ returns the exponent of $\pi(\mathsf{i})$ in the prime factorization of n. So for a Gödel number n, $\mathsf{exp}(\mathsf{n}, \mathsf{i})$ returns the code of $\mathscr{s}_i$; and for a super Gödel number n, $\mathsf{exp}(\mathsf{n}, \mathsf{i})$ returns the code of $\mathcal{Q}_i$.

Indicate individual symbol codes with angle quotes $\langle\ \rangle$ around the symbol; then for symbol $\mathscr{s}$, $\langle\overline{\mathscr{s}}\rangle$ is the corresponding numeral and $\langle\widehat{\mathscr{s}}\rangle$ the zero-place recursive function. Where $\mathcal{P}$ is any expression, let $\ulcorner\mathcal{P}\urcorner$ be its Gödel number, $\overline{\ulcorner\mathcal{P}\urcorner}$ the standard numeral for its Gödel number, and $\widehat{\ulcorner\mathcal{P}\urcorner}$ the corresponding zero-place recursive function. For a simple example, $\langle(\rangle = 3$; $\langle\overline{(}\rangle$ is the $\mathcal{L}_{NT}$ term $SSS\emptyset$ whose standard interpretation is 3, and $\langle\widehat{(}\rangle$ is the recursive function $\mathsf{suc}(\mathsf{suc}(\mathsf{suc}(\mathsf{zero}())))$ that returns the value 3. $\ulcorner(\urcorner = 2^{\langle(\rangle} = 2^3 = 8$; $\overline{\ulcorner(\urcorner}$ is the $\mathcal{L}_{NT}$ term $SSSSSSSS\emptyset$ whose standard interpretation is 8; and $\widehat{\ulcorner(\urcorner}$ is the recursive function $\mathsf{suc}(\mathsf{suc}(\mathsf{suc}(\mathsf{suc}(\mathsf{suc}(\mathsf{suc}(\mathsf{suc}($ $\mathsf{suc}(\mathsf{zero}()))))))))$ that returns the value 8. For a (slightly) more complex case, $\ulcorner x_0 \times x_5\urcorner = 2^{23} \times 3^{21} \times 5^{33}$ as described above.

Concatenation. Suppose m and n number expressions or sequences of expressions. Then the function $\mathsf{cncat}(\mathsf{m}, \mathsf{n})$—ordinarily indicated $\mathsf{m} \star \mathsf{n}$—returns the Gödel number of the expression or sequence with Gödel number m followed by the expression or sequence with Gödel number n. So for some numbered variables x, y, z, $\ulcorner x \times y\urcorner \star \ulcorner = z\urcorner = \ulcorner x \times y = z\urcorner$. This function is (primitive) recursive. Recall that $\mathsf{len}(\mathsf{n})$ is recursive and returns the number of distinct prime factors of n. Set $\mathsf{m} \star \mathsf{n}$ to,

$$(\mu x \leq B_{m,n})\{\widehat{1} \leq x \wedge (\forall i < \mathsf{len}(m))[\mathsf{exp}(x, i) = \mathsf{exp}(m, i)] \wedge (\forall i < \mathsf{len}(n))[\mathsf{exp}(x, i + \mathsf{len}(m)) = \mathsf{exp}(n, i)]\}$$

We search for the least number x (greater than or equal to one) such that exponents of initial primes in its factorization match the exponents of primes in m and exponents of primes later match exponents of primes in n. The bounded quantifiers take $\mathsf{i} < \mathsf{len}(\mathsf{m})$ and $\mathsf{i} < \mathsf{len}(\mathsf{n})$ insofar as len returns the number of primes, but $\mathsf{exp}(\mathsf{x}, \mathsf{i})$ starts the list of primes at 0; so if $\mathsf{len}(\mathsf{m}) = 3$, its primes are $\pi(0)$, $\pi(1)$, and $\pi(2)$. So the first $\mathsf{len}(\mathsf{m})$ exponents of x are the same as the exponents in m, and the next $\mathsf{len}(\mathsf{n})$ exponents of x are the same as the exponents in n.

[15] There are many ways to assign Gödel numbers. We pick just one. Our approach is like Gödel's original numbering strategy.

To ensure that the function is recursive we use bounded minimization, where $\mathsf{B}_{\mathsf{m,n}}$ is the bound under which we search for x. Let $\pi_{\mathsf{len(m)+len(n)}}$ be $\pi(\mathsf{len(m)} + \mathsf{len(n)})$; then in this case it is sufficient to set,

$$\mathsf{B}_{\mathsf{m,n}} = \left(\pi^{\mathsf{m+n}}_{\mathsf{len(m)+len(n)}}\right)^{\mathsf{len(m)+len(n)}}$$

The idea is that all the primes in x will be $< \pi_{\mathsf{len(m)+len(n)}}$. And any exponent in the factorization of m must be $\leq \mathsf{m}$ and any exponent for n must be $\leq \mathsf{n}$; so that $\mathsf{m} + \mathsf{n}$ is greater than any exponent in the factorization of x. So B results from multiplying a prime larger than any prime in x to a power greater than that of any exponent in x together as many times as there are primes in x; so x must be smaller than B.

Observe that $\star$ associates as for addition and multiplication and $(\mathsf{m} \star \mathsf{n}) \star \mathsf{o} = \mathsf{m} \star (\mathsf{n} \star \mathsf{o})$; given this we often drop parentheses for the concatenation operation. Also the requirement that $1 \leq \mathsf{m} \star \mathsf{n}$ does not usually matter since we will be interested in cases with $\mathsf{m}, \mathsf{n} > 1$; it does, however have the advantage that $\mathsf{m} \star \mathsf{n}$ is always equivalent to a product of primes—where this will smooth results down the road (if all the exponents of some primes are zero then their product is 1).

Terms. $\textsc{term}(\mathsf{n})$ is true iff n is the Gödel number of a term. Think of the trees on which we show that an expression is a term. Put formally, for any term t_n, there is a *term sequence* $t_0, t_1, \ldots, t_n$ such that each expression is either,

a. $\emptyset$

b. a variable

c. St_j where t_j occurs earlier in the sequence

d. $+t_i t_j$ where t_i and t_j occur earlier in the sequence

e. $\times t_i t_j$ where t_i and t_j occur earlier in the sequence

where we represent terms in unabbreviated form. A term is the last element of such a sequence. Let us try to say this.

First, $\textsc{var}(\mathsf{n})$ is true just in case n is the Gödel number of a variable—conceived as an expression, rather than a symbol. Then $\textsc{var}$ is (primitive) recursive. Set,

$$\textsc{var}(\mathsf{n}) = (\exists \mathsf{x} \leq \mathsf{n})\left(\mathsf{n} = \widehat{2}^{\widehat{2}3+\widehat{2}\mathsf{x}}\right)$$

If there is such an x, then n must be the Gödel number of a variable. And it is clear that this x is less than n itself. So the result is recursive.

Now $\textsc{termseq}(\mathsf{m}, \mathsf{n})$ is true when m is the super Gödel number of a sequence of terms whose last member has Gödel number n. We use the exponents in the factorization of m to number expressions in the sequence whose final member is the resulting term. Recall that $\mathsf{len(m)}$ returns the total number of primes in the factorization of m, and $\mathsf{exp(m, i)}$ the exponent of prime i in the factorization. For $\textsc{termseq}(\mathsf{m}, \mathsf{n})$ set,

$$\begin{aligned}
&\mathsf{exp(m, len(m)} \dot{-} \hat{1}) = \mathsf{n} \wedge \hat{1} < \mathsf{m} \wedge (\forall \mathsf{k} < \mathsf{len(m)})\{\\
&\mathsf{exp(m,k)} = \widehat{\ulcorner \emptyset \urcorner} \vee \textsc{var}(\mathsf{exp(m,k)}) \vee\\
&(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp(m,k)} = \widehat{\ulcorner S \urcorner} \star \mathsf{exp(m,j)}] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp(m,k)} = \widehat{\ulcorner + \urcorner} \star \mathsf{exp(m,i)} \star \mathsf{exp(m,j)}] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp(m,k)} = \widehat{\ulcorner \times \urcorner} \star \mathsf{exp(m,i)} \star \mathsf{exp(m,j)}]\}
\end{aligned}$$

So each term is zero, a variable, or put together from prior terms in the appropriate way. $\mathsf{len(m)}$ returns the number of primes in the prime factorization of m; if there is one prime it is $\pi(0)$, if there are two primes they are $\pi(0)$ and $\pi(1)$, and so forth. So the Gödel number n of the last member of the sequence is the exponent of $\pi_{\mathsf{len(m)} \dot{-} 1}$. We require that $\mathsf{m} > 1$ so that its length is other than 0 and the long quantified expression is not vacuously satisfied.

Then set $\textsc{term}(\mathsf{n})$ as follows:

$$\textsc{term}(\mathsf{n}) = (\exists \mathsf{x} \leq \mathsf{B_n})\textsc{termseq}(\mathsf{x}, \mathsf{n})$$

If some x numbers a term sequence for n, then n is a term. In this case, Gödel numbers of all prior members in a standard sequence ending in n are less than n. Further, the number of members in the sequence is the same as the number of variables and constants together with the number of function symbols in the term (one member for each variable and constant, and another corresponding to each function symbol); so the number of members in the sequence is the same as $\mathsf{len(n)}$; so all the primes in the sequence are $< \pi_{\mathsf{len(n)}}$. So multiply $\pi^{\mathsf{n}}_{\mathsf{len(n)}}$ together $\mathsf{len(n)}$ times and set $\mathsf{B_n} = \left(\pi^{\mathsf{n}}_{\mathsf{len(n)}}\right)^{\mathsf{len(n)}}$. We take a prime $\pi_{\mathsf{len(n)}}$ greater than all the primes in the sequence, to a power n greater than or equal to all the powers in the sequence, and multiply it together as many times as there are members of the sequence. The result must be greater than x, the number of the term sequence.[16]

Formulas. $\textsc{wff}(\mathsf{n})$ is to be true iff n is the number of a (well-formed) formula. For this, begin with some simple definitions. First, $\textsc{atomic}(\mathsf{n})$ is true iff n is the number of an atomic formula. The only atomic formulas of $\mathscr{L}_{\mathrm{NT}}$ are of the form $= t_1 t_2$. So it is sufficient to set,

$$\textsc{atomic}(\mathsf{n}) = (\exists \mathsf{x} \leq \mathsf{n})(\exists \mathsf{y} \leq \mathsf{n})[\textsc{term}(\mathsf{x}) \wedge \textsc{term}(\mathsf{y}) \wedge \mathsf{n} = \widehat{\ulcorner = \urcorner} \star \mathsf{x} \star \mathsf{y}]$$

[16]There are many term sequences for a given term numbered n—for members of a sequence might appear in different orders, and a sequence might include extraneous members not required for the final result. Reasoning above shows there *is* a (standard) sequence under the bound, not that all sequences for that term are under the bound.

Clearly the numbers of t_1 and t_2 are $\leq \mathsf{n}$ itself. For complex expressions, set $\mathsf{cnd}(\mathsf{n},\mathsf{o}) = \mathsf{m}$ when $\mathsf{n} = \ulcorner\mathcal{P}\urcorner$, $\mathsf{o} = \ulcorner\mathcal{Q}\urcorner$ and $\mathsf{m} = \ulcorner(\mathcal{P} \to \mathcal{Q})\urcorner$—and similarly for $\mathsf{til}(\mathsf{n})$ and $\mathsf{unv}(\mathsf{v},\mathsf{n})$.

$$\mathsf{cnd}(\mathsf{n},\mathsf{o}) = \widehat{\ulcorner(\urcorner} \star \mathsf{n} \star \widehat{\ulcorner\to\urcorner} \star \mathsf{o} \star \widehat{\ulcorner)\urcorner}$$

$$\mathsf{til}(\mathsf{n}) = \widehat{\ulcorner\sim\urcorner} \star \mathsf{n}$$

$$\mathsf{unv}(\mathsf{v},\mathsf{n}) = \widehat{\ulcorner\forall\urcorner} \star \mathsf{v} \star \mathsf{n}$$

Now think of the tree by which a formula is formed. There is a sequence of which each member is,

a. an atomic

b. $\sim\mathcal{P}$ for some previous member of the sequence $\mathcal{P}$

c. $(\mathcal{P} \to \mathcal{Q})$ for previous members of the sequence $\mathcal{P}$ and $\mathcal{Q}$

d. $\forall x\mathcal{P}$ for some previous member of the sequence $\mathcal{P}$ and variable x

So, on the model of what has gone before, we let $\textsc{formseq}(\mathsf{m},\mathsf{n})$ be true when m is the super Gödel number of a sequence of formulas whose last member has Gödel number n. We use the exponents in the factorization of m to number expressions in the sequence whose final member is the resulting formula. For $\textsc{formseq}(\mathsf{m},\mathsf{n})$ set,

$$\begin{aligned}
&\mathsf{exp}(\mathsf{m},\mathsf{len}(\mathsf{m}) \dot{-} \hat{1}) = \mathsf{n} \land \hat{1} < \mathsf{m} \land (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\
&\textsc{atomic}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \lor\\
&(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \mathsf{til}(\mathsf{exp}(\mathsf{m},\mathsf{j}))] \lor\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \mathsf{cnd}(\mathsf{exp}(\mathsf{m},\mathsf{i}),\mathsf{exp}(\mathsf{m},\mathsf{j}))] \lor\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} \leq \mathsf{n})[\textsc{var}(\mathsf{j}) \land \mathsf{exp}(\mathsf{m},\mathsf{k}) = \mathsf{unv}(\mathsf{j},\mathsf{exp}(\mathsf{m},\mathsf{i}))]\}
\end{aligned}$$

So a formula is the last member of a sequence each member of which is an atomic, or formed from previous members in the usual way. Clearly the number of a variable in an expression with number n is itself $\leq \mathsf{n}$.

Then set $\textsc{wff}(\mathsf{n})$ as follows,

$$\textsc{wff}(\mathsf{n}) = (\exists \mathsf{x} \leq \mathsf{B_n})\textsc{formseq}(\mathsf{x},\mathsf{n})$$

An expression is a formula iff there is a formula sequence of which it is the last member. Again, Gödel numbers of formulas in a standard sequence are $\leq \mathsf{n}$. And there are as many members of the sequence as there are atomics and operator symbols in the formula numbered n; so all the primes are $< \pi_{\mathsf{len}(\mathsf{n})}$.[17] So multiply $\pi^{\mathsf{n}}_{\mathsf{len}(\mathsf{n})}$ together $\mathsf{len}(\mathsf{n})$ times and set $\mathsf{B_n} = \left(\pi^{\mathsf{n}}_{\mathsf{len}(\mathsf{n})}\right)^{\mathsf{len}(\mathsf{n})}$.

[17] A formula may include more *symbols* than it has operators and atomics—consider, say, an atomic consisting of a relation symbol and terms. This makes $\mathsf{len}(\mathsf{n})$ greater than the length of the formula sequence and, again, primes in the sequence $< \pi_{\mathsf{len}(\mathsf{n})}$.

Sentential proof. $\textsc{prfads}(\mathsf{m}, \mathsf{n})$ is to be true iff m is the super Gödel number of a sequence of formulas that is an *ADs* derivation of the theorem with Gödel number n. So we revert to the relatively simple axiomatic system of Chapter 3. Thus the only rule is MP and, for example, A1 is of the sort $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$. For the sentential case, we need $\textsc{axiomads}(\mathsf{n})$ true when n is the number of an axiom. For this,

$$\begin{aligned}\textsc{axiomad1}(\mathsf{n}) &= (\exists \mathsf{p} \le \mathsf{n})(\exists \mathsf{q} \le \mathsf{n})[\textsc{wff}(\mathsf{p}) \wedge \textsc{wff}(\mathsf{q}) \wedge \mathsf{n} = \mathsf{cnd}(\mathsf{p}, \mathsf{cnd}(\mathsf{q}, \mathsf{p}))]\\ \textsc{axiomad2}(\mathsf{n}) &= \text{Homework.}\\ \textsc{axiomad3}(\mathsf{n}) &= \text{Homework.}\end{aligned}$$

Then,

$$\textsc{axiomads}(\mathsf{n}) = \textsc{axiomad1}(\mathsf{n}) \vee \textsc{axiomad2}(\mathsf{n}) \vee \textsc{axiomad3}(\mathsf{n})$$

In the next section, we will add all the logical axioms plus the axioms for Q. But these are all the axioms required for proofs of theorems in sentential logic. The rule is straightforward too. $\textsc{mp}(\mathsf{m}, \mathsf{n}, \mathsf{o})$ when m numbers a conditional whose antecedent is numbered n and consequent is numbered o.

$$\textsc{mp}(\mathsf{m}, \mathsf{n}, \mathsf{o}) = \mathsf{cnd}(\mathsf{n}, \mathsf{o}) = \mathsf{m}$$

Finally $\textsc{prfads}(\mathsf{m}, \mathsf{n})$ when m is the super Gödel number of a sequence that is a proof whose last member has Gödel number n. An *ADs* derivation is a sequence of formulas where each member is an axiom or follows from previous members by MP. This time we use the exponents in the factorization of m to number expressions in the proof—in the sequence of formulas whose final member is the one proved. This works like $\textsc{termseq}$ and $\textsc{formseq}$. For $\textsc{prfads}(\mathsf{m}, \mathsf{n})$ set,

$$\begin{aligned}&\mathsf{exp}(\mathsf{m}, \mathsf{len}(\mathsf{m}) \dot{-} \hat{1}) = \mathsf{n} \wedge \hat{1} < \mathsf{m} \wedge (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\ &\textsc{axiomads}(\mathsf{exp}(\mathsf{m}, \mathsf{k})) \vee\\ &(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})\textsc{mp}(\mathsf{exp}(\mathsf{m}, \mathsf{i}), \mathsf{exp}(\mathsf{m}, \mathsf{j}), \mathsf{exp}(\mathsf{m}, \mathsf{k}))\}\end{aligned}$$

So every formula is either an axiom or follows from previous members by MP.

It is a significant matter to have found this recursive relation! Again, in the next section, we will extend this notion to include other logical axioms, axioms of Q, and the rule Gen. Still, our construction for $\textsc{prfads}$ exhibits the essential steps required for the parallel relation $\textsc{prfq}(\mathsf{m}, \mathsf{n})$ true when m is the super Gödel number of a sequence that is a proof from the axioms of Q whose last member has Gödel number n. That discussion adds considerable detail. It is not clear that the detail is required for understanding the rest of this chapter—though of course, to the extent that results rely on the recursive $\textsc{prfq}$ relation, the detail underlies *proof* of the results!

E12.27. Find Gödel numbers for each of the following. Treat the first as an expression, rather than as simple symbol; the last is a sequence of expressions. For the latter two, you need not do the calculation!

(a) x_2 (b) $x_0 = x_1$ (c) $x_0 = x_1, \emptyset = x_0, \emptyset = x_1$

E12.28. Complete the cases for $\textsc{axiomad2}(\mathsf{n})$ and $\textsc{axiomad3}(\mathsf{n})$.

E12.29. In Chapter 8 (page 372) we define the notion of a *normal* sentential form. Let $\mathsf{wedge}(\mathsf{m},\mathsf{n}) = \mathsf{cnd}(\mathsf{til}(\mathsf{m}),\mathsf{n})$ and $\mathsf{caret}(\mathsf{m},\mathsf{n}) = \mathsf{til}(\mathsf{cnd}(\mathsf{m},\mathsf{til}(\mathsf{n})))$. Using these functions with $\textsc{atomic}$ and til from above, define a recursive relation $\textsc{norm}(\mathsf{n})$ whose application is to normal sentential forms of $\mathscr{L}_{\text{NT}}$. Hint: You will need a formula sequence to do this.

12.4.4 Completing the Construction

In this section we complete the construction of $\textsc{prfq}(\mathsf{m},\mathsf{n})$. In addition to A1–A3 and MP from *ADs*, Q includes the other logical axioms, axioms of Q, and the rule Gen. For the logical axioms, there are conditions as for A4, $\forall v \mathcal{P} \to \mathcal{P}^v_s$ where term s is substituted for variable v and s is free for v in $\mathcal{P}$. This is easy enough to apply in practice. But it takes some work to represent. We tackle the problem piece by piece.

Substitution in terms. Say $\mathsf{t} = \ulcorner t \urcorner$, $\mathsf{v} = \ulcorner v \urcorner$, and $\mathsf{s} = \ulcorner s \urcorner$ for some terms s, t, and variable v. Then $\textsc{termsub}(\mathsf{t},\mathsf{v},\mathsf{s},\mathsf{u})$ is true when u is the Gödel number of t^v_s. For this, we begin with a term sequence (with super Gödel number m) for t, and consider a parallel sequence (with super Gödel number n) that ends with u. The parallel sequence is not necessarily a term sequence, but rather includes modified versions of the terms in the sequence numbered m. For $\textsc{tsubseq}(\mathsf{m},\mathsf{n},\mathsf{t},\mathsf{v},\mathsf{s},\mathsf{u})$ set,

$$
\begin{aligned}
&\textsc{termseq}(\mathsf{m},\mathsf{t}) \land \mathsf{len}(\mathsf{m}) = \mathsf{len}(\mathsf{n}) \land \mathsf{exp}(\mathsf{n},\mathsf{len}(\mathsf{n}) \dot{-} \hat{1}) = \mathsf{u} \land (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\
&[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \widehat{\ulcorner \emptyset \urcorner} \land \mathsf{exp}(\mathsf{n},\mathsf{k}) = \widehat{\ulcorner \emptyset \urcorner}] \lor\\
&[\textsc{var}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \land \mathsf{exp}(\mathsf{m},\mathsf{k}) \neq \mathsf{v} \land \mathsf{exp}(\mathsf{n},\mathsf{k}) = \mathsf{exp}(\mathsf{m},\mathsf{k})] \lor\\
&[\textsc{var}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \land \mathsf{exp}(\mathsf{m},\mathsf{k}) = \mathsf{v} \land \mathsf{exp}(\mathsf{n},\mathsf{k}) = \mathsf{s}] \lor\\
&(\exists \mathsf{i} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \widehat{\ulcorner S \urcorner} \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \land \mathsf{exp}(\mathsf{n},\mathsf{k}) = \widehat{\ulcorner S \urcorner} \star \mathsf{exp}(\mathsf{n},\mathsf{i})] \lor\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \widehat{\ulcorner + \urcorner} \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \star \mathsf{exp}(\mathsf{m},\mathsf{j}) \land\\
&\qquad \mathsf{exp}(\mathsf{n},\mathsf{k}) = \widehat{\ulcorner + \urcorner} \star \mathsf{exp}(\mathsf{n},\mathsf{i}) \star \mathsf{exp}(\mathsf{n},\mathsf{j})] \lor\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \widehat{\ulcorner \times \urcorner} \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \star \mathsf{exp}(\mathsf{m},\mathsf{j}) \land\\
&\qquad \mathsf{exp}(\mathsf{n},\mathsf{k}) = \widehat{\ulcorner \times \urcorner} \star \mathsf{exp}(\mathsf{n},\mathsf{i}) \star \mathsf{exp}(\mathsf{n},\mathsf{j})]\}
\end{aligned}
$$

So the sequence for t^v_s (numbered by n) is like one of our "unabbreviating trees" from Chapter 2. In any place where the sequence for t (numbered by m) numbers $\emptyset$, the sequence for t^v_s numbers $\emptyset$. Where the sequence for t numbers a variable other than

v, the sequence for t^{v}_{s} numbers the same variable. But where the sequence for t numbers variable v, the sequence for t^{v}_{s} numbers s. Then later parts are built out of prior in parallel. The second sequence may not itself be a *term* sequence, insofar as it need not include all the antecedents to s (just as an unabbreviating tree would not include all the parts of a resultant term or formula).

Now set $\textsc{termsub}(t, v, s, u)$ as follows:

$$\textsc{termsub}(t, v, s, u) = (\exists x \leq X_t)(\exists y \leq Y_{t,u})\textsc{tsubseq}(x, y, t, v, s, u)$$

In this case, reasoning as for $\textsc{wff}$, the Gödel numbers in a standard sequence with number m are less than or equal to t and numbers in the sequence with number n less than or equal to u. And primes in the sequence are less than $\pi_{len(t)}$. So it is sufficient to set $X_t = \left(\pi^{t}_{len(t)}\right)^{len(t)}$ and $Y_{t,u} = \left(\pi^{u}_{len(t)}\right)^{len(t)}$.

Substitution in formulas. First, substitution into atomics. Say $p = \ulcorner \mathcal{P} \urcorner$, $v = \ulcorner v \urcorner$, and $s = \ulcorner s \urcorner$ for some atomic formula $\mathcal{P}$, variable v, and term s. Then $\textsc{atomsub}(p, v, s, q)$ is true when q is the Gödel number of $\mathcal{P}^{v}_{s}$. The condition is straightforward given $\textsc{termsub}$. For $\textsc{atomsub}(p, v, s, q)$,

$$(\exists a \leq p)(\exists b \leq p)(\exists a' \leq q)(\exists b' \leq q)[\textsc{term}(a) \wedge \textsc{term}(b) \wedge p = \widehat{\ulcorner = \urcorner} \star a \star b \wedge \\ \textsc{termsub}(a, v, s, a') \wedge \textsc{termsub}(b, v, s, b') \wedge q = \widehat{\ulcorner = \urcorner} \star a' \star b']$$

So $\mathcal{P}^{v}_{s}$ simply substitutes into the terms to which the equality applies.

Now, where $p = \ulcorner \mathcal{P} \urcorner$, $v = \ulcorner v \urcorner$, and $s = \ulcorner s \urcorner$ for an arbitrary formula $\mathcal{P}$, variable v, and term s, $\textsc{formsub}(p, v, s, q)$ is true when q is the Gödel number of $\mathcal{P}^{v}_{s}$. In the general case, $\mathcal{P}^{v}_{s}$ is complicated insofar as s replaces only *free* instances of v. Again, we build a parallel sequence with number n. When v is not free, we do not want to replace v with s—thus substitutions are made in atomics and then carried forward in all but subformulas that begin with a v-quantifier. For $\textsc{fsubseq}(m, n, p, v, s, q)$ set,

$$\begin{aligned}
&\textsc{formseq}(m, p) \wedge len(m) = len(n) \wedge exp(n, len(n) \dot{-} \hat{1}) = q \wedge (\forall k < len(m))\{\\
&[\textsc{atomic}(exp(m, k)) \wedge \textsc{atomsub}(exp(m, k), v, s, exp(n, k))] \vee\\
&(\exists i < k)[exp(m, k) = til(exp(m, i)) \wedge exp(n, k) = til(exp(n, i))] \vee\\
&(\exists i < k)(\exists j < k)[exp(m, k) = cnd(exp(m, i), exp(m, j)) \wedge exp(n, k) = cnd(exp(n, i), exp(n, j))] \vee\\
&(\exists i < k)(\exists j \leq p)[\textsc{var}(j) \wedge j \neq v \wedge exp(m, k) = unv(j, exp(m, i)) \wedge exp(n, k) = unv(j, exp(n, i))] \vee\\
&(\exists i < k)(\exists j \leq p)[\textsc{var}(j) \wedge j = v \wedge exp(m, k) = unv(j, exp(m, i)) \wedge exp(n, k) = exp(m, k)]\}
\end{aligned}$$

So substitutions are made in atomics, and carried forward in the parallel sequence—so long as no quantifier binds variable v, at which stage the sequence reverts to the form without substitution. And for $\textsc{formsub}(p, v, s, q)$,

$$\textsc{formsub}(p, v, s, q) = (\exists x \leq X_p)(\exists y \leq Y_{p,q})\textsc{fsubseq}(x, y, p, v, s, q)$$

Again, set $X_p = \left(\pi^{p}_{len(p)}\right)^{len(p)}$ and $Y_{p,q} = \left(\pi^{q}_{len(p)}\right)^{len(p)}$.

Where $\text{FORMSUB}(\mathsf{p}, \mathsf{v}, \mathsf{s}, \mathsf{q})$ is a relation which applies to the number q given p, v, and s, we may use the relation to define a function $\mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s})$ which *returns* q given p, v, and s. Set $\mathsf{formusb}(\mathsf{p}, \mathsf{v}, \mathsf{s}) = (\mu\mathsf{q} \leq \mathsf{Z}_{\mathsf{p},\mathsf{s}})\text{FORMSUB}(\mathsf{p}, \mathsf{v}, \mathsf{s}, \mathsf{q})$. In case there is no q that makes $\text{FORMSUB}(\mathsf{p}, \mathsf{v}, \mathsf{s}, \mathsf{q})$ true (as would happen if, say, p is not the number of a formula), this function reverts to the bound; otherwise $\mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s})$ is the number q of $\mathcal{P}^v_s$. In this case, the number of symbols in $\mathcal{P}^v_s$ is sure to be no greater than the number of symbols in $\mathcal{P}$ times the number of symbols in s (as though every symbol in $\mathcal{P}$ were replaced by s). And any symbol is either s or an element of $\mathcal{P}$; so the Gödel number of each symbol is no greater than the maximum of p and s and thus $\mathsf{p} + \mathsf{s}$. So q is sure to be under the bound, $\mathsf{Z}_{\mathsf{p},\mathsf{s}} = \left(\pi^{\mathsf{p}+\mathsf{s}}_{\mathsf{len}(\mathsf{p})\times\mathsf{len}(\mathsf{s})}\right)^{\mathsf{len}(\mathsf{p})\times\mathsf{len}(\mathsf{s})}$. Again, we take a prime greater than that of any symbol, to a power greater than that of any exponent, and multiply it (at least) as many times as there are symbols.

Free and bound variables. $\text{FREEt}(\mathsf{t}, \mathsf{v})$ and $\text{FREEf}(\mathsf{p}, \mathsf{v})$ are true when v is the Gödel number of a variable that is free in a term or formula that has Gödel number t or p. The idea for these relations is that substitution applies just to free variables. So if an expression changes upon a substitution of some term other than v for the variable v, v must have been free in the original expression. For a given variable x_i initially assigned number $23 + 2\mathsf{i}$, $\ulcorner x_i \urcorner = 2^{23+2\mathsf{i}}$ so that $2^{23+2\mathsf{i}+2}$ is the number of the next variable. In particular then, for v the number of a variable, $\mathsf{v} \times 2^2$ (that is $\mathsf{v} \times 4$) numbers a different variable. We use this to identify variables free in expressions numbered t and p. For terms and formulas respectively,

$$\text{FREEt}(\mathsf{t}, \mathsf{v}) = \sim\text{TERMSUB}(\mathsf{t}, \mathsf{v}, \mathsf{v} \times \hat{4}, \mathsf{t})$$

$$\text{FREEf}(\mathsf{p}, \mathsf{v}) = \sim\text{FORMSUB}(\mathsf{p}, \mathsf{v}, \mathsf{v} \times \hat{4}, \mathsf{p})$$

So v is free if the result upon substitution is other than the original expression. Observe that in Chapter 2 free and bound variables were introduced in relation to *formulas*. Now the notion is extended, in the obvious way, to terms—since terms lack quantifiers, a variable is free in a term iff it is present in the term.

Given $\text{FREEf}(\mathsf{p}, \mathsf{v})$, it is a simple matter to specify $\text{SENT}(\mathsf{n})$ true when n numbers a sentence.

$$\text{SENT}(\mathsf{n}) = \text{WFF}(\mathsf{n}) \wedge (\forall \mathsf{x} \leq \mathsf{n})[\text{VAR}(\mathsf{x}) \rightarrow \sim\text{FREEf}(\mathsf{n}, \mathsf{x})]$$

So n numbers a sentence if it numbers a formula and nothing is the number of a variable free in the formula numbered by n.

Finally, suppose $\mathsf{s} = \ulcorner s \urcorner$ and $\mathsf{v} = \ulcorner v \urcorner$; then $\text{FREEFOR}(\mathsf{s}, \mathsf{v}, \mathsf{u})$ is true iff s is free for v in the formula numbered by u. For this, we set up a sequence of formulas (not an ordinary formula sequence) including just formulas with s free for v—a sequence such that s is free for v in each member, whose last member has number u. For $\text{FFSEQ}(\mathsf{m}, \mathsf{s}, \mathsf{v}, \mathsf{u})$ set,

$$\begin{aligned}
&\mathrm{exp}(\mathrm{m}, \mathrm{len}(\mathrm{m}) \dot{-} \hat{1}) = \mathrm{u} \land \hat{1} < \mathrm{m} \land (\forall \mathrm{k} < \mathrm{len}(\mathrm{m}))\{\\
&\textsc{atomic}(\mathrm{exp}(\mathrm{m}, \mathrm{k})) \lor\\
&(\exists \mathrm{j} < \mathrm{k})[\mathrm{exp}(\mathrm{m}, \mathrm{k}) = \mathrm{til}(\mathrm{exp}(\mathrm{m}, \mathrm{j}))] \lor\\
&(\exists \mathrm{i} < \mathrm{k})(\exists \mathrm{j} < \mathrm{k})[\mathrm{exp}(\mathrm{m}, \mathrm{k}) = \mathrm{cnd}(\mathrm{exp}(\mathrm{m}, \mathrm{i}), \mathrm{exp}(\mathrm{m}, \mathrm{j}))] \lor\\
&(\exists \mathrm{p} \le \mathrm{u})(\exists \mathrm{j} \le \mathrm{u})[\textsc{var}(\mathrm{j}) \land \mathrm{j} = \mathrm{v} \land \textsc{wff}(\mathrm{p}) \land \mathrm{exp}(\mathrm{m}, \mathrm{k}) = \mathrm{unv}(\mathrm{j}, \mathrm{p})] \lor\\
&(\exists \mathrm{i} < \mathrm{k})(\exists \mathrm{j} \le \mathrm{u})[\textsc{var}(\mathrm{j}) \land \mathrm{j} \ne \mathrm{v} \land (\sim\textsc{freet}(\mathrm{s}, \mathrm{j}) \lor \sim\textsc{freef}(\mathrm{exp}(\mathrm{m}, \mathrm{i}), \mathrm{v})) \land\\
&\qquad \mathrm{exp}(\mathrm{m}, \mathrm{k}) = \mathrm{unv}(\mathrm{j}, \mathrm{exp}(\mathrm{m}, \mathrm{i}))]\}
\end{aligned}$$

For the last two clauses: First, within any subformula of the sort $\forall v \mathcal{P}$, no variable in s is bound upon substitution, for there are no free instances of v and so no substitutions. Observe that this $\mathcal{P}$ need not appear earlier in the sequence, as any formula with the v-quantifier satisfies the condition. Alternatively, if the main operator binds a different variable (and is not buried inside some $\mathcal{P}$ from the previous clause), we require either that the variable is not free in s (so that no instances are bound upon substitution) or that v is not free in the subformula (so that there are no substitutions). Given this,

$$\textsc{freefor}(\mathrm{s}, \mathrm{v}, \mathrm{u}) = (\exists \mathrm{x} \le \mathrm{B_u})\textsc{ffseq}(\mathrm{x}, \mathrm{s}, \mathrm{v}, \mathrm{u})$$

Even though this x need not number a formula sequence, whenever $\textsc{ffseq}(\mathrm{x}, \mathrm{s}, \mathrm{v}, \mathrm{u})$, then u *is* the number of a formula and every member of a standard FFSEQ is a member of a FORMSEQ; so $\mathrm{B_u}$ may be set as before.

Proof in Q. After all this work, we are finally ready for Gen, the other axioms of *AD*, and the axioms of Q.

The most challenging of these, $\textsc{axiomad4}(\mathrm{n})$ obtains when n is the Gödel number of an instance of A4. Intuitively, $\textsc{axiomad4}(\mathrm{n})$ just in case there is an s such that,

$$\begin{aligned}
&(\exists \mathrm{p} \le \mathrm{n})(\exists \mathrm{v} \le \mathrm{n})[\textsc{wff}(\mathrm{p}) \land \textsc{var}(\mathrm{v}) \land \textsc{term}(\mathrm{s}) \land \textsc{freefor}(\mathrm{s}, \mathrm{v}, \mathrm{p}) \land\\
&\qquad \mathrm{n} = \mathrm{cnd}(\mathrm{unv}(\mathrm{v}, \mathrm{p}), \mathrm{formsub}(\mathrm{p}, \mathrm{v}, \mathrm{s}))]
\end{aligned}$$

So there is a formula $\mathcal{P}$, variable v, and term s where s is free for v in $\mathcal{P}$; and the axiom is of the form, $\forall v \mathcal{P} \to \mathcal{P}^{v}_{s}$. Unfortunately, our statement is inadequate insofar as s is left free. We cannot simply supply a prefix $\exists$s as the result would not be recursively specified. It is tempting to add a bounded $(\exists \mathrm{s} \le \mathrm{n})$ with the idea that the number of s must be smaller than the number of $\mathcal{P}^{v}_{s}$. This almost works. The difficulty is the (rarely encountered) situation where the quantified variable v is not free in $\mathcal{P}$ (as when $\mathcal{P}$ is already a sentence); in this case, $\mathcal{P}^{v}_{s}$ is just $\mathcal{P}$, and there is nothing to say that s is less than n. Here is a way to do the job. Set $\textsc{axiomad4}(\mathrm{n})$ as,

$$\begin{aligned}
&(\exists \mathrm{p} \le \mathrm{n})(\exists \mathrm{v} \le \mathrm{n})\{\textsc{wff}(\mathrm{p}) \land \textsc{var}(\mathrm{v}) \land [\\
&\quad (\sim\textsc{freef}(\mathrm{p}, \mathrm{v}) \land \mathrm{n} = \mathrm{cnd}(\mathrm{unv}(\mathrm{v}, \mathrm{p}), \mathrm{p})) \lor\\
&\quad (\textsc{freef}(\mathrm{p}, \mathrm{v}) \land (\exists \mathrm{s} \le \mathrm{n})(\textsc{term}(\mathrm{s}) \land \textsc{freefor}(\mathrm{s}, \mathrm{v}, \mathrm{p}) \land \mathrm{n} = \mathrm{cnd}(\mathrm{unv}(\mathrm{v}, \mathrm{p}), \mathrm{formsub}(\mathrm{p}, \mathrm{v}, \mathrm{s}))))]\}
\end{aligned}$$

When $\sim\textsc{freef}(\mathsf{p}, \mathsf{v})$, $\mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s}) = \mathsf{p}$, so that $\mathsf{cnd}(\mathsf{unv}(\mathsf{v}, \mathsf{p}), \mathsf{p})$ is the same as $\mathsf{cnd}(\mathsf{unv}(\mathsf{v}, \mathsf{p}), \mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s}))$; and when $\textsc{freef}(\mathsf{p}, \mathsf{v})$, then s is smaller than the resultant formula. Either way, n is set to $\mathsf{cnd}(\mathsf{unv}(\mathsf{v}, \mathsf{p}), \mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s}))$. The result, then, is primitive recursive and equivalent to our original intuitive specification.

Now, mostly from homework, $\textsc{gen}$ and the other axioms follow in short order. $\textsc{gen}(\mathsf{m}, \mathsf{n})$ is true when the formula numbered n results by an application of Gen to the one numbered m; and $\textsc{axiomad5}(\mathsf{n})$ is true when n numbers an instance of A5; these are straightforward and left as exercises. Axiom six is of the sort $v = v$.

$$\textsc{axiomad6}(\mathsf{n}) = (\exists \mathsf{v} \leq \mathsf{n})[\textsc{var}(\mathsf{v}) \wedge \mathsf{n} = \widehat{\ulcorner = \urcorner} \star \mathsf{v} \star \mathsf{v}]$$

Axiom seven is of the sort, $(x_i = y) \to (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$ for function symbol h and variables $x_1 \ldots x_n$ and y. Because just a single replacement is made, we do not want to use $\textsc{termsub}$. However, we are in a position simply to list all the combinations in which one variable is replaced. In $\mathcal{L}_{NT}$ the function symbol is S, $+$, or $\times$. So the axiom is of the sort $= x y \to\, = s t$ where s is Sx, $+xz$, $+zx$, $\times xz$, or $\times zx$ and t replaces x in s with y. So for $\textsc{axiomad7}(\mathsf{n})$,

$$\begin{aligned}
&(\exists \mathsf{s} \leq \mathsf{n})(\exists \mathsf{t} \leq \mathsf{n})(\exists \mathsf{x} \leq \mathsf{n})(\exists \mathsf{y} \leq \mathsf{n})\{\textsc{var}(\mathsf{x}) \wedge \textsc{var}(\mathsf{y}) \wedge \mathsf{n} = \mathsf{cnd}(\widehat{\ulcorner = \urcorner} \star \mathsf{x} \star \mathsf{y}, \widehat{\ulcorner = \urcorner} \star \mathsf{s} \star \mathsf{t}) \wedge \\
&([\mathsf{s} = \widehat{\ulcorner S \urcorner} \star \mathsf{x} \wedge \mathsf{t} = \widehat{\ulcorner S \urcorner} \star \mathsf{y}] \vee \\
&(\exists \mathsf{z} \leq \mathsf{n})[\textsc{var}(\mathsf{z}) \wedge ((\mathsf{s} = \widehat{\ulcorner + \urcorner} \star \mathsf{x} \star \mathsf{z} \wedge \mathsf{t} = \widehat{\ulcorner + \urcorner} \star \mathsf{y} \star \mathsf{z}) \vee \\
&\qquad (\mathsf{s} = \widehat{\ulcorner + \urcorner} \star \mathsf{z} \star \mathsf{x} \wedge \mathsf{t} = \widehat{\ulcorner + \urcorner} \star \mathsf{z} \star \mathsf{y}))] \vee \\
&(\exists \mathsf{z} \leq \mathsf{n})[\textsc{var}(\mathsf{z}) \wedge ((\mathsf{s} = \widehat{\ulcorner \times \urcorner} \star \mathsf{x} \star \mathsf{z} \wedge \mathsf{t} = \widehat{\ulcorner \times \urcorner} \star \mathsf{y} \star \mathsf{z}) \vee \\
&\qquad (\mathsf{s} = \widehat{\ulcorner \times \urcorner} \star \mathsf{z} \star \mathsf{x} \wedge \mathsf{t} = \widehat{\ulcorner \times \urcorner} \star \mathsf{z} \star \mathsf{y}))])\}
\end{aligned}$$

Axiom eight is similar. It is stated in terms of atomics of the sort $\mathcal{R}^n x_1 \ldots x_n$ for relation symbol $\mathcal{R}$ and variables $x_1 \ldots x_n$. In $\mathcal{L}_{NT}$ the relation symbol is the equal sign, so these atomics are of the form, $= xy$. Again, because just a single replacement is made, we do not want to use $\textsc{formsub}$. However, we may proceed by analogy with $\textsc{axiomad7}$. This is left as an exercise. Thus we have compiled all the axioms of *AD*, and obtain $\textsc{axiomad}$ on the model of $\textsc{axiomads}$ from before.

For $\textsc{prfad}$ it is convenient to introduce a relation $\textsc{icon}(\mathsf{m}, \mathsf{n}, \mathsf{o})$ true when the formula with Gödel number o is an *immediate consequence* of ones numbered m and n.

$$\textsc{icon}(\mathsf{m}, \mathsf{n}, \mathsf{o}) = \textsc{mp}(\mathsf{m}, \mathsf{n}, \mathsf{o}) \vee (\mathsf{m} = \mathsf{n} \wedge \textsc{gen}(\mathsf{n}, \mathsf{o}))$$

Then $\textsc{prfad}(\mathsf{m}, \mathsf{n})$ is straightforward on the model of $\textsc{prfads}$.

The axioms of Q are particular formulas. So, for example, axiom Q2 is of the sort, $(Sx = Sy) \to (x = y)$. Let x and y be x_0 and x_1 respectively. Then,

$$\textsc{axiomq2}(\mathsf{n}) = \mathsf{n} = \widehat{\ulcorner (Sx = Sy) \to (x = y) \urcorner}$$

For ease of reading I do not reduce to unabbreviated form. Other axioms of Q may be treated in the same way. And now it is straightforward to produce $\textsc{axiomq}(\mathsf{n})$ that adds

Q1–Q7 to the axioms of *AD*. Then $\textsc{prfq}(\mathsf{m},\mathsf{n})$,

$$\begin{aligned}&\mathsf{exp}(\mathsf{m},\mathsf{len}(\mathsf{m}) \dot{-} \hat{1}) = \mathsf{n} \wedge \hat{1} < \mathsf{m} \wedge (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\&\textsc{axiomq}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \vee\\&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})\textsc{icon}(\mathsf{exp}(\mathsf{m},\mathsf{i}),\mathsf{exp}(\mathsf{m},\mathsf{j}),\mathsf{exp}(\mathsf{m},\mathsf{k}))\}\end{aligned}$$

works on the model of $\textsc{prfads}$ from before.

It has been our primary end in this section to find $\textsc{prfq}(\mathsf{m},\mathsf{n})$. And we have it. However, it is worth noting that with $\textsc{axiompa7}(\mathsf{n})$,

$$\begin{aligned}&(\exists \mathsf{p} \le \mathsf{n})(\exists \mathsf{v} \le \mathsf{n})\{\textsc{wff}(\mathsf{p}) \wedge \textsc{var}(\mathsf{v}) \wedge \mathsf{n} =\\&\quad \mathsf{cnd}[\mathsf{caret}(\mathsf{formsub}(\mathsf{p},\mathsf{v},\widehat{\ulcorner\emptyset\urcorner}),\mathsf{unv}(\mathsf{v},\mathsf{cnd}(\mathsf{p},\mathsf{formsub}(\mathsf{p},\mathsf{v},\widehat{\ulcorner S\urcorner}\star \mathsf{v})))),\mathsf{unv}(\mathsf{v},\mathsf{p})]\}\end{aligned}$$

we have also $\textsc{axiompa}(\mathsf{n})$ and $\textsc{prfpa}(\mathsf{m},\mathsf{n})$ for PA (for caret see E12.29). It is a significant matter to have found these recursive relations! Now we put them to work.

*E12.30. (i) Complete the construction with recursive relations for $\textsc{gen}(\mathsf{m},\mathsf{n})$, then for $\textsc{axiomad5}(\mathsf{n})$, $\textsc{axiomad8}(\mathsf{n})$, and so $\textsc{axiomad}(\mathsf{n})$ and $\textsc{prfad}(\mathsf{m},\mathsf{n})$. (ii) Complete the remaining axioms for Robinson arithmetic, and then $\textsc{axiomq}(\mathsf{n})$. (iii) Construct also $\textsc{axiompa}(\mathsf{n})$ and $\textsc{prfpa}(\mathsf{m},\mathsf{n})$.

E12.31. Suppose our numbering system is modified to include $\ulcorner\vee\urcorner$, $\ulcorner\wedge\urcorner$, $\ulcorner\exists\urcorner$, and in addition to til, cnd and unv we are given functions $\mathsf{pwedge}(\mathsf{m},\mathsf{n})$, $\mathsf{pcaret}(\mathsf{m},\mathsf{n})$, and $\mathsf{pexs}(\mathsf{v},\mathsf{n})$ for expressions that treat those operators as primitive. Augment $\textsc{formseq}$ to accommodate the new operators, and construct an $\textsc{unabbseq}(\mathsf{m},\mathsf{n},\mathsf{p},\mathsf{q})$ such that m numbers a formula sequence for p (which may contain the new operators) and n numbers a sequence whose last member is the unabbreviated version of p. Then construct $\textsc{unabb}(\mathsf{p},\mathsf{q})$ where q is the number of the unabbreviation of p.

12.5 Essential Results

In this section, we develop some first fruits of our labor. We shall need some initial theorems, important in their own right. With these theorems in hand, our results follow in short order. The results are developed and extended in later chapters. But it is worth putting them on the table at the start. (And some results at this stage provide a fitting cap to our labors.) We have expended a great deal of energy showing that, under appropriate conditions, recursive functions can be expressed and captured, and then that there exist recursive functions and relations including $\textsc{prfq}$. Now we put these results to work.

12.5.1 Diagonalization

Consider a formula $\mathcal{P}(x)$ with free variable x. The *diagonalization* of $\mathcal{P}$ is the formula $\exists x(x = \overline{\ulcorner \mathcal{P} \urcorner} \wedge \mathcal{P}(x))$. So the diagonalization of $\mathcal{P}$ is true just when $\mathcal{P}$ applies to its own Gödel number. To understand this nomenclature, consider a grid with formulas indexed by their Gödel numbers down the left and the integer Gödel numbers across the top.

	a	b	c	$\cdots$
$\mathcal{P}_{\mathsf{a}}(x)$	$\boldsymbol{\mathcal{P}_{\mathsf{a}}(\overline{\mathsf{a}})}$	$\mathcal{P}_{\mathsf{a}}(\overline{\mathsf{b}})$	$\mathcal{P}_{\mathsf{a}}(\overline{\mathsf{c}})$	
$\mathcal{P}_{\mathsf{b}}(x)$	$\mathcal{P}_{\mathsf{b}}(\overline{\mathsf{a}})$	$\boldsymbol{\mathcal{P}_{\mathsf{b}}(\overline{\mathsf{b}})}$	$\mathcal{P}_{\mathsf{b}}(\overline{\mathsf{c}})$	
$\mathcal{P}_{\mathsf{c}}(x)$	$\mathcal{P}_{\mathsf{c}}(\overline{\mathsf{a}})$	$\mathcal{P}_{\mathsf{c}}(\overline{\mathsf{b}})$	$\boldsymbol{\mathcal{P}_{\mathsf{c}}(\overline{\mathsf{c}})}$	
$\vdots$				

So, going down the main diagonal, formulas are of the sort $\mathcal{P}_{\mathsf{n}}(\overline{\mathsf{n}})$ where the formula numbered n is applied to its Gödel number n. Similarly, the diagonalization of $\mathcal{P}$ is true when $\mathcal{P}$ applies to $\overline{\ulcorner \mathcal{P} \urcorner}$.

It is easy to see that there is a recursive function $\mathsf{diag(n)}$ which takes the number of $\mathcal{P}$ and returns the number of its diagonalization. For this, let $\mathsf{num(n)}$ be the Gödel number of the standard numeral for n. So,

$$\mathsf{num(0)} = \widehat{\ulcorner \emptyset \urcorner}$$

$$\mathsf{num(Sy)} = \widehat{\ulcorner S \urcorner} \star \mathsf{num(y)}$$

So num is (primitive) recursive. Now $\mathsf{diag(n)}$ is the Gödel number of the diagonalization of the formula with Gödel number n.

$$\mathsf{diag(n)} = \widehat{\ulcorner \exists x(x = \urcorner} \star \mathsf{num(n)} \star \widehat{\ulcorner \wedge \urcorner} \star \mathsf{n} \star \widehat{\ulcorner) \urcorner}$$

It should be clear enough how to unabbreviate $\ulcorner \exists \urcorner$ and $\ulcorner \wedge \urcorner$. Since $\mathsf{diag(n)}$ is recursive, it is expressed and captured by some $Diag(x, y)$. Now we are ready for a pair of results which assert that for any formula $\mathcal{F}(y)$ there is an $\mathcal{H}$ with a sort of equivalence between $\mathcal{H}$ and $\mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$. The results come in semantical and syntactical versions.

First the semantical version. Consider a language including $\mathcal{L}_{NT}$ and the standard interpretation N.[18] Since $\mathsf{diag(n)}$ is recursive, there is a canonical formula $Diag(x, y)$ that expresses diag. Let $\mathcal{A}(x) = \exists y[Diag(x, y) \wedge \mathcal{F}(y)]$ and $\mathsf{a} = \ulcorner \mathcal{A} \urcorner$, the Gödel number of $\mathcal{A}$. Intuitively, $\mathcal{A}$ says $\mathcal{F}$ applies to the diagonalization of x. Then set $\mathcal{H} = \exists x(x = \overline{\mathsf{a}} \wedge \exists y[Diag(x, y) \wedge \mathcal{F}(y)])$ and $\mathsf{h} = \ulcorner \mathcal{H} \urcorner$, the Gödel number of $\mathcal{H}$. $\mathcal{H}$ is the diagonalization of $\mathcal{A}$; so $\mathsf{diag(a)} = \mathsf{h}$. Intuitively, $\mathcal{H}$ says that $\mathcal{F}$ applies to the diagonalization of $\mathcal{A}$, which is just to say that according to $\mathcal{H}$, $\mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$.

[18] Rather, for a language $\mathcal{L}'_{NT}$ including $\mathcal{L}_{NT}$ take an interpretation N' like N except that it makes some assignments to symbols in $\mathcal{L}'_{NT}$ but not in $\mathcal{L}_{NT}$. But we shall be interested just in assignments to the symbols of $\mathcal{L}_{NT}$, and so to assignments the same as N (compare T10.14).

T12.15. For any language including $\mathcal{L}_{NT}$ and formula $\mathcal{F}(y)$ containing just the variable y free, there is a sentence $\mathcal{H}$ such that $\mathsf{N}[\mathcal{H}] = \mathsf{N}[\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})]$. *Carnap's Equivalence*.[19]

For $\mathcal{L}$ including $\mathcal{L}_{NT}$ and any formula $\mathcal{F}(y)$, let $\mathcal{H}$ be constructed as above.

(i) Suppose $\mathsf{N}[\mathcal{H}] = \mathsf{T}$; then for any d, $\mathsf{N_d}[\mathcal{H}] = \mathsf{S}$; so $\mathsf{N_d}[\exists x(x = \bar{\mathsf{a}} \wedge \exists y[Diag(x,y) \wedge \mathcal{F}(y)])] = \mathsf{S}$; so for some m, $\mathsf{N_{d(x|m)}}[x = \bar{\mathsf{a}} \wedge \exists y(Diag(x,y) \wedge \mathcal{F}(y))] = \mathsf{S}$; from the first conjunct, this happens for $\mathsf{d}(x|\mathsf{a})$; so with T10.2, $\mathsf{N_d}[\exists y(Diag(\bar{\mathsf{a}},y) \wedge \mathcal{F}(y))] = \mathsf{S}$; so with SF$'(\exists)$ and T10.2 again, there is some m such that $\mathsf{N_d}[Diag(\bar{\mathsf{a}},\bar{\mathsf{m}})] = \mathsf{S}$ and $\mathsf{N_d}[\mathcal{F}(\bar{\mathsf{m}})] = \mathsf{S}$; from the first of these and expression $\langle \mathsf{a},\mathsf{m}\rangle \in \mathsf{diag}$; but $\mathsf{diag(a)} = \mathsf{h}$; so $\mathsf{m} = \mathsf{h}$; so $\mathsf{N_d}[\mathcal{F}(\bar{\mathsf{h}})] = \mathsf{S}$, and since d is arbitrary, $\mathsf{N}[\mathcal{F}(\bar{\mathsf{h}})] = \mathsf{T}$; which is to say $\mathsf{N}[\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})] = \mathsf{T}$.

(ii) Suppose $\mathsf{N}[\mathcal{H}] \neq \mathsf{T}$; then for some d, $\mathsf{N_d}[\mathcal{H}] \neq \mathsf{S}$; so $\mathsf{N_d}[\exists x(x = \bar{\mathsf{a}} \wedge \exists y[Diag(x,y) \wedge \mathcal{F}(y)])] \neq \mathsf{S}$; so for any m and in particular $\mathsf{m} = \mathsf{a}$, $\mathsf{N_{d(x|m)}}[x = \bar{\mathsf{a}} \wedge \exists y(Diag(x,y) \wedge \mathcal{F}(y))] \neq \mathsf{S}$; so with T10.2, $\mathsf{N_d}[\exists y(Diag(\bar{\mathsf{a}},y) \wedge \mathcal{F}(y))] \neq \mathsf{S}$; so for any m and in particular $\mathsf{m} = \mathsf{h}$, $\mathsf{N_{d(y|m)}}[Diag(\bar{\mathsf{a}},y) \wedge \mathcal{F}(y)] \neq \mathsf{S}$; so with T10.2, $\mathsf{N_d}[Diag(\bar{\mathsf{a}},\bar{\mathsf{h}})] \neq \mathsf{S}$ or $\mathsf{N_d}[\mathcal{F}(\bar{\mathsf{h}})] \neq \mathsf{S}$; so $\mathsf{N}[Diag(\bar{\mathsf{a}},\bar{\mathsf{h}})] \neq \mathsf{T}$ or $\mathsf{N}[\mathcal{F}(\bar{\mathsf{h}})] \neq \mathsf{T}$; but $\mathsf{diag(a)} = \mathsf{h}$; so by expression, $\mathsf{N}[Diag(\bar{\mathsf{a}},\bar{\mathsf{h}})] = \mathsf{T}$; so $\mathsf{N}[\mathcal{F}(\bar{\mathsf{h}})] \neq \mathsf{T}$; which is to say $\mathsf{N}[\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})] \neq \mathsf{T}$.

The reasoning skips some steps, however it is not a difficult exercise to fill in the details. Intuitively, this result should seem right: since $\mathcal{H}$ says that $\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$, $\mathcal{H}$ is true just in case $\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$ is true.

Now the syntactical version. Suppose T extends Q; since $\mathsf{diag(n)}$ is recursive, there is a canonical formula $Diag(x,y)$ that captures diag. Let $\mathcal{A}(x) = \exists y[Diag(x,y) \wedge \mathcal{F}(y)]$ and $\mathsf{a} = \ulcorner\mathcal{A}\urcorner$, the Gödel number of $\mathcal{A}$. Then set $\mathcal{H} = \exists x(x = \bar{\mathsf{a}} \wedge \exists y[Diag(x,y) \wedge \mathcal{F}(y)])$ and $\mathsf{h} = \ulcorner\mathcal{H}\urcorner$, the Gödel number of $\mathcal{H}$. $\mathcal{H}$ is the diagonalization of $\mathcal{A}$; so $\mathsf{diag(a)} = \mathsf{h}$. All this is the same as before, except that *Diag* captures rather than expresses diag. Intuitively, then, $\mathcal{H}$ says that $\mathcal{F}$ applies to the diagonalization of $\mathcal{A}$, which is just to say that according to $\mathcal{H}$, $\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$. This time we want to derive it.

T12.16. Let T be any theory that extends Q. Then for any formula $\mathcal{F}(y)$ containing just the variable y free, there is a sentence $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$. *Diagonal Lemma*.

For T extending Q and any formula $\mathcal{F}(y)$, let $\mathcal{H}$ be constructed as above. Since $\mathsf{diag(n)}$ is recursive, there is a formula $Diag(x,y)$ that captures diag; but $\mathsf{diag(a)} = \mathsf{h}$; so $T \vdash Diag(\bar{\mathsf{a}},\bar{\mathsf{h}})$ and $T \vdash \forall z(Diag(\bar{\mathsf{a}},z) \to \bar{\mathsf{h}} = z)$. See the derivation on the following page.

[19]This identification is from Smith (*An Introduction to Gödel's Theorems*, page 180), who traces the theorem's first appearance to Carnap (while in unfamiliar notation, compare Carnap, *Logical Syntax of Language* §35).

If a function f is such that $\mathsf{f}(\mathsf{n}) = \mathsf{n}$, then n is a *fixed point* for f. And by (a possibly strained) analogy, from this theorem $\mathcal{H}$ is said to be a "fixed point" for $\mathcal{F}(y)$.

*E12.32. Let T be any theory that extends Q. For any formulas $\mathcal{F}_1(z)$ and $\mathcal{F}_2(z)$, generalize the diagonal lemma to find sentences $\mathcal{H}_1$ and $\mathcal{H}_2$ such that,

$$T \vdash \mathcal{H}_1 \leftrightarrow \mathcal{F}_1(\overline{\ulcorner \mathcal{H}_2 \urcorner})$$
$$T \vdash \mathcal{H}_2 \leftrightarrow \mathcal{F}_2(\overline{\ulcorner \mathcal{H}_1 \urcorner})$$

Demonstrate your result. Hint: Generalize to the notion of a *joint diagonalization* where the *first* joint diagonalization of formulas $\mathcal{P}(x, y)$ and $\mathcal{Q}(x, y)$ is $\exists x \exists y[x = \overline{\ulcorner \mathcal{P} \urcorner} \wedge y = \overline{\ulcorner \mathcal{Q} \urcorner} \wedge \mathcal{P}(x, y)]$ and the *second* is $\exists x \exists y[x = \overline{\ulcorner \mathcal{P} \urcorner} \wedge y = \overline{\ulcorner \mathcal{Q} \urcorner} \wedge \mathcal{Q}(x, y)]$. Then given $\mathsf{p} = \ulcorner \mathcal{P} \urcorner$ and $\mathsf{q} = \ulcorner \mathcal{Q} \urcorner$, there are recursive functions $\mathsf{diag}_1(\mathsf{p}, \mathsf{q})$ and $\mathsf{diag}_2(\mathsf{p}, \mathsf{q})$ that return numbers of the joint diagonalizations, captured by some $Diag_1(x, y, z)$ and $Diag_2(x, y, z)$. Let $\mathcal{A}_1(x, y)$ =

T12.16

1.	$Diag(\overline{\mathsf{a}}, \overline{\mathsf{h}})$	from capture
2.	$\forall z(Diag(\overline{\mathsf{a}}, z) \rightarrow \overline{\mathsf{h}} = z)$	from capture
3.	$\mathcal{H}$	A (g, $\leftrightarrow$I)
4.	$\exists x(x = \overline{\mathsf{a}} \wedge \exists y[Diag(x, y) \wedge \mathcal{F}(y)])$	3 abv
5.	$j = \overline{\mathsf{a}} \wedge \exists y[Diag(j, y) \wedge \mathcal{F}(y)]$	A (g, 4$\exists$E)
6.	$j = \overline{\mathsf{a}}$	5 $\wedge$E
7.	$\exists y[Diag(j, y) \wedge \mathcal{F}(y)]$	5 $\wedge$E
8.	$Diag(j, k) \wedge \mathcal{F}(k)$	A (g, 7$\exists$E)
9.	$\mathcal{F}(k)$	8 $\wedge$E
10.	$Diag(j, k)$	8 $\wedge$E
11.	$Diag(\overline{\mathsf{a}}, k)$	10,6 =E
12.	$Diag(\overline{\mathsf{a}}, k) \rightarrow \overline{\mathsf{h}} = k$	2 $\forall$E
13.	$\overline{\mathsf{h}} = k$	12,11 $\rightarrow$E
14.	$\mathcal{F}(\overline{\mathsf{h}})$	9,13 =E
15.	$\mathcal{F}(\overline{\mathsf{h}})$	7,8-14 $\exists$E
16.	$\mathcal{F}(\overline{\mathsf{h}})$	4,5-15 $\exists$E
17.	$\mathcal{F}(\overline{\mathsf{h}})$	A (g, $\leftrightarrow$I)
18.	$Diag(\overline{\mathsf{a}}, \overline{\mathsf{h}}) \wedge \mathcal{F}(\overline{\mathsf{h}})$	1,17 $\wedge$I
19.	$\exists y[Diag(\overline{\mathsf{a}}, y) \wedge \mathcal{F}(y)]$	18 $\exists$I
20.	$\overline{\mathsf{a}} = \overline{\mathsf{a}}$	=I
21.	$\overline{\mathsf{a}} = \overline{\mathsf{a}} \wedge \exists y[Diag(\overline{\mathsf{a}}, y) \wedge \mathcal{F}(y)]$	20,19 $\wedge$I
22.	$\exists x(x = \overline{\mathsf{a}} \wedge \exists y[Diag(x, y) \wedge \mathcal{F}(y)])$	21 $\exists$I
23.	$\mathcal{H}$	22 abv
24.	$\mathcal{H} \leftrightarrow \mathcal{F}(\overline{\mathsf{h}})$	3-16,17-23 $\leftrightarrow$I
25.	$\mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$	24 abv

$\exists z[\mathit{Diag_2}(x, y, z) \wedge \mathscr{F}_1(z)]$ and $\mathscr{A}_2(x, y) = \exists z[\mathit{Diag_1}(x, y, z) \wedge \mathscr{F}_2(z)]$. You will be able to construct a sentence $\mathscr{H}_1$ that is the first joint diagonalization of $\mathscr{A}_1$ and $\mathscr{A}_2$, saying that $\mathscr{F}_1$ applies to the second; and then an $\mathscr{H}_2$ that is the second joint diagonalization of $\mathscr{A}_1$ and $\mathscr{A}_2$ that says $\mathscr{F}_2$ applies to the first.

12.5.2 The Ĩncompleteness of Arithmetic

Now we are ready for the result at which we have been aiming this whole chapter, the ĩncompleteness of arithmetic. Corresponding to Carnap's equivalence and then the diagonal lemma, the result comes in two forms. Say T is a *recursively axiomatized* formal theory if there is a recursive relation $\textsc{prft}(\mathsf{m}, \mathsf{n})$ which holds of m and n just in case m is the super Gödel number of a proof in T of the formula with Gödel number n . We have seen that Q is recursively axiomatized; but so is PA and any reasonable theory whose axioms and rules are recursively described (this is the content of 'nicely specified' in the introduction to this part).

Semantic Version

Corresponding to Carnap's equivalence, the semantic version of our argument depends on expression and then the s̃oundness of theory T.

T12.17. If T is a recursively axiomatized s̃ound theory whose language includes $\mathscr{L}_{NT}$, then there is a sentence $\mathscr{H}$ such that $T \nvdash \mathscr{H}$ and $T \nvdash \sim\mathscr{H}$.

Consider a recursively axiomatized s̃ound theory T whose language includes $\mathscr{L}_{NT}$. Since T is recursively axiomatized there is a recursive $\textsc{prft}(\mathsf{m}, \mathsf{n})$ and so $\mathit{Prft}(v, y)$ to express it. Then, where $\mathscr{F}(y)$ is $\sim\exists v \mathit{Prft}(v, y)$, by Carnap's equivalence, there is some $\mathscr{H}$ numbered h such that $\mathsf{N}[\mathscr{H}] = \mathsf{N}[\sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})]$.

(i) Suppose $T \vdash \mathscr{H}$; then since T is s̃ound, $\mathsf{N}[\mathscr{H}] = \mathsf{T}$; so by Carnap's equivalence, $\mathsf{N}[\sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] = \mathsf{T}$; so for any d, $\mathsf{N_d}[\sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] = \mathsf{S}$; so $\mathsf{N_d}[\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{S}$, and every m is such that $\mathsf{N_{d(v|m)}}[\mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{S}$; so with T10.2, $\mathsf{N_d}[\mathit{Prft}(\overline{\mathsf{m}}, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{S}$ and $\mathsf{N}[\mathit{Prft}(\overline{\mathsf{m}}, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{T}$; so by expression, $\langle \mathsf{m}, \mathsf{h} \rangle \notin \textsc{prft}$; and since this is so for every m, $T \nvdash \mathscr{H}$. Reject the assumption: $T \nvdash \mathscr{H}$.

(ii) Suppose $T \vdash \sim\mathscr{H}$; then since T is s̃ound, $\mathsf{N}[\sim\mathscr{H}] = \mathsf{T}$; so $\mathsf{N}[\mathscr{H}] \neq \mathsf{T}$; so by Carnap's equivalence, $\mathsf{N}[\sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{T}$; so for some d, $\mathsf{N_d}[\sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] \neq \mathsf{S}$; so $\mathsf{N_d}[\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] = \mathsf{S}$; so for some m, $\mathsf{N_{d(v|m)}}[\mathit{Prft}(v, \overline{\ulcorner \mathscr{H} \urcorner})] = \mathsf{S}$; and with T10.2, $\mathsf{N_d}[\mathit{Prft}(\overline{\mathsf{m}}, \overline{\ulcorner \mathscr{H} \urcorner})] = \mathsf{S}$; so by expression, $\langle \mathsf{m}, \mathsf{h} \rangle \in \textsc{prft}$; so $T \vdash \mathscr{H}$; and since T is s̃ound, $\mathsf{N}[\mathscr{H}] = \mathsf{T}$. This is impossible; reject the assumption: $T \nvdash \sim\mathscr{H}$.

By Carnap's equivalence, $\mathscr{H}$ is true iff it is not provable; with s̃oundness, this has the consequence that neither $\mathscr{H}$ nor $\sim\mathscr{H}$ is provable. So if T is a recursively axiomatized s̃ound theory whose language includes $\mathscr{L}_{NT}$, then T is ĩncomplete.

Syntactical Version

Corresponding to the diagonal lemma, the syntactical version of our argument depends on capture and then the consistency of theory T. This time we get the result in two versions. The first, simpler version is somewhat weaker than we would like—but important nonetheless. For this, we need a new concept: Say a theory T is ω-*incomplete* iff for some $\mathcal{P}(x)$, T can prove $\mathcal{P}(\overline{\mathsf{m}})$ for each $\overline{\mathsf{m}}$ in $\mathcal{L}_{NT}$ but T cannot go on to prove $\forall x\mathcal{P}(x)$. Equivalently, T is ω-incomplete iff T can prove $\sim\mathcal{P}(\overline{\mathsf{m}})$ for each $\overline{\mathsf{m}}$ in $\mathcal{L}_{NT}$ but $T \nvdash \sim\exists x\mathcal{P}(x)$. And we might generalize to more than one place. Then we have already seen that Q is ω-incomplete: we can prove say, every sentence $\overline{\mathsf{n}} \times \overline{\mathsf{m}} = \overline{\mathsf{m}} \times \overline{\mathsf{n}}$, but cannot go on to prove the corresponding universal generalization $\forall x\forall y(x \times y = y \times x)$. Say T is ω-*inconsistent* iff for some $\mathcal{P}(x)$, T proves $\mathcal{P}(\overline{\mathsf{m}})$ for every $\overline{\mathsf{m}}$ in $\mathcal{L}_{NT}$ but also proves $\sim\forall x\mathcal{P}(x)$. Equivalently, T is ω-inconsistent iff $T \vdash \sim\mathcal{P}(\overline{\mathsf{m}})$ for each $\overline{\mathsf{m}}$ in $\mathcal{L}_{NT}$ and $T \vdash \exists x\mathcal{P}(x)$. On the standard interpretation N for $\mathcal{L}_{NT}$, ω-incompleteness is a theoretical weakness—there are some things true but not provable. But ω-inconsistency is a theoretical disaster: It is not possible for the theorems of an ω-inconsistent theory to be true on N. ω-inconsistency is not itself inconsistency—for we do not have any sentence such that $T \vdash \mathcal{P}$ and $T \vdash \sim\mathcal{P}$. But we do have sentences that cannot all be true on N.[20] Observe that inconsistent theories are automatically ω-inconsistent—for from contradiction all consequences follow (including each $\mathcal{P}(\overline{\mathsf{m}})$ and also $\sim\forall x\mathcal{P}(x)$); transposing, ω-consistent theories are consistent. Now we show,

T12.18. If T is a recursively axiomatized theory extending Q, then there is a sentence $\mathcal{H}$ such that (i) if T is consistent, $T \nvdash \mathcal{H}$ and (ii) if T is ω-consistent, $T \nvdash \sim\mathcal{H}$.

Consider a recursively axiomatized theory T extending Q. Since T is recursively axiomatized there is a recursive $\textsc{prft}(\mathsf{m},\mathsf{n})$ and so $\mathit{Prft}(v, y)$ to capture it. Then, where $\mathcal{F}(y)$ is $\sim\exists v\mathit{Prft}(v, y)$, by the diagonal lemma, there is some $\mathcal{H}$ numbered h such that $T \vdash \mathcal{H} \leftrightarrow \sim\exists v\mathit{Prft}(v, \overline{\ulcorner\mathcal{H}\urcorner})$.

(i) Suppose T is consistent and $T \vdash \mathcal{H}$. Then since T is recursively axiomatized, for some m, $\textsc{prft}(\mathsf{m},\mathsf{h})$; and since T extends Q, by capture, $T \vdash \mathit{Prft}(\overline{\mathsf{m}}, \overline{\ulcorner\mathcal{H}\urcorner})$; so by ($\exists$I), $T \vdash \exists v\mathit{Prft}(v, \overline{\ulcorner\mathcal{H}\urcorner})$; so with the diagonal lemma and NB, $T \vdash \sim\mathcal{H}$; and since T is consistent, $T \nvdash \mathcal{H}$. Reject the assumption: $T \nvdash \mathcal{H}$.

(ii) Suppose T is ω-consistent and $T \vdash \sim\mathcal{H}$. Then by the diagonal lemma and NB, $T \vdash \exists v\mathit{Prft}(v, \overline{\ulcorner\mathcal{H}\urcorner})$. Since T is ω-consistent, it is consistent; so $T \nvdash \mathcal{H}$; so since T is recursively axiomatized, for all m, $\langle\mathsf{m},\mathsf{h}\rangle \notin \textsc{prft}$; and since T extends Q, by capture, $T \vdash \sim\mathit{Prft}(\overline{\mathsf{m}}, \overline{\mathsf{h}})$; so since T is ω-consistent, $T \nvdash \exists v\mathit{Prft}(v, \overline{\mathsf{h}})$; which is to say $T \nvdash \exists v\mathit{Prft}(v, \overline{\ulcorner\mathcal{H}\urcorner})$. This is impossible: $T \nvdash \sim\mathcal{H}$.

[20]From T10.16 any consistent theory has a model. So a theory that is consistent but not ω-consistent has a model. But the universe of a model for a theory that is consistent but not ω-consistent must include some member(s) to which no $\overline{\mathsf{m}}$ from $\mathcal{L}_{NT}$ is assigned.

So any recursively axiomatized ω-consistent theory extending Q is $\tilde{\text{i}}$ncomplete. But it is possible to strengthen this result by dropping the special assumption of ω-consistency.

Enhanced Syntactical Version

Again we depend upon the diagonal lemma. Without the special assumption of ω-consistency we show that no consistent, recursively axiomatized theory extending Q is negation $\tilde{\text{c}}$omplete. For this, we develop a few (independently interesting) preliminary theorems.

Say that if f is a function from (an initial segment of) $\mathbb{N}$ onto some set—so that the objects in the set are $\mathsf{f}(0), \mathsf{f}(1), \ldots$—then f *enumerates* the members of the set. A set is *recursively* enumerable if there is a recursive function that enumerates it.

T12.19. If T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable.

Consider pairs $\langle \mathsf{p}, \mathsf{t}\rangle$ where p numbers a proof of the theorem numbered t, each such pair itself associated with a number, $2^{\mathsf{p}} \times 3^{\mathsf{t}}$. Then there is a recursive function from the natural numbers to these *codes* as follows:

$$\mathsf{code}(0) = \mu z(\exists p < z)(\exists t < z)[z = \widehat{2}^{p} \times \widehat{3}^{t} \wedge \textsc{prft}(p, t)]$$
$$\mathsf{code}(Sn) = \mu z(\exists p < z)(\exists t < z)[\mathsf{code}(n) < z \wedge z = \widehat{2}^{p} \times \widehat{3}^{t} \wedge \textsc{prft}(p, t)]$$

So 0 is associated with the least integer that codes a proof of a theorem, 1 with the next, and so forth. Then,

$$\mathsf{ethrmt}(n) = \mathsf{exp}(\mathsf{code}(n), \widehat{1})$$

returns the Gödel number of theorem n in this ordering: $\mathsf{code}(\mathsf{n})$ returns the code matched to n, and exp the number of the coded theorem. For every theorem there is a proof of it, and so some code of which it is a member, and some n such that $\mathsf{ethrmt}(\mathsf{n})$ returns its Gödel number. So the theorems are recursively enumerable.

A given theorem might appear more than once in this enumeration, corresponding to codes with different proofs of it, but this is no problem, as we require only that each theorem appears in some position(s) of the list (and if it were important to eliminate duplicates, we might have added a conjunct $\sim(\exists x < p)\textsc{prft}(x, t)$ to the condition for $\mathsf{code}(Sn)$). And we might have added a conjunct $\textsc{sent}(t)$ to produce an enumeration of just sentences. Observe that we have, for the first time, made use of regular minimization—so that this function is recursive but not *primitive* recursive. Supposing that T has an infinite number of theorems, there is always some z at which the characteristic function upon which the minimization operates returns zero—so that the function is well-defined. So the theorems of a recursively axiomatized formal theory T are recursively enumerable.

Suppose we add that T is consistent and negation c̃omplete. Then the relation $\textsc{prvt}(\mathsf{p})$ true just for numbers of formulas provable in T (for theorems of T) is recursive. Intuitively, we can enumerate the theorems; then if T is consistent and negation c̃omplete, for any sentence $\mathcal{P}$, exactly one of $\mathcal{P}$ or $\sim\mathcal{P}$ must show up in the enumeration. So we can search through the list until we find either $\mathcal{P}$ or $\sim\mathcal{P}$—and if the one we find is $\mathcal{P}$, then $\mathcal{P}$ is a theorem. In particular, we find $\mathcal{P}$ or $\sim\mathcal{P}$ at the position, $\mu\mathsf{n}[\mathsf{ethrmt}(\mathsf{n}) = \widehat{\ulcorner\mathcal{P}\urcorner} \vee \mathsf{ethrmt}(\mathsf{n}) = \widehat{\ulcorner\sim\mathcal{P}\urcorner}]$. A complication is that negation completeness applies to *sentences*, while not all theorems are sentences. We overcome it by considering the universal closure $\mathsf{uclose}(\mathsf{p})$ of theorem p (see E12.33 at the end of this section): given A4 and Gen any formula $\mathcal{P}$ is provably equivalent to its universal closure and, given negation c̃ompleteness, either the closure or its negation must show up in the enumeration. Thus,

T12.20. For any recursively axiomatized, consistent, negation c̃omplete formal theory T the relation $\textsc{prvt}(\mathsf{p})$ true just in case p numbers a theorem of T is recursive. Let $\mathsf{pos}(\mathsf{p})$ be,

$$\mu\mathsf{n}([\sim\textsc{wff}(\mathsf{p}) \wedge \mathsf{n} = 0] \vee [\textsc{wff}(\mathsf{p}) \wedge (\mathsf{ethrmt}(\mathsf{n}) = \mathsf{uclose}(\mathsf{p}) \vee \mathsf{ethrmt}(\mathsf{n}) = \mathsf{til}(\mathsf{uclose}(\mathsf{p})))])$$

Then let $\textsc{prvt}(\mathsf{p})$ be,

$$\mathsf{ethrmt}(\mathsf{pos}(\mathsf{p})) = \mathsf{uclose}(\mathsf{p})$$

So $\mathsf{pos}(\mathsf{p})$ takes one of three values: if p does not number a formula it is just 0; for any p whose closure appears in the enumeration of theorems it is the position of the closure; and if the negation of its closure appears in the enumeration, it is the position of the negation. Then $\textsc{prvt}(\mathsf{p})$ is true just in case pos takes the second option—just in case p numbers a formula and the number of the formula at $\mathsf{pos}(\mathsf{p})$ is $\mathsf{uclose}(\mathsf{p})$ rather than $\mathsf{til}(\mathsf{uclose}(\mathsf{p}))$. So $\textsc{prvt}(\mathsf{p})$ is recursive and true just in case the closure, and so p itself, is a theorem of T.

Observe that $\mathsf{pos}(\mathsf{p})$ returns 0 when p does not number a formula, and when the number for the closure of p (or its negation) is the number of the first theorem in the enumeration. But when $\mathsf{pos}(\mathsf{p}) = 0$, $\mathsf{ethrmt}(\mathsf{pos}(\mathsf{p}))$ always numbers the first theorem of the enumeration—so that if p (and so $\mathsf{uclose}(\mathsf{p})$) is not the number of a formula $\textsc{prvt}(\mathsf{p})$ is false, and when $\mathsf{uclose}(\mathsf{p})$ is the number of the first theorem it is true (as it should be). Again, we appeal to regular minimization. In this case, the function to which the minimization operator applies is regular just because T is negation c̃omplete. So long as p numbers a formula, the characteristic function for the second square brackets is sure to go to zero for one disjunct or the other, and when p does not number a formula the function for the first square brackets goes to zero. So $\mathsf{pos}(\mathsf{p})$ and thus $\textsc{prvt}(\mathsf{p})$ are recursively defined.

As we have just seen, for a recursively axiomatized, consistent, negation c̃omplete theory $\textsc{prvt}(\mathsf{p})$ is recursive. Also, for *any* recursively axiomatized theory there is a recursive $\textsc{prft}(\mathsf{x}, \mathsf{y})$. But the existence of a recursive $\textsc{prft}$ for some theory does not by itself imply that $\textsc{prvt}$ for that theory is recursive—in particular, prefixing $\textsc{prft}(\mathsf{x}, \mathsf{y})$ with an existential quantifier does not result in a recursive relation insofar as unbounded quantifications are not recursive. In fact, for a consistent theory T extending Q, $\textsc{prvt}$ is not recursive. This results as a corollary to the following theorem.

T12.21. For any consistent theory T extending Q the relation $\textsc{prvt}(\mathsf{n})$, true when n numbers a theorem of T, is not captured by any formula $Prvt(y)$.

Consider a consistent theory T extending Q; and suppose the relation $\textsc{prvt}(\mathsf{n})$, true just in case n numbers a theorem of T, is captured by some $Prvt(y)$. Then there is a formula $\sim Prvt(y)$; and again since T extends Q, by the diagonal lemma there is a formula $\mathscr{H}$ with Gödel number $\ulcorner\mathscr{H}\urcorner = \mathsf{h}$ such that,

$$T \vdash \mathscr{H} \leftrightarrow \sim Prvt(\overline{\ulcorner\mathscr{H}\urcorner})$$

Suppose $T \vdash \mathscr{H}$; then $\mathsf{h} \in \textsc{prvt}$; so by capture, $T \vdash Prvt(\overline{\ulcorner\mathscr{H}\urcorner})$; so by NB, $T \vdash \sim\mathscr{H}$; and since T is consistent $T \nvdash \mathscr{H}$; this is impossible; reject the assumption: $T \nvdash \mathscr{H}$. But then $\mathscr{H}$ is not a theorem of T so that $\mathsf{h} \notin \textsc{prvt}$; so by capture, $T \vdash \sim Prvt(\overline{\ulcorner\mathscr{H}\urcorner})$; so by $\leftrightarrow$E, $T \vdash \mathscr{H}$. This is impossible; reject the original assumption: $\textsc{prvt}$ is not captured by any *Prvt*.

Corollary: For any consistent theory T extending Q the relation $\textsc{prvt}(\mathsf{n})$, true just in case n is a Gödel number of a theorem of T, is not recursive. Suppose otherwise, that $\textsc{prvt}(\mathsf{n})$ is recursive; then with T12.14 there is some formula $Prvt(y)$ that captures $\textsc{prvt}(\mathsf{n})$; but by the main result, this is impossible.

From T12.20 for any recursively axiomatized, consistent, *negation c̃omplete* formal theory the relation $\textsc{prvt}(\mathsf{n})$ is recursive. But by the corollary to T12.21 for any consistent theory extending Q the relation $\textsc{prvt}(\mathsf{n})$ is not recursive. This already suggests the ĩncompleteness result.

T12.22. No consistent, recursively axiomatized theory extending Q is negation c̃omplete.

Consider a theory T that is a consistent, recursively axiomatized extension of Q. Then since T is consistent and extends Q, by the corollary to T12.21, the relation $\textsc{prvt}(\mathsf{n})$, true iff n is the Gödel number of a theorem, is not recursive. Suppose T is negation c̃omplete; then since T is also consistent and recursively axiomatized, by T12.20, $\textsc{prvt}(\mathsf{n})$ is recursive. This is impossible; reject the assumption: T is not negation c̃omplete.

And this time we have the syntactical ĩncompleteness result without the special assumption of ω-consistency.

A theory whose axioms are true on N is $\tilde{\text{s}}$ound, and by E7.19 the axioms of Q are true on N; it follows that Q is $\tilde{\text{s}}$ound. Similarly, a theory whose axioms are true on N is both consistent and ω-consistent and, again, the axioms of Q are true on N; it follows that Q is consistent and ω-consistent. Given that it is $\tilde{\text{s}}$ound, consistent, and ω-consistent, from our theorems it immediately follows that Q is not negation $\tilde{\text{c}}$omplete. But similarly for *any* $\tilde{\text{s}}$ound recursively axiomatized theory whose language includes $\mathscr{L}_{NT}$, and for *any* consistent recursively axiomatized theory that extends Q. We already knew that there were sentences $\mathscr{P}$ such that $\mathrm{Q} \nvdash \mathscr{P}$ and $\mathrm{Q} \nvdash \sim\mathscr{P}$. One might have supposed that extensions of Q (such as PA) would remedy this problem. But our $\tilde{\text{i}}$ncompleteness results apply generally to recursively axiomatized theories extending Q—and so to any such theory. There are other ways to demonstrate $\tilde{\text{i}}$ncompleteness. We explore some in chapters that follow. However, these first arguments are sufficient to establish the point.

*E12.33. Define $\mathsf{uclose(p)}$ that returns the number of the universal closure of the formula numbered p. In order to add quantifiers in order of ascending subscripts it will be convenient to recursively define a series of expressions $\mathsf{qser(p, n)}$ numbering (up to) the n outermost quantifiers to be concatenated with p. So $\mathsf{qser(p, 0)} = \widehat{1}$ and $\mathsf{qser(p, Sn)}$ appends to $\mathsf{qser(p, n)}$ a quantifier for the next variable free in $\mathsf{qser(p, n)} \star \mathsf{p}$—remaining unchanged when there are no more free variables.

E12.34. Let T be any consistent theory extending Q and suppose $\textsc{sbthrmt}(\mathsf{n})$ is a recursive relation such that if $\textsc{sbthrmt}(\mathsf{n})$ then n numbers a theorem of T. So $\textsc{sbthrmt}(\mathsf{n})$ applies to numbers for a subset of the theorems of T. Use the diagonal lemma to show that there is a sentence $\mathscr{H}$ such that $T \vdash \mathscr{H}$ but $\ulcorner\mathscr{H}\urcorner \notin \textsc{sbthrmt}$. So a recursive relation which applies only to theorems cannot apply to all the theorems.

E12.35. Use the version of the diagonal lemma from E12.32 to provide an alternate demonstration of T12.18. Hint: You will be able to set up sentences such that the first says the second is not provable, while the second says the first is provable.

E12.36. Use the version of the diagonal lemma from E12.32 to provide an alternate demonstration of T12.21, that for any consistent theory T extending Q the relation $\textsc{prvt}(\mathsf{n})$, is not captured by any formula $Prvt(z)$.

E12.37. Consider a recursively axiomatized $\tilde{\text{s}}$ound theory whose language includes $\mathscr{L}_{NT}$. Show that reasoning parallel to that of T12.21 but using Carnap's equivalence fails to show that $\textsc{prvt}(\mathsf{n})$ is not *expressed* by some $Prvt(y)$—all you get is some $\mathscr{H}$ such that $T \nvdash \mathscr{H}$ and $\mathsf{N}[\mathscr{H}] = \mathsf{T}$. (Indeed, for a recursively axiomatized theory, $\textsc{prvt}(\mathsf{n})$ is expressed by $\exists v Prft(v, y)$.)

12.5.3 The Decision Problem

It is a short step from the result that if Q is consistent, then no recursive relation identifies the theorems of Q, to the result that if Q is consistent, then no recursive relation identifies the theorems of predicate logic.

T12.23. Suppose Q is consistent and $\mathscr{L}$ extends $\mathscr{L}_{NT}$; then the relation $\textsc{prvpl}(\mathsf{n})$ true iff n numbers an $\mathscr{L}$-theorem of predicate logic is not recursive.

Suppose otherwise, that Q is consistent and for $\mathscr{L}$ extending $\mathscr{L}_{NT}$ the relation $\textsc{prvpl}(\mathsf{n})$ true iff n numbers an $\mathscr{L}$-theorem of predicate logic is recursive. The axioms of Q are equivalent to their universal closures; with the axioms in this form, let $\mathcal{Q}$ be the conjunction of Q1–Q7; since Q1–Q7 in this form are particular sentences, $\mathcal{Q}$ is a particular sentence. Then Q $\vdash \mathcal{P}$ iff $\mathcal{Q} \vdash \mathcal{P}$; by DT iff $\vdash \mathcal{Q} \to \mathcal{P}$. Let $\mathsf{q} = \ulcorner \mathcal{Q} \urcorner$; then since $\textsc{prvpl}$ is recursive,

$$\textsc{prvq}(\mathsf{n}) = \textsc{prvpl}(\mathsf{cnd}(\hat{\mathsf{q}}, \mathsf{n}))$$

defines a recursive relation true iff n numbers a theorem of Q. But, given the consistency of Q, by the corollary to T12.21, $\textsc{prvq}(\mathsf{n})$ is not recursive. Reject the assumption: if Q is consistent, then the relation $\textsc{prvpl}(\mathsf{n})$ true iff n numbers an $\mathscr{L}$-theorem of predicate logic is not recursive.

Further, as we observed at the close of the previous section (page 632), Q *is* consistent; it follows that no recursive relation numbers the $\mathscr{L}$-theorems of predicate logic. With T12.21 no recursive relation numbers the theorems of Q. Now we see that this result extends to logical theorems. At this stage, these results may seem to be a sort of curiosity about what recursive functions do. They gain significance when, as we have already hinted can be done, we identify the recursive functions with the *computable* functions in Chapter 14.[21]

12.5.4 Tarski's Theorem

Say $\textsc{ltrue}(\mathsf{n})$ is true iff n numbers a sentence of language $\mathscr{L}$ true on the standard interpretation N. We do not assume that $\textsc{ltrue}(\mathsf{n})$ is recursive—only that, by definition, it applies to numbers of true sentences.

[21]This result applies to theorems in a language including $\mathscr{L}_{NT}$, and shows that there is no generally applicable recursive function to identify logical theorems. However in particular contexts theorems may be decidable. So for example the theorems of any sentential language are decidable; also theorems of *monadic predicate logic* which includes only one-place relation symbols are decidable. See also page 767, note 4.

T12.24. If language $\mathscr{L}$ includes $\mathscr{L}_{NT}$, then no formula *Ltrue* of $\mathscr{L}$ expresses LTRUE.

Suppose otherwise, that $\mathscr{L}$ extends $\mathscr{L}_{NT}$ and some *Ltrue*(x) expresses LTRUE(n) in $\mathscr{L}$. Then for any $\mathscr{P}$ in $\mathscr{L}$,

(A) $\mathsf{N}[Ltrue(\overline{\ulcorner\mathscr{P}\urcorner})] = \mathsf{T}$ iff $\ulcorner\mathscr{P}\urcorner \in$ LTRUE iff $\mathsf{N}[\mathscr{P}] = \mathsf{T}$

And by Carnap's equivalence there is a sentence $\mathscr{F}$ (a *false* or *liar* sentence) in $\mathscr{L}$ such that,

(B) $\mathsf{N}[\mathscr{F}] = \mathsf{T}$ iff $\mathsf{N}[\sim Ltrue(\overline{\ulcorner\mathscr{F}\urcorner})] = \mathsf{T}$ iff $\mathsf{N}[Ltrue(\overline{\ulcorner\mathscr{F}\urcorner})] \neq \mathsf{T}$

But then by (A), $\mathsf{N}[Ltrue(\overline{\ulcorner\mathscr{F}\urcorner})] = \mathsf{T}$ iff $\mathsf{N}[\mathscr{F}] = \mathsf{T}$; by (B) iff $\mathsf{N}[Ltrue(\overline{\ulcorner\mathscr{F}\urcorner})] \neq \mathsf{T}$. This is impossible; reject the original assumption: no formula *Ltrue*(x) in $\mathscr{L}$ expresses LTRUE(n).

And since every recursive relation is expressed in $\mathscr{L}_{NT}$, neither is LTRUE recursive. This theorem explains our standard jump to the metalanguage when we give conditions like SF and TI. Nothing prevents stating truth conditions. Trouble results when a theory purports to give conditions for all the sentences in its own language—if *Ltrue* is a formula of $\mathscr{L}$, Carnap's equivalence applies to it, and trouble ensues.

Observe that capture implies expression: So long as we use the same formulas for capture and expression, it is perhaps obvious that capture in a *sound* theory implies expression. Further, from T14.10 (to which you may find it interesting to refer) if a total function can be captured by a consistent recursively axiomatized theory extending Q then it is recursive; so by T12.5 it is expressed on the standard interpretation N for $\mathscr{L}_{NT}$. Thus, for some representative examples, the situation is as follows:

relation	recursive	captured	expressed
PRFQ(m, n)	✓	✓	✓
PRVQ(n)	X	X	✓
LTRUE(n)	X	X	X

Recursive relations and functions are both captured and expressed. Captured relations and functions are recursive and expressed. But expression does not imply capture. So, $\exists v Prfq(v, n)$ expresses PRVQ(n). However, as we have just observed, if PRVQ(n) is captured (in a consistent recursively axiomatized theory extending Q) then it is recursive; but by the corollary to T12.21 PRVQ(n) is not recursive; so PRVQ(n) is not captured. And now we have seen a relation LTRUE(n) not even expressed in $\mathscr{L}_{NT}$. But then we already knew from page 576 that some functions (and so relations) cannot be expressed in $\mathscr{L}_{NT}$. LTRUE(n) is a specific, significant, example.

This is a decent start into the results of Part IV of the text. In the following, we turn to deepening and extending them in different directions.

E12.38. For a language $\mathcal{L}$ that includes $\mathcal{L}_{NT}$, suppose $\textsf{SBTRUE}(n)$ is a recursive relation such that if $\textsf{SBTRUE}(n)$ then n numbers a sentence true on N. So $\textsf{SBTRUE}(n)$ applies to numbers for a subset of the truths on N. Use Carnap's equivalence to show that there is a sentence $\mathcal{H}$ such that $\mathsf{N}[\mathcal{H}] = \mathsf{T}$ but $\ulcorner\mathcal{H}\urcorner \notin \textsf{SBTRUE}$. So a recursive function which applies only to truths cannot apply to all the truths.

E12.39. Say T is a *theory of truth* for its language $\mathcal{L}$ just in case there is a formula $Ltrue(y)$ such that $T \vdash \mathcal{P} \leftrightarrow Ltrue(\overline{\ulcorner\mathcal{P}\urcorner})$ for every $\mathcal{P}$. Use the diagonal lemma to show that no recursively axiomatized consistent theory extending Q is a theory of truth for its own language $\mathcal{L}$.

E12.40. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The recursive functions and the role of the beta function in their expression and capture.

b. The essential elements from this chapter contributing to the proof of the ĩncompleteness of arithmetic.

c. The essential elements from this chapter contributing to the proof of that no recursive relation identifies the theorems of predicate logic

d. The essential elements from this chapter contributing to the proof of Tarski's theorem.

Final Results of Chapter 12

T12.15 For any language including $\mathscr{L}_{\text{NT}}$ and formula $\mathcal{F}(y)$ containing just the variable y free, there is a sentence $\mathcal{H}$ such that $\mathsf{N}[\mathcal{H}] = \mathsf{N}[\mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})]$. *Carnap's Equivalence*.

T12.16 Let T be any theory that extends Q. Then for any formula $\mathcal{F}(y)$ containing just the variable y free, there is a sentence $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$. *Diagonal Lemma*.

T12.17 If T is a recursively axiomatized s̃ound theory whose language includes $\mathscr{L}_{\text{NT}}$, then there is a sentence $\mathcal{H}$ such that $T \nvdash \mathcal{H}$ and $T \nvdash \sim\mathcal{H}$.

T12.18 If T is a recursively axiomatized theory extending Q, then there is a sentence $\mathcal{H}$ such that (i) if T is consistent, $T \nvdash \mathcal{H}$ and (ii) if T is ω-consistent, $T \nvdash \sim\mathcal{H}$.

T12.19 If T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable.

T12.20 For any recursively axiomatized, consistent, negation c̃omplete formal theory T the relation $\textsc{prvt}(\mathsf{p})$ true just in case p numbers a theorem of T is recursive.

T12.21 For any consistent theory T extending Q the relation $\textsc{prvt}(\mathsf{n})$, true when n numbers a theorem of T, is not captured by any formula $Prvt(y)$.

Corollary: For any consistent theory T extending Q the relation $\textsc{prvt}(\mathsf{n})$, true just in case n is a Gödel number of a theorem of T, is not recursive.

T12.22 No consistent, recursively axiomatized theory extending Q is negation c̃omplete.

T12.23 Suppose Q is consistent and $\mathscr{L}$ extends $\mathscr{L}_{\text{NT}}$; then the relation $\textsc{prvpl}(\mathsf{n})$ true iff n numbers an $\mathscr{L}$-theorem of predicate logic is not recursive.

T12.24 If language $\mathscr{L}$ includes $\mathscr{L}_{\text{NT}}$, then no formula *Ltrue* of $\mathscr{L}$ expresses $\textsc{ltrue}$. *Tarski's Theorem.*

Chapter 13

Gödel's Theorems

We have seen a demonstration of the ĩncompleteness of arithmetic. In this chapter, we take another run at that result, this time by Gödel's original strategy of producing particular sentences that are true iff not provable. This enables us to extend and deepen the ĩncompleteness result, and puts us in a position to take up Gödel's second ĩncompleteness theorem, according to which theories (of a certain sort) are not sufficient for demonstrations of *consistency*. We begin with a section (13.1) devoted to Gödel's first theorem. After that, sections 13.2–13.6 take up the second theorem.

13.1 Gödel's First Theorem

The arguments for ĩncompleteness from Chapter 12 depended upon Carnap's equivalence and the diagonal lemma. These are *existential* results: Under certain conditions, for a formula $\mathscr{F}$, *there is* an $\mathscr{H}$ such that $\mathscr{H}$ is equivalent to $\mathscr{F}(\overline{\ulcorner\mathscr{H}\urcorner})$. Correspondingly, our demonstrations of ĩncompleteness were demonstrations that *there is* a formula such that neither it nor its negation is provable. But we do not thereby exhibit any particular formula such that neither it nor its negation is provable. Still, our reasoning for the existential results was constructive. This suggests the possibility of finding a particular sentence $\mathscr{G}$ such that $T \nvdash \mathscr{G}$ and $T \nvdash \sim\mathscr{G}$. This is what we do. Again, the arguments come in *semantical* and *syntactical* versions and depend upon diagonal results.

13.1.1 Diagonalization

Recall that the diagonalization of $\mathscr{P}(x)$ is the formula $\exists x(x = \overline{\ulcorner\mathscr{P}\urcorner} \wedge \mathscr{P}(x))$. From section 12.5.1, there is a recursive function $\mathsf{diag}(\mathsf{n})$ that returns the number of the diagonalization of the formula with number n. We begin with semantical and syntactical results parallel to Carnap's equivalence and the diagonal lemma. Our reasoning is very much like that from section 12.5.1—to which you may find it helpful to refer.

First the semantic version. Consider some recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$. Since $\textsc{prft}(\mathsf{m},\mathsf{n})$ and $\mathsf{diag}(\mathsf{n})$ are recursive, they are expressed by canonical formulas $Prft(v,y)$ and $Diag(x,y)$. Let $\mathscr{A}(x) = \exists y(Diag(x,y) \wedge \sim\exists v Prft(v,y))$, and $\mathsf{a} = \ulcorner\mathscr{A}\urcorner$. So $\mathscr{A}$ says nothing numbers a proof of the diagonalization of a formula with number x. This construction is as from section 12.5.1, except with the particular formula $\sim\exists v Prft(v,y)$ in place of the variable $\mathscr{F}$. Then,

$$\mathscr{G} = \exists x(x = \overline{\mathsf{a}} \wedge \exists y(Diag(x,y) \wedge \sim\exists v Prft(v,y)))$$

So $\mathscr{G}$ is the diagonalization of $\mathscr{A}$, and intuitively $\mathscr{G}$ "says" that nothing numbers a proof of it. Observe that $\mathscr{G}$ is defined relative to $Prft$ for T; so each T yields its own Gödel sentence (if it were not ugly, we might sensibly introduce subscripts $\mathscr{G}_T$). Now as before,

T13.1. For any recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$, $\mathsf{N}[\mathscr{G}] = \mathsf{N}[\sim\exists v Prft(v,\overline{\ulcorner\mathscr{G}\urcorner})]$. *Carnap's result for $\mathscr{G}$.*

Consider a recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$ and the formula $\mathscr{G}$ as described above. Then by reasoning as from T12.15, $\mathsf{N}[\mathscr{G}] = \mathsf{N}[\sim\exists v Prft(v,\overline{\ulcorner\mathscr{G}\urcorner})]$. Homework.

And now the syntactical theorem. Since $\textsc{prft}(\mathsf{m},\mathsf{n})$ and $\mathsf{diag}(\mathsf{n})$ are recursive, in theories extending Q they are captured by canonical formulas $Prft(v,y)$ and $Diag(x,y)$. As above, let $\mathscr{A}(x) = \exists y(Diag(x,y) \wedge \sim\exists v Prft(v,y))$, and $\mathsf{a} = \ulcorner\mathscr{A}\urcorner$. So $\mathscr{A}$ says nothing numbers a proof of the diagonalization of a formula with number x. Then, $\mathscr{G} = \exists x(x = \overline{\mathsf{a}} \wedge \exists y(Diag(x,y) \wedge \sim\exists v Prft(v,y)))$. So $\mathscr{G}$ is the diagonalization of $\mathscr{A}$; let g be the Gödel number of $\mathscr{G}$.

*T13.2. Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathscr{G} \leftrightarrow \sim\exists v Prft(v,\overline{\ulcorner\mathscr{G}\urcorner})$. *Diagonal result for $\mathscr{G}$.*

Since T is recursively axiomatized, there is a recursive $\textsc{prft}$ and since T extends Q there are $Prft$ and $Diag$ that capture $\textsc{prft}$ and diag. From capture $T \vdash Diag(\overline{\mathsf{a}},\overline{\mathsf{g}})$, and $T \vdash \forall z(Diag(\overline{\mathsf{a}},z) \to \overline{\mathsf{g}} = z)$. It follows that $T \vdash \mathscr{G} \leftrightarrow \sim\exists v Prft(v,\overline{\mathsf{g}})$; which is to say, $T \vdash \mathscr{G} \leftrightarrow \sim\exists v Prft(v,\overline{\ulcorner\mathscr{G}\urcorner})$. Homework.

So we have results parallel to Carnap's equivalence and the diagonal lemma—only this time applied to sentence $\mathscr{G}$.

E13.1. Let $Odd(y) = \exists w(y = \overline{2} \times w + \overline{1})$. Find a sentence that is true iff its own number is odd. Motivate the stages of your construction as for the construction of $\mathscr{G}$.

E13.2. Provide reasoning for T13.1 which does not skip any steps.

*E13.3. Complete the demonstration of T13.2 by providing a derivation to show $T \vdash \mathcal{G} \leftrightarrow \sim\exists v Prft(v, \overline{\ulcorner \mathcal{G} \urcorner})$.

13.1.2 The Incompleteness of Arithmetic

Again we have a semantic result which requires expression and soundness, and a syntactical one which requires capture and consistency—where this latter result comes in two forms.

Simple Versions

Given the theorems from above, we begin with reasoning entirely parallel to that for T12.17 and T12.18.

T13.3. If T is a recursively axiomatized sound theory whose language includes $\mathcal{L}_{\text{NT}}$, then $T \nvdash \mathcal{G}$ and $T \nvdash \sim\mathcal{G}$.

Suppose T is a recursively axiomatized sound theory whose language includes $\mathcal{L}_{\text{NT}}$. Reasoning as for T12.17, $T \nvdash \mathcal{G}$ and $T \nvdash \sim\mathcal{G}$. Homework.

T13.4. If T is a recursively axiomatized theory extending Q, then if T is consistent $T \nvdash \mathcal{G}$, and if T is ω-consistent, $T \nvdash \sim\mathcal{G}$.

Suppose T is a recursively axiomatized theory extending Q. Reasoning as for T12.18, if T is consistent, $T \nvdash \mathcal{G}$, and if T is ω-consistent, $T \nvdash \sim\mathcal{G}$. Homework.

So we have the results from before only applied to the particular sentence $\mathcal{G}$. Further, it is a short step (which you have the opportunity to take in homework) from T13.1 according to which $\mathsf{N}[\mathcal{G}] = \mathsf{T}$ iff $\mathsf{N}[\sim\exists v Prft(v, \overline{\ulcorner \mathcal{G} \urcorner})] = \mathsf{T}$ to the result that $\mathcal{G}$ is true iff $T \nvdash \mathcal{G}$. Since $T \nvdash \mathcal{G}$, we have that $\mathcal{G}$ is both unprovable and true.

T13.4 is roughly the form in which Gödel proved the incompleteness of arithmetic in 1931: If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathcal{G}$; and if T is an ω-consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathcal{G}$. Gödel himself did not show the result for Q, but rather for the significantly stronger theory of Russell and Whitehead's 1910–13 *Principia Mathematica*—but his argument does not require all the power of that theory and, as we have seen, the reasoning goes through for theories extending Q. Insofar as standard theories including Q and PA are consistent and ω-consistent, the results are sufficient for the incompleteness of arithmetic (compare note 2 on page 642).

Rosser's Sentence

But again it is possible to drop the special assumption of ω-consistency. This time we proceed by means of a sentence somewhat different from $\mathcal{G}$.[1] Let $\overline{\mathsf{PRFT}}(\mathsf{m},\mathsf{n}) = \mathsf{PRFT}(\mathsf{m},\mathsf{til}(\mathsf{n}))$; so $\overline{\mathsf{PRFT}}(\mathsf{m},\mathsf{n})$ obtains when m numbers a proof of the negation of the formula numbered n. Since it is recursive, it is captured by some $\overline{Prft}(w,y)$. Set,

$$RPrft(v,y) = Prft(v,y) \wedge (\forall w \leq v){\sim}\overline{Prft}(w,y)$$

So $RPrft(v,y)$ just in case v numbers a proof of the formula numbered y, and no w less than or equal to v numbers a proof of the negation of that formula. Now, working as before, set $\mathcal{A}'(x) = \exists y(Diag(x,y) \wedge {\sim}\exists v RPrft(v,y))$, and $\mathsf{a} = \ulcorner\mathcal{A}'\urcorner$. So $\mathcal{A}'$ says nothing numbers an R-proof of the diagonalization of a formula with number x. Then,

$$\mathcal{R} = \exists x(x = \bar{\mathsf{a}} \wedge \exists y(Diag(x,y) \wedge {\sim}\exists v RPrft(v,y)))$$

So $\mathcal{R}$ is the diagonalization of $\mathcal{A}'$. And $\mathcal{R}$ has the key syntactic property just like $\mathcal{G}$. Again, reasoning as for the diagonal lemma,

T13.5. Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathcal{R} \leftrightarrow {\sim}\exists v RPrft(v, \overline{\ulcorner\mathcal{R}\urcorner})$. *Diagonal result for* $\mathcal{R}$.

You can show this just as for T13.2.

And now we can show that a consistent, recursively axiomatized theory extending Q proves neither $\mathcal{R}$ nor ${\sim}\mathcal{R}$. Reasoning is somewhat more involved than before, but still straightforward.

T13.6. If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathcal{R}$ and $T \nvdash {\sim}\mathcal{R}$.

Suppose T is a consistent recursively axiomatized theory extending Q, and let $\mathsf{r} = \overline{\ulcorner\mathcal{R}\urcorner}$.

(i) Suppose $T \vdash \mathcal{R}$. Then by T13.5, $T \vdash {\sim}\exists v RPrft(v, \overline{\ulcorner\mathcal{R}\urcorner})$. Since T is recursively axiomatized, for some m, $\mathsf{PRFT}(\mathsf{m},\mathsf{r})$; and since T extends Q, by capture, $T \vdash Prft(\overline{\mathsf{m}},\bar{\mathsf{r}})$. But by consistency, $T \nvdash {\sim}\mathcal{R}$; so there is no n such that $\mathsf{PRFT}(\mathsf{n},\mathsf{til}(\mathsf{r}))$; so for all n, and in particular all $\mathsf{n} \leq \mathsf{m}$, $\langle \mathsf{n},\mathsf{r}\rangle \notin \overline{\mathsf{PRFT}}$; so by capture, $T \vdash {\sim}\overline{Prft}(\bar{\mathsf{n}},\bar{\mathsf{r}})$; so by T8.25, $T \vdash (\forall w \leq \overline{\mathsf{m}}){\sim}\overline{Prft}(w,\bar{\mathsf{r}})$; so $T \vdash Prft(\overline{\mathsf{m}},\bar{\mathsf{r}}) \wedge (\forall w \leq \overline{\mathsf{m}}){\sim}\overline{Prft}(w,\bar{\mathsf{r}})$; so $T \vdash RPrft(\overline{\mathsf{m}},\bar{\mathsf{r}})$; so $T \vdash \exists v RPrft(v,\bar{\mathsf{r}})$; and since T is consistent, $T \nvdash {\sim}\exists v RPrft(v,\bar{\mathsf{r}})$, which is to say, $T \nvdash {\sim}\exists v RPrft(v,\overline{\ulcorner\mathcal{R}\urcorner})$. This is impossible; reject the assumption: $T \nvdash \mathcal{R}$.

(ii) Suppose $T \vdash {\sim}\mathcal{R}$. Then since T is recursively axiomatized, for some m, $\langle \mathsf{m},\mathsf{r}\rangle \in \overline{\mathsf{PRFT}}$; and since T extends Q, by capture, $T \vdash \overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$. By consistency,

[1] Barkley Rosser, "Extensions of Some Theorems of Gödel and Church."

$T \nvdash \mathcal{R}$; so for any n, and in particular, any $\mathsf{n} \le \mathsf{m}$, $\langle \mathsf{n}, \mathsf{r}\rangle \notin$ PRFT; so by capture, $T \vdash \sim Prft(\overline{\mathsf{n}}, \overline{\mathsf{r}})$; and by T8.25, $T \vdash (\forall w \le \overline{\mathsf{m}}) \sim Prft(w, \overline{\mathsf{r}})$. Now reason as follows:

1.	$\sim\mathcal{R}$	from T
2.	$\overline{Prft}(\overline{\mathsf{m}}, \overline{\mathsf{r}})$	capture
3.	$(\forall w \le \overline{\mathsf{m}}) \sim Prft(w, \overline{\mathsf{r}})$	capture and T8.25
4.	$\mathcal{R} \leftrightarrow \sim\exists v RPrft(v, \overline{\mathsf{r}})$	T13.5
5.	$\exists v RPrft(v, \overline{\mathsf{r}})$	1,4 NB
6.	$RPrft(j, \overline{\mathsf{r}})$	A (c, 5∃E)
7.	$Prft(j, \overline{\mathsf{r}}) \wedge (\forall w \le j) \sim \overline{Prft}(w, \overline{\mathsf{r}})]$	6 abv
8.	$j \le \overline{\mathsf{m}} \vee \overline{\mathsf{m}} \le j$	T8.23
9.	$j \le \overline{\mathsf{m}}$	A (c, 8∨E)
10.	$Prft(j, \overline{\mathsf{r}})$	7 ∧E
11.	$\sim Prft(j, \overline{\mathsf{r}})$	3,9 (∀E)
12.	$\bot$	10,11 ⊥I
13.	$\overline{\mathsf{m}} \le j$	A (c, 8∨E)
14.	$(\forall w \le j) \sim \overline{Prft}(w, \overline{\mathsf{r}})]$	7 ∧E
15.	$\sim \overline{Prft}(\overline{\mathsf{m}}, \overline{\mathsf{r}})$	14,13 (∀E)
16.	$\bot$	2,15 ⊥I
17.	$\bot$	8,9-12,13-16 ∨E
18.	$\bot$	5,6-17 ∃E

So T is inconsistent. Reject the assumption: $T \nvdash \sim\mathcal{R}$.

In reasoning for T13.4, from $T \vdash \sim\mathcal{G}$ and the diagonal result we had $\exists v Prft(v, \overline{\mathsf{g}})$, but no way to convert it to a contradiction with $\sim Prft(\overline{\mathsf{0}}, \overline{\mathsf{g}}), \sim Prft(\overline{\mathsf{1}}, \overline{\mathsf{g}}), \ldots$ without the appeal to ω-consistency. We can, however, move from $\sim Prft(\overline{\mathsf{0}}, \overline{\mathsf{r}}), \sim Prft(\overline{\mathsf{1}}, \overline{\mathsf{r}}), \ldots, \sim Prft(\overline{\mathsf{m}}, \overline{\mathsf{r}})$ to a *bounded* quantification $(\forall w \le \overline{\mathsf{m}}) \sim Prft(w, \overline{\mathsf{r}})$. Then the special nature of $\mathcal{R}$ aids the argument: Suppose $j \le \overline{\mathsf{m}}$; from $RPrft(j, \overline{\mathsf{r}})$ it follows that $Prft(j, \overline{\mathsf{r}})$, and we contradict the bounded quantification in the usual way. Suppose $\overline{\mathsf{m}} \le j$; from $RPrft(j, \overline{\mathsf{r}})$ it follows that nothing less than or equal to j (including $\overline{\mathsf{m}}$) numbers a proof of til(r); but from the assumption that $T \vdash \sim\mathcal{R}$ we have $\overline{Prft}(\overline{\mathsf{m}}, \overline{\mathsf{r}})$ and we contradict again. So $T \nvdash \mathcal{R}$ and $T \nvdash \sim\mathcal{R}$.

Let us close this section with some reflections on what we have shown: First, from the semantic argument a s̃ound recursively axiomatized theory whose language includes $\mathcal{L}_{NT}$ is ĩncomplete; from the syntactic argument a consistent recursively axiomatized theory extending Q is ĩncomplete. Both apply to recursively axiomatized theories. The arguments work because a theory whose language includes $\mathcal{L}_{NT}$ expresses the recursive functions; and a theory extending Q captures the recursive functions. So the semantic result requires s̃oundness and expression, and the syntactic requires consistency with capture. For s̃oundness and consistency we have,

$$T \text{ is s̃ound} \Longrightarrow T \text{ is } \omega\text{-consistent} \Longrightarrow T \text{ is consistent}$$

So our results are progressively stronger as the assumptions have become correspondingly weaker. But for expression and capture,

$$\text{capture} \Longrightarrow \text{expression}$$

So the requirement is increased as we move from expression to capture. These relations serve to locate the theories to which our results apply: If a recursively axiomatized theory is s̃ound and it's language expresses the (primitive) recursive functions then it is ĩncomplete, and if a recursively axiomatized theory is consistent and captures the recursive functions then it is ĩncomplete. But when a theory does not meet these conditions, we will not have shown that it is ĩncomplete.[2]

Second, we have not shown that there are truths of $\mathscr{L}_{NT}$ not provable in any recursively axiomatized consistent theory extending Q. Rather, what we have shown is that for any recursively axiomatized consistent theory extending Q, there are some truths of $\mathscr{L}_{NT}$ not provable in that theory. For a given recursively axiomatized theory, there will be a given relation $\textsf{PRFT}(\mathsf{m},\mathsf{n})$ and $\mathit{Prft}(v, y)$ depending on the particular axioms of that theory—and so unique sentences $\mathscr{G}$ and $\mathscr{R}$ constructed as above. In particular, given that a theory cannot prove, say, $\mathscr{R}$, we might simply *add* $\mathscr{R}$ to its axioms; then of course there is a derivation of $\mathscr{R}$ from the axioms of the revised theory! But then the new theory T' will generate a new relation $\textsf{PRFT}'(\mathsf{m},\mathsf{n})$ and a new $\mathit{Prft}'(v, y)$ and so a new unprovable sentence $\mathscr{R}'$. So any consistent theory extending Q is negation ĩncomplete.

But it is worth a word about what are theories extending Q. Any such theory should build in equivalents of the $\mathscr{L}_{NT}$ vocabulary $\emptyset$, S, $+$, and $\times$—and should have a predicate $\mathit{Nat}(x)$ to identify a class of objects to count as the natural numbers. For Q and PA this predicate *Nat* may just be $x = x$, since the axioms apply to all the members of the intended universe. But there may be cases where the natural numbers are a subset of the domain. Then if the theory makes the axioms of Q true on those objects, it is ĩncomplete. Straightforward extensions of Q are ones like PA which simply add to its axioms. But ordinary ZFC set theory also falls into this category—for it is possible to define a class of sets, say, $\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}, \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}, \ldots,$ where $\mathsf{0}$ is the empty set $\varnothing$ and any successor is the set of all the numbers prior to it, along with operations on sets which obey the axioms of Q.[3] It follows that ZFC is negation ĩncomplete. In contrast, the domain for the theory of real closed fields

[2]The historical order of Q and PA is the reverse of the order in which we have developed them. PA emerged in the late 1800s, and Q only in 1950 (after the 1931 publication of Gödel's result). Q was never supposed to be c̃omplete; rather it was an explicit weakening of PA (and *Principia Mathematica*), sufficient to capture the recursive functions and so to support Gödel's theorem. Thus, again, we locate the theories to which our results apply: the syntactical result applies to recursively axiomatized consistent theories extending Q, but not to ones weaker than Q. (In fact, there are ĩncompletenesss results for systems weaker than Q—but that is is a topic beyond the scope of our discussion.)

[3]For discussion, see any introduction to set theory, for example, Enderton, *Elements of Set Theory*, Chapter 4. See also page 650.

(RCF) includes all the entities required to do arithmetic; however the language of this theory does not have a predicate $Nat(x)$ to pick out the natural numbers, and RCF cannot recapitulate the theory of natural numbers on any subclass of its domain; so our ĩncompleteness theorem does not get a grip—and, in fact, this theory is c̃omplete (compare pages 557 and 565). Observe, though, that it is a *weakness* of RCF, its inability to specify a certain class, that makes room for its c̃ompleteness.

E13.4. Provide the reasoning to show T13.3 and T13.4.

E13.5. For T a recursively axiomatized sound theory whose language includes $\mathscr{L}_{\text{NT}}$, fill out the reasoning mentioned on page 639 to show that $\mathsf{N}[\mathscr{G}] = \mathsf{T}$ iff $T \nvdash \mathscr{G}$, and so that $\mathscr{G}$ is both unprovable and true.

E13.6. Demonstrate T13.5.

13.2 Gödel's Second Theorem: Overview

We turn now to Gödel's second ĩncompleteness theorem on the unprovability of consistency. The discussion is divided into five main sections. First, in this section, Gödel's second theorem is proved subject to three *derivability conditions*. Then we turn to the derivability conditions themselves. The first is easy. But the second and third require extended discussion. There is some background (section 13.3). Then discussion of the second condition (section 13.4), and the third condition (section 13.5). This completes the proof. We conclude with some reflections and consequences from our results (section 13.6). Textbooks ordinarily end their discussion of the second theorem with the demonstration from the derivability conditions, offering just some general perspective on how the conditions are to be obtained.[4] However, even if you decide to bypass details, this general perspective will be enhanced by survey of a particular instance that exhibits what is involved.

The main argument developed in this section applies generally to recursively axiomatized theories extending Q that satisfy the derivability conditions, to show that such a theory cannot prove its own consistency. From the sections that follow, PA is one such theory. The result is that PA and its extensions cannot prove their own

[4]So, for example, Boolos, Burgess, and Jeffrey omit demonstrations with a remark that, "the proofs of the [second and third derivability conditions] are omitted from virtually all books on the level of this one, not because they involve any terribly difficult new ideas, but because the innumerable routine verifications they—and especially the last—require would take up too much time and patience" (*Computability and Logic*, page 234). Smith devotes about three pages (*An Introduction to Gödel's Theorems*, pages 258–60). Gödel himself outlines but does not give a proof of the second theorem in his 1931, "On the Formally Undecidable Propositions of *Principia Mathematica* and Related Systems." The proof was first carried out in 1939 by Hilbert and Bernays, *Grundlagen der Mathematik, Vol II*.

consistency. The reason for the switch to PA will become vivid in demonstration of the derivability conditions.[5]

Main argument. We have seen that for recursively axiomatized theories there is a recursive relation $\mathsf{PRFT}(\mathsf{m}, \mathsf{n})$. Since it is recursive, in theories extending Q, this relation is captured by a corresponding $\mathit{Prft}(v, y)$. Let,

$$\mathit{Prvt}(y) = \exists v \mathit{Prft}(v, y)$$

So $\mathit{Prvt}(y)$ just when something numbers a proof of the formula numbered y—when the formula numbered by y is provable.[6] Insofar as the quantifier is unbounded, there is no suggestion that there is a corresponding recursive relation—in fact, we have seen from T12.21 that no recursive relation is true just of numbers for the theorems of Q. Let,

$$\mathit{Cont} = \sim\mathit{Prvt}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$$

So *Cont* is true just in case there is no proof of $\overline{0} = \overline{1}$. There are different ways to express consistency, but for theories extending Q this does as well as any other. Let T extend Q. (i) Suppose $T \vdash \overline{0} = \overline{1}$; since T extends Q, $T \vdash \overline{0} \neq \overline{1}$; so $T \vdash \overline{0} = \overline{1}$ and $T \vdash \overline{0} \neq \overline{1}$; so T is inconsistent. (ii) Suppose T is inconsistent; then it proves anything; so $T \vdash \overline{0} = \overline{1}$. So $T \vdash \overline{0} = \overline{1}$ iff T is inconsistent; and, transposing, T is consistent iff $T \nvdash \overline{0} = \overline{1}$. So T is consistent iff *Cont* is true (for further discussion, see section 13.6.1). Notice that consistency sentences vary with the provability predicate—so instances of *Cont* are *Conq* for Q and *Conpa* for PA.

Gödel's second ĩncompleteness theorem is this simple result: Under certain conditions, if T is consistent, then $T \nvdash \mathit{Cont}$. If it is consistent, then T cannot prove its own consistency. Suppose the first incompleteness theorem (T13.4) applies to T, and suppose we could show,

$$(**) \qquad T \vdash \mathit{Cont} \rightarrow \sim\mathit{Prvt}(\overline{\ulcorner \mathscr{G} \urcorner})$$

Then, given what has gone before, we could make the following very simple argument. Suppose T is a recursively axiomatized theory extending Q.

> By T13.2, $T \vdash \mathscr{G} \leftrightarrow \sim\exists v \mathit{Prft}(v, \overline{\ulcorner \mathscr{G} \urcorner})$, which is to say, $T \vdash \mathscr{G} \leftrightarrow \sim\mathit{Prvt}(\overline{\ulcorner \mathscr{G} \urcorner})$; from this and $(**)$, $T \vdash \mathit{Cont} \rightarrow \mathscr{G}$; so if $T \vdash \mathit{Cont}$ then $T \vdash \mathscr{G}$; but from the first theorem (T13.4), if T is consistent, then $T \nvdash \mathscr{G}$; so if T is consistent, $T \nvdash \mathit{Cont}$.

[5]But the argument goes through for certain theories weaker than PA. Of relevance to Hilbert, it goes through for *primitive recursive arithmetic* (PRA)—whose theorems are like those of a system which adds to the axioms of Q the induction schema but restricted to Π_1 formulas. Though he is not entirely clear, arguably, PRA is Hilbert's real theory *R* (see page 562). We set aside such concerns.

[6]Following Gödel, formula *Prvt* is often labeled '*Bew*'—short for the German word "provable."

So the argument reduces to showing (∗∗). Observe that in reasoning for T13.4 we have already shown,

$$T \text{ is consistent} \Longrightarrow T \nvdash \mathscr{G}$$

So the argument reduces to showing that T *proves* what we have already seen is so. There is nothing mysterious about this: *Cont*, *Prvt*, and the like are formulas, and so just the sort of thing to which our proof apparatus applies. Given the parallel between what has gone before and what we require, it is not surprising that much of what we shall do is (roughly) parallel to what has gone before: Corresponding to the recursive functions of Chapter 12, our idea is to define "coordinate" functions into the theory T and then, within T, demonstrate matching results about them. So we "push" reasoning from the metalanguage into the theory. It is this that motivates the switch from Q to PA—as many of the arguments that would have been by induction are forced into the theory and so are by IN.

Let us abbreviate $Prvt(\overline{\ulcorner \mathscr{P} \urcorner})$ by $\Box\mathscr{P}$. Observe that this obscures the corner quotes. Still, we shall find it useful. So we need $T \vdash Cont \to \sim\Box\mathscr{G}$, which is just to say, $T \vdash \sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}$. Suppose T satisfies the following *derivability conditions:*

D1. If $T \vdash \mathscr{P}$ then $T \vdash \Box\mathscr{P}$

D2. $T \vdash \Box(\mathscr{P} \to \mathscr{Q}) \to (\Box\mathscr{P} \to \Box\mathscr{Q})$

D3. $T \vdash \Box\mathscr{P} \to \Box\Box\mathscr{P}$

Then we shall be able to show $T \vdash Cont \to \sim\Box\mathscr{G}$.

The utility of $\Box$ in this context is that D1–D3 characterize a standard modal logic, K4—and it is not surprising that *provability* should correspond to a kind of necessity. There is an elegant natural derivation system for this modal logic. For this you might check out Roy, "Natural Derivations for Priest" §2 (but in the nomenclature there borrowed from Priest, the system is $NK\tau$). However rather than introduce and explain a new derivation system, we obtain a version of K4 simply by adding D1–D3 to A1–A3 and MP from *ADs*. So K4 has D1 as a new rule, and D2 and D3 as new axioms.[7] Since A1–A3 and MP remain, we have all the theorems from before. Our demonstration of DT does not, however, extend to include the new rule D1; DT is fine for derivations with just A1–A3 and MP; but in any place where the new rule is involved, we set DT to the side. As a simple K4 example, $\vdash_{K4} \Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})$.

(A)

1.	$\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})$	T3.9
2.	$\Box[\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})]$	1 D1
3.	$\Box[\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})] \to [\Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})]$	D2
4.	$\Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})$	3,2 MP

[7]While K4 correctly represents the derivability conditions, it is not a c̊omplete logic of provability. We get a c̊omplete system if we add to K4 a rule according to which from $\Box\mathscr{P} \to \mathscr{P}$ we may infer $\mathscr{P}$. For discussion see section 13.6.2 and Boolos, *The Logic of Provability*.

Now, given that $T \vdash \mathscr{G} \to \sim\exists v Prft(v, \overline{\ulcorner\mathscr{G}\urcorner})$ from T13.2, we shall be able to show that $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

T13.7. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic for provability, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

1.	$\mathscr{G} \to \sim\Box\mathscr{G}$	from T13.2
2.	$\Box(\mathscr{G} \to \sim\Box\mathscr{G})$	1 D1
3.	$\Box(\mathscr{G} \to \sim\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box\sim\Box\mathscr{G})$	D2
4.	$\Box\mathscr{G} \to \Box\sim\Box\mathscr{G}$	3,2 MP
5.	$\Box\sim\Box\mathscr{G} \to \Box(\Box\mathscr{G} \to \overline{0} = \overline{1})$	(A)
6.	$\Box\mathscr{G} \to \Box(\Box\mathscr{G} \to \overline{0} = \overline{1})$	4,5 T3.2
7.	$\Box(\Box\mathscr{G} \to \overline{0} = \overline{1}) \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	D2
8.	$\Box\mathscr{G} \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	6,7 T3.2
9.	$[\Box\mathscr{G} \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))] \to [(\Box\mathscr{G} \to \Box\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))]$	A2
10.	$(\Box\mathscr{G} \to \Box\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	9,8 MP
11.	$\Box\mathscr{G} \to \Box\Box\mathscr{G}$	D3
12.	$\Box\mathscr{G} \to \Box(\overline{0} = \overline{1})$	10,11 MP
13.	$[\Box\mathscr{G} \to \Box(\overline{0} = \overline{1})] \to [\sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}]$	T3.13
14.	$\sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}$	13,12 MP

So $T \vdash \sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}$ which is to say, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

As usual, reasoning from the axiomatic derivation is not especially intuitive. Still, with the derivability conditions, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$. Given this, reason as before:

T13.8. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions, if T is consistent, $T \nvdash Cont$.

Suppose T is a recursively axiomatized theory extending Q that satisfies the derivability conditions. Then by T13.7, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$; and by T13.2, $T \vdash \mathscr{G} \leftrightarrow \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$; so $T \vdash Cont \to \mathscr{G}$; so if $T \vdash Cont$ then $T \vdash \mathscr{G}$; but from the first ĩncompleteness theorem T13.4, if T is consistent, then $T \nvdash \mathscr{G}$; so if T is consistent, $T \nvdash Cont$.

One might wonder about the significance of this theorem: If T were inconsistent, it *would* prove *Cont*. Further, suppose T is a recursively axiomatized theory extending Q that satisfies the derivability conditions: From the theorem, if T is consistent then $T \nvdash Cont$—so, transposing, if $T \vdash Cont$ then T is inconsistent! So a failure to prove *Cont* is no reason to think that T is inconsistent. The interesting point here results from using one theory to prove the consistency of another. Recall the main Hilbert strategy as outlined in the introduction to Part IV; a key component is the demonstration by means of some real theory R that an ideal theory I is consistent. But supposing that PA (say) cannot prove its own consistency, we can be sure that no *weaker* theory can prove the consistency of PA. And if PA cannot prove even the consistency of PA, then PA and theories weaker than PA cannot be used to prove the consistency of theories *stronger* than PA.[8] So a leg of the Hilbert strategy seems to be removed. Observe, however, that the theorem does not show that the consistency of PA is unprovable: A theory stronger than PA at least in some respects might still prove the consistency of PA. So for example, we might consider a theory PA* like PA but with the addition of *Conpa* as an axiom. Then trivially PA* proves *Conpa*. Of course, as a means of demonstrating the consistency of PA such an argument assumes that which is to be shown. More seriously, G. Gentzen proves the consistency of PA by a theory that is not a simple extension of PA (Gentzen, "Consistency of Number Theory" and "New Version of the Consistency Proof"). But related concerns apply. A non-question-begging demonstration of the consistency of PA by a strengthened theory requires some reason for accepting the s̃oundness of the stronger theory that is not already a reason to think that PA is consistent.[9]

Another theorem is easy to show, and left as an exercise.

T13.9. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic for provability, $T \vdash Cont \leftrightarrow \sim Prvt(\overline{\ulcorner Cont \urcorner})$.

Hints: (i) Show that $T \vdash Cont \rightarrow \sim\Box Cont$; you can do this starting with $Cont \rightarrow \sim\Box\mathcal{G}$ from T13.7 and $\sim\Box\mathcal{G} \rightarrow \mathcal{G}$ from T13.2. Then (ii) show $T \vdash \sim\Box Cont \rightarrow Cont$; for this, use T6.46 with T3.9 to show $T \vdash \overline{0} = \overline{1} \rightarrow Cont$; then you should be able to obtain $\sim\Box Cont \rightarrow \sim\Box(\overline{0} = \overline{1})$ which is to say $\sim\Box Cont \rightarrow Cont$.

From this theorem, supposing the derivability conditions, *Cont* is another $\mathcal{P}$ which, like $\mathcal{G}$, is such that $T \vdash \mathcal{P} \leftrightarrow \sim Prvt(\overline{\ulcorner \mathcal{P} \urcorner})$; so *Cont* is another fixed point for $\sim Prvt(x)$. It follows that *Cont* is another sentence such that both it and its negation are unprovable. Interestingly, *Cont* uses the notion of provability, but is not constructed so as to say anything about its *own* provability—and so this instance of ĩncompleteness does not depend on self-reference for the unprovable sentence.

Proving the conditions. We have shown that the second theorem holds for a theory if it meets the derivability conditions. But this is not to show that the theorem holds for any theories! In order to tie the result to something concrete, we turn now to showing that PA meets the derivability conditions, and so that PA satisfies the theorem. It will be clear how to extend the result to recursively axiomatized theories extending PA. Coinciding with the move to PA we revert to considering original rather than

[8] And the same goes for Hilbert's PRA (see note 5 on page 644).

[9] But this is not the end of the matter. See for example Detlefsen, "Interpreting Gödel's Second Theorem" and *Hilbert's Program*. Again, this is a topic in philosophy of mathmatics (compare page 563 note 3).

canonical formulas: this avoids some complication—and, insofar as results for capture and then the first ĩncompleteness theorem go through with cannonical formulas in theories extending $\mathrm{Q_s}$ (and PA is a theory extending $\mathrm{Q_s}$) all those results are preserved for us (see section 12.3.2).

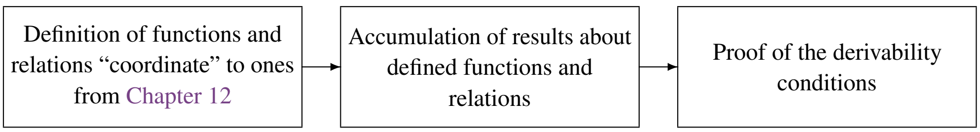
Demonstration of the first derivability condition is simple.

T13.10. Suppose T is a recursively axiomatized theory extending $\mathrm{Q_s}$. Then if $T \vdash \mathcal{P}$, then $T \vdash \Box\mathcal{P}$.

Suppose T is a recursively axiomatized theory extending $\mathrm{Q_s}$ and $T \vdash \mathcal{P}$; since T is recursively axiomatized, for some m, $\textsc{prft}(\mathsf{m}, \ulcorner\mathcal{P}\urcorner)$; and since T extends $\mathrm{Q_s}$, by T12.9 there is a(n original) $Prft$ that captures $\textsc{prft}$; so $T \vdash Prft(\overline{\mathsf{m}}, \overline{\ulcorner\mathcal{P}\urcorner})$; so by $\exists$I, $T \vdash \exists x Prft(x, \overline{\ulcorner\mathcal{P}\urcorner})$; so $T \vdash Prvt(\overline{\ulcorner\mathcal{P}\urcorner})$; so $T \vdash \Box\mathcal{P}$.

$\mathrm{Q_s}$ strengthens Q insofar as it proves uniqueness of remainder. But, as for *Def*[*rm*] on page 661, PA is a theory that proves uniqueness of remainder—so PA qualifies as a recursively axiomatized theory extending $\mathrm{Q_s}$. So (D1), if PA $\vdash \mathcal{P}$ then PA $\vdash \Box\mathcal{P}$.

Proving the second and third conditions is considerably more difficult. As remarked above, much of what we shall do is (roughly) parallel to reasoning applied to recursive functions: we define coordinate functions into PA and demonstrate parallel results about them. Thus reasoning includes the following stages:



We begin with definitions. Then we accumulate a series of results about the defined notions that finally put us in a position to demonstrate the derivability conditions themselves. Section 13.3 develops the first box. Then section 13.4 develops boxes two and three with respect to the second condition and, building on that, section 13.5 with respect to the third condition.

E13.7. Using corner quotes and overlines, unabbreviate $\Box\mathcal{P} \to \Box\Box\mathcal{P}$.

E13.8. Show that $\vdash_{K4} \Box(\mathcal{P} \wedge \mathcal{Q}) \to (\Box\mathcal{P} \wedge \Box\mathcal{Q})$. Hint: As a preliminary result, use T9.4 to show $\mathcal{A} \to \mathcal{B}, \mathcal{A} \to \mathcal{C} \vdash_{ADs} \mathcal{A} \to (\mathcal{B} \wedge \mathcal{C})$.

E13.9. (a) Produce derivations to show both directions of the biconditional in T13.9. (b) Use your result to demonstrate that T is negation ĩncomplete—that if T is recursively axiomatized theory extending Q that satisfies the derivability conditions, then if T is consistent, $T \nvdash Cont$, and if T is ω-consistent, $T \nvdash {\sim}Cont$.

Additional Theorems of PA

*T13.11. *The following are theorems of PA.*

(a) PA $\vdash (r \leq s \wedge s \leq t) \rightarrow r \leq t$
(b) PA $\vdash (r < s \wedge s < t) \rightarrow r < t$
(c) PA $\vdash (r < s \wedge s \leq t) \rightarrow r < t$
(d) PA $\vdash (r \leq s \wedge s < t) \rightarrow r < t$
(e) PA $\vdash \emptyset \leq t$
(f) PA $\vdash \emptyset < St$
(g) PA $\vdash \emptyset \neq t \leftrightarrow \emptyset < t$
(h) PA $\vdash \overline{1} \neq St \leftrightarrow \emptyset < t$
(i) PA $\vdash \emptyset < t \rightarrow \exists w(t = Sw)$ $\quad$ w not in t.
(j) PA $\vdash t < St$
(k) PA $\vdash s \leq t \leftrightarrow Ss \leq St$
(l) PA $\vdash s < t \leftrightarrow Ss < St$
(m) PA $\vdash s < t \leftrightarrow Ss \leq t$
(n) PA $\vdash s \leq t \leftrightarrow s < t \vee s = t$
(o) PA $\vdash s < St \leftrightarrow s < t \vee s = t$
(p) PA $\vdash s < St \leftrightarrow s \leq t$
(q) PA $\vdash s \leq St \leftrightarrow s \leq t \vee s = St$
(r) PA $\vdash s < t \vee s = t \vee t < s$
(s) PA $\vdash s \leq t \vee t < s$
(t) PA $\vdash t < s \rightarrow t \neq s$
(u) PA $\vdash s \leq t \leftrightarrow t \not< s$
(v) PA $\vdash (s \leq t \wedge t \leq s) \rightarrow s = t$
(w) PA $\vdash s \leq s + t$
(x) PA $\vdash r \leq s \leftrightarrow r + t \leq s + t$
(y) PA $\vdash r < s \leftrightarrow r + t < s + t$
(z) PA $\vdash (q \leq r \wedge s \leq t) \rightarrow q + s \leq r + t$
(aa) PA $\vdash (q < r \wedge s \leq t) \rightarrow q + s < r + t$
(ab) PA $\vdash \emptyset < t \rightarrow s \leq s \times t$
(ac) PA $\vdash r \leq s \rightarrow r \times t \leq s \times t$
(ad) PA $\vdash \emptyset < r \times s \rightarrow \emptyset < s$
(ae) PA $\vdash (\overline{1} < r \wedge \emptyset < s) \rightarrow s < r \times s$
(af) PA $\vdash (\emptyset < t \wedge r < s) \rightarrow r \times t < s \times t$
(ag) PA $\vdash (\forall x \leq y)\exists v \mathcal{F}xv \leftrightarrow \exists z(\forall x \leq y)(\exists v \leq z)\mathcal{F}xv$
(ah) PA $\vdash \forall x[(\forall z < x)\mathcal{P}^x_z \rightarrow \mathcal{P}] \rightarrow \forall x \mathcal{P}$ $\quad$ *strong induction (a)*
(ai) PA $\vdash (\mathcal{P}^x_\emptyset \wedge \forall x[(\forall z \leq x)\mathcal{P}^x_z \rightarrow \mathcal{P}^x_{Sx}]) \rightarrow \forall x \mathcal{P}$ $\quad$ *strong induction (b)*
(aj) PA $\vdash \exists x \mathcal{P} \rightarrow \exists x[\mathcal{P} \wedge (\forall z < x)\sim\mathcal{P}^x_z]$ $\quad$ *least number principle*

Demonstrations for (a) – (af) were E6.37. The final four are E13.10. Except for (r) and a couple at the end, these are simple arguments that draw upon definitions and prior theorems, without separate application of PA7 or IN.

*E13.10. Show T13.11ag. Then without separate appeal to PA7 (or IN) show that PA proves conditionals (i) from PA7 (applied to $(\forall z < x)\mathcal{P}(z)$) to ah; (ii) from ah (applied to $\sim\mathcal{P}(x)$) to aj; (iii) from aj (applied to $\sim\mathcal{P}(x)$) to ai; and (iv) from ai to PA7. You may appeal to prior theorems from Chapter 6 and T13.11. Thus, in the context of prior results to which you appeal, these principles are connected in a "loop" so that each follows from the others. And, of course, this proves each of T13.11ah, ai, aj in PA.

13.3 The Derivability Conditions: Definition

Our aim in this section is to show that PA defines function and relation symbols corresponding to functions and relations from Chapter 12. We begin with some remarks on what is *required* to introduce function and relation symbols into PA. Then we turn to showing that PA in fact defines functions and relations corresponding to the recursive functions and relations of Chapter 12.

13.3.1 Remarks on Definition

In theories extending Q_s, a recursive function $\mathsf{rec}(\vec{\mathsf{x}})$ is captured by an original formula $Rec(\vec{x}, y)$. Now we shall want a defined function symbol $\mathscr{rec}(\vec{x})$ that is matched to $Rec(\vec{x}, y)$ so that PA $\vdash y = \mathscr{rec}(\vec{x}) \leftrightarrow Rec(\vec{x}, y)$. Then we shall be able to operate on the term $\mathscr{rec}(\vec{x})$ very much as upon the recursive $\mathsf{rec}(\vec{\mathsf{x}})$. Up to this point, we have taken a language, as $\mathcal{L}_q$ or $\mathcal{L}_{NT}$, as basic and introduced any additional symbols, for example $\exists$ or $\leq$, as means of abbreviation for expressions in the original language. But in the present context it will be convenient to *extend* the language by the definition of new symbols.

So, for example, given a theory T in language $\mathcal{L}$, we might introduce symbols and corresponding axioms to obtain T' and $\mathcal{L}'$ as follows:

Symbol	*Axiom*	*Condition*
$\leq$	$x \leq y \leftrightarrow \exists z(z + x = y)$	
$\varnothing$	$y = \varnothing \leftrightarrow \forall x(x \notin y)$	$T \vdash \exists! y \forall x(x \notin y)$
S	$y = Sx \leftrightarrow \forall z[z \in y \leftrightarrow (z \in x \vee z = x)]$	$T \vdash \exists! y \forall z[z \in y \leftrightarrow (z \in x \vee z = x)]$

We are familiar with the first case. So far, we have thought of this as an *abbreviation*—and as such the listed axiom is of the sort $\mathcal{Q}' \leftrightarrow \mathcal{Q}$ with the abbreviated form on one side, and the unabbreviated on the other. A theory is not extended by the addition of an "axiom" of this sort. But it is possible to see the symbol as *new* vocabulary. In all three cases T' includes an axiom to define the symbol. The last two require also a uniqueness condition in the original T. For these, let $\exists! y\mathcal{P}(y)$ abbreviate $\exists y[\mathcal{P}(y) \wedge \forall z(\mathcal{P}(z) \to z = y)]$ or equivalently $\exists y\mathcal{P}(y) \wedge \forall y \forall z[(\mathcal{P}(y) \wedge \mathcal{P}(z)) \to$

$y = z]$ so that *exactly one* thing is $\mathcal{P}$. Then the cases for a constant and function symbol are standard examples from set theory, where zero and successor are defined (taken together, these work as described on page 642). Details of the examples are not important; examples are meant only to illustrate the idea of definition. We begin with conditions under which new vocabulary is introduced, and turn to some basic applications.

Conditions for Definition

Consider some theory T and language $\mathcal{L}$. We will consider a language $\mathcal{L}'$ extended with some new symbol and theory T' extended with a corresponding axiom. There are separate cases for a relation symbol, constant symbol, and function symbol.

Relation symbol. To introduce a new relation symbol $\mathcal{R}\vec{x}$ we require an axiom in the extended theory such that,

$$T' \vdash \mathcal{R}(\vec{x}) \leftrightarrow \mathcal{Q}(\vec{x})$$

where $\mathcal{Q}(\vec{x})$ is in $\mathcal{L}$. So $\mathcal{R}$ is defined by formula $\mathcal{Q}$. Then for a formula $\mathcal{F}'$ including the new symbol, there should be a conversion $\mathfrak{C}$ such that $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}$ for $\mathcal{F}$ in the original $\mathcal{L}$ and,

$$T' \vdash \mathcal{F}' \qquad \text{iff} \qquad T \vdash \mathfrak{C}[\mathcal{F}']$$

So $\mathfrak{C}[\mathcal{F}']$ is like our unabbreviated formula, always available in the original T when $\mathcal{F}'$ is a theorem of T'. The conversion for a relation $\mathcal{R}(\vec{\mathfrak{z}})$ is straightforward. For arbitrary $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, say $\mathcal{A}^{\mathcal{B}}_{\mathcal{C}}$ replaces each instance of $\mathcal{B}$ in $\mathcal{A}$ with $\mathcal{C}$. Then make sure the bound variables of $\mathcal{Q}$ do not overlap the variables of $\vec{\mathfrak{z}}$—so that from the axiom $T' \vdash \mathcal{R}(\vec{\mathfrak{z}}) \leftrightarrow \mathcal{Q}(\vec{\mathfrak{z}})$—and set $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}(\vec{\mathfrak{z}})}_{\mathcal{Q}(\vec{\mathfrak{z}})}$. Thus, from the example above,

$$T' \vdash x \leq y \leftrightarrow \exists z(z + x = y)$$

Suppose $\mathcal{F}' = \forall z(a \leq z)$. Then we want to instantiate x and y from the axiom to a and z. But z is not free for y in the axiom. We solve the problem by revising bound variables; so $T' \vdash x \leq y \leftrightarrow \exists w(w + x = y)$ and then $T' \vdash a \leq z \leftrightarrow \exists w(w + a = z)$. So $\mathfrak{C}[\mathcal{F}']$ replaces $(a \leq z)$ in $\mathcal{F}'$ with $\exists w(w + a = z)$ to obtain $\forall z \exists w(w + a = z)$.

Constant symbol. To introduce a new constant symbol we require an axiom in the extended theory, along with a condition in the original theory such that,

$$T' \vdash y = c \leftrightarrow \mathcal{Q}(y) \qquad \text{and} \qquad T \vdash \exists! y \mathcal{Q}(y)$$

where $\mathcal{Q}(y)$ is in $\mathcal{L}$. Again for a formula $\mathcal{F}'$ including the new symbol, we expect a conversion $\mathfrak{C}$ such that $\mathfrak{C}[\mathcal{F}']$ is a formula of $\mathcal{L}$, and $T' \vdash \mathcal{F}'$ iff $T \vdash \mathfrak{C}[\mathcal{F}']$. Let z be a variable that does not appear in $\mathcal{F}'$ or $\mathcal{Q}$. Then from the axiom, $T' \vdash z = c \leftrightarrow \mathcal{Q}(z)$ and,

$$\mathfrak{C}[\mathcal{F}'] = \exists z(\mathcal{Q}(z) \wedge \mathcal{F}'^{c}_{z})$$

So, from the example above, we are given $T' \vdash y = \varnothing \leftrightarrow \forall x(x \notin y)$; suppose $\mathcal{F}' = \exists y(\varnothing \in y)$. Then z is a variable that does not appear in $\mathcal{F}'$ or in $\mathcal{Q}(y) = \forall x(x \notin y)$. So $T' \vdash z = \varnothing \leftrightarrow \forall x(x \notin z)$ and $\mathfrak{C}[\mathcal{F}'] = \exists z[\forall x(x \notin z) \wedge \exists y(z \in y)]$.

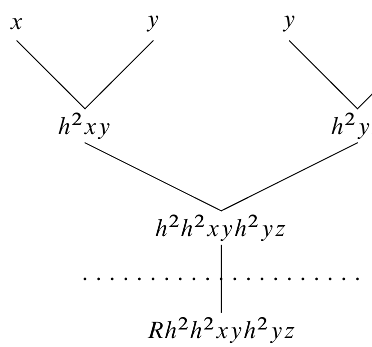
Function symbol. To introduce a function symbol, there is an axiom and condition,

$$T' \vdash y = \mathscr{h}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y) \qquad \text{and} \qquad T \vdash \exists! y\mathcal{Q}(\vec{x}, y)$$

Begin with an atomic $\mathcal{R}'$. When a single instance of $\mathscr{h}\vec{\mathfrak{t}}$ appears in $\mathcal{R}'$, the conversion for a function symbol works like that for constants: Again, make sure the bound variables of $\mathcal{Q}$ do not overlap the variables of $\vec{\mathfrak{t}}$ and let z be a variable that does not appear in $\mathcal{R}'$ or in $\mathcal{Q}$. Then it is sufficient to set $\mathfrak{C}[\mathcal{R}'] = \exists z(\mathcal{Q}(\vec{\mathfrak{t}}, z) \wedge \mathcal{R}'^{\mathscr{h}\vec{\mathfrak{t}}}_{z})$, and for arbitrary $\mathcal{F}'$, $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}'}_{\mathfrak{C}[\mathcal{R}']}$. In general, however, $\mathcal{R}'$ may include multiple instances of $\mathscr{h}$, including one in the scope of another. In this case, we replace instances of the function symbol beginning with ones that have widest scope. Begin where $\mathcal{R}' = \mathcal{R}t_1 \ldots t_n$ and $t_1 \ldots t_n$ may involve instances of $\mathscr{h}\vec{\mathfrak{t}}$. Order instances of $\mathscr{h}\vec{\mathfrak{t}}$ in $\mathcal{R}'$ from the left (or, on a Chapter 2 tree, from the bottom) into a list $\mathscr{h}\vec{\mathfrak{t}}_1, \mathscr{h}\vec{\mathfrak{t}}_2, \ldots, \mathscr{h}\vec{\mathfrak{t}}_m$, so that when $i < j$, no $\mathscr{h}\vec{\mathfrak{t}}_i$ appears in the scope of $\mathscr{h}\vec{\mathfrak{t}}_j$. Then set $\mathcal{R}_0 = \mathcal{R}'$, and for $i \geq 1$ and some new variable z, $\mathcal{R}_i = \exists z(\mathcal{Q}(\vec{\mathfrak{t}}_i, z) \wedge (\mathcal{R}_{i-1})^{\mathscr{h}\vec{\mathfrak{t}}_i}_{z})$. Then $\mathfrak{C}[\mathcal{R}'] = \mathcal{R}_m$ and for arbitrary $\mathcal{F}'$, $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}'}_{\mathcal{R}_m}$.

In case a single replacement is made, this is no different than before. For a case with more than one replacement, consider $\mathcal{R}' = \mathcal{R}_0 = Rh^2h^2xyh^2yz$; then the tree is as follows:

(B)



So instances of $\mathscr{h}\mathscr{q}\mathscr{r}$ are ordered $\langle h^2h^2xyh^2yz, h^2xy, h^2yz\rangle$. Make sure the bound variables of $\mathcal{Q}(\vec{x}, y)$ do not overlap the variables of $\mathcal{R}'$. In this example $\mathscr{h}$ is a two-place function symbol so that $\vec{x}$ consists of some variables m, n and the axiom is of the sort $y = \mathscr{h}(m, n) \leftrightarrow \mathcal{Q}(m, n, y)$. And we use $\mathcal{Q}$ to replace instances of $\mathscr{h}$, working our way up through the tree. So,

$$\mathcal{R}_0 = Rh^2h^2xyh^2yz$$

$$\mathcal{R}_1 = \exists u[\mathcal{Q}h^2xyh^2yzu \wedge Ru]$$

$$\mathcal{R}_2 = \exists v(\mathcal{Q}xyv \wedge \exists u[\mathcal{Q}vh^2yzu \wedge Ru])$$

$$\mathcal{R}_3 = \exists w[\mathcal{Q}yzw \wedge \exists v(\mathcal{Q}xyv \wedge \exists u[\mathcal{Q}vwu \wedge Ru])]$$

$\mathcal{R}_1$ uses $\mathcal{Q}$ to replace all of $h^2h^2xyh^2yz$, operating on the terms h^2xy and h^2yz; $\mathcal{R}_2$ uses $\mathcal{Q}$ to replace h^2xy in $\mathcal{R}_1$ operating on the terms x and y; and $\mathcal{R}_3$ uses $\mathcal{Q}$ to replace h^2yz in $\mathcal{R}_2$ operating on the terms y and z. Observe that free variables of the result are the same as in $\mathcal{R}'$.

To show that our definitions work, that $T' \vdash \mathcal{F}'$ iff $T \vdash \mathfrak{C}[\mathcal{F}']$ we need a couple of theorems. The first establishes a connection between $\mathcal{F}'$ and $\mathfrak{C}[\mathcal{F}']$ within T'.

T13.12. For some defined symbol, with its associated axiom and conversion procedure, $T' \vdash \mathcal{F}' \leftrightarrow \mathfrak{C}[\mathcal{F}']$.

(r) For a relation symbol, $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}(\vec{\mathfrak{z}})}_{\mathcal{Q}(\vec{\mathfrak{z}})}$; and we are given $T' \vdash \mathcal{R}(\vec{x}) \leftrightarrow \mathcal{Q}(\vec{x})$. Revise bound variables of $\mathcal{Q}$ so that they do not overlap the variables of $\vec{\mathfrak{z}}$; then $\vec{\mathfrak{z}}$ is free for $\vec{x}$ in $\mathcal{Q}$, so $T' \vdash \mathcal{R}(\vec{\mathfrak{z}}) \leftrightarrow \mathcal{Q}(\vec{\mathfrak{z}})$; so with T9.9, $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}'^{\mathcal{R}(\vec{\mathfrak{z}})}_{\mathcal{Q}(\vec{\mathfrak{z}})}$; so $T' \vdash \mathcal{F}' \leftrightarrow \mathfrak{C}[\mathcal{F}']$.

(c) The case for constants is left as an exercise. In this case, $\mathfrak{C}[\mathcal{F}'] = \exists z(\mathcal{Q}(z) \wedge \mathcal{F}'^{c}_{z})$; and we are given $T' \vdash y = c \leftrightarrow \mathcal{Q}(y)$. Reasoning simplifies that for the case (f) that follows.

(f) For an atomic $\mathcal{R}' = \mathcal{R}_0$, function symbol $\mathscr{h}$, and sequence $\mathcal{R}_0 \dots \mathcal{R}_m$ that replaces instances of $\mathscr{h}$ from $\mathcal{R}'$, $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}'}_{\mathcal{R}_m}$; and we are given that $T' \vdash y = \mathscr{h}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)$. Where $\mathcal{R}[\mathscr{h}(\vec{\mathfrak{z}})]$ has some term $\mathscr{h}\vec{\mathfrak{z}}$ and $\mathcal{R}[z]$ replaces that instance of $\mathscr{h}\vec{\mathfrak{z}}$ with z, begin showing equivalence between each member of the sequence and the next—that $T' \vdash \mathcal{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})] \leftrightarrow \mathcal{R}_i(\vec{\mathfrak{z}})$, where $\mathcal{R}_i(\vec{\mathfrak{z}}) = \exists z(\mathcal{Q}(\vec{\mathfrak{z}}, z) \wedge \mathcal{R}_{i-1}[z])$. For this, see the derivation in the box on the next page. Thus for members of the sequence, $T' \vdash \mathcal{R}_{i-1} \leftrightarrow \mathcal{R}_i$; and by repeated applications of this result, $T' \vdash \mathcal{R}' \leftrightarrow \mathcal{R}_m$; so with T9.9, $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}'^{\mathcal{R}'}_{\mathcal{R}_m}$; so $T' \vdash \mathcal{F}' \leftrightarrow \mathfrak{C}[\mathcal{F}']$.

So far, so good, but this only says what the extended T' proves—that the richer T' proves $\mathfrak{C}[\mathcal{F}']$ iff it proves $\mathcal{F}'$. But we want to see that the original T proves $\mathfrak{C}[\mathcal{F}']$ iff T' proves $\mathcal{F}'$. We bridge between T' and T by a semantic theorem that, together with ^{s}oundness and ^{c}ompleteness, yields the desired result. Recall that $\mathfrak{C}[\mathcal{F}']$ is a formula in language $\mathcal{L}$.

T13.13. Consider T in language $\mathscr{L}$ and T' with some defined symbol, axiom, and condition; then for any formula $\mathcal{F}$ in $\mathscr{L}$, $T' \vDash \mathcal{F}$ iff $T \vDash \mathcal{F}$.

Since the entailments of T' include all the entailments of T, the direction from right to left is obvious. So suppose $T' \vDash \mathcal{F}$. To show $T \vDash \mathcal{F}$, consider an arbitrary model M such that $\mathsf{M}[T] = \mathsf{T}$; our aim is to show $\mathsf{M}[\mathcal{F}] = \mathsf{T}$, and so that $T \vDash \mathcal{F}$.

(r) Relation symbol. Extend M to a model M′ like M except that for arbitrary d, $\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n] \rangle \in \mathsf{M}'[\mathcal{R}]$ iff $\mathsf{M_d}[\mathcal{Q}(x_1 \ldots x_n)] = \mathsf{S}$; iff $\mathsf{M'_d}[\mathcal{Q}(x_1 \ldots x_n)] = \mathsf{S}$ (the latter by T10.14 since M and M′ agree on assignments to symbols in $\mathcal{Q}$). Since M′ and M agree on assignments to symbols other than $\mathcal{R}$, by T10.14 $\mathsf{M}'[T] = \mathsf{T}$. And $\mathsf{M}'[\mathcal{R}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x})] = \mathsf{T}$: suppose otherwise; then by TI there is some d such that $\mathsf{M'_d}[\mathcal{R}x_1 \ldots x_n \leftrightarrow \mathcal{Q}(x_1 \ldots x_n)] \neq \mathsf{S}$; so by SF′($\leftrightarrow$), $\mathsf{M'_d}[\mathcal{R}x_1 \ldots x_n] \neq \mathsf{S}$ and $\mathsf{M'_d}[\mathcal{Q}(x_1 \ldots x_n)] = \mathsf{S}$ (or the other way around); from the former by SF(r), $\langle \mathsf{M'_d}[x_1] \ldots \mathsf{M'_d}[x_n] \rangle \notin \mathsf{M}'[\mathcal{R}]$; so $\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n] \rangle \notin \mathsf{M}'[\mathcal{R}]$; so by construction, $\mathsf{M'_d}[\mathcal{Q}(x_1 \ldots x_n)] \neq \mathsf{S}$; this is impossible, and similarly in the other case; reject the assumption: $\mathsf{M}'[\mathcal{R}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x})] = \mathsf{T}$. So $\mathsf{M}'[T'] = \mathsf{T}$; so since $T' \vDash \mathcal{F}$, $\mathsf{M}'[\mathcal{F}] = \mathsf{T}$; and by T10.14 again, $\mathsf{M}[\mathcal{F}] = \mathsf{T}$.

T13.12(f)

1.	$y = h\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)$	from T'
2.	$\quad \mathcal{R}_{i-1}[h(\vec{\mathfrak{z}})]$	A (g, $\leftrightarrow$I)
3.	$\quad h(\vec{\mathfrak{z}}) = h(\vec{\mathfrak{z}}) \leftrightarrow \mathcal{Q}(\vec{\mathfrak{z}}, h(\vec{\mathfrak{z}}))$	from 1
4.	$\quad h\vec{\mathfrak{z}} = h\vec{\mathfrak{z}}$	=I
5.	$\quad \mathcal{Q}(\vec{\mathfrak{z}}, h(\vec{\mathfrak{z}}))$	3,4 $\leftrightarrow$E
6.	$\quad \mathcal{Q}(\vec{\mathfrak{z}}, h(\vec{\mathfrak{z}})) \wedge \mathcal{R}_{i-1}[h(\vec{\mathfrak{z}})]$	5,2 $\wedge$I
7.	$\quad \exists z(\mathcal{Q}(\vec{\mathfrak{z}}, z) \wedge \mathcal{R}_{i-1}[z])$	6 $\exists$I
8.	$\quad \mathcal{R}_i(\vec{\mathfrak{z}})$	7 abv
9.	$\quad \mathcal{R}_i(\vec{\mathfrak{z}})$	A (g, $\leftrightarrow$I)
10.	$\quad \exists z(\mathcal{Q}(\vec{\mathfrak{z}}, z) \wedge \mathcal{R}_{i-1}[z])$	9 abv
11.	$\quad\quad \mathcal{Q}(\vec{\mathfrak{z}}, j) \wedge \mathcal{R}_{i-1}[j]$	A (g, 10$\exists$E)
12.	$\quad\quad \mathcal{Q}(\vec{\mathfrak{z}}, j)$	11 $\wedge$E
13.	$\quad\quad j = h(\vec{\mathfrak{z}}) \leftrightarrow \mathcal{Q}(\vec{\mathfrak{z}}, j)$	from 1
14.	$\quad\quad j = h(\vec{\mathfrak{z}})$	13,12 $\leftrightarrow$E
15.	$\quad\quad \mathcal{R}_{i-1}[j]$	11 $\wedge$E
16.	$\quad\quad \mathcal{R}_{i-1}[h(\vec{\mathfrak{z}})]$	15,14 =E
17.	$\quad \mathcal{R}_{i-1}[h(\vec{\mathfrak{z}})]$	10,11-16 $\exists$E
18.	$\mathcal{R}_{i-1}[h(\vec{\mathfrak{z}})] \leftrightarrow \mathcal{R}_i(\vec{\mathfrak{z}})$	2-8,9-17 $\leftrightarrow$I

Lines (3) and (13) are from (1) by $\forall$I and then $\forall$E. Things are arranged so that the variables of $\vec{\mathfrak{z}}$ are not bound upon substitution into $\mathcal{Q}$. So instances of the axiom at (3) and (13) along with $\exists$I and =E at (7) and (16) satisfy constraints.

(c) The case for constants is left as an exercise.

(f) Function symbol. Extend M to a model M' like M except that for arbitrary d, $\langle\langle \mathsf{d}[x_1]\ldots\mathsf{d}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\mathscr{h}]$ iff $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(x_1\ldots x_n, y)] = \mathsf{S}$; by T10.14 iff $\mathsf{M}'_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(x_1\ldots x_n, y)] = \mathsf{S}$. Since M' and M agree on assignments to symbols other than $\mathscr{h}$, by T10.14 $\mathsf{M}'[T] = \mathsf{T}$. And $\mathsf{M}'[y = \mathscr{h}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)] = \mathsf{T}$: suppose otherwise; then by TI there is some h such that $\mathsf{M}'_{\mathsf{h}}[y = \mathscr{h}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)] \neq \mathsf{S}$; so by SF$'(\leftrightarrow)$, $\mathsf{M}'_{\mathsf{h}}[y = \mathscr{h}\vec{x}] \neq \mathsf{S}$ and $\mathsf{M}'_{\mathsf{h}}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$ (or the other way around). Say $\mathsf{h}[y] = \mathsf{a}$; then $\mathsf{M}'_{\mathsf{h}}[y] = \mathsf{a}$ and $\mathsf{h} = \mathsf{h}(y|\mathsf{a})$; with the latter, $\mathsf{M}'_{\mathsf{h}(y|\mathsf{a})}[\mathcal{Q}(x_1\ldots x_n, y)] = \mathsf{S}$; so by construction $\langle\langle \mathsf{h}[x_1]\ldots\mathsf{h}[x_n]\rangle, \mathsf{a}\rangle \in \mathsf{M}'[\mathscr{h}]$; so $\langle\langle \mathsf{M}'_{\mathsf{h}}[x_1]\ldots\mathsf{M}'_{\mathsf{h}}[x_n]\rangle, \mathsf{a}\rangle \in \mathsf{M}'[\mathscr{h}]$, and by TA(f), $\mathsf{M}'_{\mathsf{h}}[\mathscr{h}x_1\ldots x_n] = \mathsf{a}$; so $\mathsf{M}'_{\mathsf{h}}[y] = \mathsf{a} = \mathsf{M}'_{\mathsf{h}}[\mathscr{h}\vec{x}]$; so $\langle \mathsf{M}'_{\mathsf{h}}[y], \mathsf{M}'_{\mathsf{h}}[\mathscr{h}\vec{x}]\rangle \in \mathsf{M}'[=]$; so by SF(r), $\mathsf{M}'_{\mathsf{h}}[y = \mathscr{h}\vec{x}] = \mathsf{S}$; this is impossible, and similarly in the other case; reject the assumption: $\mathsf{M}'[y = \mathscr{h}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)] = \mathsf{T}$. So $\mathsf{M}'[T'] = \mathsf{T}$; so since $T' \vDash \mathcal{F}$, $\mathsf{M}'[\mathcal{F}] = \mathsf{T}$; and by T10.14 again, $\mathsf{M}[\mathcal{F}] = \mathsf{T}$.

This argument repeatedly constructs from M an M' on which all the axioms of T' are true; then since T' entails $\mathcal{F}$, $\mathsf{M}'[\mathcal{F}] = \mathsf{T}$; and since M' and M agree on assignments to all the symbols in $\mathcal{F}$, $\mathsf{M}[\mathcal{F}] = \mathsf{T}$. The reasoning is interesting insofar as it exhibits how an interpretation M for T in $\mathscr{L}$ extends to an M' for T' with the symbols of $\mathscr{L}'$.

For T13.13 it is, in fact, important to show our specifications result in legitimate interpretations (compare pages 483–484). This is the point at which the uniqueness conditions matter. Here the most interesting case.

(f) Since $T \vdash \exists! y\mathcal{Q}(\vec{x}, y)$, by s̊oundness $T \vDash \exists! y\mathcal{Q}(\vec{x}, y)$; so since $\mathsf{M}[T] = \mathsf{T}$, $\mathsf{M}[\exists! y\mathcal{Q}(\vec{x}, y)] = \mathsf{T}$; so by TI, for any d, $\mathsf{M}_{\mathsf{d}}[\exists! y\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$, and $(*)$ for any d there is exactly one $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$. Given this:

(i) Each function has at least one output object: Consider some objects $\mathsf{o}_1\ldots\mathsf{o}_n$ and an assignment d such that $\mathsf{d}[x_1] = \mathsf{o}_1$ and … and $\mathsf{d}[x_n] = \mathsf{o}_n$; then from $(*)$ there is an m such that $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$; so by construction $\langle\langle \mathsf{d}[x_1]\ldots\mathsf{d}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\mathscr{h}]$; so $\langle\langle \mathsf{o}_1\ldots\mathsf{o}_n\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\mathscr{h}]$, and $\mathsf{M}'[\mathscr{h}]$ has an output object for input $\langle \mathsf{o}_1\ldots\mathsf{o}_n\rangle$.

(ii) Each function has at most one output object: Suppose there there are some objects $\mathsf{o}_1\ldots\mathsf{o}_n, \mathsf{m}, \mathsf{n}$ such that $\langle\langle \mathsf{o}_1\ldots\mathsf{o}_n\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\mathscr{h}]$ and $\langle\langle \mathsf{o}_1\ldots\mathsf{o}_n\rangle, \mathsf{n}\rangle \in \mathsf{M}'[\mathscr{h}]$; then by construction, there are assignments d and h, $\mathsf{d}[x_1] = \mathsf{h}[x_1] = \mathsf{o}_1$ and … and $\mathsf{d}[x_n] = \mathsf{h}[x_n] = \mathsf{o}_n$ such that both $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$ and $\mathsf{M}_{\mathsf{h}(y|\mathsf{n})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$. But $\mathsf{d}[y|\mathsf{n}]$ and $\mathsf{h}[y|\mathsf{n}]$ assign all the same objects to variables free in $\mathcal{Q}(\vec{x}, y)$; so from $\mathsf{M}_{\mathsf{h}(y|\mathsf{n})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$ and T8.5, $\mathsf{M}_{\mathsf{d}(y|\mathsf{n})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$; so both $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$ and $\mathsf{M}_{\mathsf{d}(y|\mathsf{n})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$; so by $(*)$, $\mathsf{m} = \mathsf{n}$. So $\mathsf{M}'[\mathscr{h}]$ has at most one output object for input, $\langle \mathsf{o}_1\ldots\mathsf{o}_n\rangle$.

And now our desired result combines T13.12 and T13.13 with s̊oundness and c̊ompleteness.

T13.14. For some defined symbol, with its associated axiom and conversion procedure, $T' \vdash \mathcal{F}'$ iff $T \vdash \mathfrak{C}[\mathcal{F}']$.

From T13.12, $T' \vdash \mathcal{F}'$ iff $T' \vdash \mathfrak{C}[\mathcal{F}']$; by s̊oundness and c̊ompleteness iff $T' \vDash \mathfrak{C}[\mathcal{F}']$; by T13.13 iff $T \vDash \mathfrak{C}[\mathcal{F}']$; by s̊oundness and c̊ompleteness iff $T \vdash \mathfrak{C}[\mathcal{F}']$.

And there may be a sequence of theories with new symbols such that our results apply to each member of the series. In the following, we will be clear about when new symbols and associated axioms are introduced, and about the conditions under which this may be done. In light of the results we have achieved however, we will not generally distinguish between a theory and its definitional extensions.

E13.11. Supposing that $T' \vdash y = \hbar^2 uv \leftrightarrow \mathcal{Q}(u, v, y)$ use the method of the text to find $\mathfrak{C}[A \wedge Bh^2ch^2xy]$.

E13.12. Complete the unfinished cases for constants in T13.12 and T13.13 (including legitimacy of the specification).

First Applications

Here are some quick results that will be helpful as we move forward. We specify conditions under which PA defines functions by composition, and then by regular and bounded minimization.

First, if PA defines some functions $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines their composition $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$. We are introducing a function symbol, so we introduce an axiom and then show that the condition is met. This pattern will repeat many times.

T13.15. If PA defines $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$.

Suppose PA defines $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$.

Def[*cmp*(g, h)] Let PA $\vdash v = f(\vec{x}, \vec{y}, \vec{z}) \leftrightarrow v = h(\vec{x}, g(\vec{y}), \vec{z})$. Then,

(i) PA $\vdash \exists v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$

1.	$h(\vec{x}, g(\vec{y}), \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$	=I
2.	$\exists v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$	1 ∃I

(ii) PA $\vdash \forall u \forall v[(u = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow u = v]$.

1.	$j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge k = h(\vec{x}, g(\vec{y}), \vec{z})$	A ($g, \rightarrow$I)
2.	$j = h(\vec{x}, g(\vec{y}), \vec{z})$	1 ∧E
3.	$k = h(\vec{x}, g(\vec{y}), \vec{z})$	1 ∧E
4.	$j = k$	2,3 =E
5.	$(j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge k = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow j = k$	1-4 →I
6.	$\forall v[(j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow j = v]$	5 ∀I
7.	$\forall u \forall v[(u = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow u = v]$	6 ∀I

So PA $\vdash \exists! v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$. And PA defines $f(\vec{x}, \vec{y}, \vec{z})$.

In addition, we can introduce functions for *minimization*. First, for unbounded minimization, the idea is to set $v = \mu y \mathcal{Q}(\vec{x}, y)$ just in case $\mathcal{Q}(\vec{x}, v)$, and no $z < v$ is such that $\mathcal{Q}(\vec{x}, z)$. In the ordinary case, a new function symbol $\hbar$ is introduced with an axiom of the sort $v = \hbar\vec{x} \leftrightarrow \mathcal{P}(\vec{x}, v)$ under the condition $T \vdash \exists! v \mathcal{P}(\vec{x}, v)$. But, in this case, the situation is simplified by the following theorem:

T13.16. If PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$, then PA defines $\mu y \mathcal{Q}(\vec{x}, y)$. Suppose PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$.

Def$[\mu y \mathcal{Q}(\vec{x}, y)]$ Let PA $\vdash v = \mu y \mathcal{Q}(\vec{x}, y) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)]$.

(i) PA $\vdash \exists v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)]$. Since PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$, by the least number principle T13.11aj, PA $\vdash \exists v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)]$.

(ii) PA $\vdash \forall u \forall v[(\mathcal{Q}(\vec{x}, u) \wedge (\forall z < u) {\sim} \mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)) \rightarrow u = v]$.

1.	$\mathcal{Q}(\vec{x}, j) \wedge (\forall z < j) {\sim} \mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, k) \wedge (\forall z < k) {\sim} \mathcal{Q}(\vec{x}, z)$	A ($g, \rightarrow$I)
2.	$j < k \vee j = k \vee k < j$	T13.11r
3.	$j < k$	A ($c, \sim$I)
4.	$(\forall z < k) {\sim} \mathcal{Q}(\vec{x}, z)$	1 $\wedge$E
5.	${\sim} \mathcal{Q}(\vec{x}, j)$	4,3 ($\forall$E)
6.	$\mathcal{Q}(\vec{x}, j)$	1 $\wedge$E
7.	$\bot$	6,5 $\bot$I
8.	$j \not< k$	3-7 $\sim$I
9.	$k < j$	A ($c, \sim$I)
10.	$(\forall z < j) {\sim} \mathcal{Q}(\vec{x}, z)$	1 $\wedge$E
11.	${\sim} \mathcal{Q}(\vec{x}, k)$	10,9 ($\forall$E)
12.	$\mathcal{Q}(\vec{x}, k)$	1 $\wedge$E
13.	$\bot$	12,11, $\bot$I
14.	$k \not< j$	9-13 $\sim$I
15.	$j = k$	2,8,14 DS
16.	$(\mathcal{Q}(\vec{x}, j) \wedge (\forall z < j) {\sim} \mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, k) \wedge (\forall z < k) {\sim} \mathcal{Q}(\vec{x}, z)) \rightarrow j = k$	1-15 $\rightarrow$I
17.	$\forall u \forall v[(\mathcal{Q}(\vec{x}, u) \wedge (\forall z < u) {\sim} \mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)) \rightarrow u = v]$	17 $\forall$I

So both (i) and (ii) follow with PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$. (i) results with the least number principle. And, as with the uniqueness result for the canonical $\mathcal{B}'$ (T12.12), (ii) results because the bounded quantifier of $\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, z)$ already builds in that at most one thing satisfies the conjunction.

For bounded minimization, we want the least $y \leq z$ such that $\mathcal{Q}(\vec{x}, y)$ if one exists and otherwise z—where z may or may not appear in $\vec{x}$ and so among the variables of $\mathcal{Q}$. Again, the situation is simplified as follows:

T13.17. For any formula $\mathcal{Q}(\vec{x}, y)$, PA defines $(\mu y \leq z)\mathcal{Q}(\vec{x}, y)$.

Def$[(\mu y \leq z)\mathcal{Q}(\vec{x}, y)]$ Let PA $\vdash v = (\mu y \leq z)\mathcal{Q}(\vec{x}, y) \leftrightarrow$
$v = \mu y[y = z \vee \mathcal{Q}(\vec{x}, y)]$.

Let $m(\vec{x}, z) = \mu y[y = z \vee \mathcal{Q}(\vec{x}, y)]$; then we require,

(i) PA $\vdash \exists v(v = m(\vec{x}, z))$

(ii) PA $\vdash \forall u \forall v([u = m(\vec{x}, z) \wedge v = m(\vec{x}, z)] \rightarrow u = v)$

These conditions are trivially met so long as $m(\vec{x}, z)$ is defined. And by T13.16 $m(\vec{x}, z)$ is defined given just the existential condition PA $\vdash \exists v[v = z \vee \mathcal{Q}(\vec{x}, v)]$, which follows immediately from PA $\vdash z = z$; so the conditions for bounded minimization are always satisfied.

Given these notions, we may obtain some immediate results.

*T13.18. Let $m(\vec{x}) = \mu y\mathcal{Q}(\vec{x}, y)$; then,

(a) PA $\vdash \mathcal{Q}(\vec{x}, m(\vec{x}))$

(b) PA $\vdash (\forall z < m(\vec{x}))\sim\mathcal{Q}(\vec{x}, z)$

*(c) PA $\vdash \mathcal{Q}(\vec{x}, v) \rightarrow m(\vec{x}) \leq v$

(d) PA $\vdash (\mu y \leq \emptyset)\mathcal{Q}(\vec{x}, y) = \emptyset$

(e) If PA $\vdash (\exists v \leq t)\mathcal{Q}(\vec{x}, v)$ then PA $\vdash (\mu y \leq t)\mathcal{Q}(\vec{x}, y) = m(\vec{x})$.

Because it is always possible to switch bound variables so that $\mathcal{Q}$ is converted to an equivalent $\mathcal{Q}'$ whose bound variables do not overlap with variables free in $m(\vec{x})$, we simply assume $m(\vec{x})$ is free for v in $\mathcal{Q}(\vec{x}, v)$—and we will generally make this move. Given this, (a)–(d) are straightforward. For (e) see the derivation on the following page.

Of these, (a) and (b) simply observe that the definition applies to the function defined. From (c), the least y such that $\mathcal{Q}(\vec{x}, y)$ is always $\leq$ an arbitrary v such that $\mathcal{Q}(\vec{x}, v)$. From (d) it does not matter about $\mathcal{Q}$, the least y under the bound $\emptyset$ is always $\emptyset$. Given a bounded existential, (e) converts between bounded and unbounded minimization: Intuitively, either $(\mu y \leq t)\mathcal{Q}(\vec{x}, y)$ reverts to the bound or it does not; if it does not, then the bounded minimization returns the same value as the unbounded; if it does, then no y under the bound is such that $\mathcal{Q}(\vec{x}, y)$, and the bounded minimization returns t—but then with the premise $\mathcal{Q}(\vec{x}, t)$ so that, again, the bounded minimization returns the same value as the unbounded.

*E13.13. Show (c) and (d) of T13.18. Hard-core: Show all the unfinished parts of T13.18.

T13.18e

Suppose PA $\vdash (\exists v \leq t)\mathcal{Q}(\vec{x}, v)$; then trivially, PA $\vdash \exists v\mathcal{Q}(\vec{x}, v)$. So by T13.16, PA defines $\mu y\mathcal{Q}(\vec{x}, y)$ and PA $\vdash v = \mu y\mathcal{Q}(\vec{x}, y) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall w < v)\sim\mathcal{Q}(\vec{x}, w)]$.

1.	$(\exists v \leq t)\mathcal{Q}(\vec{x}, v)$	given
2.	$v = \mu y\mathcal{Q}(\vec{x}, y) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall w < v)\sim\mathcal{Q}(\vec{x}, w)]$	as above
3.	$n(\vec{x}, t) = (\mu y \leq t)\mathcal{Q}(\vec{x}, y)$	def
4.	$n(\vec{x}, t) = \mu y[y = t \vee \mathcal{Q}(\vec{x}, y)]$	3 *Def*[$\mu y \leq$]
5.	$n(\vec{x}, t) = t \vee \mathcal{Q}(\vec{x}, n(\vec{x}, t))$	4 T13.18a
6.	$\mathcal{Q}(\vec{x}, j)$	A (g, 1($\exists$E))
7.	$j \leq t$	
8.	$j < t \vee j = t$	7 T13.11n
9.	$j = t$	A (g, 8$\vee$E)
10.	$t = n(\vec{x}, t) \vee t \neq n(\vec{x}, t)$	T3.1
11.	$t = n(\vec{x}, t)$	A (g, 10$\vee$E)
12.	$\mathcal{Q}(\vec{x}, t)$	6,9 =E
13.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t))$	12,11 =E
14.	$t \neq n(\vec{x}, t)$	A (g, 10$\vee$E)
15.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t))$	5,14 DS
16.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t))$	10,11-13,14-15 $\vee$E
17.	$j < t$	A (g, 8$\vee$E)
18.	$j = t \vee \mathcal{Q}(\vec{x}, j)$	6 $\vee$I
19.	$n(\vec{x}, t) \leq j$	4,18 T13.18c
20.	$n(\vec{x}, t) < t$	19,17 T13.11d
21.	$n(\vec{x}, t) \neq t$	20 T13.11t
22.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t))$	5,21 DS
23.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t))$	8,9-16,17-22 $\vee$E
24.	$(\forall w < n(\vec{x}, t))\sim[w = t \vee \mathcal{Q}(\vec{x}, w)]$	4 T13.18b
25.	$l < n(\vec{x}, t)$	A (g, ($\forall$I))
26.	$\sim[l = t \vee \mathcal{Q}(\vec{x}, l)]$	24,25 ($\forall$E)
27.	$l \neq t \wedge \sim\mathcal{Q}(\vec{x}, l)$	26 DeM
28.	$\sim\mathcal{Q}(\vec{x}, l)$	27 $\wedge$E
29.	$(\forall w < n(\vec{x}, t))\sim\mathcal{Q}(\vec{x}, w)$	25-28 ($\forall$I)
30.	$\mathcal{Q}(\vec{x}, n(\vec{x}, t)) \wedge (\forall w < n(\vec{x}, t))\sim\mathcal{Q}(\vec{x}, w)$	23,29 $\wedge$I
31.	$n(\vec{x}, t) = \mu y\mathcal{Q}(\vec{x}, y) \leftrightarrow$ $[\mathcal{Q}(\vec{x}, n(\vec{x}, t)) \wedge (\forall w < n(\vec{x}, t))\sim\mathcal{Q}(\vec{x}, w)]$	from 2
32.	$n(\vec{x}, t) = \mu y\mathcal{Q}(\vec{x}, y)$	31,30 $\leftrightarrow$E
33.	$n(\vec{x}, t) = \mu y\mathcal{Q}(\vec{x}, y)$	1,6-32 ($\exists$E)
34.	$(\mu y \leq t)\mathcal{Q}(\vec{x}, y) = \mu y\mathcal{Q}(\vec{x}, y)$	33,3 =E

The key to this derivation is to obtain $\mathcal{Q}(\vec{x}, n(\vec{x}, t))$ at (23) and $(\forall w < n(\vec{x}, t))\sim\mathcal{Q}(\vec{x}, w)$ at (29). Then result comes from the definition (2). Observe that we permit a step (def) at (3), very much as in Chapter 7.

First Theorems of Chapter 13

T13.1 For any recursively axiomatized theory T whose language includes $\mathscr{L}_{\text{NT}}$, $\mathsf{N}[\mathscr{G}] = \mathsf{N}[\sim\exists v Prft(v, \overline{\ulcorner\mathscr{G}\urcorner})]$. *Carnap's result for* $\mathscr{G}$.

T13.2 Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathscr{G} \leftrightarrow \sim\exists v Prft(v, \overline{\ulcorner\mathscr{G}\urcorner})$. *Diagonal result for* $\mathscr{G}$.

T13.3 If T is a recursively axiomatized $\tilde{\text{s}}$ound theory whose language includes $\mathscr{L}_{\text{NT}}$, then $T \nvdash \mathscr{G}$ and $T \nvdash \sim\mathscr{G}$.

T13.4 If T is a recursively axiomatized theory extending Q, then if T is consistent $T \nvdash \mathscr{G}$, and if T is ω-consistent, $T \nvdash \sim\mathscr{G}$.

T13.5 Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathscr{R} \leftrightarrow \sim\exists v RPrft(v, \overline{\ulcorner\mathscr{R}\urcorner})$. *Diagonal result for* $\mathscr{R}$.

T13.6 If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{R}$ and $T \nvdash \sim\mathscr{R}$.

T13.7 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic for provability, $T \vdash Cont \rightarrow \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

T13.8 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions, if T is consistent, $T \nvdash Cont$.

T13.9 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic for provability, $T \vdash Cont \leftrightarrow \sim Prvt(\overline{\ulcorner Cont\urcorner})$.

T13.10 Suppose T is a recursively axiomatized theory extending Q_{s}. Then if $T \vdash \mathscr{P}$, then $T \vdash \Box\mathscr{P}$.

Corollary (D1): If PA $\vdash \mathscr{P}$ then PA $\vdash \Box\mathscr{P}$.

T13.11 Some theorems of PA including for inequality and strong induction.

T13.12 For some defined symbol, with its associated axiom and conversion procedure, $T' \vdash \mathscr{F}' \leftrightarrow \mathfrak{C}[\mathscr{F}']$.

T13.13 Consider T in language $\mathscr{L}$ and T' with some defined symbol, axiom, and condition; then for any formula $\mathscr{F}$ in $\mathscr{L}$, $T' \vDash \mathscr{F}$ iff $T \vDash \mathscr{F}$.

T13.14 For some defined symbol, with its associated axiom and conversion procedure, $T' \vdash \mathscr{F}'$ iff $T \vdash \mathfrak{C}[\mathscr{F}']$.

T13.15 If PA defines $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$.

T13.16 If PA $\vdash \exists v \mathscr{Q}(\vec{x}, v)$, then PA defines $\mu y \mathscr{Q}(\vec{x}, y)$.

T13.17 For any formula $\mathscr{Q}(\vec{x}, y)$, PA defines $(\mu y \leq z)\mathscr{Q}(\vec{x}, y)$.

T13.18 Some preliminary results for bounded and unbounded minimization.

13.3.2 Definitions for Recursive Functions

Our aim is to define and manipulate functions and relations corresponding to the recursive functions and relations of Chapter 12. Having said something about the conditions under which functions and relations are defined, in this section we set out to show that PA in fact defines such functions and relations. We begin with the core argument to show that PA defines functions corresponding to recursive functions of Chapter 12. Then a series of results to show that this argument goes through for the case when functions arise by recursion. Finally, the main theorem is extended to show that PA defines functions and relations *coordinate* to ones in Chapter 12.

Insofar as we understand what a theorem of PA *is*, not all of the *demonstrations* of the theorems are required to understand the argument—and some may obscure the overall flow. Thus, for our main argument, we often list results, shifting hints and demonstrations into exercises and answers to exercises. To retain demonstration of results, a great many exercises are in fact included in the answers.

The Core Result

To define functions and relations corresponding to the recursive functions and relations of Chapter 12, the main argument is an induction on the sequence of recursive functions. However, with an eye to the β-function, we begin showing that PA defines remainder $rm(m,n)$ and quotient $qt(m,n)$ functions corresponding to $\mathsf{m}/(\mathsf{n}+1)$. Division is by $\mathsf{n}+1$ to avoid the possibility of division by zero.[10]

**Def*[*rm*] Let PA $\vdash v = rm(m,n) \leftrightarrow (\exists w \leq m)[m = Sn \times w + v \wedge v < Sn]$.

(i) PA $\vdash \exists x(\exists w \leq m)[m = Sn \times w + x \wedge x < Sn]$

(ii) PA $\vdash \forall x \forall y[((\exists w \leq m)[m = Sn \times w + x \wedge x < Sn] \wedge$
$(\exists w \leq m)[m = Sn \times w + y \wedge y < Sn]) \rightarrow x = y]$

Def[*qt*] Let PA $\vdash v = qt(m,n) \leftrightarrow m = Sn \times v + rm(m,n)$.

(i) PA $\vdash \exists x[m = Sn \times x + rm(m,n)]$

(ii) PA $\vdash \forall x \forall y[(m = Sn \times x + rm(m,n) \wedge m = Sn \times y + rm(m,n)) \rightarrow x = y]$

Def[β] Let PA $\vdash v = \beta(p,q,i) \leftrightarrow v = rm(p, q \times Si)$.

Since this is a composition of functions, immediate from T13.15. In this simple case we might as well have asserted, PA $\vdash \beta(p,q,i) = rm(p, q \times Si)$.

[10] A choice is made: Another option is define the functions so that an arbitrary value is assigned for division by zero (as for example Boolos, *The Logic of Provability*, page 27). Our selection makes for somewhat unintuitive statements of that which is intuitively true—rather than (relatively) intuitive statements including that which is intuitively undefined or false.

For hints on the first two see the associated exercise E13.14. Substituting p and $q \times Si$ from β into *rm*, PA $\vdash v = \beta(p,q,i) \leftrightarrow (\exists w \leq p)[p = S(q \times Si) \times w + v \wedge v < S(q \times Si)]$, which is to say PA $\vdash v = \beta(p,q,i) \leftrightarrow \mathscr{B}(p,q,i,v)$, where $\mathscr{B}$ is the original formula to express the beta function.[11]

And now our main argument that PA defines functions corresponding to recursive functions. The main result is for functions—it will extend to relations as an easy corollary. However we shall not be able to show that PA defines functions corresponding to *all* the recursive functions: When a recursive function $\mathsf{g(x,y)}$ captured by some original $\mathscr{G}(x,y,z)$ is regular, $\exists y \mathscr{G}(x,y,\overline{0})$ is true. Say an application of regular minimization to generate $\mathsf{f}(\vec{\mathsf{x}})$ from $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$ is (PA) *friendly* iff we can prove it—iff PA $\vdash \exists y \mathscr{G}(\vec{x},y,\overline{0})$. Then an arbitrary recursive function is (PA) *friendly* just in case it is an initial function or arises by applications of composition, recursion, or friendly regular minimization. Observe that all *primitive* recursive functions are automatically friendly insofar as they involve no applications of minimization at all.

*T13.19. For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathscr{R}(\vec{x},v)$ by which it is expressed and captured, PA defines a function $\mathbb{r}(\vec{x})$ such that PA $\vdash v = \mathbb{r}(\vec{x}) \leftrightarrow \mathscr{R}(\vec{x},v)$.

Suppose $\mathsf{r}(\vec{\mathsf{x}})$ is a friendly recursive function. By induction on the sequence of recursive functions,

Basis: $\mathsf{r}_0(\vec{\mathsf{x}})$ is an initial function $\mathsf{suc(x)}$, $\mathsf{zero()}$, or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{j})$.

(s) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{suc(x)}$. Let PA $\vdash v = \mathbb{suc}(x) \leftrightarrow Sx = v$. But $Sx = v$ is the original formula $Suc(x,v)$ by which $\mathsf{suc(x)}$ is expressed and captured; so PA $\vdash v = \mathbb{suc}(x) \leftrightarrow Suc(x,v)$. And by reasoning as follows,

1.	$Sx = Sx$	=I
2.	$\exists y(Sx = y)$	1 ∃I

1.	$\quad Sx = j \wedge Sx = k$	A (g, →I)
2.	$\quad Sx = j$	1 ∧E
3.	$\quad Sx = k$	1 ∧E
4.	$\quad j = k$	3,2 =E
5.	$(Sx = j \wedge Sx = k) \to j = k$	1-4 →I
6.	$\forall z[(Sx = j \wedge Sx = z) \to j = z]$	5 ∀I
7.	$\forall y \forall z[(Sx = y \wedge Sx = z) \to y = z]$	6 ∀I

PA $\vdash \exists! y(Sx = y)$. So PA defines $\mathbb{suc}(x)$.

(z) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{zero()}$. Let PA $\vdash v = \mathbb{zero}() \leftrightarrow \overline{0} = v$. Then PA $\vdash v = \mathbb{zero}() \leftrightarrow Zero(v)$. And by [homework] PA defines $\mathbb{zero}()$.

(i) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x}_\mathsf{j})$. Let PA $\vdash v = \mathbb{idnt}^j_k(x_1 \ldots x_j) \leftrightarrow [(x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v]$. Then PA $\vdash v = \mathbb{idnt}^j_k(x_1 \ldots x_j) \leftrightarrow Idnt^j_k(x_1 \ldots x_j, v)$. And by [homework] PA defines $\mathbb{idnt}^j_k(x_1 \ldots x_j)$.

[11] In section 12.2.3 we considered an intuitive $\mathsf{rem(p,q)}$ and said $\beta(\mathsf{p,q,i}) = \mathsf{rem[p,S(q \times S(i))]}$. That rem is partial insofar as it returns no value when $\mathsf{q} = 0$, though β remains total insofar division is by successors. Here, *rm* corresponds to an $\mathsf{rm(p,q)} = \mathsf{rem(p,S(q))}$. So rm is total and $\mathsf{rem(p,S(q \times S(i))]} = \mathsf{rm(p,q \times Si)}$ so that the β-function comes out the same either way.

Assp: For any i, $0 \le i < k$, and $\mathsf{r_i}(\vec{\mathsf{x}})$ with $\mathcal{R}_i(\vec{x}, v)$, PA defines $\mathscr{r}_i(\vec{x})$ such that PA $\vdash v = \mathscr{r}_i(\vec{x}) \leftrightarrow \mathcal{R}_i(\vec{x}, v)$.

Show: PA defines $\mathscr{r}_k(\vec{x})$ such that PA $\vdash v = \mathscr{r}_k(\vec{x}) \leftrightarrow \mathcal{R}_k(\vec{x}, v)$.

$\mathsf{r_k}(\vec{\mathsf{x}})$ is either an initial function or arises by composition, recursion, or friendly regular minimization. If $\mathsf{r_k}(\vec{\mathsf{x}})$ is an initial function, then reason as in the basis. So suppose one of the other cases.

(c) $\mathsf{r_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{g}(\vec{\mathsf{y}}), \vec{\mathsf{z}})$ for some $\mathsf{h_i}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ and $\mathsf{g_j}(\vec{\mathsf{y}})$ where $\mathsf{i, j < k}$. By assumption PA defines $\mathscr{h}(\vec{x}, w, \vec{z})$ such that PA $\vdash v = \mathscr{h}(\vec{x}, w, \vec{z}) \leftrightarrow \mathcal{H}(\vec{x}, w, \vec{z}, v)$ and PA defines $\mathscr{g}(\vec{y})$ such that PA $\vdash w = \mathscr{g}(\vec{y}) \leftrightarrow \mathcal{G}(\vec{y}, w)$. Let PA $\vdash \mathscr{r}_k(\vec{x}, \vec{y}, \vec{z}) = \mathscr{h}(\vec{x}, \mathscr{g}(\vec{y}), \vec{z})$. Then by T13.15 PA defines $\mathscr{r}_k$. And, where the original $\mathcal{R}_k$ is of the sort $\exists w[\mathcal{G}(\vec{y}, w) \wedge \mathcal{H}(\vec{x}, w, \vec{z}, v)]$, PA $\vdash v = \mathscr{r}_k(\vec{x}, \vec{y}, \vec{z}) \leftrightarrow \mathcal{R}_k(\vec{x}, \vec{y}, \vec{z}, v)$. Dropping $\vec{x}$ and $\vec{z}$ and reducing $\vec{y}$ to a single variable, the derivation is as on the next page.

(r) $\mathsf{r_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from some $\mathsf{g_i}(\vec{\mathsf{x}})$ and $\mathsf{h_j}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ where $\mathsf{i, j < k}$. By assumption PA defines $\mathscr{g}(\vec{x})$ such that PA $\vdash v = \mathscr{g}(\vec{x}) \leftrightarrow \mathcal{G}(\vec{x}, v)$ and PA defines $\mathscr{h}(\vec{x}, y, u)$ such that PA $\vdash v = \mathscr{h}(\vec{x}, y, u) \leftrightarrow \mathcal{H}(\vec{x}, y, u, v)$. Let PA $\vdash z = \mathscr{r}_k(\vec{x}, y) \leftrightarrow$

$$\exists p \exists q[\beta(p, q, \emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, y) = z]$$

By the argument of the next section, PA defines $\mathscr{r}(\vec{x}, y)$. And where the original $\mathcal{R}_k(\vec{x}, y, z) =$

$$\exists p \exists q\{\exists v[\mathcal{B}(p, q, \emptyset, v) \wedge \mathcal{G}(\vec{x}, v)] \wedge$$
$$(\forall i < y)\exists u \exists v[\mathcal{B}(p, q, i, u) \wedge \mathcal{B}(p, q, Si, v) \wedge \mathcal{H}(\vec{x}, i, u, v)] \wedge \mathcal{B}(p, q, y, z)\}$$

we need PA $\vdash z = \mathscr{r}_k(\vec{x}, y) \leftrightarrow \mathcal{R}_k(\vec{x}, y, z)$. To manage long formulas let,

$$\mathcal{P}(p, q, \vec{x}) = \beta(p, q, \emptyset) = \mathscr{g}(\vec{x})$$

$$\mathcal{Q}(p, q, \vec{x}, y) = (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si)$$

Then PA $\vdash z = \mathscr{r}_k(\vec{x}, y) \leftrightarrow \exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, y) \wedge \beta(p, q, y) = z]$. A derivation to show PA $\vdash z = \mathscr{r}_k(\vec{x}, y) \to \mathcal{R}_k(\vec{x}, y, z)$ is on page 665. The other direction is homework.

(m) $\mathsf{r_k}(\vec{\mathsf{x}})$ arises by friendly regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption PA defines $\mathscr{g}(\vec{x}, y)$ such that $(*)$ PA $\vdash v = \mathscr{g}(\vec{x}, y) \leftrightarrow \mathcal{G}(\vec{x}, y, v)$ where $\mathcal{G}$ is the original formula to express and capture g. Let PA $\vdash \mathscr{r}_k(\vec{x}) = \mu y[\mathscr{g}(\vec{x}, y) = \overline{0}]$. Since the minimization is friendly, PA $\vdash \exists y \mathcal{G}(\vec{x}, y, \overline{0})$; so with $(*)$ PA $\vdash \exists y(\mathscr{g}(\vec{x}, y) = \overline{0})$; and by T13.16, PA defines $\mathscr{r}_k(\vec{x})$. By *Def*[μy], PA $\vdash v = \mathscr{r}_k(\vec{x}) \leftrightarrow [\mathscr{g}(\vec{x}, v) = \overline{0} \wedge (\forall z < v)(\mathscr{g}(\vec{x}, z) \neq \overline{0})]$. And PA $\vdash \mathcal{R}_k(\vec{x}, v) \leftrightarrow [\mathcal{G}(\vec{x}, v, \overline{0}) \wedge (\forall y < v)\exists z(\mathcal{G}(\vec{x}, y, z) \wedge \overline{0} \neq z)]$. Then with $(*)$ it is easy to show PA $\vdash v = \mathscr{r}_k(\vec{x}) \leftrightarrow \mathcal{R}_k(\vec{x}, v)$. Homework.

Indct: For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and the original formula $\mathcal{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathscr{r}(\vec{x})$ such that PA $\vdash v = \mathscr{r}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$.

Of course it remains to show that PA defines $\mathscr{r}(\vec{x}, y)$ in the case when $\mathsf{r}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion.

*E13.14. Complete the justifications for *Def*[*rm*] and *Def*[*qt*].

Hints for remainder. (i): This is an argument by IN on m. The zero case is easy from $\emptyset = Sn \times \emptyset + \emptyset \wedge \emptyset < Sn$. Then under the assumption $\exists x(\exists w \leq j)[j = Sn \times w + x \wedge x < Sn]$ for $\to$I, you need the result for Sj. Given $j = Sn \times q + r \wedge r < Sn$ by assumptions for $\exists$E and ($\exists$E), $r < n \vee r = n$. In the first case Sj is divided by leaving the quotient q the same, and incrementing r; in the second case Sj is divided by Sq with remainder zero. (ii): This does

T13.19(c)

1.	$\mathscr{r}(y) = \mathscr{h}(\mathscr{g}(y))$	def $\mathscr{r}$
2.	$v = \mathscr{h}(w) \leftrightarrow \mathcal{H}(w, v)$	by assp
3.	$w = \mathscr{g}(y) \leftrightarrow \mathcal{G}(y, w)$	by assp
4.	$\quad v = \mathscr{r}(y)$	A (g, $\leftrightarrow$I)
5.	$\quad v = \mathscr{h}(\mathscr{g}(y))$	1,4 =E
6.	$\quad \mathscr{g}(y) = \mathscr{g}(y)$	=I
7.	$\quad \mathscr{g}(y) = \mathscr{g}(y) \leftrightarrow \mathcal{G}(y, \mathscr{g}(y))$	from 3
8.	$\quad \mathcal{G}(y, \mathscr{g}(y))$	7,6 $\leftrightarrow$E
9.	$\quad \mathscr{h}(\mathscr{g}(y)) = \mathscr{h}(\mathscr{g}(y))$	=I
10.	$\quad \mathscr{h}(\mathscr{g}(y)) = \mathscr{h}(\mathscr{g}(y)) \leftrightarrow \mathcal{H}(\mathscr{g}(y), \mathscr{h}(\mathscr{g}(y)))$	from 2
11.	$\quad \mathcal{H}(\mathscr{g}(y), \mathscr{h}(\mathscr{g}(y)))$	10,9 $\leftrightarrow$E
12.	$\quad \mathcal{H}(\mathscr{g}(y), v)$	11,5 =E
13.	$\quad \mathcal{G}(y, \mathscr{g}(y)) \wedge \mathcal{H}(\mathscr{g}(y), v)$	8,12 $\wedge$I
14.	$\quad \exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	13 $\exists$I
15.	$\quad \exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	A (g, $\leftrightarrow$I)
16.	$\qquad \mathcal{G}(y, j) \wedge \mathcal{H}(j, v)$	A (g, 15$\exists$E)
17.	$\qquad j = \mathscr{g}(y) \leftrightarrow \mathcal{G}(y, j)$	from 3
18.	$\qquad \mathcal{G}(y, j)$	16 $\wedge$E
19.	$\qquad j = \mathscr{g}(y)$	17,18 $\leftrightarrow$E
20.	$\qquad v = \mathscr{h}(j) \leftrightarrow \mathcal{H}(j, v)$	from 2
21.	$\qquad \mathcal{H}(j, v)$	16 $\wedge$E
22.	$\qquad v = \mathscr{h}(j)$	20,21 $\leftrightarrow$E
23.	$\qquad v = \mathscr{h}(\mathscr{g}(y))$	22,19 =E
24.	$\qquad v = \mathscr{r}(y)$	1,23 =E
25.	$\quad v = \mathscr{r}(y)$	15,16-24 $\exists$E
26.	$v = \mathscr{r}(y) \leftrightarrow \exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	4-14,15-25 $\leftrightarrow$I
27.	$v = \mathscr{r}(y) \leftrightarrow \mathcal{R}(y, v)$	26 abv

With the use of (2) and (3) by $\forall$I and $\forall$E as on lines (7), (10), (17), and (20) this derivation is straightforward. As usual, we suppose that quantifiers are arranged so that substitutions are allowed—and so, for example, that $\mathscr{g}(y)$ is free for w in $\mathcal{H}(w, v)$ and $\mathcal{G}(y, w)$.

not require IN, but is an involved derivation all the same. Once you instantiate the bounded existential quantifiers to quotients p with remainder j and q with remainder k, you have $p < q \vee p = q \vee q < p$. When $p = q$, $j = k$ follows easily with cancellation for addition. And the other cases contradict.

T13.19(r)

1.	$v = \beta(p,q,i) \leftrightarrow \mathcal{B}(p,q,i,v)$	*Def*[β]
2.	$v = g(\vec{x}) \leftrightarrow \mathcal{G}(\vec{x},v)$	by assp
3.	$v = h(\vec{x},y,u) \leftrightarrow \mathcal{H}(\vec{x},y,u,v)$	by assp
4.	$z = r(\vec{x},y)$	A (g, $\rightarrow$I)
5.	$\exists p \exists q[\mathcal{P}(p,q,\vec{x}) \wedge \mathcal{Q}(p,q,\vec{x},y) \wedge \beta(p,q,y) = z]$	4 def r
6.	$\mathcal{P}(a,b,\vec{x}) \wedge \mathcal{Q}(a,b,\vec{x},y) \wedge \beta(a,b,y) = z$	A (g, 5$\exists$E)
7.	$\beta(a,b,\emptyset) = g(\vec{x})$	6 $\wedge$E ($\mathcal{P}$)
8.	$\mathcal{G}(\vec{x}, g(\vec{x}))$	from 2
9.	$\mathcal{B}(a,b,\emptyset,\beta(a,b,\emptyset))$	from 1
10.	$\mathcal{B}(a,b,\emptyset,g(\vec{x}))$	9,7 $=$E
11.	$\mathcal{B}(a,b,\emptyset,g(\vec{x})) \wedge \mathcal{G}(\vec{x},g(\vec{x}))$	10,8 $\wedge$I
12.	$\exists v[\mathcal{B}(a,b,\emptyset,v) \wedge \mathcal{G}(\vec{x},v)]$	11 $\exists$I
13.	$(\forall i < y) h(\vec{x},i,\beta(a,b,i)) = \beta(a,b,Si)$	6 $\wedge$E ($\mathcal{Q}$)
14.	$l < y$	A (g, ($\forall$I))
15.	$h(\vec{x},l,\beta(a,b,l)) = \beta(a,b,Sl)$	13,14 ($\forall$E)
16.	$\mathcal{B}(a,b,l,\beta(a,b,l))$	from 1
17.	$\mathcal{B}(a,b,Sl,\beta(a,b,Sl))$	from 1
18.	$\mathcal{H}(\vec{x},l,\beta(a,b,l),h(\vec{x},l,\beta(a,b,l)))$	from 3
19.	$\mathcal{H}(\vec{x},l,\beta(a,b,l),\beta(a,b,Sl))$	18,15 $=$E
20.	$\mathcal{B}(a,b,l,\beta(a,b,l)) \wedge \mathcal{B}(a,b,Sl,\beta(a,b,Sl)) \wedge$ $\mathcal{H}(\vec{x},l,\beta(a,b,l),\beta(a,b,Sl))$	16,17,19 $\wedge$I
21.	$\exists u \exists v[\mathcal{B}(a,b,l,u) \wedge \mathcal{B}(a,b,Sl,v) \wedge \mathcal{H}(\vec{x},l,u,v)]$	20 $\exists$I
22.	$(\forall i < y)\exists u \exists v[\mathcal{B}(a,b,i,u) \wedge \mathcal{B}(a,b,Si,v) \wedge \mathcal{H}(\vec{x},i,u,v)]$	14-21 ($\forall$I)
23.	$\beta(a,b,y) = z$	6 $\wedge$E
24.	$\mathcal{B}(a,b,y,\beta(a,b,y))$	from 1
25.	$\mathcal{B}(a,b,y,z)$	24,23 $=$E
26.	$\exists v[\mathcal{B}(a,b,\emptyset,v) \wedge \mathcal{G}(\vec{x},v)] \wedge$ $(\forall i < y)\exists u \exists v[\mathcal{B}(a,b,i,u) \wedge \mathcal{B}(a,b,Si,v) \wedge \mathcal{H}(\vec{x},i,u,v)] \wedge$ $\mathcal{B}(a,b,y,z)$	12,22,25 $\wedge$I
27.	$\exists p \exists q\{\exists v[\mathcal{B}(p,q,\emptyset,v) \wedge \mathcal{G}(\vec{x},v)] \wedge$ $(\forall i < y)\exists u \exists v[\mathcal{B}(p,q,i,u) \wedge \mathcal{B}(p,q,Si,v) \wedge \mathcal{H}(\vec{x},i,u,v)] \wedge$ $\mathcal{B}(p,q,y,z)\}$	26 $\exists$I
28.	$\mathcal{R}(\vec{x},y,z)$	27 def $\mathcal{R}$
29.	$\mathcal{R}(\vec{x},y,z)$	5,6-28 $\exists$E
30.	$z = r(\vec{x},y) \rightarrow \mathcal{R}(\vec{x},y,z)$	4-29 $\rightarrow$I

$\mathcal{P}$ and $\mathcal{Q}$ are as described on page 663. Again, with the use of (1), (2), and (3) by $\forall$I, $\forall$E, and then $=$I and $\leftrightarrow$E as on lines (8), (9), and so forth, this derivation is straightforward.

Hints for quotient. (i) With *Def*[*rm*], PA $\vdash (\exists w \leq m)[m = Sn \times w + rm(m,n) \wedge rm(m,n) < Sn]$; and the result follows easily. (ii) This is easy with cancellation laws for addition and multiplication.

*E13.15. Complete the cases left to homework from T13.19. You should set up the entire induction, but may refer to the text as the text refers unfinished cases to homework.

The Recursion Clause

We turn now to a series of definitions and results with the aim of showing that PA defines $\mathscr{r}$ in the case when r arises by recursion. Some of the functions so-defined are equivalent to ones that will result by T13.19. However, insofar as we have not yet proved T13.19, we cannot use it! So we are showing directly that PA gives the required results.

Uniqueness. It will be easiest to begin with the uniqueness clause. Where $\mathcal{F}(\vec{x}, y, z)$ is our formula,

$$\exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$$

*T13.20. PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \rightarrow m = n]$

By IN on the value of y. For the zero case you need PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, \emptyset, m) \wedge \mathcal{F}(\vec{x}, \emptyset, n)) \rightarrow m = n]$. This is simple enough and left as homework. Given the zero case, see the main argument by IN on the following page.

*E13.16. Complete the demonstration for T13.20 by completing the demonstration of the zero case.

Existence. Considerably more difficult is the existential condition. To show that PA $\vdash \exists z \mathcal{F}(\vec{x}, y, z)$,

$$\text{PA} \vdash \exists z \exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$$

we shall have to show there are p and q that yield the right result for the β-function. And for this we require the Chinese remainder theorem in PA. Though we have resources to state the β-function, we do not yet have all that is required to duplicate reasoning as from the page 583 beta function reference (for example, factorial). Thus we shall have to proceed in a different way. In particular, we specially depend on *least common multiple*—which we shall be able to define directly, apart from T13.19. Again, we build by a series of results.

T13.20

1.	$\forall m \forall n[(\mathcal{F}(\vec{x}, \emptyset, m) \wedge \mathcal{F}(\vec{x}, \emptyset, n)) \to m = n]$	zero case
2.	$\forall m \forall n[(\mathcal{F}(\vec{x}, j, m) \wedge \mathcal{F}(\vec{x}, j, n)) \to m = n]$	A (g, $\to$I)
3.	$\mathcal{F}(\vec{x}, Sj, u) \wedge \mathcal{F}(\vec{x}, Sj, v)$	A (g, $\to$I)
4.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, Sj) \wedge \beta(p, q, Sj) = u]$	3 $\wedge$E
5.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, Sj) \wedge \beta(p, q, Sj) = v]$	3 $\wedge$E
6.	$\mathcal{P}(a, b, \vec{x}) \wedge \mathcal{Q}(a, b, \vec{x}, Sj) \wedge \beta(a, b, Sj) = u$	A (g, 4$\exists$E)
7.	$\mathcal{P}(a, b, \vec{x})$	6 $\wedge$E
8.	$(\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	6 $\wedge$E ($\mathcal{Q}$)
9.	$\beta(a, b, Sj) = u$	6 $\wedge$E
10.	$\mathcal{P}(c, d, \vec{x}) \wedge \mathcal{Q}(c, d, \vec{x}, Sj) \wedge \beta(c, d, Sj) = v$	A (g, 5$\exists$E)
11.	$\mathcal{P}(c, d, \vec{x})$	10 $\wedge$E
12.	$(\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si)$	10 $\wedge$E ($\mathcal{Q}$)
13.	$\beta(c, d, Sj) = v$	10 $\wedge$E
14.	$j < Sj$	T13.11j
15.	$\mathbb{h}(\vec{x}, j, \beta(a, b, j)) = \beta(a, b, Sj)$	8,14 ($\forall$E)
16.	$\mathbb{h}(\vec{x}, j, \beta(c, d, j)) = \beta(c, d, Sj)$	12,14 ($\forall$E)
17.	$k < j$	A (g, ($\forall$I))
18.	$k < Sj$	17 T13.11o
19.	$\mathbb{h}(\vec{x}, k, \beta(a, b, k)) = \beta(a, b, Sk)$	8,18 ($\forall$E)
20.	$(\forall i < j)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	17-19 ($\forall$I)
21.	$\beta(a, b, j) = \beta(a, b, j)$	$=$I
22.	$\mathcal{P}(a, b, \vec{x}) \wedge \mathcal{Q}(a, b, \vec{x}, j) \wedge \beta(a, b, j) = \beta(a, b, j)$	7,20,21 $\wedge$I
23.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, j) \wedge \beta(p, q, j) = \beta(a, b, j)]$	22 $\exists$I
24.	$\mathcal{F}(\vec{x}, j, \beta(a, b, j))$	23 abv
25.	$k < j$	A (g, ($\forall$I))
26.	$k < Sj$	25 T13.11o
27.	$\mathbb{h}(\vec{x}, k, \beta(c, d, k)) = \beta(c, d, Sk)$	12,26 ($\forall$E)
28.	$(\forall i < j)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si)$	25-27 ($\forall$I)
29.	$\beta(c, d, j) = \beta(c, d, j)$	$=$I
30.	$\mathcal{P}(c, d, \vec{x}) \wedge \mathcal{Q}(c, d, \vec{x}, j) \wedge \beta(c, d, j) = \beta(c, d, j)$	11,28,29 $\wedge$I
31.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, j) \wedge \beta(p, q, j) = \beta(c, d, j)]$	30 $\exists$I
32.	$\mathcal{F}(\vec{x}, j, \beta(c, d, j))$	31 abv
33.	$\beta(a, b, j) = \beta(c, d, j)$	2,24,32 $\forall$E
34.	$\mathbb{h}(\vec{x}, j, \beta(c, d, j)) = \beta(a, b, Sj)$	15,33 $=$E
35.	$\beta(a, b, Sj) = \beta(c, d, Sj)$	34,16 $=$E
36.	$u = v$	35,9,13 $=$E
37.	$u = v$	5,10-36 $\exists$E
38.	$u = v$	4,6-37 $\exists$E
39.	$(\mathcal{F}(\vec{x}, Sj, u) \wedge \mathcal{F}(\vec{x}, Sj, v)) \to u = v$	3-38 $\to$I
40.	$\forall m \forall n[(\mathcal{F}(\vec{x}, Sj, m) \wedge \mathcal{F}(\vec{x}, Sj, n)) \to m = n]$	39 $\forall$I
41.	$\forall m \forall n[(\mathcal{F}(\vec{x}, j, m) \wedge \mathcal{F}(\vec{x}, j, n)) \to m = n] \to$ $\forall m \forall n[(\mathcal{F}(\vec{x}, Sj, m) \wedge \mathcal{F}(\vec{x}, Sj, n)) \to m = n]$	2-40 $\to$I
42.	$\forall y\{\forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \to m = n] \to$ $\forall m \forall n[(\mathcal{F}(\vec{x}, Sy, m) \wedge \mathcal{F}(\vec{x}, Sy, n)) \to m = n]\}$	41 $\forall$I
43.	$\forall y \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \to m = n]$	1,42 IN
44.	$\forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \to m = n]$	43 $\forall$E

Again, $\mathcal{P}$ and $\mathcal{Q}$ are as on page 663. The key to this argument is attaining $\mathcal{F}(\vec{x}, j, \beta(a, b, j))$ and $\mathcal{F}(\vec{x}, j, \beta(c, d, j))$ on lines (24) and (32). From these the assumption on (2) comes into play, and the result follows with other equalities.

First, *subtraction with cutoff*. The definition is not by recursion as before. However the effect is the same: $x \dot{-} y$ works like subtraction when $x \geq y$, and otherwise goes to $\emptyset$.

*Def[$\dot{-}$] Let PA $\vdash v = x \dot{-} y \leftrightarrow [x = y + v \vee (x < y \wedge v = \emptyset)]$.

(i) PA $\vdash \exists v[x = y + v \vee (x < y \wedge v = \emptyset)]$

(ii) PA $\vdash \forall m \forall n[([x = y + m \vee (x < y \wedge m = \emptyset)] \wedge [x = y + n \vee (x < y \wedge n = \emptyset)]) \rightarrow m = n]$

The proof of (i) and (ii) is left as an exercise. So PA defines ($\dot{-}$). And it proves a series of intuitive results.

*T13.21. The following are theorems of PA.

(a) PA $\vdash b \leq a \rightarrow a = b + (a \dot{-} b)$

*(b) PA $\vdash a \leq b \rightarrow a \dot{-} b = \emptyset$

(c) PA $\vdash a \dot{-} b \leq a$

*(d) PA $\vdash (a \leq r \wedge r \leq s) \rightarrow r \dot{-} a \leq s \dot{-} a$

(e) PA $\vdash (a \leq r \wedge r < s) \rightarrow r \dot{-} a < s \dot{-} a$

*(f) PA $\vdash b < a \leftrightarrow \emptyset < a \dot{-} b$

(g) PA $\vdash \emptyset < a \rightarrow a \dot{-} \overline{1} < a$

(h) PA $\vdash a \dot{-} \emptyset = a$

(i) PA $\vdash Sa \dot{-} a = \overline{1}$

(j) PA $\vdash a = Sa \dot{-} \overline{1}$

(k) PA $\vdash \emptyset < a \rightarrow a = S(a \dot{-} \overline{1})$

*(l) PA $\vdash Sb \leq a \rightarrow a \dot{-} b = S(a \dot{-} Sb)$

*(m) PA $\vdash c \leq a \rightarrow (a \dot{-} c) + b = (a + b) \dot{-} c$

*(n) PA $\vdash (b \leq a \wedge c \leq b) \rightarrow a \dot{-} (b \dot{-} c) = (a \dot{-} b) + c$

*(o) PA $\vdash (a \dot{-} b) \dot{-} c = a \dot{-} (b + c)$

(p) PA $\vdash (a + c) \dot{-} (b + c) = a \dot{-} b$

*(q) PA $\vdash a \times (b \dot{-} c) = (a \times b) \dot{-} (a \times c)$

(a) and (b) are from the definition and the basis upon which the rest depend. (c)–(l) are simple subtraction facts—except where the inequalities are required to protect against cases when $a \dot{-} b$ goes to $\emptyset$. And (m)–(q) are some results for association and distribution. For hints see E13.17 along with answers to the exercise.

Next *factor*. As for the recursive relation of Chapter 12, we say $m|n$ when $m+1$ divides n.

Def[|] Let PA $\vdash m|n \leftrightarrow \exists q(Sm \times q = n)$.

Since factor is a relation, no condition is required over and above the axiom so that the definition is good as it stands. And, again, PA proves a series of results. These are reasonably intuitive. Observe however that our choice to divide by $m+1$ means that, as in T13.22a below, $\emptyset|a$.

*T13.22. The following are theorems of PA.

(a) PA $\vdash \emptyset|a$

(b) PA $\vdash a|Sa$

(c) PA $\vdash a|\emptyset$

(d) PA $\vdash a|b \rightarrow a|(b \times c)$

*(e) PA $\vdash [(a \dot{-} \overline{1})|c \wedge (b \dot{-} \overline{1})|d] \rightarrow (ab \dot{-} \overline{1})|cd$

(f) PA $\vdash (a|Sb \wedge b|c) \rightarrow a|c$

*(g) PA $\vdash a|b \rightarrow [a|(b+c) \leftrightarrow a|c]$

(h) PA $\vdash (c \leq b \wedge a|b) \rightarrow [a|(b \dot{-} c) \leftrightarrow a|c]$

(i) PA $\vdash a < b \rightarrow b \nmid Sa$

(j) PA $\vdash a \leq b \rightarrow rm(a,b) = a$

*(k) PA $\vdash a|b \leftrightarrow rm(b,a) = \emptyset$

*(l) PA $\vdash rm[a + (y \times Sd), d] = rm(a,d)$

*(m) PA $\vdash Sd \times z \leq a \rightarrow z \leq qt(a,d)$

*(n) PA $\vdash y \times Sd \leq a \rightarrow rm[a \dot{-} (y \times Sd), d] = rm(a,d)$

So (a) (the successor of) $\emptyset$ divides any number; (b) (the successor of) a divides Sa; and (c) any number divides into $\emptyset$ zero times. (d) if a divides b then it divides $b \times c$; (e) where subtraction compensates for successor, if a divides c and b divides d, ab divides cd; and (f) if a divides Sb and (the successor of) b divides c, then a divides c. (g) is like $(b + c)/a = b/a + c/a$ so that dividing the sum breaks into dividing the members; (h) is the comparable principle for subtraction. From (i) if $a < b$, then (the successor of) b is not a factor of Sa; and (j) if $a \leq b$, then (the successor of) b divides into a zero times with remainder a. Then (k) makes the obvious connection between remainder and factor—given that remainder and factor (along with quotient and the recursive factor relation) divide by successor and so are consistently defined. In (l) the remainder of the second part $y \times Sd$ is $\emptyset$ so that the remainder of the sum is just whatever there is from the first part; (n) is the comparable principle for subtraction. The intervening (m) is required for (n) and tells us that if z multiples of (the successor of) d come to $\leq a$, then $z \leq qt(a, d)$—since the quotient maximizes the multiples of (the successor of) d that are $\leq a$.

And now PA defines relations *prime* and *relatively prime*. Prime has its usual sense. And numbers are relatively prime when they have no common divisor other than one—though they may not therefore individually be prime. Though division is by successor, these notions are given their usual sense by adjusting the numbers that are said to "divide."

Def[*Pr*] Let PA $\vdash Pr(n) \leftrightarrow [\overline{1} < n \wedge \forall x(x|n \to (Sx = \overline{1} \vee Sx = n))]$.

Def[*Rp*] Let PA $\vdash Rp(a, b) \leftrightarrow \forall x[(x|a \wedge x|b) \to Sx = \overline{1}]$.

Since these are relations, no condition is required over and above the axioms. At the limits, for any b we get $Rp(\overline{1}, b)$ since the only number that divides both $\overline{1}$ and b is (the successor of) $\emptyset$. By this reasoning, $Rp(\overline{1}, \emptyset)$. But for $a \neq \emptyset$ and so $Sa \neq \overline{1}$, $\sim Rp(Sa, \emptyset)$, for when $a \neq \emptyset$, both Sa and $\emptyset$ are divided by (the successor of) a and so by a number other than (the successor of) $\emptyset$.

To make progress with *Pr* and *Rp*, it will be helpful to introduce a couple of subsidiary notions. When $G(a, b, i)$ we say that i is *good*. Then supposing a and b are greater than zero, $d(a, b)$ is the *least* v such that its successor is good.

Def[G] Let PA $\vdash G(a, b, i) \leftrightarrow \exists x \exists y(ax + i = by)$.

Def[d] Let PA $\vdash d(a, b) = \mu v[(\emptyset < a \wedge \emptyset < b) \to G(a, b, Sv)]$.

(i) PA $\vdash \exists v[(\emptyset < a \wedge \emptyset < b) \to G(a, b, Sv)]$

Because d is defined by minimization, only the existence condition is required. Although we shall not prove the general result (showing it only for the special case when a and b are relatively prime), $G(a,b,i)$ when i is some multiple of the greatest common divisor of a and b (this is *Bézout's lemma*). If a or b is not greater than $\emptyset$ then vacuously $d(a,b)$ is just $\emptyset$. Otherwise $d(a,b)$ is the least i such that $G(a,b,Si)$ and so (in our sense of "division") the greatest common divisor itself.

And PA proves a series of results for these notions. Observe again that if we are interested in whether a prime divides some b we are interested in whether $Pr(Sa) \wedge a|b$ since it is the successor that is divided into b.

*T13.23. The following are theorems of PA.

(a) PA $\vdash \sim Pr(\emptyset)$

(b) PA $\vdash \sim Pr(\overline{1})$

(c) PA $\vdash Pr(\overline{2})$

*(d) PA $\vdash \overline{1} < a \rightarrow \exists z(Pr(Sz) \wedge z|a)$

*(e) PA $\vdash Rp(a,b) \leftrightarrow \sim\exists x[Pr(Sx) \wedge x|a \wedge x|b]$

(f) PA $\vdash G(a,b,m) \rightarrow G(a,b,m \times n)$

*(g) PA $\vdash (\emptyset < a \wedge \emptyset < b) \rightarrow$
$\forall m \forall n[(G(a,b,m) \wedge G(a,b,n) \wedge n \leq m) \rightarrow G(a,b,m \dot{-} n)]$

*(h) PA $\vdash [Rp(a,b) \wedge \emptyset < a \wedge \emptyset < b] \rightarrow G(a,b,\overline{1})$

*(i) PA $\vdash [Pr(Sa) \wedge a|(b \times c)] \rightarrow (a|b \vee a|c)$

The argument for (h) is relatively complex; its main stages are as in the box on the following page.

T13.23(a)–(c) are simple particular facts. From (d) every number greater than one is divided by some prime (which may or may not be itself). From (e), a and b are relatively prime iff there is no prime that divides them both; in one direction this is obvious—if a prime divides them both, then they are divided by a number other than (the successor of) zero; in the other direction, if some number other than (the successor of) zero divides them both, then some prime of it divides them both. (f), (g), and (h) are motivated insofar as $G(a,b,i)$ when i is a multiple of the greatest common divisor of a and b: (f) is so motivated directly; (g) if a and b have greatest common divisor g, then m is some gi and n is some gj and so $m - n = g(i - j)$ remains a multiple of g; then (h), if a and b are relatively prime, their greatest common divisor is one. (f) and (g) are required for (h), which is in turn required for (i). Then (i) is *Euclid's lemma* according to which if Sa is prime and Sa divides $b \times c$ then Sa divides b or divides c; if Sa is prime and divides $b \times c$ then it must appear in the factorization of b or the factorization of c—so that it divides one or the other.

Now *least common multiple*. Given a function $m(i)$, $lcm\{m(i) \mid i < k\}$ is the least $v > \emptyset$ such that for any $i < k$, (the successor of) $m(i)$ divides v. We avoid worries about the case when $m(i) = \emptyset$ by our usual account of factor.

*Def[lcm] Let PA $\vdash lcm\{m(i) \mid i < k\} = \mu v[\emptyset < v \wedge (\forall i < k)m(i)|v]$.

(i) PA $\vdash \exists x[\emptyset < x \wedge (\forall i < k)m(i)|x]$

As a matter of notation, let $l[m]_k = lcm\{m(i) \mid i < k\}$ and, where m is clear from context, let $l_k = lcm\{m(i) \mid i < k\}$. Also, it will be convenient (and easy) to define a predecessor to the least common multiple p_k, such that PA $\vdash Sp_k = l_k$.

*T13.24. The following are theorems of PA.

(a) PA $\vdash l_\emptyset = \overline{1}$

(b) PA $\vdash j < k \rightarrow m(j)|l_k$

*(c) PA $\vdash (\forall i < k)m(i)|x \rightarrow p_k|x$

*(d) PA $\vdash \forall n[(Pr(Sn) \wedge n|l_k) \rightarrow (\exists i < k)n|Sm(i)]$

So (a) for any function $m(i)$, the least common multiple under the (impossible) condition $i < \emptyset$ defaults to $\overline{1}$. (b) applies the definition for the result that when $j < k$,

T13.23h

1.	$(\emptyset < a \wedge \emptyset < b) \rightarrow G(a,b,Sd(a,b))$	Def[d] T13.18a
2.	$\quad Rp(a,b) \wedge \emptyset < a \wedge \emptyset < b$	A (g, $\rightarrow$I)
3.	$\quad Rp(a,b)$	2 $\wedge$E
4.	$\quad \forall x[(x|a \wedge x|b) \rightarrow Sx = \overline{1}]$	3 Def[Rp]
5.	$\quad \emptyset < a \wedge \emptyset < b$	2 $\wedge$E
6.	$\quad G(a,b,Sd(a,b))$	1,5 $\rightarrow$E
7.	$\quad G(a,b,a)$	[a]
8.	$\quad G(a,b,b)$	[b]
9.	$\quad \forall x[G(a,b,x) \rightarrow d(a,b)|x]$	[c]
10.	$\quad d(a,b)|a$	9,7 $\forall$E
11.	$\quad d(a,b)|b$	9,8 $\forall$E
12.	$\quad d(a,b)|a \wedge d(a,b)|b$	10,11 $\wedge$I
13.	$\quad Sd(a,b) = \overline{1}$	4,12 $\forall$E
14.	$\quad G(a,b,\overline{1})$	6,13 =E
15.	$[Rp(a,b) \wedge \emptyset < a \wedge \emptyset < b] \rightarrow G(a,b,\overline{1})$	2-14 $\rightarrow$I

So the argument reduces to [a]–[c]. For these, intuitively, a and b are automatically multiples of the greatest common divisor of a and b; and their greatest common divisor divides any multiple of the greatest common divisor. For hints, see the associated E13.19.

$m(j)$ divides $lcm\{m(i) \mid i < k\}$. (c) is perhaps best conceived by prime factorization: the least common multiple of some collection has all the primes of each member and no more; but any number into which all the members of the collection divide must include all those primes; so the least common multiple divides it as well. (d) is the related result that if a prime divides the least common multiple of some collection, then it divides some member of the collection.

Finally we arrive at the Chinese remainder theorem. Say $h(\vec{x}, i)$ and $m(\vec{x}, i)$ are functions with at least variable i free (but, for simplicity, we omit $\vec{x}$ from our specifications when it will do no harm). Then the theorem tells us that if for all $i < k$, $h(i)$ is less than or equal to $m(i)$, and if for all $i < j < k$, $Sm(i)$ and $Sm(j)$ are relatively prime, then $\exists p(\forall i < k)rm(p, m(i)) = h(i)$. So there is a p such that the remainder of p and $m(i)$ matches the value of $h(i)$.

*T13.25. PA $\vdash [(\forall i < k)h(i) \leq m(i) \wedge \forall i \forall j((i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j)))] \rightarrow \exists p(\forall i < k)rm(p, m(i)) = h(i)$.

Let,

$\mathcal{A}(k) = (\forall i < k)h(i) \leq m(i) \wedge \forall i \forall j((i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j)))$

$\mathcal{B}(k) = \exists p(\forall i < k)rm(p, m(i)) = h(i)$.

So we want PA $\vdash \mathcal{A}(k) \rightarrow \mathcal{B}(k)$. The argument is to show $\forall n(\mathcal{A}(n) \rightarrow \mathcal{B}(n))$ by induction on the value of n. See its overall structure on the next page.

Having obtained this important result, we are almost done! It remains to show that (i) in the case where $m(i) = q \times Si$ and so $rm(p, m(i)) = rm(p, q \times Si) = \beta(p, q, i)$, we can obtain the antecedent to T13.25 and so detach its consequent; and (ii) the values $h(i)$ to which remainders are matched may be derived from the original recursive conditions for the function $\mathscr{r}$—and so that there are p and q that satisfy the recursive conditions.

As preliminaries to showing that we can detach the consequent of T13.25, we require a couple notions for maximum value: First *maxp* for the greatest of a *pair* of values, and then *maxs* for the maximum from a *set*.

Def[*maxp*] Let PA $\vdash maxp(x, y) = \mu v[x \leq v \wedge y \leq v]$.

(i) PA $\vdash \exists v[x \leq v \wedge y \leq v]$

Hint: By T13.11s, PA $\vdash y \leq x \vee x < y$; in either case the result is easy.

Def[*maxs*] Let PA $\vdash maxs\{m(i) \mid i < k\} = \mu v[(\forall i < k)m(i) \leq v]$.

(i) PA $\vdash \exists v[(\forall i < k)m(i) \leq v]$

For hints see the associated exercise E13.22.

T13.25

1.	$\mathscr{A}(\emptyset) \to \mathscr{B}(\emptyset)$	[a]
2.	$\mathscr{A}(a) \to \mathscr{B}(a)$	A (g, $\to$I)
3.	$\mathscr{A}(Sa)$	A (g, $\to$I)
4.	$[(\forall i < a)h(i) \le m(i) \land \forall i \forall j((i < j \land j < a) \to \mathcal{R}p(Sm(i), Sm(j)))]$ $\to \exists p(\forall i < a)\mathcal{rm}(p, m(i)) = h(i)$	2 abv
5.	$(\forall i < Sa)h(i) \le m(i) \land \forall i \forall j((i < j \land j < Sa) \to \mathcal{R}p(Sm(i), Sm(j)))$	3 abv
6.	$(\forall i < Sa)h(i) \le m(i)$	5 $\land$E
7.	$\forall i \forall j((i < j \land j < Sa) \to \mathcal{R}p(Sm(i), Sm(j)))$	5 $\land$E
8.	$\exists p(\forall i < a)\mathcal{rm}(p, m(i)) = h(i)$	[b]
9.	$(\forall i < a)\mathcal{rm}(r, m(i)) = h(i)$	A (g, 8$\exists$E)
10.	$\mathcal{R}p(l[m]_a, Sm(a))$	[c]
11.	$\emptyset < Sm(a)$	T13.11f
12.	$\emptyset < l_a$	*Def*[*lcm*]
13.	$\mathcal{R}p(l[m]_a, Sm(a)) \land \emptyset < l_a \land \emptyset < Sm(a)$	10,12,11 $\land$I
14.	$G(l_a, Sm(a), \overline{1})$	13 T13.23h
15.	$G(l_a, Sm(a), r + (l_a \dot{-} \overline{1}) \times h(a))$	14 T13.23f
16.	$\exists x \exists y(l_a \times x + [r + (l_a \dot{-} \overline{1}) \times h(a)] = Sm(a) \times y)$	15 *Def*[G]
17.	$l_a \times b + [r + (l_a \dot{-} \overline{1}) \times h(a)] = Sm(a) \times c$	A (g, 16$\exists$E)
18.	$s = l_a \times (b + h(a)) + r$	def
19.	$s = Sm(a) \times c + h(a)$	[d]
20.	$(\forall i < Sa)\mathcal{rm}(s, m(i)) = h(i)$	[e]
21.	$\exists p(\forall i < Sa)\mathcal{rm}(p, m(i)) = h(i)$	20 $\exists$I
22.	$\mathscr{B}(Sa)$	21 abv
23.	$\mathscr{B}(Sa)$	16,17-22 $\exists$E
24.	$\mathscr{B}(Sa)$	8,9-23 $\exists$E
25.	$\mathscr{A}(Sa) \to \mathscr{B}(Sa)$	3-24 $\to$I
26.	$(\mathscr{A}(a) \to \mathscr{B}(a)) \to (\mathscr{A}(Sa) \to \mathscr{B}(Sa))$	2-25 $\to$I
27.	$\forall n[(\mathscr{A}(n) \to \mathscr{B}(n)) \to (\mathscr{A}(Sn) \to \mathscr{B}(Sn))]$	26 $\forall$I
28.	$\forall n(\mathscr{A}(n) \to \mathscr{B}(n))$	1,27 IN
29.	$\mathscr{A}(k) \to \mathscr{B}(k)$	28 $\forall$E

[b,c] First lines (8) and so (9) use (6) and (7) to obtain the antecedent of (4) and so to detach its consequent. Then from (7) we get (10) according to which $l[m]_a$, and $Sm(a)$ are relatively prime: intuitively if, as from (7), some $d \dots e$ and f are relatively prime, they have no primes in common; but the least common multiple of $d \dots e$ has just the primes of $d \dots e$; so their least common multiple has no primes in common with f. Then from (10), we obtain (16) and so (17).

[d,e] Term s is constructed so that for any $i < Sa$ the remainder of s and $m(i)$ is $h(i)$: For $i < a$ and any z, $m(i)$ divides $l_a \times z$ evenly; so $m(i)$ divides the first term from (18) evenly; so the remainder of s and $m(i)$ is the same as the remainder of r with $m(i)$—and with (9) this is just $h(i)$. But by (17), (18), and simple arithmetic we get (19); so when $i = a$, $m(i)$ divides the first term evenly, and since with (6), $h(a) \le m(a)$, again the remainder of s and $m(i)$ is $h(i)$. And from these, for any $i < Sa$, the remainder of s and $m(i)$ is $h(i)$.

The "trick" to this is the construction of s so that remainders for $i < a$ stay the same, but the remainder at a is $h(a)$. For this construction see Boolos, *The Logic of Provability*, pages 30–31. Hints and derivations for [a]–[e] are associated with E13.21.

So $maxp(x, y)$ is the maximum of x and y, and $maxs\{m(i) \mid i < k\}$ is the maximum from $m(i)$ with $i < k$. As a matter of notation, let $maxs[m]_k = maxs\{m(i) \mid i < k\}$ and where m is understood, $maxs_k = maxs\{m(i) \mid i < k\}$. A couple of results are immediate with T13.18a.

T13.26. The following are theorems of PA.

(a) PA $\vdash x \leq maxp(x, y) \wedge y \leq maxp(x, y)$

(b) PA $\vdash (\forall i < k)m(i) \leq maxs_k$

These simply state the obvious: that the maximum is greater than or equal to the rest.

Now with values of q and $m(i)$ as below, we demonstrate the antecedent to T13.25, and so obtain its consequent—where this is a result for β.

*T13.27. PA $\vdash \exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$. Let,

$r = maxp(k, maxs[h]_k)$

$q = lcm\{i \mid i < Sr\}$

$m(i) = q \times Si$

Recall from *Def*[*beta*] that PA $\vdash \beta(p, q, i) = rm(p, q \times Si)$. Then we may reason,

1.	$m(i) = q \times Si$	def m
2.	$\beta(p, q, i) = rm(p, q \times Si)$	*Def*[*beta*]
3.	$(\forall i < k)h(i) \leq m(i)$	[i]
4.	$\forall i \forall j[(i < j \wedge j < k) \to Rp(Sm(i), Sm(j))]$	[ii]
5.	$\exists p(\forall i < k)rm(p, m(i)) = h(i)$	3,4 T13.25
6.	$\quad (\forall i < k)rm(p, m(i)) = h(i)$	A (g 5$\exists$E)
7.	$\quad\quad i < k$	A (g, ($\forall$I))
8.	$\quad\quad rm(p, m(i)) = h(i)$	6,7 ($\forall$E)
9.	$\quad\quad rm(p, q \times Si) = h(i)$	8,1 =E
10.	$\quad\quad \beta(p, q, i) = h(i)$	9,2 =E
11.	$\quad (\forall i < k)\beta(p, q, i) = h(i)$	7-10 ($\forall$I)
12.	$\quad \exists q(\forall i < k)\beta(p, q, i) = h(i)$	11 $\exists$I
13.	$\quad \exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$	12 $\exists$I
14.	$\exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$	5,6-13 $\exists$E

So the demonstration reduces to that of [i] and [ii], the two conjuncts to the antecedent of T13.25. For the main structure of [ii], see the derivation on page 677.

Now, moving toward the result that values for $h(i)$ may derive from the recursive conditions, a theorem that uses the above result to show that a β-function for values $< k$ can always be extended to another like it but with an arbitrary k^{th} value. We show that given $\beta(a,b,i)$ there are sure to be p and q such that $\beta(p,q,i)$ is like $\beta(a,b,i)$ for $i < k$ and for arbitrary n, $\beta(p,q,k) = n$. This is because we may *define* a function h which is like $\beta(a,b,i)$ for $i < k$ and otherwise n—and find p,q such that $\beta(p,q,i)$ matches it. Thus,

**Def*[$h(i)$] Let PA $\vdash v = h(i) \leftrightarrow [(i < k \wedge v = \beta(a,b,i)) \vee (k \leq i \wedge v = n)]$.

(i) PA $\vdash \exists v[(i < k \wedge v = \beta(a,b,i)) \vee (k \leq i \wedge v = n)]$

(ii) PA $\vdash \forall x \forall y[([(i < k \wedge x = \beta(a,b,i)) \vee (k \leq i \wedge x = n)] \wedge$
$[(i < k \wedge y = \beta(a,b,i)) \vee (k \leq i \wedge y = n)]) \rightarrow x = y]$

Observe that our notation $h(i)$ and $m(i)$ obscures the point that, in addition to variable i, function h has k, n, a, b free (and from definitions for T13.27, free variables of m take over all the free variables from h). Through T13.27, the the extra variables play no role; now they matter.

*T13.28. PA $\vdash \exists p \exists q[(\forall i < k)\beta(p,q,i) = \beta(a,b,i) \wedge \beta(p,q,k) = n]$

From T13.27—for hints see the associated exercise E13.24.

For application of this theorem it is important that the free variables k, n, a, b, may be instantiated in the usual way—in particular, we shall be interested in a case with k instantiated to Sj and n to $\mathscr{h}(\vec{x}, j, \beta(a,b,j))$. This theorem already suggests that we can take an arbitrary sequence and *extend* it according to recursive conditions.

Finally, then, the result we have been after in this section: As before, let $\mathcal{F}(\vec{x}, y, v)$ be our formula,

$$\exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = v]$$

*T13.29. For $\mathcal{F}$ as above, PA $\vdash \exists v \mathcal{F}(\vec{x}, y, v)$.

Let $\mathcal{F}(\vec{x}, y, v)$ be as above; the argument is by IN on y. The zero case is left as an exercise. See the derivation on page 678 for the main argument.

This completes the demonstration of T13.19! So for any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathcal{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathscr{r}(\vec{x})$ such that PA $\vdash v = \mathscr{r}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$. In particular, then, PA defines functions corresponding to all the primitive recursive functions from Chapter 12.

Exercises that follow include some (quite) extended derivations. For this, a good strategy is to set up a main page for overall structure, with subparts shifted to auxiliary sheets, as we have done in our examples. Alternatively, you might work in an electronic context without size limits.[12]

[12] Again, one such option is SLAPP. But its capacities are stretched by exercises in this volume. Though it requires some startup effort and is not a "what you see is what you get" processor, an especially flexible computer option is LaTeX; for this see `https://tonyroyphilosophy.net/symbolic-logic/`.

T13.27[ii]

1.	$i < j \wedge j < k$	A (g →I)
2.	$i < j$	1 ∧E
3.	$j < k$	1 ∧E
4.	$\sim Rp(Sm(i), Sm(j))$	A (c, ∼E)
5.	$\exists x[Pr(Sx) \wedge x \mid Sm(i) \wedge x \mid Sm(j)]$	4 T13.23e
6.	$Pr(Sa) \wedge a \mid Sm(i) \wedge a \mid Sm(j)$	A (c, 5∃E)
7.	$Pr(Sa) \wedge a \mid S(q \times Si) \wedge a \mid S(q \times Sj)$	6 def $m(i)$
8.	$Pr(Sa)$	7 ∧E
9.	$a \mid S(q \times Si)$	7 ∧E
10.	$a \mid S(q \times Sj)$	7 ∧E
11.	$a \mid q(j \dot{-} i)$	[a]
12.	$a \mid q \vee a \mid (j \dot{-} i)$	8,11 T13.23i
13.	$a \mid q$	A (g, 12∨E)
14.	$a \mid q$	13 R
15.	$a \mid (j \dot{-} i)$	A (g, 12∨E)
16.	$a \mid q$	[b]
17.	$a \mid q$	12,13-14,15-16 ∨E
18.	$a \mid (q \times Si)$	17 T13.22d
19.	$q \times Si < S(q \times Si)$	T13.11j
20.	$q \times Si \leq S(q \times Si)$	19 T13.11n
21.	$a \mid (S(q \times Si) \dot{-} (q \times Si)) \leftrightarrow a \mid (q \times Si)$	20,9 T13.22h
22.	$a \mid (S(q \times Si) \dot{-} (q \times Si))$	21,18 ↔E
23.	$a \mid \overline{1}$	22 T13.21i
24.	$S\emptyset < Sa$	8 *Def*[*Pr*]
25.	$\emptyset < a$	24 T13.11l
26.	$a \nmid \overline{1}$	25 T13.22i
27.	$\perp$	23,26 ⊥I
28.	$\perp$	5,6-27 ∃E
29.	$Rp(Sm(i), Sm(j))$	4-28 ∼E
30.	$(i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j))]$	1-29 →I
31.	$\forall i \forall j[(i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j))]$	30 ∀I

Hints. [a]: With $i < j$ you will be able to show $a \mid (S(q \times Sj) \dot{-} S(q \times Si))$; and with some work that $S(q \times Sj) \dot{-} S(q \times Si) = q(j \dot{-} i)$. [b]: With $i < j$, you have $\emptyset < j \dot{-} i$; so there is an l such that $j \dot{-} i = Sl$; then $a \mid Sl$ and with T13.24b, $l \mid q$ so with T13.22f, $a \mid q$.

T13.29

1.	$\exists v \mathcal{F}(\vec{x}, \emptyset, v)$	zero case
2.	$\exists v \mathcal{F}(\vec{x}, j, v)$	A (g, $\rightarrow$I)
3.	$\exists v \exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, j) \wedge \beta(p, q, j) = v]$	2 abv
4.	$\mathcal{P}(a, b, \vec{x}) \wedge \mathcal{Q}(a, b, \vec{x}, j) \wedge \beta(a, b, j) = z$	A (g, 3$\exists$E)
5.	$\beta(a, b, \emptyset) = \mathbb{g}(\vec{x})$	4 $\wedge$E ($\mathcal{P}$)
6.	$(\forall i < j)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	4 $\wedge$E ($\mathcal{Q}$)
7.	$\exists p \exists q[(\forall i < Sj)\beta(p, q, i) = \beta(a, b, i) \wedge \beta(p, q, Sj) = \mathbb{h}(\vec{x}, j, \beta(a, b, j))]$	T13.28
8.	$(\forall i < Sj)\beta(c, d, i) = \beta(a, b, i) \wedge \beta(c, d, Sj) = \mathbb{h}(\vec{x}, j, \beta(a, b, j))$	A (g, 7$\exists$E)
9.	$(\forall i < Sj)\beta(c, d, i) = \beta(a, b, i)$	8 $\wedge$E
10.	$\beta(c, d, Sj) = \mathbb{h}(\vec{x}, j, \beta(a, b, j))$	8 $\wedge$E
11.	$\emptyset < Sj$	T13.11f
12.	$\beta(c, d, \emptyset) = \beta(a, b, \emptyset)$	9,11 ($\forall$E)
13.	$\beta(c, d, \emptyset) = \mathbb{g}(\vec{x})$	5,12 =E
14.	$l < Sj$	A (g, ($\forall$I))
15.	$\beta(c, d, l) = \beta(a, b, l)$	9,14 ($\forall$E)
16.	$l < j \vee l = j$	14 T13.11o
17.	$l < j$	A (g, 16$\vee$E)
18.	$\mathbb{h}(\vec{x}, l, \beta(a, b, l)) = \beta(a, b, Sl)$	6,17 ($\forall$E)
19.	$Sl < Sj$	17 T13.11l
20.	$\beta(c, d, Sl) = \beta(a, b, Sl)$	9,19 ($\forall$E)
21.	$\mathbb{h}(\vec{x}, l, \beta(a, b, l)) = \beta(c, d, Sl)$	18,20 =E
22.	$l = j$	A (g, 16$\vee$E)
23.	$\mathbb{h}(\vec{x}, l, \beta(a, b, l)) = \beta(c, d, Sl)$	10,22 =E
24.	$\mathbb{h}(\vec{x}, l, \beta(a, b, l)) = \beta(c, d, Sl)$	16,17-23 $\vee$E
25.	$\mathbb{h}(\vec{x}, l, \beta(c, d, l)) = \beta(c, d, Sl)$	24,15 =E
26.	$(\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si)$	14-25 ($\forall$I)
27.	$\beta(c, d, Sj) = \beta(c, d, Sj)$	=I
28.	$\mathcal{P}(c, d, \vec{x}) \wedge \mathcal{Q}(c, d, \vec{x}, Sj) \wedge \beta(c, d, Sj) = \beta(c, d, Sj)$	13,26,27 $\wedge$I
29.	$\exists v \exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, Sj) \wedge \beta(p, q, Sj) = v]$	28 $\exists$I
30.	$\exists v \mathcal{F}(\vec{x}, Sj, v)$	29 abv
31.	$\exists v \mathcal{F}(\vec{x}, Sj, v)$	7,8-30 $\exists$E
32.	$\exists v \mathcal{F}(\vec{x}, Sj, v)$	3,4-31 $\exists$E
33.	$\exists v \mathcal{F}(\vec{x}, j, v) \rightarrow \exists v \mathcal{F}(\vec{x}, Sj, v)$	2-32 $\rightarrow$I
34.	$\forall y[\exists v \mathcal{F}(\vec{x}, y, v) \rightarrow \exists v \mathcal{F}(\vec{x}, Sy, v)]$	33 $\forall$I
35.	$\exists v \mathcal{F}(\vec{x}, y, v)$	1,34 IN

$\mathcal{P}$ and $\mathcal{Q}$ are as on page 663. From the assumption (2), there are a, b such that the β-function has the right features for every $i < j$. With T13.28 there are c, d such that the β-function has the right features for $i < Sj$. The derivation establishes that this is so and generalizes.

*E13.17. Show (i) and (ii) for $\mathit{Def}[\dot{-}]$. Then show T13.21 (a) and (p). Hard-core: show all of the results in T13.21.

Hints for T13.21. (f): From left to right, with the assumption for $\leftrightarrow$I and then $Sj + b = a$ for $\exists$E, you can obtain $a = b + (a \dot{-} b)$; then you have what you need with T6.73; in the other direction, $a < b$ or $b \leq a$ one is impossible, the other gives what you want. (m): With the assumption $c \leq a$, you have also $c \leq a + b$; so that both $a = c + (a \dot{-} c)$ and $a + b = c + [(a + b) \dot{-} c]$; then =E and T6.73 do the work. (n): You can get this with a couple applications of (m). (o): Begin with $b + c \leq a \vee a < b + c$; in the first case, you will be able to show that $(b + c) + [(a \dot{-} b) \dot{-} c] = (b + c) + [a \dot{-} (b + c)]$ and apply T6.73; for the other, $b \leq a \vee a < b$. (q): Begin with $\emptyset = a \vee \emptyset < a$; in the first case, both sides equal $\emptyset$; for in the second, $c \leq b \vee b < c$; in the first of these cases, you will be able to show $ac + a(b \dot{-} c) = ac + (ab \dot{-} ac)$ and apply T6.73; for the second, again both sides equal $\emptyset$.

*E13.18. Show T13.22d and T13.22i. Hard-core: show all of the results in T13.22.

Hints for T13.22. (g): Under assumptions for $\rightarrow$I and then $Sa \times j = b$ for $\exists$E, set up for $\leftrightarrow$I; the argument from right to left is not hard; in the other direction, with $a|(b + c)$ and then $Sa \times k = b + c$ for $\exists$E, you will be able to show $j \leq k$, so that for some l, $l + j = k$; $a|c$ follows. (l): Let $r = rm(a, d)$; then from the definition and $\exists$E you have $a = (Sd \times j) + r \wedge r < Sd$; if you assert $a + (y \times Sd) = a + (y \times Sd)$ by =I you should be able to show $a + (y \times Sd) = Sd \times (j + y) + r \wedge r < Sd$, and the result from this. (m): Let $r = rm(a, d)$ and $q = qt(a, d)$; then by the definitions you have $a = Sd \times q + r$ and $r < Sd$; assume $Sd \times z \leq a$ for $\rightarrow$I and $q < z$ for $\sim$I; then you should be able to show $a < Sd \times z$ to contradict the assumption for $\rightarrow$I. (n): Again let $r = rm(a, d)$ and $q = qt(a, d)$; then by the definitions you have $a = Sd \times q + r$ and $r < Sd$; assume $y \times Sd \leq a$ for $\rightarrow$I; you should be able to show $a \dot{-} (y \times Sd) = Sd(q \dot{-} y) + r \wedge r < Sd$; then you need $(\exists w < a \dot{-} (y \times Sd))[a \dot{-} (y \times Sd) = Sd \times w + r \wedge r < Sd]$ to apply $\mathit{Def}[rm]$.

*E13.19. Provide a demonstration of the condition for $\mathit{Def}[d]$ along with a complete demonstration of T13.23h. Hard-core: Show all of the results from T13.23.

Hint for $\mathit{Def}[d]$. Begin with $\emptyset \not< b \vee \emptyset < b$ and go for the existentially quantified goal. In the second case, there is some j such that $b = Sj$ and it is easy to show $a \times \emptyset + Sj = b \times \bar{1}$ and generalize.

Hints for T13.23h. [a]: Show $a \times (b \dot{-} \bar{1}) + a = b \times a$ and generalize. [b]: Show $a \times \emptyset + b = b \times \bar{1}$ and generalize. [c]: Go for $G(a, b, i) \rightarrow d(a, b)|i$ toward an application of $\forall$I; then under the assumption $G(a, b, i)$ for $\rightarrow$I, let

$q = qt(i, d(a,b))$ and $r = rm(i, d(a,b))$; then from the definitions you have $i = (Sd(a,b) \times q) + r$, and $r < Sd(a,b)$, and $(\forall y < d(a,b))\sim[(\emptyset < a \wedge \emptyset < b) \rightarrow G(a,b,Sy)]$; and you should be able to show $G(a,b,i \dot{-} (Sd(a,b) \times q))$ using (6) with T13.23f and T13.23g; but also $i \dot{-} (Sd(a,b) \times q) = r$ so that $G(a,b,r)$; now an assumption that r is a successor leads to contradiction; so $r = \emptyset$ and $d(a,b)|i$.

Hints for the rest of T13.23. (c): This is straightforward with T13.22i. (d): You can do this by the second form of strong induction T13.11ai; the zero case is trivial; to reach $\forall x\{(\forall y \leq x)[\overline{1} < y \rightarrow \exists z(Pr(Sz) \wedge z|y)] \rightarrow [\overline{1} < Sx \rightarrow \exists z(Pr(Sz) \wedge z|Sx)]\}$ assume $(\forall y \leq k)[\overline{1} < y \rightarrow \exists z(Pr(Sz) \wedge z|y)]$ and $\overline{1} < Sk$; then Sk is prime or not; if it is prime, the result is immediate; if it is not, for some j you will be able to show $Sj \leq k$ and apply the assumption. (e): From left to right, under the assumption for $\leftrightarrow$I assume $\exists x[Pr(Sx) \wedge x|a \wedge x|b]$ and $Pr(Sj) \wedge j|a \wedge j|b$ for $\sim$I and $\exists$E; then you should be able to show that $\overline{1} < Sj$ and $\overline{1} \not< Sj$; in the other direction, under the assumption for $\leftrightarrow$I and then $j|a \wedge j|b$ for $\rightarrow$I, suppose $\emptyset < j$; this is impossible, which gives the result you want. (g): Under the assumptions $\emptyset < a \wedge \emptyset < b$ and then $G(a,b,i) \wedge G(a,b,j) \wedge j \leq i$ for $\rightarrow$I and then $ap + i = bq$ and $ar + j = bs$ for $\exists$E, starting with $(bq + bar) + (bsa \dot{-} bs) = (bq + bar) + (bsa \dot{-} bs)$ by =I, with some effort, you will be able to show $a[(p + bs) + (br \dot{-} r)] + (i \dot{-} j) = b[(q + ar) + (sa \dot{-} s)]$ and generalize. (i): Under the assumption $Pr(Sa) \wedge a|(b \times c)$ assume $a \nmid b$ with the idea of obtaining $a \nmid b \rightarrow a|c$ for Impl; set out to show $Rp(b, Sa)$ for an application of T13.23h to get $\exists x \exists y[bx + \overline{1} = Sa \times y]$; with this, you will have $bp + \overline{1} = Sa \times q$ by $\exists$E; and you should be able to show $a|cbp$ and $a|(cbp + c)$ for an application of T13.22g.

*E13.20. Show the condition for *Def*[*lcm*] and provide a demonstration for T13.24d. Hard-core: show the rest of the results T13.24.

Hint for *Def*[*lcm*]. This is an argument by IN on k. For the basis, show $\emptyset < \overline{1} \wedge (\forall i < \emptyset)m(i)|\overline{1}$ and generalize. For the main argument, under the assumptions $\exists x[\emptyset < x \wedge (\forall i < j)m(i)|x]$ for $\rightarrow$I and $\emptyset < a \wedge (\forall i < j)m(i)|a$ for $\exists$E, set out to show $\emptyset < a \times Sm(j) \wedge (\forall i < Sj)m(i)|(a \times Sm(j))$ and generalize.

Hints for T13.24. For (c) and (d) define predecessor to the least common multiple; for this, let PA $\vdash v = plcm\{m(i) \mid i < k\} \leftrightarrow Sv = lcm\{m(i) \mid i < k\}$; then $p[m]_k = plcm\{m(i) \mid i < k\}$ and, where m is clear from context, $p_k = plcm\{m(i) \mid i < k\}$; show that PA defines *plcm*. Then (c): Under the assumption $(\forall i < k)m(i)|x$ for $\rightarrow$I, let $r = rm(x, p_k)$ and $q = qt(x, p_k)$; you have $(\forall y < l_k)\sim[\emptyset < y \wedge (\forall i < k)m(i)|y]$ from *Def*[*lcm*] with T13.18b; you should be able to apply this to show that $r = \emptyset$ and so that $p_k|x$. (d): This is an induction on k; for the show, you have $(\forall i < j)m(i)|l_j$ from *Def*[*lcm*]; then under assumptions $\forall n[(Pr(Sn) \wedge n|l_j) \rightarrow (\exists i < j)n|Sm(i)]$ and $Pr(Sa) \wedge a|l_{Sj}$ for $\rightarrow$I, you should be able to use T13.24c to show $p_{Sj}|(l_j \times Sm(j))$; and from this $a|l_j \vee a|Sm(j)$; in either case, you have your result.

*E13.21. Provide derivations to show parts [c] and [e] to the derivation for T13.25. Hard-core: complete the entire derivation.

Hints. With line umbers from the text outline, [a]: Trivially $(\forall i < \emptyset)rm(\emptyset, m(i)) = h(i)$; this gives you $\mathscr{B}(\emptyset)$ and so the result. [b]: You will be able to use (6) and (7) to generate the antecedent to (4). [c]: Suppose otherwise; with T13.23e there is a u such that $Pr(Su) \wedge u|l_a \wedge u|Sm(a)$; then with T13.24d there is a $v < a$ such that $u|Sm(v)$ so that with (7) $Rp(Sm(v), Sm(a))$. But this is impossible with $u|Sm(a)$, $u|Sm(v)$ and T13.23e. [d]: By *Def*[*lcm*], $\emptyset < l_a$ so that $h(a) < h(a)l_a$. Then with T13.21a and T13.21q you can show $s = (l_a \times b + [r + (l_a \dot{-} \overline{1}) \times h(a)]) + h(a)$ and apply (17). [e]: Suppose for $(\forall$I$)$ $u < Sa$; then $u < a \vee u = a$. In the first case, with T13.24b and T13.22d $m(u)|l_a(b + h(a))$; so that there is a v such that $Sm(u)v = l_a(b + h(a))$; then using (18) and T13.22l, $rm(s, m(u)) = rm(r, m(u))$; so that you can apply (9). In the second case, with (19) and T13.22l $rm(s, m(u)) = rm(h(u), m(u))$; but from (6), $h(u) \leq m(u)$ and with T13.22j $rm(h(u), m(u)) = h(u)$.

E13.22. Given *maxp* and T13.26a provide a derivation to show the condition of *Def*[*maxs*]. Hard-core: Provide justification *Def*[*maxp*]; and show the results in T13.26 as well.

Hint for *Def*[*maxs*]. First obtain *maxp* and T13.26a. Then the argument is by IN on k. For the show you will have assumptions of the sort $(\forall i < j)m(i) \leq l$ and $a < Sj$; then $a < j \vee a = j$; in either case you will be able to show that $m(a) \leq maxp(l, m(j))$.

*E13.23. Provide a demonstration for [i] of T13.27. Hard-core: Provide a complete demonstration for T13.27.

Hints for [i]. Under the assumption $j < k$ for $(\forall$I$)$ you will be able to show $h(j) < Sr$, and with T13.24b $r|q$; from this, $Sr \leq q$ so that $Sr \leq q \times Sj$ which gives the result you want. For [ii] see page 677.

*E13.24. Show T13.28. Hard-core: show also the conditions for *Def*[$h(i)$].

Hint for *Def*[$h(i)$]. (i) is straightforward under $i < k \vee k \leq i$ from T13.11s.

Hints for T13.28. From T13.27 applied to Sk, $\exists p \exists q(\forall i < Sk)\beta(p, q, i) = h(i)$; then with $(\forall i < Sk)\beta(c, d, i) = h(i)$ for $\exists$E, you will be able to obtain both

$(\forall i < k)\beta(c,d,i) = \beta(a,b,i)$ and $\beta(c,d,k) = n$. For the second of these, you have $(k < k \wedge h(k) = \beta(a,b,k)) \vee (k \leq k \wedge h(k) = n)$ from *Def*$[h(i)]$; for the first, under $l < k$ for ($\forall$I) you can apply *Def*$[h(i)]$ again to show that $\beta(c,d,l) = \beta(a,b,l)$.

*E13.25. Complete the demonstration of T13.29 by showing the zero case.

Hint: From *Def*$[beta]$ and T13.22j, you have $\beta(g(\vec{x}), g(\vec{x}), \emptyset) = g(\vec{x})$; so $g(\vec{x})$ is an a such that $\beta(a,a,\emptyset) = g(\vec{x})$. Then the result is easy.

Coordinate Functions and Relations

We conclude this section showing first that the functions we have defined in PA are *coordinate* to friendly recursive functions. Then, turning to relations, we show that PA defines relations coordinate to the recursive relations of Chapter 12.

To show that functions defined in PA are coordinate to the friendly recursive functions, we begin with a couple of preliminaries: First, as sort of addendum to T13.19, if some recursive function $\mathsf{f}(\vec{\mathsf{x}})$ just is some previously defined $\mathsf{g}(\vec{\mathsf{x}})$, let PA $\vdash \mathscr{f}(\vec{x}) = \mathscr{g}(\vec{x})$. Then PA defines $\mathscr{f}(\vec{x})$. Second, a result fundamental to every case where a function is defined by recursion. As above let $\mathscr{F}(\vec{x}, y, v)$ be,

$$\exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = v]$$

and suppose PA $\vdash v = \mathscr{f}(\vec{x}, y) \leftrightarrow \mathscr{F}(\vec{x}, y, v)$ so that $\mathscr{f}(\vec{x}, y)$ is defined by recursion. Then the standard recursive conditions apply. That is,

T13.30. Suppose $\mathscr{f}(\vec{x}, y)$ is defined by $\mathscr{g}(\vec{x})$ and $\mathscr{h}(\vec{x}, y, u)$ so that PA $\vdash v = \mathscr{f}(\vec{x}, y) \leftrightarrow \mathscr{F}(\vec{x}, y, v)$. Then,

(a) PA $\vdash \mathscr{f}(\vec{x}, \emptyset) = \mathscr{g}(\vec{x})$

(b) PA $\vdash \mathscr{f}(\vec{x}, Sy) = \mathscr{h}(\vec{x}, y, \mathscr{f}(\vec{x}, y))$

Hint: (a) follows easily in just a few lines with PA $\vdash \mathscr{F}(\vec{x}, \emptyset, \mathscr{f}(\vec{x}, \emptyset))$ from the definition for $\mathscr{f}(\vec{x}, y)$. For (b), see the derivation on the next page.

From this theorem, our functions defined by recursion behave like ones we have seen before, with clauses for the basis and successor. This lets us manipulate the functions very much as before. The importance of this point emerges immediately below.

Now we can show that definitions of functions from Chapter 12 are coordinate with definitions in PA. Coordinate definitions result in a sort of structural similarity:

Cf The definition of a recursive function (except a function defined by bounded minimization) is *coordinate* with its definition in PA iff,

(i) $\mathsf{f}(\vec{\mathsf{x}})$ is an initial function zero, suc, or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}$ and $\mathscr{f}(\vec{x})$ is $\mathscr{zero}$, $\mathscr{suc}$, or $\mathscr{idnt}^{j}_{k}$, and PA $\vdash v = \mathscr{zero}() \leftrightarrow \emptyset = v$, PA $\vdash v = \mathscr{suc}(x) \leftrightarrow Sx = v$, and PA $\vdash v = \mathscr{idnt}^{j}_{k}(x_1 \dots x_j) \leftrightarrow [(x_1 = x_1 \wedge \dots \wedge x_j = x_j) \wedge x_k = v]$.

(c) $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is defined from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ by composition so that $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}}) = \mathsf{h}(\vec{\mathsf{x}}, \mathsf{g}(\vec{\mathsf{y}}), \vec{\mathsf{z}})$, and for coordinate $\mathscr{g}(\vec{x})$ and $\mathscr{h}(\vec{x}, w, \vec{z})$, PA $\vdash$ $\mathscr{f}(\vec{x}, \vec{y}, \vec{z}) = \mathscr{h}(\vec{x}, \mathscr{g}(\vec{y}), \vec{z})$.

(r) $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ is defined from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by recursion so that $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{0}) = \mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{Sy}) = \mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{f}(\vec{\mathsf{x}}, \mathsf{y}))$ and for coordinate $\mathscr{g}(\vec{x})$ and $\mathscr{h}(\vec{x}, y, u)$, PA $\vdash$ $\mathscr{f}(\vec{x}, \emptyset) = \mathscr{g}(\vec{x})$ and PA $\vdash$ $\mathscr{f}(\vec{x}, Sy) = \mathscr{h}(\vec{x}, y, \mathscr{f}(\vec{x}, y))$.

(m) $\mathsf{f}(\vec{\mathsf{x}})$ is defined from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ by friendly regular minimization so that $\mathsf{f}(\vec{\mathsf{x}}) = \mu\mathsf{y}[\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y}) = \mathsf{zero}()]$ and for coordinate $\mathscr{g}(\vec{x}, y)$, PA $\vdash$ $\mathscr{f}(\vec{x}) = \mu y[\mathscr{g}(\vec{x}, y) = \mathscr{zero}()]$.

(e) $\mathsf{f}(\vec{\mathsf{x}})$ just is some $\mathsf{g}(\vec{\mathsf{x}})$ so that $\mathsf{f}(\vec{\mathsf{x}}) = \mathsf{g}(\vec{\mathsf{x}})$, and for coordinate $\mathscr{g}(\vec{x})$, PA $\vdash$ $\mathscr{f}(\vec{x}) = \mathscr{g}(\vec{x})$.

T13.30(b)

1.	$\mathcal{F}(\vec{x}, Sy, \mathscr{f}(\vec{x}, Sy))$	def $\mathscr{f}(\vec{x}, y)$
2.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, Sy) \wedge \beta(p, q, Sy) = \mathscr{f}(\vec{x}, Sy)]$	1 abv
3.	$\mathcal{P}(a, b, \vec{x}) \wedge \mathcal{Q}(a, b, \vec{x}, Sy) \wedge \beta(a, b, Sy) = \mathscr{f}(\vec{x}, Sy)$	A (g, 2$\exists$E)
4.	$\mathcal{P}(a, b, \vec{x})$	3 $\wedge$E
5.	$(\forall i < Sy)\mathscr{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	3 $\wedge$E ($\mathcal{Q}$)
6.	$y < Sy$	T13.11j
7.	$\mathscr{h}(\vec{x}, y, \beta(a, b, y)) = \beta(a, b, Sy)$	5,6 ($\forall$E)
8.	$\beta(a, b, Sy) = \mathscr{f}(\vec{x}, Sy)$	3 $\wedge$E
9.	$\mathscr{f}(\vec{x}, Sy) = \mathscr{h}(\vec{x}, y, \beta(a, b, y))$	7,8 =E
10.	$j < y$	A (g, ($\forall$I))
11.	$j < Sy$	10,6 T13.11b
12.	$\mathscr{h}(\vec{x}, j, \beta(a, b, j)) = \beta(a, b, Sj)$	5,11 ($\forall$E)
13.	$(\forall i < y)\mathscr{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	10-12 ($\forall$I)
14.	$\beta(a, b, y) = \beta(a, b, y)$	=I
15.	$\mathcal{P}(a, b, \vec{x}) \wedge \mathcal{Q}(a, b, \vec{x}, y) \wedge \beta(a, b, y) = \beta(a, b, y)$	4,13,14 $\wedge$I
16.	$\exists p \exists q[\mathcal{P}(p, q, \vec{x}) \wedge \mathcal{Q}(p, q, \vec{x}, y) \wedge \beta(p, q, y) = \beta(a, b, y)]$	15 $\exists$I
17.	$\mathscr{f}(\vec{x}, y) = \beta(a, b, y)$	16 def $\mathscr{f}(\vec{x}, y)$
18.	$\mathscr{f}(\vec{x}, Sy) = \mathscr{h}(\vec{x}, y, \mathscr{f}(\vec{x}, y))$	9,17 =E
19.	$\mathscr{f}(\vec{x}, Sy) = \mathscr{h}(\vec{x}, y, \mathscr{f}(\vec{x}, y))$	2,3-18 $\exists$E

Again, $\mathcal{P}$ and $\mathcal{Q}$ are as on page 663. The key stages of this argument are at (9) which has the result with $\beta(a, b, y)$ where we want $\mathscr{f}(\vec{x}, y)$ and then (17) which shows they are one and the same.

The definition of a recursive function is coordinate with its definition in PA just in case PA proves theorems syntactically "congruent" to the recursive definitions. And now we simply observe that PA in fact defines functions coordinate to the friendly recursive functions.

T13.31. For any friendly recursive function, $\mathsf{r}(\vec{\mathsf{x}})$, PA defines a coordinate function $\mathit{r}(\vec{x})$.

By review of Cf together with T13.19 and T13.30. To connect case (m) with definition RM and T13.19(m), observe that $\widehat{\mathsf{0}} = \mathsf{zero}()$ and PA $\vdash \overline{\mathsf{0}} = \mathit{zero}()$. For a more formal demonstration of this theorem, see E13.30.

This works because PA proves theorems entirely parallel to the recursive definitions. So, for example, from Chapter 12, $\mathsf{plus}(\mathsf{x},\mathsf{y})$ is defined by recursion from $\mathsf{gplus}(\mathsf{x}) = \mathsf{idnt}^1_1(\mathsf{x})$ and $\mathsf{hplus}(\mathsf{x},\mathsf{y},\mathsf{u}) = \mathsf{suc}(\mathsf{idnt}^3_3(\mathsf{x},\mathsf{y},\mathsf{u}))$. Now from T13.31: with Cf(i), PA defines coordinate identity and successor functions, $\mathit{idnt}^1_1(x)$, $\mathit{idnt}^3_3(x,y,u)$, and $\mathit{suc}(w)$; with Cf(e), PA $\vdash \mathit{gplus}(x) = \mathit{idnt}^1_1(x)$; with Cf(c), PA $\vdash \mathit{hplus}(x,y,u) = \mathit{suc}(\mathit{idnt}^3_3(x,y,u))$; and then with Cf(r), PA $\vdash \mathit{plus}(x,\emptyset) = \mathit{gplus}(x)$ and PA $\vdash \mathit{plus}(x,Sy) = \mathit{hplus}(x,y,\mathit{plus}(x,y))$. Given T13.31, from recursive definitions we are positioned simply to "write down" the structurally parallel theorems for the defined functions. For some additional examples,

T13.32. The following are theorems of PA.

(a) PA $\vdash \mathit{gtimes}(x) = \mathit{zero}(x)$

(b) PA $\vdash \mathit{htimes}(x,y,u) = \mathit{plus}(\mathit{idnt}^3_3(x,y,u),x)$

(c) PA $\vdash \mathit{times}(x,\emptyset) = \mathit{gtimes}(x)$

(d) PA $\vdash \mathit{times}(x,Sy) = \mathit{htimes}(x,y,\mathit{times}(x,y))$

(e) PA $\vdash \mathit{gpower}(x) = \mathit{suc}(\mathit{zero}(x))$

(f) PA $\vdash \mathit{hpower}(x,y,u) = \mathit{times}(\mathit{idnt}^3_3(x,y,u),x)$

(g) PA $\vdash \mathit{power}(x,\emptyset) = \mathit{gpower}(x)$

(h) PA $\vdash \mathit{power}(x,Sy) = \mathit{hpower}(x,y,\mathit{power}(x,y))$

(a)–(d) are like the definition for $\mathsf{times}(\mathsf{x},\mathsf{y})$ on page 569; (e)–(h) are like definitions for $\mathsf{power}(\mathsf{x},\mathsf{y})$ from E12.1.

And the range of theorems we allow ourselves to "write down" is extended by some additional simple results. Observe that $\mathit{plus}(x,y)$, say, is defined by a complex expression through T13.31, and so is not the same expression as our old friend $x + y$. Thus it is not obvious that our standard means for manipulation of $+$ apply to plus.

We could recover our ordinary results if we could show PA $\vdash \mathit{plus}(x, y) = x + y$. And similar comments apply to other ordinary functions. Thus initially we seek to show that defined functions are equivalent to ones with which we are familiar. As a preliminary, distinguish $\overline{\overline{\mathsf{n}}} = \mathit{suc}(\ldots \mathit{suc}(\mathit{zero}())\ldots)$ in addition to $\overline{\mathsf{n}} = S \ldots S\emptyset$ and $\hat{\mathsf{n}} = \mathsf{suc}(\ldots \mathsf{suc}(\mathsf{zero}())\ldots)$. By Cf(i), $\mathit{zero}()$ is coordinate with $\mathsf{zero}()$ and $\mathit{suc}(w)$ with $\mathsf{suc}(\mathsf{w})$; then by multiple applications of Cf(c), $\overline{\overline{\mathsf{n}}}$ is coordinate with $\hat{\mathsf{n}}$.

*T13.33. The following are theorems of PA.

(a) PA $\vdash \mathit{suc}(x) = Sx$

(b) PA $\vdash \mathit{idnt}^j_k(x_1 \ldots x_j) = x_k$

(c) PA $\vdash \mathit{zero}() = \emptyset$

(d) PA $\vdash \mathit{zero}^n(x_1 \ldots x_n) = \emptyset$

*(e) PA $\vdash \overline{\overline{\mathsf{n}}} = \overline{\mathsf{n}}$

(f) PA $\vdash \mathit{pred}(\emptyset) = \emptyset$

(g) PA $\vdash \mathit{pred}(Sy) = y$

*(h) PA $\vdash \mathit{pred}(y) = y \dot{-} \overline{1}$
corollary: PA $\vdash \emptyset < y \rightarrow S\,\mathit{pred}(y) = y$
corollary: PA $\vdash (\mathit{pred}(x)|a \wedge \mathit{pred}(y)|b) \rightarrow \mathit{pred}(xy)|ab$

(i) PA $\vdash \mathit{plus}(x, y) = x + y$

(j) PA $\vdash \mathit{times}(x, y) = x \times y$

*(k) PA $\vdash \mathit{subc}(x, y) = x \dot{-} y$

(l) PA $\vdash \mathit{absval}(x - y) = (x \dot{-} y) + (y \dot{-} x)$

(e) is by an easy induction in the metalanguage; otherwise arguments for (a)–(g) are very much the same and nearly trivial. (h) is from (f) and (g), and its corollaries with T13.21k and T13.22e. Arguments for (i)–(k) are by IN. As examples, (a) and (i) are worked on page 689. For additional hints see the associated exercise, E13.27.

So this theorem establishes the equivalences we expect for the defined symbols suc, idnt, zero, $\overline{\overline{\mathsf{n}}}$, pred, plus, times, subc, and absval. Again, $+$, $\times$, and the like are primitive symbols of $\mathscr{L}_{NT}$ where plus and times result by T13.19. And this theorem extends the range of results we write down from recursive definitions. Thus if some $\mathsf{f(x, y, z) = times(plus(x, y), z) = (x + y) \times z}$, then with T13.31, PA $\vdash \mathit{f}(x, y, z) = \mathit{times}(\mathit{plus}(x, y), z)$ and now, PA $\vdash \mathit{f}(x, y, z) = (x + y) \times z$.

Now say a recursive relation is *friendly* iff it has a friendly characteristic function. Then PA defines relations corresponding to each friendly recursive relation. At one level, this is easy: Suppose $\textsf{R}$ is a friendly recursive relation. Since $\textsf{R}$ is friendly, its characteristic function $\textsf{ch}_\textsf{R}$ is friendly and so PA defines $\mathcal{ch}_\textsf{R}$. Set,

$$\text{PA} \vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x}) = \emptyset$$

Then PA defines $\mathbb{R}(\vec{x})$. In fact, however, for relations coordinate to ones from Chapter 12 we shall require specifications in PA that "track" with the recursive definitions. For this, it will be helpful to show that PA defines relations by a short induction. As developed in Chapter 12, recursive relations are either atomic $\textsf{EQ}$, $\textsf{LEQ}$, $\textsf{LESS}$, or arise by operations $\textsf{NEG}$, $\textsf{DSJ}$, $(\exists \textsf{y} \le \textsf{z})$. Then,

T13.34. For any recursive $\textsf{R}$ defined in Chapter 12, PA defines $\mathbb{R}$ such that PA $\vdash$ $\mathbb{R}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x}) = \emptyset$.

Suppose $\textsf{R}$ is one of the recursive relations defined in Chapter 12. By induction on the number of recursive operators in the definition of $\textsf{R}(\vec{\textsf{x}})$,

Basis: $\textsf{R}$ is defined without recursive operators. Then $\textsf{R}$ is $\textsf{EQ}(\textsf{x},\textsf{y})$, $\textsf{LEQ}(\textsf{x},\textsf{y})$, or $\textsf{LESS}(\textsf{x},\textsf{y})$. Let PA $\vdash$ $\mathbb{E}q(x,y) \leftrightarrow \mathcal{sg}(\mathcal{absval}(x \mathbin{\text{-}} y)) = \emptyset$; and PA $\vdash$ $\mathbb{L}eq(x,y) \leftrightarrow \mathcal{sg}(\mathcal{subc}(x,y)) = \emptyset$; and PA $\vdash$ $\mathbb{L}ess(x,y) \leftrightarrow \mathcal{sg}(\mathcal{subc}(\mathcal{suc}(x), y)) = \emptyset$. Then PA defines $\mathbb{E}q$, $\mathbb{L}eq$, and $\mathbb{L}ess$. And, in each case, PA $\vdash$ $\mathbb{R}(x,y) \leftrightarrow \mathcal{ch}_\textsf{R}(x,y) = \emptyset$.

Assp: For any i, $0 \le i < k$, if $\textsf{R}$ is defined by i recursive operators, then PA defines $\mathbb{R}$ such that PA $\vdash$ $\mathbb{R}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x}) = \emptyset$.

Show: If $\textsf{R}$ is defined by k recursive operators, then PA defines $\mathbb{R}$ such that PA $\vdash$ $\mathbb{R}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x}) = \emptyset$.

If $\textsf{R}$ is defined by k recursive operators, then it is $\textsf{NEG}(\textsf{P}(\vec{\textsf{x}}))$, $\textsf{DSJ}(\textsf{P}(\vec{\textsf{x}}), \textsf{Q}(\vec{\textsf{y}}))$, or $(\exists \textsf{y} \le \textsf{z})\textsf{P}(\vec{\textsf{x}},\textsf{y})$, for $\textsf{P}$ and $\textsf{Q}$ with $< k$ recursive operators.

($\textsf{NEG}$) $\textsf{R}(\vec{\textsf{x}})$ is $\textsf{NEG}(\textsf{P}(\vec{\textsf{x}}))$, and so has characteristic function, $\textsf{csg}(\textsf{ch}_\textsf{P}(\vec{\textsf{x}}))$. By assumption, PA defines $\mathbb{P}(\vec{x})$ such that PA $\vdash$ $\mathbb{P}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{P}(\vec{x}) = \emptyset$. Let PA $\vdash$ $\mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow \mathcal{csg}(\mathcal{ch}_\textsf{P}(\vec{x})) = \emptyset$. Then PA defines $\mathbb{R}(\vec{x})$ and PA $\vdash$ $\mathbb{R}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x}) = \emptyset$.

($\textsf{DSJ}$) $\textsf{R}(\vec{\textsf{x}},\vec{\textsf{y}})$ is $\textsf{DSJ}(\textsf{P}(\vec{\textsf{x}}), \textsf{Q}(\vec{\textsf{y}}))$, and so has characteristic function, $\textsf{times}(\textsf{ch}_\textsf{P}(\vec{\textsf{x}}), \textsf{ch}_\textsf{Q}(\vec{\textsf{y}}))$. By assumption, PA defines $\mathbb{P}(\vec{x})$ and $\mathbb{Q}(\vec{y})$ such that PA $\vdash$ $\mathbb{P}(\vec{x}) \leftrightarrow \mathcal{ch}_\textsf{P}(\vec{x}) = \emptyset$ and PA $\vdash$ $\mathbb{Q}(\vec{y}) \leftrightarrow \mathcal{ch}_\textsf{Q}(\vec{y}) = \emptyset$. Let PA $\vdash$ $\mathbb{D}sj(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow \mathcal{times}(\mathcal{ch}_\textsf{P}(\vec{x}), \mathcal{ch}_\textsf{Q}(\vec{y})) = \emptyset$. Then PA defines $\mathbb{R}(\vec{x},\vec{y})$ and PA $\vdash$ $\mathbb{R}(\vec{x},\vec{y}) \leftrightarrow \mathcal{ch}_\textsf{R}(\vec{x},\vec{y}) = \emptyset$.

($\exists \le$) $\textsf{R}(\vec{\textsf{x}},\textsf{z})$ is $(\exists \textsf{y} \le \textsf{z})\textsf{P}(\vec{\textsf{x}},\textsf{y})$. Then given $\textsf{ch}_\textsf{P}(\vec{\textsf{x}},\textsf{y})$, for some $\textsf{v}$ not in $\vec{\textsf{x}}$ and not $\textsf{y}$ there is an $\textsf{eleq}(\vec{\textsf{x}},\textsf{v})$ such that $\textsf{geleq}(\vec{\textsf{x}}) = \textsf{ch}_\textsf{P}(\vec{\textsf{x}},\hat{0})$ and $\textsf{heleq}(\vec{\textsf{x}},\textsf{v},\textsf{u}) = \textsf{u} \times \textsf{ch}_\textsf{P}(\vec{\textsf{x}},\textsf{Sv})$; and the characteristic function of $(\exists \textsf{y} \le \textsf{z})\textsf{P}(\vec{\textsf{x}},\textsf{y})$ is $\textsf{eleq}(\vec{\textsf{x}},\textsf{z})$. By assumption, PA defines $\mathbb{P}(\vec{x})$ such that PA $\vdash$ $\mathbb{P}(\vec{x}) \leftrightarrow$

$\mathscr{ch}_{\mathsf{P}}(\vec{x}) = \emptyset$; then with T13.31 PA defines $\mathscr{eleq}(x, v)$. Let PA $\vdash (\exists y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow \mathscr{eleq}(\vec{x}, z) = \emptyset$. Then PA defines $\mathbb{R}(\vec{x}, z)$ and PA $\vdash \mathbb{R}(\vec{x}, z) \leftrightarrow \mathscr{ch}_{\mathsf{R}}(\vec{x}, z) = \emptyset$.

Indct: For any recursive R as defined in Chapter 12, PA defines $\mathbb{R}$ such that PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$.

As a quick corollary: From T13.19, function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$ is originally captured by an $\mathcal{R}(\vec{x}, v)$ such that PA $\vdash \mathscr{ch}_{\mathsf{R}}(\vec{x}) = v \leftrightarrow \mathcal{R}(\vec{x}, v)$; so PA $\vdash \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$; and with this theorem, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$. Also, as addenda to this theorem: If a relation $\mathsf{R}(\vec{\mathsf{x}})$ just is some previously defined $\mathsf{Q}(\vec{\mathsf{x}})$, let PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{Q}(\vec{x})$; then PA defines $\mathbb{R}$. And for a function $(\mu\mathsf{y} \leq \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{y})$ with its $\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}}, \mathsf{y})$, for some v not in $\vec{\mathsf{x}}$ and not y there is an $\mathsf{eleq}(\vec{\mathsf{x}}, \mathsf{v})$ as above, and then $\mathsf{mleq}(\vec{\mathsf{x}}, \mathsf{v})$ such that $\mathsf{gmleq}(\vec{\mathsf{x}}) = \mathsf{zero}(\vec{\mathsf{x}})$ and $\mathsf{hmleq}(\vec{\mathsf{x}}, \mathsf{v}, \mathsf{u}) = \mathsf{u} + \mathsf{eleq}(\vec{\mathsf{x}}, \mathsf{v})$; and then $(\mu\mathsf{y} \leq \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{y}) = \mathsf{mleq}(\vec{\mathsf{x}}, \mathsf{z})$. But PA defines $\mathscr{mleq}(\vec{x}, v)$; let PA $\vdash (\mu y \leq z)\mathbb{P}(\vec{x}, y) = \mathscr{mleq}(\vec{x}, z)$; so PA defines $(\mu y \leq z)\mathbb{P}(\vec{x}, y)$.

Now as a preliminary to showing that recursive relations defined in Chapter 12 are *coordinate* with their definitions in PA, a couple of theorems: First, some easy results for functions $\mathscr{sg}(y)$ and $\mathscr{csg}(y)$—these are not equivalences (because no equivalents have previously been defined), but result directly for the defined functions. Then some equivalences for basic defined relations.

*T13.35. The following are theorems of PA.

*(a) PA $\vdash \mathscr{sg}(\emptyset) = \emptyset$

*(b) PA $\vdash \mathscr{sg}(Sy) = \overline{1}$

(c) PA $\vdash y = \emptyset \leftrightarrow \mathscr{sg}(y) = \emptyset$

*(d) PA $\vdash \emptyset < y \leftrightarrow \mathscr{sg}(y) = \overline{1}$

(e) PA $\vdash \mathscr{csg}(\emptyset) = \overline{1}$

(f) PA $\vdash \mathscr{csg}(Sy) = \emptyset$

*(g) PA $\vdash y = \emptyset \leftrightarrow \mathscr{csg}(y) = \overline{1}$

*(h) PA $\vdash \emptyset < y \leftrightarrow \mathscr{csg}(y) = \emptyset$

(i) PA $\vdash \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset \vee \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \overline{1}$.

(a), (b), (e), (f) are from the definitions; then (c), (d), (g), (h) result easily from them. For (i), recall from (CF) that a characteristic function is (officially) of the sort $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}}))$ so that from T13.31, PA $\vdash \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \mathscr{sg}(\mathscr{p}(\vec{x}))$.

*T13.36. The following are theorems of PA.

*(a) PA $\vdash \mathbb{E}q(x,y) \leftrightarrow x = y$

(b) PA $\vdash \mathbb{L}eq(x,y) \leftrightarrow x \leq y$

*(c) PA $\vdash \mathbb{L}ess(x,y) \leftrightarrow x < y$

*(d) PA $\vdash \mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow {\sim}\mathbb{P}(\vec{x})$

(e) PA $\vdash \mathbb{D}sj(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\mathbb{P}(\vec{x}) \vee \mathbb{Q}(\vec{y}))$

*(f) PA $\vdash (\exists y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y \leq z)\mathbb{P}(\vec{x}, y)$

*(g) PA $\vdash (\mu y \leq z)\mathbb{P}(\vec{x}, y) = (\mu y \leq z)\mathbb{P}(\vec{x}, y)$

The argument for (g) is particularly involved; see the box on page 690 for the main outlines of that argument. For hints see the associated exercise E13.29.

So this theorem delivers the equivalences we expect for $\mathbb{E}q$, $\mathbb{L}eq$, $\mathbb{L}ess$, $\mathbb{N}eg$, $\mathbb{D}sj$, $(\exists y \leq z)$, and $(\mu y \leq z)$.

Now we can show that definitions of recursive relations from Chapter 12 are coordinate with definitions in PA (and similarlily for bounded minimization). Again, coordinate definitions result in a sort of structural similarity.

Cr The definition of a recursive relation is *coordinate* with its definition in PA iff,

(a) $\mathsf{R}(\vec{\mathsf{x}})$ is an atomic function $\textsc{eq}(\mathsf{x},\mathsf{y})$, or $\textsc{leq}(\mathsf{x},\mathsf{y})$, or $\textsc{less}(\mathsf{x},\mathsf{y})$, and PA $\vdash \mathbb{E}q(x,y) \leftrightarrow x = y$, and PA $\vdash \mathbb{L}eq(x,y) \leftrightarrow x \leq y$, and PA $\vdash \mathbb{L}ess(x,y) \leftrightarrow x < y$.

(n) $\mathsf{R}(\vec{\mathsf{x}})$ is $\textsc{neg}(\mathsf{P}(\vec{\mathsf{x}}))$ and for coordinate $\mathbb{P}(\vec{x})$, PA $\vdash \mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow {\sim}\mathbb{P}(\vec{x})$.

(d) $\mathsf{R}(\vec{\mathsf{x}}, \vec{\mathsf{y}})$ is $\textsc{dsj}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{y}}))$, and for coordinate $\mathbb{P}(\vec{x})$ and $\mathbb{Q}(\vec{y})$, we have PA $\vdash \mathbb{D}sj(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\mathbb{P}(\vec{x}) \vee \mathbb{Q}(\vec{y}))$.

(q) $\mathsf{R}(\vec{\mathsf{x}})$ is a bounded quantification $(\exists \mathsf{y} \leq \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{y})$ and for coordinate $\mathbb{P}(\vec{x}, y)$, PA $\vdash (\exists y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y \leq z)\mathbb{P}(\vec{x}, y)$.

(e) $\mathsf{R}(\vec{\mathsf{x}})$ just is some $\mathsf{Q}(\vec{\mathsf{x}})$ and for coordinate $\mathbb{Q}(\vec{x})$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{Q}(\vec{x})$.

Cm The definition of a recursive function $\mathsf{f}(\vec{\mathsf{x}}) = (\mu\mathsf{y} \leq \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{y})$ defined by bounded minimization is *coordinate* with its definition in PA iff for coordinate $\mathbb{P}$, PA $\vdash (\mu y \leq z)\mathbb{P}(\vec{x}, y) = (\mu y \leq z)\mathbb{P}(\vec{x}, y)$.

Again, the defination of a recursive relation is coordinate with its definition in PA just in case PA proves theorems syntactically "congruent" to the recursive definitions. And now we simply observe that PA in fact defines relations coordinate to the recursive relations.

T13.33a

1.	$v = \mathit{suc}(x) \leftrightarrow Sx = v$	T13.31
2.	$\mathit{suc}(x) = \mathit{suc}(x) \leftrightarrow Sx = \mathit{suc}(x)$	from 1
3.	$\mathit{suc}(x) = \mathit{suc}(x)$	=I
4.	$\mathit{suc}(x) = Sx$	2,3 $\leftrightarrow$E

Once we write down the expression on (1) with T13.31, the condition for *suc* is like that for S.

T13.33i

1.	$\mathit{gplus}(x) = \mathit{idnt}^1_1(x)$	T13.31
2.	$\mathit{gplus}(x) = x$	1 T13.33b
3.	$\mathit{plus}(x, \emptyset) = \mathit{gplus}(x)$	T13.31
4.	$\mathit{plus}(x, \emptyset) = x$	3,2 =E
5.	$x + \emptyset = x$	T6.48
6.	$\mathit{plus}(x, \emptyset) = x + \emptyset$	4,5 =E
7.	$\quad \mathit{plus}(x, j) = x + j$	A (g, $\rightarrow$I)
8.	$\quad \mathit{plus}(x, Sj) = \mathit{hplus}(x, j, \mathit{plus}(x, j))$	T13.31
9.	$\quad \mathit{hplus}(x, j, u) = \mathit{suc}(\mathit{idnt}^3_3(x, j, u))$	T13.31
10.	$\quad \mathit{hplus}(x, j, u) = \mathit{suc}(u)$	9 T13.33b
11.	$\quad \mathit{hplus}(x, j, u) = Su$	10 T13.33a
12.	$\quad \mathit{hplus}(x, j, \mathit{plus}(x, j)) = S\mathit{plus}(x, j)$	from 11
13.	$\quad \mathit{plus}(x, Sj) = S\mathit{plus}(x, j)$	8,12 =E
14.	$\quad \mathit{plus}(x, Sj) = S(x + j)$	13,7 =E
15.	$\quad S(x + j) = x + Sj$	T6.49
16.	$\quad \mathit{plus}(x, Sj) = x + Sj$	14,15 =E
17.	$[\mathit{plus}(x, j) = x + j] \rightarrow [\mathit{plus}(x, Sj) = x + Sj]$	7-16 $\rightarrow$I
18.	$\forall y([\mathit{plus}(x, y) = x + y] \rightarrow [\mathit{plus}(x, Sy) = x + Sy])$	17 $\forall$I
19.	$\mathit{plus}(x, y) = x + y$	6,18 IN

Again, once we write down the expressions on (1) and (9) with T13.31 and then on (3) and (8), the conditions for $\mathit{plus}(x, y)$ work like the ones for $x + y$—so that the equivalence results by IN.

T13.36g

1.	$\mathit{mleq}(\vec{x}, \emptyset) = (\mu y \leq \emptyset)\mathbb{P}(\vec{x}, y)$	[a]
2.	$\mathit{eleq}(\vec{x}, j) = \emptyset \leftrightarrow (\exists y \leq j)\mathbb{P}(x, y)$	T13.34
3.	$\mathit{mleq}(\vec{x}, Sj) = \mathit{hmleq}(\vec{x}, j, \mathit{mleq}(\vec{x}, j))$	T13.31
4.	$\mathit{hmleq}(\vec{x}, j, u) = \mathit{plus}(u, \mathit{eleq}(\vec{x}, j))$	T13.31
5.	$(\mu y \leq z)\mathbb{P}(\vec{x}, y) = \mathit{mleq}(\vec{x}, z)$	T13.34
6.	$\mathit{eleq}(\vec{x}, j) = \emptyset \leftrightarrow (\exists y \leq j)\mathbb{P}(x, y)$	2 T13.36f
7.	$\mathit{hmleq}(\vec{x}, j, u) = u + \mathit{eleq}(\vec{x}, j)$	4 T13.33i
8.	$\mathit{hmleq}(\vec{x}, j, \mathit{mleq}(\vec{x}, j)) = \mathit{mleq}(\vec{x}, j) + \mathit{eleq}(\vec{x}, j)$	from 7
9.	$\mathit{mleq}(\vec{x}, Sj) = \mathit{mleq}(\vec{x}, j) + \mathit{eleq}(\vec{x}, j)$	3,8 =E
10.	$\quad \mathit{mleq}(\vec{x}, j) = (\mu y \leq j)\mathbb{P}(\vec{x}, y)$	A (g, →I)
11.	$\quad a = \mathit{mleq}(\vec{x}, j)$	def
12.	$\quad b = \mathit{mleq}(\vec{x}, Sj)$	def
13.	$\quad b = a + \mathit{eleq}(\vec{x}, j)$	9,11,12 =E
14.	$\quad a = (\mu y \leq j)\mathbb{P}(\vec{x}, y)$	10,11 =E
15.	$\quad a = \mu y[y = j \vee \mathbb{P}(\vec{x}, y)]$	14 *Def*$[\mu y \leq]$
16.	$\quad (\forall w < a)[w \neq j \wedge \sim\mathbb{P}(\vec{x}, w)]$	15 T13.18b
17.	$\quad a = j \vee \mathbb{P}(\vec{x}, a)$	15 T13.18a
18.	$\quad\quad a = j$	A (g, 17∨E)
19.	$\quad\quad \sim\mathbb{P}(\vec{x}, j) \vee \mathbb{P}(\vec{x}, j)$	T3.1
20.	$\quad\quad\quad \sim\mathbb{P}(\vec{x}, j)$	A (g, 19∨E)
21.	$\quad\quad\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, w))$	[b]
22.	$\quad\quad\quad \mathbb{P}(\vec{x}, j)$	A (g 19∨E)
23.	$\quad\quad\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, w))$	[c]
24.	$\quad\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, w))$	19,20-21,22-23 ∨E
25.	$\quad\quad \mathbb{P}(\vec{x}, a)$	A (g, 17∨E)
26.	$\quad\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, w))$	[d]
27.	$\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, w))$	17,18-24,25-26 ∨E
28.	$\quad [b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)\sim(w = Sj \vee \mathbb{P}(\vec{x}, w))$	27 DeM
29.	$\quad b = \mu y[y = Sj \vee \mathbb{P}(\vec{x}, y)]$	28 *Def*$[\mu y]$
30.	$\quad b = (\mu y \leq Sj)\mathbb{P}(\vec{x}, y)$	29 *Def*$[\mu y \leq]$
31.	$\quad \mathit{mleq}(\vec{x}, Sj) = (\mu y \leq Sj)\mathbb{P}(\vec{x}, y)$	30,12 =E
32.	$[\mathit{mleq}(\vec{x}, j) = (\mu y \leq j)\mathbb{P}(\vec{x}, y)] \rightarrow$ $[\mathit{mleq}(\vec{x}, Sj) = (\mu y \leq Sj)\mathbb{P}(\vec{x}, y)]$	10-31 →I
33.	$\forall n([\mathit{mleq}(\vec{x}, n) = (\mu y \leq n)\mathbb{P}(\vec{x}, y)] \rightarrow$ $[\mathit{mleq}(\vec{x}, Sn) = (\mu y \leq Sn)\mathbb{P}(\vec{x}, y)])$	32 ∀I
34.	$\mathit{mleq}(\vec{x}, z) = (\mu y \leq z)\mathbb{P}(\vec{x}, y)$	1,33 IN
35.	$(\mu y \leq z)\mathbb{P}(\vec{x}, y) = (\mu y \leq z)\mathbb{P}(\vec{x}, y)$	5,34 =E

Hints: The zero case [a] is straightforward with T13.18d. For the show, the main argument is to obtain $[b = Sj \vee \mathbb{P}(\vec{x}, b)] \wedge (\forall w < b)\sim(w = Sj \vee \mathbb{P}(\vec{x}, w))$ at (28) toward the application of *Def*$[\mu y]$ at (29). For the left conjunct, at [b] you will be able to show that $b = Sj$; and for that conjunct at [c] and [d] you will be able to show $b = j$ and $b = a$ respectively, and so with the assumptions, $\mathbb{P}(\vec{x}, b)$.

T13.37. For any friendly recursive function $\mathsf{f}(\mathsf{x})$ PA defines a coordinate function $\mathbb{f}(\vec{x})$. And for any recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ as defined in Chapter 12, PA defines a coordinate relation $\mathbb{R}(\vec{x})$.

By T13.31, and then review of Cr and Cm together with T13.34 and T13.36. For a more formal demonstration, see E13.30.

So, for example, from Chapter 12, $\mathsf{FCTR}(\mathsf{m},\mathsf{n}) = (\exists \mathsf{y} \leq \mathsf{n})\mathsf{EQ}(\mathsf{times}(\mathsf{suc}(\mathsf{m}),\mathsf{y}),\mathsf{n}) = (\exists \mathsf{y} \leq \mathsf{n})(\mathsf{Sm} \times \mathsf{y} = \mathsf{n})$. By T13.31, PA defines $\mathbb{times}(\mathbb{suc}(m), y)$; and with T13.33, PA $\vdash \mathbb{times}(\mathbb{suc}(m), y) = Sm \times y$. So by T13.34, PA defines $\mathbb{Eq}(\mathbb{times}(\mathbb{suc}(m), y), n)$; and by T13.36, PA $\vdash \mathbb{Eq}(\mathbb{times}(\mathbb{suc}(m), y), n) \leftrightarrow Sm \times y = n$. And again, PA defines $(\exists y \leq n)\mathbb{Eq}(\mathbb{times}(\mathbb{suc}(m), y), n)$ so that PA $\vdash (\exists y \leq n)\mathbb{Eq}(\mathbb{times}(\mathbb{suc}(m), y), n) \leftrightarrow (\exists y \leq n)(Sm \times y = n)$. And since this just is $\mathbb{Fctr}(m, n)$, by the addendum to T13.34, PA $\vdash \mathbb{Fctr}(m, n) \leftrightarrow (\exists y \leq n)(Sm \times y = n)$.

At this stage, we have defined in PA functions and relations corresponding to all the recursive functions and relations defined in Chapter 12. Again, then, we are in a position to "write down" results in PA directly from the recursive definitions. And again we expand the range of relations that we "write down" by demonstrating some final equivalents.

*T13.38. The following are theorems of PA.

(a) PA $\vdash \mathbb{Imp}(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\mathbb{P}(\vec{x}) \rightarrow \mathbb{Q}(\vec{y}))$

(b) PA $\vdash \mathbb{Cnj}(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\mathbb{P}(\vec{x}) \wedge \mathbb{Q}(\vec{y}))$

(c) PA $\vdash (\mathbb{\exists} y < z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y < z)\mathbb{P}(\vec{x}, y)$

*(d) PA $\vdash (\mathbb{\forall} y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\forall y \leq z)\mathbb{P}(\vec{x}, y)$

(e) PA $\vdash (\mathbb{\forall} y < z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\forall y < z)\mathbb{P}(\vec{x}, y)$

(f) PA $\vdash \mathbb{Fctr}(m, n) \leftrightarrow m|n$

*(g) PA $\vdash \mathbb{Prime}(n) \leftrightarrow Pr(n)$

With T13.37, (a), (b), (d), and (e) are nearly trivial (and the others are not hard). As examples, (a) and (c) are worked on on page 693.

Given these results, here are some last examples to illustrate theorems we simply write down. By T13.37, relations and functions including $\mathbb{Var}(v)$, $\mathbb{Wff}(n)$, $\mathbb{exp}(n, i)$, $\mathbb{len}(n)$, and $\mathbb{cncat}(m, n)$ have coordinate definitions in PA; let $\mathbb{cncat}(m, n) = m * n$. Given this,

*T13.39. The following are theorems of PA.

(a) PA $\vdash$ $\mathbb{til}(n) = \overline{\ulcorner\sim\urcorner} * n$

(b) PA $\vdash$ $\mathbb{cnd}(n, o) = \overline{\ulcorner(\urcorner} * n * \overline{\ulcorner\rightarrow\urcorner} * o * \overline{\ulcorner)\urcorner}$

(c) PA $\vdash$ $\mathbb{unv}(v, n) = \overline{\ulcorner\forall\urcorner} * v * n$

(d) PA $\vdash$ $\mathbb{caret}(m, n) = \mathbb{til}(\mathbb{cnd}(m, \mathbb{til}(n)))$

(e) PA $\vdash$ $\mathbb{Mp}(m, n, o) \leftrightarrow \mathbb{cnd}(n, o) = m$

(f) PA $\vdash$ $\mathbb{Gen}(m, n) \leftrightarrow (\exists v \leq n)[\mathbb{Var}(v) \wedge n = \mathbb{unv}(v, m)]$

(g) PA $\vdash$ $\mathbb{Icon}(m, n, o) \leftrightarrow [\mathbb{Mp}(m, n, o) \vee (m = n \wedge \mathbb{Gen}(n, o))]$

*(h) PA $\vdash$ $\mathbb{Axiomad1}(n) \leftrightarrow (\exists p \leq n)(\exists q \leq n)[\mathbb{Wff}(p) \wedge \mathbb{Wff}(q) \wedge n = \mathbb{cnd}(p, \mathbb{cnd}(q, p)]$

and similarly for the other axioms

(i) PA $\vdash$ $\mathbb{Axiompa}(n) \leftrightarrow [\mathbb{Axiomad1}(n) \vee \ldots \vee \mathbb{Axiomad8}(n) \vee \mathbb{Axiompa1}(n) \vee \ldots \vee \mathbb{Axiompa7}(n)]$

(j) PA $\vdash$ $\mathbb{Prfpa}(m, n) \leftrightarrow \{\mathbb{exp}(m, \mathbb{len}(m) \dot{-} \overline{1}) = n \wedge \overline{1} < m \wedge$
$(\forall k < \mathbb{len}(m))[\mathbb{Axiompa}(\mathbb{exp}(m, k)) \vee (\exists i < k)(\exists j < k)\mathbb{Icon}(\mathbb{exp}(m, i), \mathbb{exp}(m, j), \mathbb{exp}(m, k))]\}$

This theorem suggests the range of notions we shall be able to reason about in PA. Officially the instances of concatenation should be grouped in pairs; however T13.46g below is an association result which tells us that the grouping does not matter. Given that PA defines $\mathbb{Var}$, $\mathbb{cncat}$, and the like, it is important that '=' and the *operators* in expressions above are ordinary expressions of $\mathscr{L}_{NT}$. Thus we shall be able to manipulate the expressions in the usual ways. In general, from the recursive definitions we simply assert such results, citing just T13.37 as justification.

E13.26. Produce the derivation to show T13.30a.

*E13.27. Produce a derivation to show T13.33j. Hard-core: show all the unworked cases from T13.33.

Hints: For (h) it will be helpful to assert $y = \emptyset \vee \emptyset < y$. (k) works in the usual way up to the point in the show stage where you get $\mathbb{subc}(x, Sj) = \mathbb{pred}(x \dot{-} j)$; then it will take some work to show $x \dot{-} Sj = \mathbb{pred}(x \dot{-} j)$; for this begin with $x \leq j \vee j < x$ by T13.11s; the first case is straightforward; for the second, you will be able to show $S(x \dot{-} Sj) = S\,\mathbb{pred}(x \dot{-} j)$ and apply T6.47.

*E13.28. Show T13.35 (e)–(g), Hard-core: Show each of the results in T13.35.

T13.38a

1.	$\mathbb{I}mp(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\sim\mathbb{P}(\vec{x}) \vee \mathbb{Q}(\vec{y}))$	T13.37
2.	$\mathbb{I}mp(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow (\mathbb{P}(\vec{x}) \rightarrow \mathbb{Q}(\vec{y}))$	1 Impl

From page 607, $\mathsf{IMP}(\mathsf{P}(\vec{x}), \mathsf{Q}(\vec{y}))$ is $\sim\mathsf{P}(\vec{x}) \vee \mathsf{Q}(\vec{y})$. Then (1) is the structurally parallel theorem of PA.

T13.38c

1.	$(\exists y < z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y \leq z)(y \neq z \wedge \mathbb{P}(\vec{x}, y))$	T13.37
2.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	A (g, $\leftrightarrow$I)
3.	$(\exists y \leq z)(y \neq z \wedge \mathbb{P}(\vec{x}, y))$	1,2 $\leftrightarrow$E
4.	$j \neq z \wedge \mathbb{P}(\vec{x}, j)$	A (g, 3($\exists$E))
5.	$j \leq z$	
6.	$j \neq z$	4 $\wedge$E
7.	$j < z$	5,6 T13.11n
8.	$\mathbb{P}(\vec{x}, j)$	4 $\wedge$E
9.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	8,7 ($\exists$I)
10.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	3,4-9 ($\exists$E)
11.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	A (g, $\leftrightarrow$I)
12.	$\mathbb{P}(\vec{x}, j)$	A (g, 11($\exists$E))
13.	$j < z$	
14.	$j \neq z$	13 T13.11t
15.	$j \neq z \wedge \mathbb{P}(\vec{x}, j)$	14,12 $\wedge$I
16.	$j \leq z$	13 T13.11n
17.	$(\exists y \leq z)(y \neq z \wedge \mathbb{P}(\vec{x}, y))$	15,16 ($\exists$I)
18.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	1,17 $\leftrightarrow$E
19.	$(\exists y < z)\mathbb{P}(\vec{x}, y)$	11,12-18 ($\exists$E)
20.	$(\exists y < z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y < z)\mathbb{P}(\vec{x}, y)$	2-10,11-19 $\leftrightarrow$I

From page 607, $(\exists y < z)\mathsf{P}(\vec{x}, y)$ is $(\exists y \leq z)(y \neq z \wedge \mathsf{P}(\vec{x}, y))$. Then (1) results by T13.37 (and T13.38b).

Font Conventions

At different stages, we employ different fonts for items of different sorts. For the most part, this is straightforward. Here we collect some conventions together.

1. Expressions of symbolic object languages are given in italics; these include the function (lowercase) and relation (first letter uppercase) symbols abbreviated or defined in Q and PA.

   A, a, β, *function*, *Relation*

2. Objects from the semantic account are indicated by upright and sans serif fonts; these include recursive functions (lowercase) and relations (small capitals)—and bold when special symbols are used.

   J, d, o, $\upbeta$, $\mathsf{function}$, $\textsc{relation}$,

3. '$=$', '$<$', '$+$', and '$\times$' are symbols in symbolic languages, and in the metalanguage names for themselves; when bold they pick out recursively defined relations and functions. Narrowed versions are used to express the relations in the metalanguage.

   $=$, $\mathbf{=}$, $=$, $<$, $\mathbf{<}$, $<$, $+$, $\mathbf{+}$, $+$, $\times$, $\mathbf{\times}$, $\times$

4. Expressions may be indicated by quotation as, 'Bob is happy', and '$\forall x$'; but often and where confusion will not arise, distinguished just by vocabulary and font and indicated by simple display, $\forall x(Ax \to Bx)$. The (meta-)language for description of object expressions includes script variables.

   $\mathscr{P}$, $\mathscr{p}$

5. Variables in the Fraktur font range over metalinguistic expressions and over classes (whose members are themselves identified in the metalanguage). Metalinguistic operators have versions $\Rightarrow$, $\Leftrightarrow$, $\neg$, $\vartriangle$, $\triangledown$, $\perp$, A, and S.

   $\mathfrak{A}$, $\mathfrak{a}$, $\mathfrak{M}$, $\mathfrak{m}$

6. Function and relation symbols introduced into PA by T13.37 have their first character in a "hollow" blackboard font—these are not automatically equivalent to ones that may be described in (1), though we may set out to demonstrate equivalence.

   $\mathbb{f}unction$, $\mathbb{R}elation$

7. Object expressions for computer languages are given in a typewriter font,

   `Expression`

8. n is a natural number and $\hat{\mathsf{n}}$ the recursive function $\mathsf{suc}(\ldots\mathsf{suc}(\mathsf{zero}())\ldots)$ that returns n. In $\mathscr{L}_{NT}$, $\bar{\bar{\mathsf{n}}}$ is $\mathbb{s}uc(\ldots\mathbb{s}uc(\mathbb{z}ero())\ldots)$, and $\bar{\mathsf{n}}$ is $S\ldots S\emptyset$.

*E13.29. Show (b), (e), and (f) from T13.36. Hard-core: Demonstrate each of the results in T13.36.

Hints for T13.36. (a): For this relation, you have $\mathbb{E}q(x, y) \leftrightarrow sg(absval(x - y)) = \emptyset$ from T13.34; this gives $\mathbb{E}q(x, y) \leftrightarrow [(x \dot{-} y) + (y \dot{-} x)] = \emptyset$; now for $\leftrightarrow$I, the case from $x = y$ is easy; from $\mathbb{E}q(x, y)$, you have $y \leq x \vee x < y$ from T13.11s; the cases are not hard and similar (since $x < y$ gives $x \leq y$). (d): This is straightforward with $\mathbb{P}(\vec{x}) \leftrightarrow ch_{\mathsf{P}}(\vec{x}) = \emptyset$ and $\mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow csg(ch_{\mathsf{P}}(\vec{x})) = \emptyset$ from T13.34. (f): Recall from Chapter 12 that $(\exists y \leq z)\mathsf{P}(\vec{x}, y)$ is defined by means of an $\mathsf{eleq}(\vec{x}, v)$ corresponding to $(\exists y \leq v)\mathsf{P}(\vec{x}, y)$; the main argument is to show by IN that PA $\vdash \forall v[eleq(\vec{x}, v) = \emptyset \leftrightarrow (\exists y \leq v)\mathbb{P}(\vec{x}, y)]$. You have $\mathbb{P}(\vec{x}, y) \leftrightarrow ch_{\mathsf{P}}(\vec{x}, y) = \emptyset$ by T13.34; for the zero case, you have $eleq(\vec{x}, \emptyset) = geleq(\vec{x})$, and $geleq(\vec{x}) = ch_{\mathsf{P}}(\vec{x}, \emptyset)$; for the main reasoning, you have $eleq(\vec{x}, Sj) = heleq(\vec{x}, j, eleq(\vec{x}, j))$, and $heleq(\vec{x}, j, u) = times[u, ch_{\mathsf{P}}(\vec{x}, suc(j))]$; once you have finished the induction, it is a simple matter of applying $(\exists y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow eleq(\vec{x}, z) = \emptyset$ to get $(\exists y \leq z)\mathbb{P}(\vec{x}, y) \leftrightarrow (\exists y \leq z)\mathbb{P}(\vec{x}, y)$.

*E13.30. We have justified T13.31 and T13.37 by "review" of relevant definitions and theorems. This review is really by induction. (i) For T13.31, by induction on the sequence of recursive functions produce the review of Cf with T13.19 and T13.30 to show that PA defines functions coordinate to friendly recursive functions. (ii) For T13.37, by induction on the number of recursive operators in the definition of a recursive relation produce the review of Cr with T13.34 and T13.36 to show that PA defines relations coordinate to recursive relations defined in Chapter 12.

*E13.31. Prove T13.38f and then, from the next theorem, finish T13.39h. Hard-core: Work the remaining results in T13.38. For T13.39h you may take it that included relations and functions such as $\mathbb{T}erm$, $\mathbb{F}reefor$, $formsub$ are defined.

Hints: The left and right sides of T13.38f,g have nearly matching definitions except that the recursive side includes a bounded quantifier—so that in each case you have to work for one direction of the biconditional.

13.4 The Second Condition: $\Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$

Given functions and relations defined in PA, we turn now to demonstration of the second derivability condition. Again there is some background—after which demonstration of the condition itself is straightforward. The overall idea is simple: Suppose that PA $\vdash \Box(\mathcal{P} \to \mathcal{Q})$ and then that PA $\vdash \Box\mathcal{P}$. Then there are j and k such that $\mathsf{PRFPA}(\mathsf{j}, \widehat{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner})$ and $\mathsf{PRFPA}(\mathsf{k}, \widehat{\ulcorner\mathcal{P}\urcorner})$. Intuitively, then, $\mathsf{l} = \mathsf{j} \star \mathsf{k} \star \widehat{2^{\ulcorner\mathcal{Q}\urcorner}}$ numbers a proof of $\mathcal{Q}$—for we prove $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$, so that $\mathcal{Q}$ follows immediately as the last line by MP. So if PA $\vdash \Box(\mathcal{P} \to \mathcal{Q})$, then if PA $\vdash \Box\mathcal{P}$ then PA $\vdash \Box\mathcal{Q}$. The task is to prove all of this in PA. Thus, having shown that PA defines recursive functions and relations, we set out to obtain some further results about them.

13.4.1 Further Results

We have seen that PA defines functions and relations coordinate to ones from Chapter 12. Some of the elementary functions and relations so defined are equivalent to those already in PA. Now we require results for defined functions and relations beyond the elementary ones. Thus, proceeding roughly in the order from Chapter 12, we begin with results for exponentiation, factorial, and the like, and continue through to complex notions including $\mathbb{W}\!f\!f$ and $\mathcal{f}ormsub$. At this stage, we are acquiring results, not by demonstrating equivalence to expressions already defined (since there are no such expressions already defined), but by showing them directly for the coordinate functions and relations.

Let $\mathbb{p}ower(x, y) = x^y$; so we revert to the standard notation. Then,

*T13.40. The following are theorems of PA.

(a) (i) PA $\vdash m^{\emptyset} = \overline{1}$
(ii) PA $\vdash m^{Sn} = m^n \times m$

(b) PA $\vdash m^{\overline{1}} = m$

(c) PA $\vdash \emptyset < a \to \emptyset^a = \emptyset$

(d) PA $\vdash m^a \times m^b = m^{a+b}$

*(e) PA $\vdash m \le n \to m^a \le n^a$

(f) PA $\vdash \mathbb{p}red(m^b) | m^{a+b}$

(g) PA $\vdash \emptyset < m \to \emptyset < m^a$

*(h) PA $\vdash (\emptyset < m \wedge b \le a) \to m^b \le m^a$

(i) PA $\vdash (\overline{1} < m \wedge b < a) \to m^b < m^a$

(j) PA $\vdash \emptyset < a \to m \le m^a$

*(k) PA $\vdash (\emptyset < a \wedge \overline{1} < m) \to \mathbb{p}red(m^{a+b}) \nmid m^b$

*(l) PA $\vdash \overline{1} < m \to a < m^a$

*(m) PA $\vdash \overline{1} < m \to (m^a = m^b \to a = b)$

The first result (a) is immediate from the recursive definition and T13.37; it forms the basis for the rest. Then (b)–(m) are basic results that should be accessible from ordinary arithmetic.

*T13.41. The following are theorems of PA.

(a) (i) PA $\vdash fact(\emptyset) = \overline{1}$
(ii) PA $\vdash fact(Sn) = fact(n) \times Sn$

*(b) PA $\vdash \emptyset < fact(n)$

(c) PA $\vdash (\forall y < n) y | fact(n)$

*(d) $(\exists v \leq S fact(n))[n < v \wedge Pr(v)]$

These are some basic results for factorial. Again (a) gives the recursive conditions from which the rest follow. (b) is obvious. (c) is a consequence of the way the factorial includes successors of all the numbers less than it. (d) is like a result we have seen before according to which the successor of a product of primes is not divisible by any of those primes; so there is a prime not among them and less than or equal to the successor of the product (see G2 in the arithmetic for Gödel numbering reference). Take all the primes up to n—so other primes are greater than n; since $\mathsf{n!}$ includes the product of all the primes up to n, there is a prime greater than n but less than or equal to $\mathsf{S(n!)}$.

Observe that it is easy to obtain, say, PA $\vdash fact(\overline{1}) = \overline{1}$ from (a). But also $\mathsf{fact(n)}$ is captured by some $Fact(n, v)$, and since $\langle 1, 1 \rangle \in \mathsf{fact}$, PA $\vdash Fact(\overline{1}, \overline{1})$; but by T13.19 PA $\vdash \overline{1} = fact(\overline{1}) \leftrightarrow Fact(\overline{1}, \overline{1})$; so again, PA $\vdash fact(\overline{1}) = \overline{1}$. For sentences of this sort, we often assert the relevant fact with justification, 'cap(ture)'.

*T13.42. The following are theorems of PA.

(a) (i) PA $\vdash \mathit{pi}(\emptyset) = \overline{2}$
(ii) PA $\vdash \mathit{pi}(Sn) = (\mu z \leq S fact(\mathit{pi}(n)))[\mathit{pi}(n) < z \wedge Pr(z)]$

(b) PA $\vdash (\exists v \leq S fact(\mathit{pi}(n)))[\mathit{pi}(n) < v \wedge Pr(v)]$

*(c) PA $\vdash \mathit{pi}(Sn) = \mu z[\mathit{pi}(n) < z \wedge Pr(z)]$

(d) PA $\vdash \mathit{pi}(n) < \mathit{pi}(Sn) \wedge Pr(\mathit{pi}(Sn))$

(e) PA $\vdash (\forall w < \mathit{pi}(Sn)) \sim [\mathit{pi}(n) < w \wedge Pr(w)]$

*(f) PA $\vdash Pr(\mathit{pi}(n))$

(g) PA $\vdash \overline{1} < \mathit{pi}(n)$

(h) PA $\vdash \emptyset < \mathit{pi}(n)^a$

(i) PA $\vdash \emptyset < a \rightarrow \overline{1} < \mathit{pi}(n)^a$
corollary: PA $\vdash \emptyset < a \rightarrow \overline{1} < \overline{2}^a$

(j) PA $\vdash S\,\mathit{pred}(\mathit{pi}(n)^a) = \mathit{pi}(n)^a$

(k) PA $\vdash (\forall m < n)\mathit{pi}(m) < \mathit{pi}(n)$

(l) PA $\vdash (\forall m \leq n)Sm < \mathit{pi}(n)$

*(m) PA $\vdash \forall y[Pr(y) \rightarrow \exists j\, \mathit{pi}(j) = y]$

*(n) PA $\vdash m \neq n \rightarrow \mathit{pred}(\mathit{pi}(m)) \nmid \mathit{pi}(n)^a$

*(o) PA $\vdash m \neq n \rightarrow \mathit{pred}(\mathit{pi}(m)^{Sb}) \nmid \mathit{pi}(n)^a$

*(p) PA $\vdash [m \neq n \wedge \mathit{pred}(\mathit{pi}(m)^b)|(s \times \mathit{pi}(n)^a)] \rightarrow \mathit{pred}(\mathit{pi}(m)^b)|s$

These are some basic results for prime sequences. (a) gives the basic recursive conditions. (b) is an existential result that simply instantiates T13.41d; (c) uses it to extract the successor condition from bounded to unbounded minimization; this allows application of the definition in (d) and (e). (f)–(j) are some simple consequences of the fact that $\mathit{pi}(n)$ is prime. Then (k) $m < n$ implies $\mathit{pi}(m) < \mathit{pi}(n)$. And (l) each prime is greater than the successor of its index. (m) every prime appears as some $\mathit{pi}(j)$. And (n)–(p) echo results for factor except combined with primes and exponentiation.

In this theorem (b) and then (c)–(e) are a first instance of a pattern we shall see repeatedly: Given a bounded condition $\mathbb{a} = (\mu x \leq t)\mathcal{P}(x)$ from some primitive recursive definition, we show there exists an x less than or equal to the bound such that $\mathcal{P}(x)$; this allows application of T13.18e to "extract" the bounded to an unbounded minimization, and then T13.18 (a) and (b) to obtain $\mathcal{P}(\mathbb{a})$ and that for $z < \mathbb{a}$, $\sim\mathcal{P}(z)$; this forms the basis for further results.

*T13.43. The following are theorems of PA.

(a) PA $\vdash \mathit{exp}(n, i) = (\mu x \leq n)[\mathit{pred}(\mathit{pi}(i)^{Sx}) \nmid n]$

(b) PA $\vdash \mathit{exp}(\emptyset, i) = \emptyset$

*(c) PA $\vdash \mathit{exp}(Sn, i) = \mu x[\mathit{pred}(\mathit{pi}(i)^{Sx}) \nmid Sn]$

(d) PA $\vdash \mathit{pred}(\mathit{pi}(i)^{S\mathit{exp}(Sn,i)}) \nmid Sn$

(e) PA $\vdash (\forall w < \mathit{exp}(Sn, i))\mathit{pred}(\mathit{pi}(i)^{Sw})|Sn]$

*(f) PA $\vdash \mathit{pred}(\mathit{pi}(i)^{\mathit{exp}(Sn,i)})|Sn$

(g) PA $\vdash [\mathit{pred}(\mathit{pi}(i)^a)|Sn \wedge \mathit{pred}(\mathit{pi}(i)^{Sa}) \nmid Sn] \rightarrow \mathit{exp}(Sn, i) = a$

*(h) PA $\vdash \mathit{exp}(m, j) \leq m$

(i) PA $\vdash n \leq j \rightarrow \mathit{exp}(Sn, j) = \emptyset$

(j) PA $\vdash \mathit{exp}(\mathit{pi}(i)^p, i) = p$

*(k) PA $\vdash i \neq j \rightarrow \mathit{exp}(\mathit{pi}(i)^p, j) = \emptyset$

*(l) PA $\vdash$ $\mathit{pred}(\mathit{pi}(i))|Sm \leftrightarrow \overline{1} \leq \mathit{exp}(Sm, i)$

*(m) PA $\vdash$ $\exists q[\mathit{pi}(i)^{\mathit{exp}(Sn,i)} \times q = Sn \wedge \mathit{pred}(\mathit{pi}(i)) \nmid q \wedge$
$\forall y(y \neq i \rightarrow \mathit{exp}(q, y) = \mathit{exp}(Sn, y))]$

*(n) PA $\vdash$ $\mathit{exp}(Sm \times Sn, i) = \mathit{exp}(Sm, i) + \mathit{exp}(Sn, i)$

(a) is from the definition. (b) is the standard result for minimization with bound $\emptyset$. (c) extracts the successor case from the bounded to an unbounded minimization; this allows application of the definition in (d) and (e). From (f) a prime to the power of its exponent in the factorization of Sn divides Sn. From (g) if some a has features of the exponent as in both (f) and (d) then a is the exponent. Then (h) the exponent of some prime in the factorization of m cannot be greater than m; and (i) a prime whose index is greater than or equal to n does not divide into Sn. (j) and (k) make an obvious connection for the exponent of a prime, and (l) between exponent and factor. According (m) once you divide Sn by $\mathit{pi}(i)^{\mathit{exp}(Sn,i)}$, you are left with a q such that $\mathit{pi}(i)$ does not divide into it any more, and such that the exponents of all the other primes remain the same as in Sn. From (n) the i^{th} exponent of a product sums the i^{th} exponents of its factors.

*T13.44. The following are theorems of PA.

(a) PA $\vdash$ $\mathit{len}(n) = (\mu y \leq n)(\forall z \leq n)[y \leq z \rightarrow \mathit{exp}(n, z) = \emptyset]$

*(b) PA $\vdash$ $\mathit{len}(Sn) = \mu y(\forall z \leq Sn)[y \leq z \rightarrow \mathit{exp}(Sn, z) = \emptyset]$

(c) PA $\vdash$ $(\forall z \leq Sn)[\mathit{len}(Sn) \leq z \rightarrow \mathit{exp}(Sn, z) = \emptyset]$

(d) PA $\vdash$ $(\forall w < \mathit{len}(Sn)){\sim}(\forall z \leq Sn)[w \leq z \rightarrow \mathit{exp}(Sn, z) = \emptyset]$

(e) PA $\vdash$ $\emptyset < \mathit{len}(m) \rightarrow \overline{1} < m$

*(f) PA $\vdash$ $\emptyset < \mathit{exp}(m, i) \rightarrow i < \mathit{len}(m)$

*(g) PA $\vdash$ $(\forall k : l < k)\mathit{exp}(Sm, k) = \emptyset \rightarrow \mathit{len}(Sm) \leq Sl$

*(h) PA $\vdash$ $\overline{1} < m \rightarrow \emptyset < \mathit{len}(m)$

*(i) PA $\vdash$ $\emptyset < p \rightarrow \mathit{len}(\mathit{pi}(i)^p) = Si$
corollary: PA $\vdash$ $\emptyset < p \rightarrow \mathit{len}(\overline{2}^p) = \overline{1}$

(j) PA $\vdash$ $(\forall z : \mathit{len}(n) \leq z)\mathit{exp}(n, z) = \emptyset$

*(k) PA $\vdash$ $\mathit{len}(n) = Sl \rightarrow \overline{1} \leq \mathit{exp}(n, l)$

Again (a) is from the definition. (b) extracts the successor case from bounded to unbounded minimization; (c) and (d) then apply the definition. (e) is immediate from $\mathit{len}(\emptyset) = \emptyset$ and $\mathit{len}(\overline{1}) = \emptyset$. From (f) if an exponent of some prime in the factorization of m is greater than zero, that prime is involved in the factorization of m; (g) length is set up so that it finds the first prime such that it and all the ones after have exponent zero; so if all the primes after some l have exponent zero, then the length is no greater than Sl; (h) gives the biconditional from (e); (i) gives the length for a prime to any power; and from (j) primes $\geq$ the length of n must all have exponent $\emptyset$; (k) the prime prior to the length has exponent $\geq \overline{1}$.

For the rest of this section including results for concatenation to follow, it will be helpful to introduce a pair of auxiliary notions. First $exc(m,n,i)$ takes the value of the i^{th} exponent in the concatenation of m and n. Let PA $\vdash exc(m,n,i) =$

$$\mu y([i < \mathbb{len}(m) \wedge y = \mathbb{exp}(m,i)] \vee [\mathbb{len}(m) \leq i \wedge y = \mathbb{exp}(n, i \dot{-} \mathbb{len}(m))])$$

It is left as an exercise to show that PA proves the existential condition, and so defines exc. The idea is simply to set y to one or the other of $\mathbb{exp}(m,i)$ or $\mathbb{exp}(n, i \dot{-} \mathbb{len}(m))$ so that y takes the value of the i^{th} exponent in the concatenation. Next, $\mathbb{val}(n,i)$ returns the product of the first i members of the prime factorization of n. $\mathbb{val}$ is defined by recursion so that,

PA $\vdash \mathbb{val}(n,\emptyset) = \overline{1}$
PA $\vdash \mathbb{val}(n,Sy) = \mathbb{val}(n,y) \times \mathbb{pi}(y)^{\mathbb{exp}(n,y)}$

Similarly $\mathbb{val}^*(m,n,i)$ is defined by recursion and,

PA $\vdash \mathbb{val}^*(m,n,\emptyset) = \overline{1}$
PA $\vdash \mathbb{val}^*(m,n,Sy) = \mathbb{val}^*(m,n,y) \times \mathbb{pi}(y)^{exc(m,n,y)}$

So $\mathbb{val}^*(m,n,i)$ returns the product of the first i primes in the factorization of the concatenation of m and n. Here are some results for these notions. Let $l = \mathbb{len}(m) + \mathbb{len}(n)$.

*T13.45. The following are theorems of PA.

(a) PA $\vdash i < \mathbb{len}(m) \rightarrow exc(m,n,i) = \mathbb{exp}(m,i)$

(b) PA $\vdash \mathbb{len}(m) \leq i \rightarrow exc(m,n,i) = \mathbb{exp}(n, i \dot{-} \mathbb{len}(m))$

(c) PA $\vdash \emptyset < \mathbb{val}^*(m,n,i)$

*(d) PA $\vdash (\forall i : a \leq i)\mathbb{pred}(\mathbb{pi}(i)) \nmid \mathbb{val}^*(m,n,a)$

*(e) PA $\vdash (\forall j < i)\mathbb{exp}(\mathbb{val}^*(m,n,i), j) = exc(m,n,j)$

*(f) PA $\vdash (\forall i < \mathbb{len}(m))[\mathbb{exp}(\mathbb{val}^*(m,n,l),i) = \mathbb{exp}(m,i)] \wedge$
$(\forall i < \mathbb{len}(n))[\mathbb{exp}(\mathbb{val}^*(m,n,l), i + \mathbb{len}(m)) = \mathbb{exp}(n,i)]$

*(g) PA $\vdash \mathbb{val}^*(m,n,l) \leq [\mathbb{pi}(l)^{m+n}]^l$

(h) PA $\vdash \emptyset < \mathbb{val}(m,i)$

*(i) $(\forall i : a \leq i)\mathbb{pred}(\mathbb{pi}(i)) \nmid \mathbb{val}(m,a)$

*(j) PA $\vdash (\forall j < i)\mathit{exp}(\mathit{val}(m,i), j) = \mathit{exp}(m, j)$

(k) PA $\vdash \mathit{len}(\mathit{val}(a, j)) \leq j$

*(l) PA $\vdash \mathit{len}(\mathit{val}(a, j)) \leq \mathit{len}(a)$

(m) PA $\vdash (\forall i < k)\mathit{exp}(a,i) = \mathit{exp}(b,i) \to \mathit{val}(a,k) = \mathit{val}(b,k)$

*(n) PA $\vdash \mathit{len}(Sn) \leq x \to \mathit{val}(Sn, x) = Sn$
corollary: PA $\vdash \mathit{val}(Sn, \mathit{len}(Sn)) = Sn$

*(o) PA $\vdash [\mathit{len}(n) \leq q \wedge (\forall k < \mathit{len}(n))\mathit{exp}(n,k) \leq r] \to$
$\mathit{val}(n, \mathit{len}(n)) \leq [\mathit{pi}(q)^r]^q$

(a) and (b) apply the definition for *exc*. (c) is obvious. (d) results because $\mathit{val}^*(m,n,a)$ is a product of primes prior to $\mathit{pi}(a)$ so that greater primes do not divide it. Then (e) the exponents in val^* are like the exponents in *exc*. This gives us (f) that the exponents in val^* are like the exponents in m and n. But (g) val^* is constructed so that $\mathit{val}^*(m,n,l)$ is always less than or equal to $[\mathit{pi}(l)^{m+n}]^l$. Then (h)–(o) are related results for *val*. In cases to follow, (g) and the closely related (o) will be crucial for finding bounds and so extracting results from bounded minimization.

We are now ready for some results about concatenation. Again let $m * n$ be the defined correlate to $\mathsf{m} \star \mathsf{n}$; and as above let $l = \mathit{len}(m) + \mathit{len}(n)$.

*T13.46. The following are theorems of PA.

(a) (i) PA $\vdash m * n = (\mu x \leq B_{m,n})\{\overline{1} \leq x \wedge$
$(\forall i < \mathit{len}(m))[\mathit{exp}(x,i) = \mathit{exp}(m,i)] \wedge$
$(\forall i < \mathit{len}(n))[\mathit{exp}(x, i + \mathit{len}(m)) = \mathit{exp}(n,i)]\}$
(ii) PA $\vdash B_{m,n} = [\mathit{pi}(l)^{m+n}]^l$

(b) PA $\vdash m * n = \mu x\{\overline{1} \leq x \wedge (\forall i < \mathit{len}(m))[\mathit{exp}(x,i) = \mathit{exp}(m,i)] \wedge$
$(\forall i < \mathit{len}(n))[\mathit{exp}(x, i + \mathit{len}(m)) = \mathit{exp}(n,i)]\}$

(c) PA $\vdash \overline{1} \leq m * n \wedge (\forall i < \mathit{len}(m))[\mathit{exp}(m * n, i) = \mathit{exp}(m,i)] \wedge$
$(\forall i < \mathit{len}(n))[\mathit{exp}(m * n, i + \mathit{len}(m)) = \mathit{exp}(n,i)]$

(d) PA $\vdash (\forall w < m * n){\sim}\{\overline{1} \leq w \wedge (\forall i < \mathit{len}(m))[\mathit{exp}(w,i) = \mathit{exp}(m,i)] \wedge$
$(\forall i < \mathit{len}(n))[\mathit{exp}(w, i + \mathit{len}(m)) = \mathit{exp}(n,i)]\}$

*(e) PA $\vdash \mathit{len}(m * n) = l$

(f) PA $\vdash \mathit{exp}(m * n, i + \mathit{len}(m)) = \mathit{exp}(n,i)$

*(g) PA $\vdash (a * b) * c = a * (b * c)$

(h) PA $\vdash n \leq \overline{1} \to Sm * n = Sm$

(i) PA $\vdash n \leq \overline{1} \to n * Sm = Sm$

*(j) PA $\vdash (\mathbb{len}(c) = \mathbb{len}(d) \wedge Sa * c = Sb * d) \rightarrow Sa = Sb$
corollary: PA $\vdash Sa * c = Sb * c \rightarrow Sa = Sb$
corollary: PA $\vdash (\mathbb{len}(Sa) = \mathbb{len}(Sb) \wedge Sa * c = Sb * d) \rightarrow Sa = Sb$

(k) PA $\vdash (\mathbb{len}(c) = \mathbb{len}(d) \wedge c * Sa = d * Sb) \rightarrow Sa = Sb$
corollary: PA $\vdash c * Sa = c * Sb \rightarrow Sa = Sb$
corollary: PA $\vdash (\mathbb{len}(Sa) = \mathbb{len}(Sb) \wedge c * Sa = d * Sb) \rightarrow Sa = Sb$

*(l) PA $\vdash \mathbb{val}(Sm * Sn, a) = \mathbb{val}(Sm, a) * \mathbb{val}(Sn, a \dot{-} \mathbb{len}(Sm))$

*(m) PA $\vdash \mathbb{val}(m, \mathbb{len}(m)) \leq \mathbb{val}(m * n, a + \mathbb{len}(m))$
corollary: PA $\vdash m \leq m * n$

(n) PA $\vdash \mathbb{val}(n, a) \leq \mathbb{val}(m * n, a + \mathbb{len}(m))$
corollary: PA $\vdash n \leq m * n$

(a) is from the definition. T13.45g enables us to extract $m * n$ from bounded to unbounded minimization to get (b) and then (c) and (d). From (e) the length of $m * n$ sums the lengths of m and n. (f) generalizes the last conjunct of (c). (g) is an association result—and with this, we typically drop parentheses for concatenations (of course, although it associates, $*$ does not commute). From (h) and (i) concatenation with a number less than or equal to one results in no change. (j) and (k) enable a sort of cancellation law for concatenation. (l) distributes $\mathbb{val}$ over concatenation; then (m) and (n) tell us that the number of a concatenation is greater than or equal to the numbers of its parts.

The idea for application of T13.45g to get (b) is the same as behind the intuitive account of the bound from Chapter 12: $\mathbb{pi}(l)^{m+n}$ is greater than every term in the factorization of $m * n$; so $[\mathbb{pi}(l)^{m+n}]^i$ remains greater than $\mathbb{val}^*(m, n, i)$; and $\mathbb{val}^*(m, n, l)$ is therefore both under the bound and, with T13.45f, satisfies the definition T13.46a for $m * n$—so the existential condition is satisfied, and we may extract the bounded to an unbounded minimization. Once this is accomplished, we are most of the way home.

To manipulate $\mathbb{Termseq}$ it will be convenient to let,

$$\begin{array}{lcl} A(m,k) & = & \mathbb{exp}(m,k) = \overline{\ulcorner \emptyset \urcorner} \vee \mathbb{Var}(\mathbb{exp}(m,k)) \\ B(m,k) & = & (\exists j < k)\mathbb{exp}(m,k) = \overline{\ulcorner S \urcorner} * \mathbb{exp}(m,j) \\ C(m,k) & = & (\exists i < k)(\exists j < k)\mathbb{exp}(m,k) = \overline{\ulcorner + \urcorner} * \mathbb{exp}(m,i) * \mathbb{exp}(m,j) \\ D(m,k) & = & (\exists i < k)(\exists j < k)\mathbb{exp}(m,k) = \overline{\ulcorner \times \urcorner} * \mathbb{exp}(m,i) * \mathbb{exp}(m,j) \end{array}$$

*T13.47. The following are theorems of PA.

(a) PA $\vdash \mathbb{Var}(v) \leftrightarrow (\exists x \leq v)\big(v = \overline{2}^{\overline{23} + \overline{2}x}\big)$

(b) PA $\vdash \mathbb{Termseq}(m, t) \leftrightarrow \{\mathbb{exp}(m, \mathbb{len}(m) \dot{-} \overline{1}) = t \wedge \overline{1} < m \wedge$
$(\forall k < \mathbb{len}(m))[A(m,k) \vee B(m,k) \vee C(m,k) \vee D(m,k)]\}$

(c) (i) PA $\vdash \mathbb{T}erm(t) \leftrightarrow (\exists x \leq B_t)\mathbb{T}ermseq(x,t)$
(ii) PA $\vdash B_t = [\mathbb{pi}(\mathbb{len}(t))^t]^{\mathbb{len}(t)}$

(d) PA $\vdash \mathbb{V}ar(v) \to \mathbb{len}(v) = \overline{1}$

(e) PA $\vdash \mathbb{V}ar(v) \to (\mathbb{V}ar(v \times \overline{4}) \wedge v \times \overline{4} \neq v)$

*(f) PA $\vdash \mathbb{T}ermseq(m,t) \to (\forall k < \mathbb{len}(m))(\overline{1} < \mathbb{exp}(m,k))$

(g) PA $\vdash \mathbb{T}erm(t) \to \overline{1} < t$

*(h) PA $\vdash t = \overline{\ulcorner \emptyset \urcorner} \to \mathbb{T}ermseq(\overline{2}^t, t)$

(i) PA $\vdash \mathbb{V}ar(t) \to \mathbb{T}ermseq(\overline{2}^t, t)$

*(j) PA $\vdash \mathbb{T}ermseq(m,t) \to \mathbb{T}ermseq(m * \overline{2}^{\overline{\ulcorner S \urcorner} * t}, \overline{\ulcorner S \urcorner} * t)$

*(k) PA $\vdash [\mathbb{T}ermseq(m,t) \wedge \mathbb{T}ermseq(n,q)] \to$
$\mathbb{T}ermseq(m * n * \overline{2}^{\overline{\ulcorner + \urcorner} * t * q}, \overline{\ulcorner + \urcorner} * t * q)$

(l) PA $\vdash [\mathbb{T}ermseq(m,t) \wedge \mathbb{T}ermseq(n,q)] \to$
$\mathbb{T}ermseq(m * n * \overline{2}^{\overline{\ulcorner \times \urcorner} * t * q}, \overline{\ulcorner \times \urcorner} * t * q)$

*(m) PA $\vdash \mathbb{T}ermseq(m,t) \to (\forall k < \mathbb{len}(m))\exists n[\mathbb{T}ermseq(n, \mathbb{exp}(m,k)) \wedge$
$\mathbb{len}(n) \leq \mathbb{len}(\mathbb{exp}(m,k)) \wedge (\forall i < \mathbb{len}(n))\mathbb{exp}(n,i) \leq \mathbb{exp}(m,k)]$

(n) PA $\vdash \mathbb{T}ermseq(m,t) \to (\forall i < \mathbb{len}(m))\mathbb{T}erm(\mathbb{exp}(m,i))$
corollary: PA $\vdash \mathbb{T}ermseq(m,t) \to \mathbb{T}erm(t)$

(o) PA $\vdash \mathbb{V}ar(v) \to \mathbb{T}erm(v)$

*(p) PA $\vdash \mathbb{T}erm(t) \to \mathbb{T}erm(\overline{\ulcorner S \urcorner} * t)$

(q) PA $\vdash (\mathbb{T}erm(s) \wedge \mathbb{T}erm(t)) \to \mathbb{T}erm(\overline{\ulcorner + \urcorner} * s * t)$

(r) PA $\vdash (\mathbb{T}erm(s) \wedge \mathbb{T}erm(t)) \to \mathbb{T}erm(\overline{\ulcorner \times \urcorner} * s * t)$

(a), (b), and (c) are from the definitions variable, term sequence, and term. (d)–(g) are simple results. (h)–(l) generate term sequences; they are important for (m) according to which each member of a term sequence has a term sequence constrained by bounds realted to B_t. (m) yields (n), that anything with a term sequence is a term; the rest follow from that.

From its definition, $\mathbb{T}erm(t)$ does not immediately follow from $\mathbb{T}ermseq(m,t)$ insofar as a sequence might build in extraneous terms not required for t—with the result that m is not less than B_t (compare page 615, note 16). The general idea for these theorems is that given a term sequence, there is a *standard* term sequence containing just the elements you would have included in a Chapter 2 tree, adequate to yield $\mathbb{T}erm(t)$. Thus we move from the existence of a term sequence through (m) to a term sequence of the right sort, and so to (n). Something new happens in (m) insofar as the induction is not on the length of m but on the length of its *exponents*. Reasoning, as it were, "down" through the tree and using (h)–(l), we show that for each member of the original sequence there is a "standard" sequence that comes in under its bound (and so a sequence under the bound for t).

We continue with some results for $\mathbb{F}ormseq$ and $\mathbb{W}\!f\!f$ that are closely related to T13.47. Let,

$$\begin{array}{lcl}
E(m,k) & = & \mathbb{A}tomic(\mathbb{exp}(m,k)) \\
F(m,k) & = & (\exists j < k)[\mathbb{exp}(m,k) = \mathbb{til}(\mathbb{exp}(m,j))] \\
G(m,k) & = & (\exists i < k)(\exists j < k)[\mathbb{exp}(m,k) = \mathbb{cnd}(\mathbb{exp}(m,i),\mathbb{exp}(m,j))] \\
H(p,m,k) & = & (\exists i < k)(\exists j \leq p)[\mathbb{V}ar(j) \wedge \mathbb{exp}(m,k) = \mathbb{unv}(j,\mathbb{exp}(m,i))]
\end{array}$$

*T13.48. The following are theorems of PA.

(a) PA $\vdash \mathbb{A}tomic(p) \leftrightarrow$
$(\exists x \leq p)(\exists y \leq p)[\mathbb{T}erm(x) \wedge \mathbb{T}erm(y) \wedge p = \overline{\ulcorner = \urcorner} * x * y]$

(b) PA $\vdash \mathbb{F}ormseq(m,p) \leftrightarrow \{\mathbb{exp}(m,\mathbb{len}(m) \dot{-} \overline{1}) = p \wedge \overline{1} < m \wedge$
$(\forall k < \mathbb{len}(m))[E(m,k) \vee F(m,k) \vee G(m,k) \vee H(p,m,k)]\}$

(c) (i) PA $\vdash \mathbb{W}\!f\!f(p) \leftrightarrow (\exists x \leq B_p)\mathbb{F}ormseq(x,p)$
(ii) PA $\vdash B_p = [\mathbb{pi}(\mathbb{len}(p))^p]^{\mathbb{len}(p)}$

(d) PA $\vdash \mathbb{F}ormseq(m,p) \to (\forall k < \mathbb{len}(m))(\overline{1} < \mathbb{exp}(m,k))$

(e) PA $\vdash \mathbb{W}\!f\!f(p) \to \overline{1} < p$

*(f) PA $\vdash \mathbb{A}tomic(p) \to \mathbb{F}ormseq(\overline{2}^p, p)$

(g) PA $\vdash \mathbb{F}ormseq(m,p) \to \mathbb{F}ormseq(m * \overline{2}^{\mathbb{til}(p)}, \mathbb{til}(p))$

(h) PA $\vdash [\mathbb{F}ormseq(m,p) \wedge \mathbb{F}ormseq(n,q)] \to$
$\mathbb{F}ormseq(m * n * \overline{2}^{\mathbb{cnd}(p,q)}, \mathbb{cnd}(p,q))$

*(i) PA $\vdash [\mathbb{F}ormseq(m,p) \wedge \mathbb{V}ar(v)] \to \mathbb{F}ormseq(m * \overline{2}^{\mathbb{unv}(v,p)}, \mathbb{unv}(v,p))$

*(j) PA $\vdash \mathbb{F}ormseq(m,p) \to (\forall k < \mathbb{len}(m))\exists n[\mathbb{F}ormseq(n,\mathbb{exp}(m,k)) \wedge$
$\mathbb{len}(n) \leq \mathbb{len}(\mathbb{exp}(m,k)) \wedge (\forall i < \mathbb{len}(n))\mathbb{exp}(n,i) \leq \mathbb{exp}(m,k)]$

(k) PA $\vdash \mathbb{F}ormseq(m,p) \to (\forall i < \mathbb{len}(m))\mathbb{W}\!f\!f(\mathbb{exp}(m,i))$
corollary: PA $\vdash \mathbb{F}ormseq(m,p) \to \mathbb{W}\!f\!f(p)$

(l) PA $\vdash \mathbb{A}tomic(p) \to \mathbb{W}\!f\!f(p)$

(m) PA $\vdash \mathbb{W}\!f\!f(p) \to \mathbb{W}\!f\!f(\mathbb{til}(p))$

*(n) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{W}\!f\!f(q)] \to \mathbb{W}\!f\!f(\mathbb{cnd}(p,q))$

(o) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{V}ar(v)] \to \mathbb{W}\!f\!f(\mathbb{unv}(v,p))$

(p) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{W}\!f\!f(q)] \to \mathbb{W}\!f\!f(\mathbb{caret}(p,q))$

Again, from its definition, $\mathbb{W}\!f\!f(p)$ does not immediately follow from $\mathbb{F}ormseq(m, p)$ insofar as the sequence might build in extraneous elements not required for p—with the result that m is not less than B_p. And again the general idea is that given a formula sequence, there is a *standard* formula sequence containing just the elements you would have included in a Chapter 2 tree, adequate to yield $\mathbb{W}\!f\!f(p)$. Thus we move from the existence of a formula sequence through (j) to a formula sequence of the required sort.

Continuing roughly in the order of Chapter 12, we move on to some substitution results for terms. For $\mathbb{T}subseq$ let,

$$\begin{array}{lcl}
I(m,n,k) & = & \mathbb{e}xp(m,k) = \overline{\ulcorner\emptyset\urcorner} \wedge \mathbb{e}xp(n,k) = \overline{\ulcorner\emptyset\urcorner} \\
J(v,m,n,k) & = & \mathbb{V}ar(\mathbb{e}xp(m,k)) \wedge \mathbb{e}xp(m,k) \neq v \wedge \mathbb{e}xp(n,k) = \mathbb{e}xp(m,k) \\
K(v,s,m,n,k) & = & \mathbb{V}ar(\mathbb{e}xp(m,k)) \wedge \mathbb{e}xp(m,k) = v \wedge \mathbb{e}xp(n,k) = s \\
L(m,n,k) & = & (\exists i < k)[\mathbb{e}xp(m,k) = \overline{\ulcorner S\urcorner} * \mathbb{e}xp(m,i) \wedge \mathbb{e}xp(n,k) = \overline{\ulcorner S\urcorner} * \mathbb{e}xp(n,i)] \\
M(m,n,k) & = & (\exists i < k)(\exists j < k)[\mathbb{e}xp(m,k) = \overline{\ulcorner +\urcorner} * \mathbb{e}xp(m,i) * \mathbb{e}xp(m,j) \wedge \\
 & & \quad \mathbb{e}xp(n,k) = \overline{\ulcorner +\urcorner} * \mathbb{e}xp(n,i) * \mathbb{e}xp(n,j)] \\
N(m,n,k) & = & (\exists i < k)(\exists j < k)[\mathbb{e}xp(m,k) = \overline{\ulcorner \times\urcorner} * \mathbb{e}xp(m,i) * \mathbb{e}xp(m,j) \wedge \\
 & & \quad \mathbb{e}xp(n,k) = \overline{\ulcorner \times\urcorner} * \mathbb{e}xp(n,i) * \mathbb{e}xp(n,j)]
\end{array}$$

*T13.49. The following are theorems of PA.

(a) PA $\vdash \mathbb{T}subseq(m,n,t,v,s,u) \leftrightarrow \{\mathbb{T}ermseq(m,t) \wedge \mathbb{l}en(m) = \mathbb{l}en(n) \wedge \mathbb{e}xp(n, \mathbb{l}en(n) \dot{-} \overline{1}) = u \wedge (\forall k < \mathbb{l}en(m))[I(m,n,k) \vee J(v,m,n,k) \vee K(v,s,m,n,k) \vee L(m,n,k) \vee M(m,n,k) \vee N(m,n,k)]\}$

(b) (i) PA $\vdash \mathbb{T}ermsub(t,v,s,u) \leftrightarrow (\exists x \leq X_t)(\exists y \leq Y_{t,u})\mathbb{T}subseq(x,y,t,v,s,u)$

(ii) PA $\vdash X_t = [\mathbb{p}i(\mathbb{l}en(t))^t]^{\mathbb{l}en(t)}$

(iii) PA $\vdash Y_{t,u} = [\mathbb{p}i(\mathbb{l}en(t))^u]^{\mathbb{l}en(t)}$

*(c) PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{T}subseq(m,n,t,v,s,u)] \to (\forall j < \mathbb{l}en(n))\mathbb{T}erm(\mathbb{e}xp(n,j))$

corollary: PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{T}ermsub(t,v,s,u)] \to \mathbb{T}erm(u)$

(d) PA $\vdash t = \overline{\ulcorner\emptyset\urcorner} \to \mathbb{T}subseq(\overline{2}^t, \overline{2}^t, t, v, s, t)$

(e) PA $\vdash (\mathbb{V}ar(t) \wedge t \neq v) \to \mathbb{T}subseq(\overline{2}^t, \overline{2}^t, t, v, s, t)$

*(f) PA $\vdash (\mathbb{T}erm(s) \wedge \mathbb{V}ar(t) \wedge t = v) \to \mathbb{T}subseq(\overline{2}^t, \overline{2}^s, t, v, s, s)$

*(g) PA $\vdash \mathbb{T}subseq(m,n,t,v,s,u) \to \mathbb{T}subseq(m * \overline{2}^{\overline{\ulcorner S\urcorner}*t}, n * \overline{2}^{\overline{\ulcorner S\urcorner}*u}, \overline{\ulcorner S\urcorner} * t, v, s, \overline{\ulcorner S\urcorner} * u)$

(h) PA $\vdash [\mathbb{T}subseq(m,n,t,v,s,u) \wedge \mathbb{T}subseq(m',n',t',v,s,u')] \to \mathbb{T}subseq(m*m'*\overline{2}^{\overline{\ulcorner +\urcorner}*t*t'}, n*n'*\overline{2}^{\overline{\ulcorner +\urcorner}*u*u'}, \overline{\ulcorner +\urcorner}*t*t', v, s, \overline{\ulcorner +\urcorner}*u*u')$

(i) PA $\vdash [\mathbb{T}subseq(m,n,t,v,s,u) \wedge \mathbb{T}subseq(m',n',t',v,s,u')] \to \mathbb{T}subseq(m*m'*\overline{2}^{\overline{\ulcorner \times\urcorner}*t*t'}, n*n'*\overline{2}^{\overline{\ulcorner \times\urcorner}*u*u'}, \overline{\ulcorner \times\urcorner}*t*t', v, s, \overline{\ulcorner \times\urcorner}*u*u')$

*(j) PA $\vdash \mathbb{T}subseq(m, n, t, v, s, u) \rightarrow \mathbb{T}ermsub(t, v, s, u)$

*(k) PA $\vdash [\mathbb{T}erm(t) \wedge \mathbb{T}erm(s)] \rightarrow \exists u[\mathbb{T}ermsub(t, v, s, u) \wedge$
$\mathbb{l}en(u) \leq \mathbb{l}en(t) \times \mathbb{l}en(s) \wedge (\forall k < \mathbb{l}en(u))\mathbb{e}xp(u, k) \leq t + s]$

(a)–(b) are from the definitions. (c) follows directly. (d)–(i) generate sequences to yield (j). Then (k) establishes bounds on a term substitution, required for corresponding bounds related to $\mathcal{f}ormsub$ as for T13.50n,o below.

Some substitution results for formulas are closely related to the previous theorem. Let,

$$
\begin{aligned}
O(v,s,m,n,k) &= \mathbb{A}tomic(\mathbb{e}xp(m,k)) \wedge \mathbb{A}tomsub(\mathbb{e}xp(m,k), v, s, \mathbb{e}xp(n,k)) \\
P(m,n,k) &= (\exists i < k)[\mathbb{e}xp(m,k) = \mathbb{t}il(\mathbb{e}xp(m,i)) \wedge \mathbb{e}xp(n,k) = \mathbb{t}il(\mathbb{e}xp(n,i))] \\
Q(m,n,k) &= (\exists i < k)(\exists j < k)[\mathbb{e}xp(m,k) = \mathbb{c}nd(\mathbb{e}xp(m,i), \mathbb{e}xp(m,j)) \wedge \\
&\qquad \mathbb{e}xp(n,k) = \mathbb{c}nd(\mathbb{e}xp(n,i), \mathbb{e}xp(n,j))] \\
R(v,p,m,n,k) &= (\exists i < k)(\exists j \leq p)[\mathbb{V}ar(j) \wedge j \neq v \wedge \mathbb{e}xp(m,k) = \mathbb{u}nv(j, \mathbb{e}xp(m,i)) \wedge \\
&\qquad \mathbb{e}xp(n,k) = \mathbb{u}nv(j, \mathbb{e}xp(n,i))] \\
S(v,p,m,n,k) &= (\exists i < k)(\exists j \leq p)[\mathbb{V}ar(j) \wedge j = v \wedge \mathbb{e}xp(m,k) = \mathbb{u}nv(j, \mathbb{e}xp(m,i)) \wedge \\
&\qquad \mathbb{e}xp(n,k) = \mathbb{e}xp(m,k)]
\end{aligned}
$$

*T13.50. The following are theorems of PA.

(a) PA $\vdash \mathbb{A}tomsub(p, v, s, q) \leftrightarrow$
$(\exists a \leq p)(\exists b \leq p)(\exists a' \leq q)(\exists b' \leq q)[\mathbb{T}erm(a) \wedge \mathbb{T}erm(b) \wedge$
$p = \overline{\ulcorner = \urcorner} * a * b \wedge \mathbb{T}ermsub(a, v, s, a') \wedge \mathbb{T}ermsub(b, v, s, b') \wedge$
$q = \overline{\ulcorner = \urcorner} * a' * b']$

(b) PA $\vdash \mathbb{F}subseq(m, n, p, v, s, q) \leftrightarrow \{\mathbb{F}ormseq(m, p) \wedge \mathbb{l}en(m) = \mathbb{l}en(n) \wedge$
$\mathbb{e}xp(n, \mathbb{l}en(n) \dot{-} \overline{1}) = q \wedge (\forall k < \mathbb{l}en(m))[O(v, s, m, n, k) \vee P(m, n, k) \vee$
$Q(m, n, k) \vee R(v, p, m, n, k) \vee S(v, p, m, n, k)]\}$

(c) (i) PA $\vdash \mathbb{F}ormsub(p, v, s, q) \leftrightarrow$
$(\exists x \leq X_p)(\exists y \leq Y_{p,q})\mathbb{F}subseq(x, y, p, v, s, q)$
(ii) PA $\vdash X_p = [\mathbb{p}i(\mathbb{l}en(p))^p]^{\mathbb{l}en(p)}$
(iii) PA $\vdash Y_{p,q} = [\mathbb{p}i(\mathbb{l}en(p))^q]^{\mathbb{l}en(p)}$

(d) (i) PA $\vdash \mathcal{f}ormsub(p, v, s) = (\mu q \leq Z_{p,s})\mathbb{F}ormsub(p, v, s, q)$
(ii) PA $\vdash Z_{p,s} = [\mathbb{p}i(\mathbb{l}en(p) \times \mathbb{l}en(s))^{p+s}]^{\mathbb{l}en(p) \times \mathbb{l}en(s)}$

(e) PA $\vdash \mathbb{A}tomsub(p, v, s, q) \rightarrow \overline{1} < q$

(f) PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{A}tomsub(p, v, s, q)] \rightarrow \mathbb{A}tomic(q)$

(g) PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{F}subseq(m, n, p, v, s, q)] \rightarrow (\forall j < \mathbb{l}en(n))\mathbb{W}\!ff(\mathbb{e}xp(n, j))$
corollary: PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{F}ormsub(p, v, s, q)] \rightarrow \mathbb{W}\!ff(q)$

(h) PA $\vdash [\mathbb{A}tomic(p) \wedge \mathbb{A}tomsub(p, v, s, q)] \rightarrow \mathbb{F}subseq(\overline{2}^p, \overline{2}^q, p, v, s, q)$

*(i) PA $\vdash \mathbb{F}subseq(m,n,p,v,s,q) \rightarrow$
$\mathbb{F}subseq(m * \overline{2}^{\mathbb{t}il(p)}, n * \overline{2}^{\mathbb{t}il(q)}, \mathbb{t}il(p), v, s, \mathbb{t}il(q))$

(j) PA $\vdash [\mathbb{F}subseq(m,n,p,v,s,q) \wedge \mathbb{F}subseq(m',n',p',v,s,q')] \rightarrow$
$\mathbb{F}subseq(m * m' * \overline{2}^{\mathbb{c}nd(p,p')}, n * n' * \overline{2}^{\mathbb{c}nd(q,q')}, \mathbb{c}nd(p,p'), v, s, \mathbb{c}nd(q,q'))$

(k) PA $\vdash [\mathbb{F}subseq(m,n,p,v,s,q) \wedge \mathbb{V}ar(u) \wedge u \neq v] \rightarrow$
$\mathbb{F}subseq(m * \overline{2}^{\mathbb{u}nv(u,p)}, n * \overline{2}^{\mathbb{u}nv(u,q)}, \mathbb{u}nv(u,p), v, s, \mathbb{u}nv(u,q))$

(l) PA $\vdash [\mathbb{F}subseq(m,n,p,v,s,q) \wedge \mathbb{V}ar(u) \wedge u = v] \rightarrow$
$\mathbb{F}subseq(m * \overline{2}^{\mathbb{u}nv(u,p)}, n * \overline{2}^{\mathbb{u}nv(u,p)}, \mathbb{u}nv(u,p), v, s, \mathbb{u}nv(u,p))$

(m) PA $\vdash \mathbb{F}subseq(m,n,p,v,s,q) \rightarrow$
$(\forall i < \mathbb{l}en(m))\mathbb{F}ormsub(\mathbb{e}xp(m,i), v, s, \mathbb{e}xp(n,i))$

corollary: PA $\vdash \mathbb{F}subseq(m,n,p,v,s,q) \rightarrow \mathbb{F}ormsub(p,v,s,q)$

*(n) PA $\vdash [\mathbb{A}tomic(p) \wedge \mathbb{T}erm(s)] \rightarrow \exists q[\mathbb{A}tomsub(p,v,s,q) \wedge$
$\mathbb{l}en(q) \leq \mathbb{l}en(p) \times \mathbb{l}en(s) \wedge (\forall k < \mathbb{l}en(q))\mathbb{e}xp(q,k) \leq p + s]$

*(o) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{T}erm(s)] \rightarrow \exists q[\mathbb{F}ormsub(p,v,s,q) \wedge$
$\mathbb{l}en(q) \leq \mathbb{l}en(p) \times \mathbb{l}en(s) \wedge (\forall k < \mathbb{l}en(q))\mathbb{e}xp(q,k) \leq p + s]$

*(p) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{T}erm(s)] \rightarrow \mathbb{F}ormsub(p,v,s,\mathit{formsub}(p,v,s))$

(q) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{T}erm(s)] \rightarrow \mathbb{W}\!f\!f(\mathit{formsub}(p,v,s))$

(a)–(m) are like results from the previous theorem. Then (n)–(q) move from the $\mathbb{F}ormsub$ relation to the $\mathit{formsub}$ function.

Finally we extend our results by means of a pair of matched theorems whose results are related to unique readability for terms and then for formulas (as from section 11.2).

*T13.51. The following are theorems of PA.

First, as a preliminary to T13.51f and then T13.52h it will be helpful to show the following. We are thinking of $c*a*c'*b*c''$ as for example, $\overline{\ulcorner(\urcorner}*a*\overline{\ulcorner\rightarrow\urcorner}*b*\overline{\ulcorner)\urcorner}$.

*(a)

a.	$\forall u[(\mathcal{P}(u) \wedge \mathbb{l}en(u) \leq x) \rightarrow (\forall k < \mathbb{l}en(u))\sim\mathcal{P}(\mathbb{v}al(u,k))]$	P
b.	$\mathcal{P}(a) \wedge \mathcal{P}(b) \wedge \mathcal{P}(d) \wedge \mathcal{P}(e)$	P
c.	$\mathbb{v}al(c * a * c' * b * c'', j) = c * d * c' * e * c''$	P
d.	$\mathbb{l}en(c * a * c' * b * c'') \leq Sx$	P
e.	$j < \mathbb{l}en(c * a * c' * b * c'')$	P
f.	$\emptyset < c \wedge \emptyset < c' \wedge \emptyset < c''$	P
g.	$\forall v(\mathcal{P}(v) \rightarrow \overline{1} < v)$	P
	$\vdots$	
h.	$\bot$	

Given boundaries from (f) and (g): If (a) a term (formula) of length less than or equal to x does not have an initial segment that is a term (formula); and (b) a, b, d, e number terms (formulas); then a concatenation whose length is less than or equal to Sx with terms (formulas) a and b cannot have an initial segment (of length j) equal to a concatenation with the terms (formulas) d and e. As a corollary, when $c' = c'' = \overline{1}$, by T13.46h,i these terms drop out of the concatenations and the theorem reduces to a version where (c) is $\mathbb{val}(c * a * b, j) = c * d * e$, and the only substantive conjunct of (f) is the first.

(b) PA $\vdash [\mathbb{T}erm(a) \wedge \mathbb{T}erm(b)] \rightarrow [\overline{\ulcorner S \urcorner} * a = \overline{\ulcorner S \urcorner} * b \rightarrow a = b]$

(c) PA $\vdash \mathbb{T}erm(\overline{\ulcorner S \urcorner} * a) \rightarrow \exists r[\overline{\ulcorner S \urcorner} * a = \overline{\ulcorner S \urcorner} * r \wedge \mathbb{T}erm(r)]$

*(d) PA $\vdash \mathbb{T}erm(\overline{\ulcorner + \urcorner} * a) \rightarrow \exists r \exists s[\overline{\ulcorner + \urcorner} * a = \overline{\ulcorner + \urcorner} * r * s \wedge \mathbb{T}erm(r) \wedge \mathbb{T}erm(s)]$

(e) PA $\vdash \mathbb{T}erm(\overline{\ulcorner \times \urcorner} * a) \rightarrow \exists r \exists s[\overline{\ulcorner \times \urcorner} * a = \overline{\ulcorner \times \urcorner} * r * s \wedge \mathbb{T}erm(r) \wedge \mathbb{T}erm(s)]$

*(f) PA $\vdash \mathbb{T}erm(t) \rightarrow (\forall k < \mathbb{len}(t)) \sim \mathbb{T}erm(\mathbb{val}(t, k))$

*(g) PA $\vdash [\mathbb{T}erm(a) \wedge \mathbb{T}erm(b) \wedge \mathbb{T}erm(c) \wedge \mathbb{T}erm(d)] \rightarrow$
$[r * a * b = r * c * d \rightarrow (a = c \wedge b = d)]$

Returning to our Chapter 11 discussion of unique readability, reasoning for (c)–(e) is like that for T11.3. Then (f) is like T11.4. (g) applies especially in the case when r is $\overline{\ulcorner + \urcorner}$ or $\overline{\ulcorner \times \urcorner}$ or $\overline{\ulcorner = \urcorner}$; then it gives a uniqueness result for $r * a * b$ like T11.5.

And now there are the parallel results for formulas. For the final result (l) let $\mathbb{P}rvpa(n) = \exists x \mathbb{P}rfpa(x, n)$.

*T13.52. The following are theorems of PA.

(a) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{W}ff(q)] \rightarrow [\mathbb{til}(p) = \mathbb{til}(q) \rightarrow p = q]$

(b) PA $\vdash [\mathbb{V}ar(u) \wedge \mathbb{V}ar(v)] \rightarrow [\mathbb{unv}(u, p) = \mathbb{unv}(v, q) \rightarrow u = v]$

(c) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{W}ff(q) \wedge \mathbb{V}ar(u) \wedge \mathbb{V}ar(v)] \rightarrow$
$[\mathbb{unv}(u, p) = \mathbb{unv}(v, q) \rightarrow p = q]$

*(d) PA $\vdash \mathbb{W}ff(\overline{\ulcorner = \urcorner} * a) \rightarrow \exists r \exists s[\overline{\ulcorner = \urcorner} * a = \overline{\ulcorner = \urcorner} * r * s \wedge \mathbb{T}erm(r) \wedge \mathbb{T}erm(s)]$

*(e) PA $\vdash \mathbb{W}ff(\overline{\ulcorner \sim \urcorner} * p) \rightarrow \exists r[\overline{\ulcorner \sim \urcorner} * p = \mathbb{til}(r) \wedge \mathbb{W}ff(r)]$

(f) PA $\vdash \mathbb{W}ff(\overline{\ulcorner (\urcorner} * p) \rightarrow \exists r \exists s[\overline{\ulcorner (\urcorner} * p = \mathbb{cnd}(r, s) \wedge \mathbb{W}ff(r) \wedge \mathbb{W}ff(s)]$

(g) PA $\vdash \mathbb{W}ff(\overline{\ulcorner \forall \urcorner} * p) \rightarrow \exists w \exists r[\overline{\ulcorner \forall \urcorner} * p = \mathbb{unv}(w, r) \wedge \mathbb{V}ar(w) \wedge \mathbb{W}ff(r)]$

(h) PA $\vdash \mathbb{W}ff(p) \rightarrow (\forall k < \mathbb{len}(p)) \sim \mathbb{W}ff(\mathbb{val}(p, k))$

*(i) PA $\vdash [\mathbb{W}ff(p)) \wedge \mathbb{W}ff(q) \wedge \mathbb{W}ff(a) \wedge \mathbb{W}ff(b)] \rightarrow$
$[\mathbb{cnd}(p, q) = \mathbb{cnd}(a, b) \rightarrow (p = a \wedge q = b)]$

*(j) PA $\vdash [\mathbb{Wff}(\mathbb{cnd}(p,q)) \wedge \mathbb{Wff}(p)] \to \mathbb{Wff}(q)$

*(k) PA $\vdash \mathbb{Axiompa}(a) \to \mathbb{Wff}(a)$

*(l) PA $\vdash \mathbb{Prvpa}(a) \to \mathbb{Wff}(a)$

Again reasoning for (d)–(g) is like that for T11.3; then (h) is like T11.4; and (i) like T11.5.

*E13.32. Show (d) and (i) from T13.40. Hard-core: show each of the results from T13.40.

Hints for T13.40. (a) is immediate from the definition of power and T13.37. (d) uses IN on the value of b. (e) uses IN on a. (f) is straightforward with cases for $m^b = \emptyset$ and $\emptyset < m^b$. (g) and (l) are by IN. For (h) and (i) under the assumption for $\to$I unabbreviate the inequality between b and a. For (m), $a < b \vee a = b \vee b < a$; but the first and last are impossible.

*E13.33. Show (c) and (d) from T13.41. Hard-core: show each of the results from T13.41.

Hints for T13.41. (a) is from the definition of fact and T13.37. (b) and (c) are straightforward by IN. Reasoning for (d) is like (G2) in the arithmetic for Gödel numbering reference once you realize that all the primes less than n are included in $fact(n)$.

*E13.34. Show (k) and (l) from T13.42. Hard-core: show each of the results from T13.42.

Hints for T13.42. (a) is from definition pi and T13.37. (b) is from T13.41d; (c) applies T13.18.e; and then (d) and (e) are by T13.18(a) and (b). (k) and (l) are simple inductions. (m) is by using IN on k to show $(\forall y \leq \mathbb{pi}(k))[Pr(y) \to \exists j\, \mathbb{pi}(j) = y]$; the result then follows easily with (l). Under the assumption for $\to$I, (n) is by IN on a. For (o) under assumptions for $\to$I and $\sim$I, you will be able to show $\mathbb{pred}(\mathbb{pi}(m)) | \mathbb{pi}(n)^a$ and use (n). For (p) under the assumption for $\to$I you will be able to show $i \leq b \to \mathbb{pred}(\mathbb{pi}(m)^i) | s$ by induction on i; the result then follows easily with $b \leq b$.

*E13.35. Show (c) and (g) from T13.43. Hard-core: show each of the results from T13.43.

Hints for T13.43. As a preliminary to (c), let PA $\vdash ex(n,i) = \mu x[\mathbb{pred}(\mathbb{pi}(i)^{Sx}) \nmid Sn]$, and show that PA defines ex; then you will be able to obtain $(\exists x \leq Sn)[\mathbb{pred}(\mathbb{pi}(i)^{Sx}) \nmid Sn]$ and apply T13.18e; for this, $\emptyset = ex(n,i) \vee \emptyset < ex(n,i)$;

in either case, $\mathit{ex}(n,i) \leq Sn$; the first case is easy; for the other, apply T13.11i and go for the goal by $\exists$E. (g) is by showing that $a = \mu x[\mathit{pred}(\mathit{pi}(i)^{Sx}) \nmid Sn]$. (m): With T13.43f it is easy to show $\exists q[\mathit{pi}(i)^{\mathit{exp}(Sn,i)} \times q = Sn]$ and so to obtain a j such that $\mathit{pi}(i)^{\mathit{exp}(Sn,i)} \times j = Sn$; then it is easy enough to obtain $\mathit{pred}(\mathit{pi}(i)) \nmid j$; the hard part is to show $\forall y(y \neq i \rightarrow \mathit{exp}(j,y) = \mathit{exp}(Sn,y))$—for this, it will be helpful to establish that j is a successor. (n): Toward an application of T13.43g it will be easy to obtain the left conjunct, $\mathit{pred}(\mathit{pi}(i)^{\mathit{exp}(Sm,i)+\mathit{exp}(Sn,i)})$ $|(Sm \times Sn)$; for the right, it will be helpful to begin with T13.43m applied to Sm, and again to Sn.

*E13.36. Show (e) and (j) from T13.44. Hard-core: show each of the results from T13.44.

Hints for T13.44. (b): with T13.43i you will be able to obtain $(\forall z \leq Sn)[Sn \leq z \rightarrow \mathit{exp}(Sn,z) = \emptyset]$ and existentially generalize on Sn. Under the assumption for $\rightarrow$I, (f) divides into cases for $m = \emptyset$ and $\emptyset < m$; for the latter, suppose $i \not< \mathit{len}(m)$; then you will be able to make use of (c). (h) is straightforward with T13.23d and ultimately (f). For (i), begin with $\mathit{len}(\mathit{pi}(i)^p) < Si \vee \mathit{len}(\mathit{pi}(i)^p) = Si \vee Si < \mathit{len}(\mathit{pi}(i)^p)$ by T13.11r; the first is easily eliminated with T13.44f; then, supposing $Si < \mathit{len}(\mathit{pi}(i)^p)$, you will be able to obtain a contradiction using T13.44d. (j): Under the assumption $\mathit{len}(n) \leq a$ for ($\forall$I), either $n = \emptyset$ or $\emptyset < n$; the first case is easy; for the second, there is some m such that $n = Sm$; your main reasoning will be to show $\mathit{exp}(Sm,a) = \emptyset$. (k): Under the assumption for $\rightarrow$I, $\emptyset = n$ or $\emptyset < n$; the first is impossible; so there is some m such that $n = Sm$; with this, suppose $\overline{1} \not\leq \mathit{exp}(Sm,l)$; then with T13.44b you will be able to reach $\mathit{len}(Sm) = l$ and contradiction from this.

*E13.37. Show the existential condition to the definition of *exc*, and then T13.45m. Hard-core: show each of the results from T13.45.

Hints for T13.45. (d) is by IN on a. (e) is by IN on i; in the show under $(\forall j < i)\mathit{exp}(\mathit{val}^*(m,n,i),j) = \mathit{exc}(m,n,j)$ and $a < Si$ you will have separate cases for $a < i$ and $a = i$. (f) is straightforward with applications of (e), (a), and (b). For (g) you may obtain $i \leq l \rightarrow \mathit{val}^*(m,n,i) \leq [\mathit{pi}(l)^{m+n}]^i$ by induction on i; in the show it will be useful to obtain $\mathit{exc}(m,n,i) \leq m+n$ and from this $\mathit{pi}(l)^{\mathit{exc}(m,n,i)} \leq \mathit{pi}(l)^{m+n}$. (k) is easy with (i). For (n) you will be able to show $\forall n[\mathit{len}(Sn) \leq x \rightarrow \mathit{val}(Sn,x) = Sn]$ by induction on x: the $\emptyset$-case is straightforward; then under the inductive assumption with $\mathit{len}(Sa) \leq Sx$ for $\rightarrow$I you have $\mathit{len}(Sa) \leq x \vee \mathit{len}(Sa) = Sx$; the first case is straightforward; the second is an extended argument—you will be able to apply T13.43m to obtain a $q > \emptyset$ and so an Sr whose prime factorization is like that of Sa but without $\mathit{pi}(x)$—show $\mathit{len}(Sr) \leq x$ so that from the assumption, $\mathit{val}(Sr,x) = Sr$, and then the result from that. For (o) under the assumption for $\rightarrow$I, you will be able to get $i \leq q \rightarrow \mathit{val}(n,i) \leq [\mathit{pi}(q)^r]^i$ by IN on i.

*E13.38. Show T13.46b and, towards a demonstration of T13.46e, show that PA $\vdash$ $l \leq \mathit{len}(m * n)$ (see hint below). Hard-core: show each of the results from T13.46.

Hints for T13.46. (b) is easy with theorems from T13.45. (e) divides into showing (i) $l \leq \mathit{len}(m * n)$ and (ii) $\mathit{len}(m * n) \leq l$; part (i) divides into cases for $\emptyset = \mathit{len}(n)$ and $\emptyset < \mathit{len}(n)$; and within the first, again, cases for $\emptyset = \mathit{len}(m)$ and $\emptyset < \mathit{len}(m)$; for (ii), assume $\mathit{len}(m * n) \not\leq l$, then $\mathit{len}(m * n)$ is some Sp and for some a, $\mathit{len}(Sp) = S(a + l)$, and using T13.43m you will be able to find a j like $m * n$ but without $\mathit{pi}(a + l)$ in its factorization to contradict an instance of T13.46d. (g): where $l' = \mathit{len}(a) + \mathit{len}(b) + \mathit{len}(c)$, you will be able to show $(\forall i < l')\mathit{exp}((a * b) * c), i) = \mathit{exp}(a * (b * c), i)$. (j) and (k) are straightforward with T13.46c. For (l) you will be able to obtain $\mathit{val}(Sm * Sn, a) = \mathit{val}(\mathit{val}(Sm, a) * \mathit{val}(Sn, a \dot{-} \mathit{len}(Sm)), a)$ by T13.45m, and from this the result you want. (m) and (n) are by induction on a.

*E13.39. Work T13.47n and then T13.48g including, in the extended $\vee$E, at least the E and F cases. Hard-core: show each of the results from T13.47 and T13.48.

Hints for T13.47. (f) is straightforward by an extended $\vee$E under assumptions for $\rightarrow$I and ($\forall$I). For (h), under the assumption for $\rightarrow$I, work systematically through its conjuncts to apply (b). Similarly (j)–(l) are disjunctive but straightforward. For (m) let $\mathcal{P} = \exists n[\mathbb{T}\mathit{ermseq}(n, \mathit{exp}(m,k)) \wedge \mathit{len}(n) \leq \mathit{len}(\mathit{exp}(m,k)) \wedge (\forall i < \mathit{len}(n))\mathit{exp}(n,i) \leq \mathit{exp}(m,k)]$; under the assumption for $\rightarrow$I, show $\forall x(\forall k < \mathit{len}(m))(\mathit{len}(\mathit{exp}(m,k)) \leq x \rightarrow \mathcal{P})$ by IN; the basis is straightforward; then, under the inductive assumption along with $a < \mathit{len}(m)$ for ($\forall$I) and $\mathit{len}(\mathit{exp}(m,a)) \leq Sx$ for $\rightarrow$I, apply (b); the derivation is then a (long) argument by cases where you will be able to apply (h)–(l). (n) follows easily with (m) and T13.45o.

Hints for T13.48. (a)–(k) work very much like the parallel theorems from T13.47. In particular, T13.48g parallels T13.47j.

*E13.40. Work T13.50m including, in the extended $\vee$E, at least the O case. Hard-core: show each of the results from T13.49 and T13.50.

Hints for T13.49. For (c) under the assumption for $\rightarrow$I go for $\forall j(j < \mathit{len}(n) \rightarrow \mathbb{T}\mathit{erm}(\mathit{exp}(n, j)))$ by T13.11ah. For (j) let $\mathcal{P}(m,n,v,s,k) = \exists a \exists b[\mathbb{T}\mathit{subseq}(a, b, \mathit{exp}(m,k), v, s, \mathit{exp}(n,k)) \wedge \mathit{len}(a) \leq \mathit{len}(\mathit{exp}(m,k)) \wedge (\forall i < \mathit{len}(a))(\mathit{exp}(a,i) \leq \mathit{exp}(m,k) \wedge \mathit{exp}(b,i) \leq \mathit{exp}(n,k))]$; then under the assumption for $\rightarrow$I, show $\forall x(\forall k < \mathit{len}(m))[\mathit{len}(\mathit{exp}(m,k) \leq x \rightarrow \mathcal{P}]$ by IN; the result follows from this. For (k) let $\mathcal{P}(m,i,v,s) = \exists x \exists y \exists u[\mathbb{T}\mathit{subseq}(x, y, \mathit{exp}(m,i), v, s, u) \wedge \mathit{len}(u) \leq \mathit{len}(\mathit{exp}(m,i)) \times \mathit{len}(s) \wedge (\forall k < \mathit{len}(u))\mathit{exp}(u,k) \leq \mathit{exp}(m,i) + s]$; under the assumption $\mathbb{Term}(t) \wedge \mathbb{Term}(s)$ given $\mathbb{Termseq}(m,t)$ you will be able to show $\forall i[i < \mathbb{len}(m) \rightarrow \mathcal{P}]$ by strong induction on i; the result follows easily from this.

Hints for T13.50. (a)–(m) and (o) work very much like the parallel theorems from T13.49. (p) follows easily with (o).

*E13.41. Work T13.52h including, in the extended $\vee$E, at least the E and F cases. Hard-core: show each of the results from T13.51 and T13.52.

Hints for T13.51. For (a) suppose $j < \mathbb{len}(c)$, this leads to contradiction so that $\mathbb{len}(c) \leq j$ and you can "pick off" the first concatenated terms from premise (c) to get $\mathbb{val}(a * c' * b * c'', j \dot{-} \mathbb{len}(c)) = d * c' * e * c''$; suppose $j \dot{-} \mathbb{len}(c) < \mathbb{len}(a)$, again this leads to contradiction so that $\mathbb{len}(a) \leq j \dot{-} \mathbb{len}(c)$ and $\mathbb{val}(a, j \dot{-} \mathbb{len}(c)) = a$; either $\mathbb{len}(d) < \mathbb{len}(a) \vee \mathbb{len}(d) = \mathbb{len}(a) \vee \mathbb{len}(a) < \mathbb{len}(d)$; the first and last lead to contradiction, and with the other you will be able to pick off the next terms; contiue to $\mathbb{val}(c'', (((j \dot{-} \mathbb{len}(c)) \dot{-} \mathbb{len}(a)) \dot{-} \mathbb{len}(c')) \dot{-} \mathbb{len}(b)) = c''$; then you will have the makings to contradict premise (e). For (f) show $\forall x \forall t[(\mathbb{Term}(t) \wedge \mathbb{len}(t) \leq x) \rightarrow (\forall k < \mathit{len}(t)){\sim}\mathbb{Term}(\mathbb{val}(t,k))]$ by IN on x; the zero case is easy; then under the inductive assumption, with $\mathbb{Term}(a) \wedge \mathbb{len}(a) \leq Sx$ for $\rightarrow$I and $j < \mathit{len}(a)$ for ($\forall$I), $\emptyset = j \vee \emptyset < j$; the first case is easy; for the second you have $(\exists x \leq B)\mathbb{Termseq}(x,a)$ and then with assumptions $\mathbb{Termseq}(m,a)$ for ($\exists$E) and $\mathbb{Term}(\mathbb{val}(a,j))$ for $\sim$I, the argument becomes an extended disjunction from $A(m, \mathbb{len}(m) \dot{-} \overline{1}) \vee B(m, \mathbb{len}(m) \dot{-} \overline{1}) \vee C(m, \mathbb{len}(m) \dot{-} \overline{1}) \vee D(m, \mathbb{len}(m) \dot{-} \overline{1})$ where you can reach contradiction in each.

Hints for T13.52. (j) is straightforward starting with (f); at some stage you will need to worry about the case $q = \emptyset$. Given T13.39 (and E13.31), (k) and (l) are not hard; for (l) you can use T13.11ah.

13.4.2 The Condition

After all our preparation, we are ready to turn to the second derivability condition, that PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$. Again, given both PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q})$ and PA $\vdash \Box\mathcal{P}$ the idea is that there are j and k such that $\textsf{PRFPA}(\mathsf{j}, \widehat{\ulcorner \mathcal{P} \rightarrow \mathcal{Q} \urcorner})$ and $\textsf{PRFPA}(\mathsf{k}, \widehat{\ulcorner \mathcal{P} \urcorner})$ so that $\mathsf{l} = \mathsf{j} \star \mathsf{k} \star \widehat{2^{\ulcorner \mathcal{Q} \urcorner}}$ numbers a proof of $\mathcal{Q}$. We show PA $\vdash$ $\mathit{Prvpa}(\mathbb{cnd}(p,q)) \rightarrow (\mathit{Prvpa}(p) \rightarrow \mathit{Prvpa}(q))$; the second condition (without free variables) follows as an immediate corollary.

Observe again that we have on the table expressions of the sort, $+$, *Plus*, and $\mathbb{plus}$—where the first is a primitive symbol of $\mathcal{L}_{\text{NT}}$, the second the original relation to capture the recursive function plus, and the last a function symbol defined through T13.37. In view of demonstrated equivalences, we will tend to slide between them without notice. In particular by the corollary to T13.34, $\mathbb{Prvpa}(n) = \exists x \mathbb{Prfpa}(x,n)$ is equivalent to $\mathit{Prvpa}(n) = \exists x \mathit{Prfpa}(x,n)$.

*T13.53. PA $\vdash$ *Prvpa*$(\mathbb{c}nd(p,q)) \rightarrow (Prvpa(p) \rightarrow Prvpa(q))$. Corollary: PA $\vdash$ $\Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$.

To manage long formulas let,

$\mathcal{Q}(m,k) = (\exists i < k)(\exists j < k)\mathbb{I}con(\mathbb{e}xp(m,i), \mathbb{e}xp(m,j), \mathbb{e}xp(m,k))$

Then T13.39j appears in the form, PA $\vdash \mathbb{P}rfpa(m,n) \leftrightarrow$

$\mathbb{e}xp(m, \mathbb{l}en(m) \dot{-} \overline{1}) = n \wedge \overline{1} < m \wedge (\forall k < \mathbb{l}en(m))(\mathbb{A}xiompa(\mathbb{e}xp(m,k)) \vee \mathcal{Q}(m,k))$

Now see the derivation on the following page.

The derivation is long and skips steps; but it should be enough for you to see how the argument works—and to fill in the details if you choose. First, from the part up to the line labeled (a), under assumptions for →I, there are derivations numbered j, k, and a longer sequence numbered l (at lines 9, 10, 11). And the last member of this longer sequence is an immediate consequence of last members from the derivations numbered j and k. At (b) the results from (16) are all applied to the sequence numbered l; so the last formula in the longer sequence is an immediate consequence of its earlier members. From lines up to (c), the different fragments of the longer sequence have the character of a proof. And at (d), the whole sequence numbered l has the character of a proof. Finally, from lines up to (e) we observe that this longer sequence yields $\mathbb{P}rvpa(q)$ and discharge the assumptions for the result that $\mathbb{P}rvpa(\mathbb{c}nd(p,q)) \rightarrow [\mathbb{P}rvpa(p) \rightarrow \mathbb{P}rvpa(q)]$ so that PA $\vdash Prvpa(\mathbb{c}nd(p,q)) \rightarrow (Prvpa(p) \rightarrow Prvpa(q))$.

But now we have PA $\vdash Prvpa(\mathbb{c}nd(\overline{\ulcorner\mathcal{P}\urcorner}, \overline{\ulcorner\mathcal{Q}\urcorner})) \rightarrow [Prvpa(\overline{\ulcorner\mathcal{P}\urcorner}) \rightarrow Prvpa(\overline{\ulcorner\mathcal{Q}\urcorner})]$ as an instance, and by capture, PA $\vdash Prvpa(\overline{\ulcorner\mathcal{P} \rightarrow \mathcal{Q}\urcorner}) \rightarrow [Prvpa(\overline{\ulcorner\mathcal{P}\urcorner}) \rightarrow Prvpa(\overline{\ulcorner\mathcal{Q}\urcorner})]$ so that PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$. Thus the second derivability condition is established.

*E13.42. As a start to a complete demonstration of T13.53, provide a demonstration through part (c) that does not skip any steps. You may find it helpful to divide your demonstration into separate parts for (a), (b), and then for lines (22) and (24). Hard-core: complete the entire derivation.

Hint: As a preliminary to (24) it will be helpful to show PA proves $(\forall i < s)[\mathcal{P}(t+i) \vee (\exists m < i)(\exists n < i)\mathcal{Q}(t+m, t+n, t+i)] \rightarrow (\forall i : t \leq i < t+s)[\mathcal{P}(i) \vee (\exists m < i)(\exists n < i)\mathcal{Q}(m,n,i)]$. Where l is a fixed parameter, let $\mathcal{P}(t+i)$ be $\mathbb{A}xiompa(\mathbb{e}xp(l, t+i))$ and $\mathcal{Q}(t+m, t+n, t+i)$ be $\mathbb{I}con(\mathbb{e}xp(l, t+m), \mathbb{e}xp(l, t+n), \mathbb{e}xp(l, t+i))$. Then (23) is of the sort to which the preliminary theorem applies.

T13.53

	1.	$\mathbb{P}rvpa(\mathbb{c}nd(p,q))$	A $(g, \to\text{I})$
	2.	$\mathbb{W}f\!f(\mathbb{c}nd(p,q))$	1 T13.52l
	3.	$\mathbb{P}rvpa(p)$	A $(g, \to\text{I})$
	4.	$\mathbb{W}f\!f(p)$	3 T13.52l
	5.	$\mathbb{W}f\!f(q)$	2,4 T13.52j
	6.	$\mathbb{I}con(\mathbb{c}nd(p,q), p, q)$	T13.39e,g
	7.	$\exists v\, \mathbb{P}rfpa(v, \mathbb{c}nd(p,q))$	1 def $\mathbb{P}rvpa$
	8.	$\exists v\, \mathbb{P}rfpa(v, p)$	3 def $\mathbb{P}rvpa$
	9.	$\mathbb{P}rfpa(j, \mathbb{c}nd(p,q))$	A $(g, 7\exists\text{E})$
	10.	$\mathbb{P}rfpa(k, p)$	A $(g, 8\exists\text{E})$
	11.	$l = j * k * \overline{2}^q$	def
	12.	$\emptyset < \mathbb{l}en(j) \wedge \emptyset < \mathbb{l}en(k)$	from 9,10
	13.	$\mathbb{e}xp(j, \mathbb{l}en(j) \dot{-} \overline{1}) = \mathbb{c}nd(p,q)$	9 T13.39j
	14.	$\mathbb{e}xp(k, \mathbb{l}en(k) \dot{-} \overline{1}) = p$	10 T13.39j
	15.	$\mathbb{e}xp(l, \mathbb{l}en(j) + \mathbb{l}en(k)) = q$	11 T13.46e,f
a	16.	$\mathbb{I}con[\mathbb{e}xp(j, \mathbb{l}en(j) \dot{-} \overline{1}), \mathbb{e}xp(k, \mathbb{l}en(k) \dot{-} \overline{1}), \mathbb{e}xp(l, \mathbb{l}en(j) + \mathbb{l}en(k))]$	6,13,14,15 =E
	17.	$(\forall i < \mathbb{l}en(j))[\mathbb{e}xp(l,i) = \mathbb{e}xp(j,i)]$	11 T13.46c
	18.	$(\forall i < \mathbb{l}en(k))[\mathbb{e}xp(l, \mathbb{l}en(j) + i) = \mathbb{e}xp(k,i)]$	11 T13.46c
	19.	$\mathbb{e}xp(l, \mathbb{l}en(j) \dot{-} \overline{1}) = \mathbb{e}xp(j, \mathbb{l}en(j) \dot{-} \overline{1})$	17,12 T13.21g
	20.	$\mathbb{e}xp(l, \mathbb{l}en(j) + \mathbb{l}en(k) \dot{-} \overline{1}) = \mathbb{e}xp(k, \mathbb{l}en(k) \dot{-} \overline{1})$	18,12 T13.21g
b	21.	$Icon[\mathbb{e}xp(l, \mathbb{l}en(j) \dot{-} \overline{1}), \mathbb{e}xp(l, \mathbb{l}en(j) + \mathbb{l}en(k) \dot{-} \overline{1}),$ $\mathbb{e}xp(l, \mathbb{l}en(j) + \mathbb{l}en(k))]$	16,19,20 =E
	22.	$(\forall i < \mathbb{l}en(j))[\mathbb{A}xiompa(\mathbb{e}xp(l,i)) \vee$ $(\exists m < i)(\exists n < i)Icon(\mathbb{e}xp(l,m), \mathbb{e}xp(l,n), \mathbb{e}xp(l,i))]$	9,17 T13.39j
	23.	$(\forall i < \mathbb{l}en(k))[\mathbb{A}xiompa(\mathbb{e}xp(l, \mathbb{l}en(j) + i)) \vee$ $(\exists m < i)(\exists n < i)\mathbb{I}con(\mathbb{e}xp(l, \mathbb{l}en(j) + m), \mathbb{e}xp(l, \mathbb{l}en(j) + n),$ $\mathbb{e}xp(l, \mathbb{l}en(j) + i))]$	10,18 T13.39j
c	24.	$(\forall i : \mathbb{l}en(j) \le i < \mathbb{l}en(j) + \mathbb{l}en(k))[\mathbb{A}xiompa(\mathbb{e}xp(l,i)) \vee$ $(\exists m < i)(\exists n < i)\mathbb{I}con(\mathbb{e}xp(l,m), \mathbb{e}xp(l,n), \mathbb{e}xp(l,i))]$	from 23
	25.	$x < \mathbb{l}en(l)$	A $(g, (\forall\text{I}))$
	26.	$x < \mathbb{l}en(j) \vee \mathbb{l}en(j) \le x < \mathbb{l}en(j) + \mathbb{l}en(k) \vee x = \mathbb{l}en(j) + \mathbb{l}en(k)$	from 11,25
	27.	$x < \mathbb{l}en(j)$	A $(g, 26\vee\text{E})$
	28.	$\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)$	22,27 $(\forall\text{E})$
	29.	$\mathbb{l}en(j) \le x < \mathbb{l}en(j) + \mathbb{l}en(k)$	A $(g, 26\vee\text{E})$
	30.	$\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)$	24,29 $(\forall\text{E})$
	31.	$x = \mathbb{l}en(j) + \mathbb{l}en(k)$	A $(g, 26\vee\text{E})$
	32.	$\mathbb{Q}(l,x)$	from 21,31
	33.	$\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)$	32 $\vee$I
	34.	$\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)$	26,27-33 $\vee$E
d	35.	$(\forall x < \mathbb{l}en(l))[\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)]$	25-34 $(\forall\text{I})$
	36.	$\overline{1} < l$	with 11
	37.	$\mathbb{e}xp(l, \mathbb{l}en(l) \dot{-} \overline{1}) = q$	15 T13.46e
	38.	$\mathbb{e}xp(l, \mathbb{l}en(l) \dot{-} \overline{1}) = q \wedge \overline{1} < l \wedge$ $(\forall x < \mathbb{l}en(l))[\mathbb{A}xiompa(\mathbb{e}xp(l,x)) \vee \mathbb{Q}(l,x)]$	37,36,35 $\wedge$I
	39.	$\mathbb{P}rfpa(l,q)$	38 T13.39j
	40.	$\mathbb{P}rvpa(q)$	39 $\exists$I
	41.	$\mathbb{P}rvpa(q)$	8,10-40 $\exists$E
	42.	$\mathbb{P}rvpa(q)$	7,9-41 $\exists$E
	43.	$\mathbb{P}rvpa(p) \to \mathbb{P}rvpa(q)$	3-42 $\to$I
e	44.	$\mathbb{P}rvpa(\mathbb{c}nd(p,q)) \to [\mathbb{P}rvpa(p) \to \mathbb{P}rvpa(q)]$	1-43 $\to$I

Second Theorems of Chapter 13

T13.19. For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathcal{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathbb{r}(\vec{x})$ such that PA $\vdash v = \mathbb{r}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$. This theorem depends on conditions for the recursion clause and so on T13.20 and T13.29.

T13.20. Where $\mathcal{F}(\vec{x}, y, v)$ is the formula for recursion, PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \rightarrow m = n]$.

T13.21–T13.24. T13.21 Results for $a \dot{-} b$. T13.22 results for $a|b$. T13.23 results for $Pr(a)$ and $Rp(a)$. T13.24 results for $lcm(a)$.

T13.25. PA $\vdash [(\forall i < k)h(i) \leq m(i) \wedge \forall i \forall j((i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j)))] \rightarrow \exists p(\forall i < k)rm(p, m(i)) = h(i)$.

T13.26–T13.28. T13.26 results for *maxp* and *maxs*. T13.27 PA $\vdash \exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$. T13.28 PA $\vdash \exists p \exists q[(\forall i < k)\beta(p, q, i) = \beta(a, b, i) \wedge \beta(p, q, k) = n]$.

T13.29. PA $\vdash \exists v \exists p \exists q[\beta(p, q, \emptyset) = g(\vec{x}) \wedge (\forall i < y)h(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, y) = v]$.

T13.30. Suppose $\mathbb{f}(\vec{x}, y)$ is defined by $\mathbb{g}(\vec{x})$ and $\mathbb{h}(\vec{x}, y, u)$ so that PA $\vdash v = \mathbb{f}(\vec{x}, y) \leftrightarrow \mathcal{F}(\vec{x}, y, v)$; then, (a) PA $\vdash \mathbb{f}(\vec{x}, \emptyset) = \mathbb{g}(\vec{x})$ and (b) PA $\vdash \mathbb{f}(\vec{x}, Sy) = \mathbb{h}(\vec{x}, y, \mathbb{f}(\vec{x}, y))$.

T13.31. For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$, PA defines a coordinate function $\mathbb{r}(\vec{x})$.

T13.32–T13.33. T13.32 is some sample applications of T13.31. T13.33 extends applications by equivalences for $\mathbb{suc}$, $\mathbb{idnt}^j_k$, $\mathbb{zero}$, $\bar{\bar{\mathsf{n}}}$, $\mathbb{pred}$, $\mathbb{plus}$, $\mathbb{times}$, $\mathbb{subc}$, and $\mathbb{absval}$.

T13.34–T13.36. T13.34 for any recursive R defined in Chapter 12, PA defines $\mathbb{R}$ such that PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$. T13.35 Results for $\mathbb{sg}$, $\mathbb{csg}$, and $\mathbb{ch}_{\mathsf{R}}$. T13.36 Equivalences for $\mathbb{Eq}$, $\mathbb{Leq}$, $\mathbb{Less}$, $\mathbb{Neg}$, $\mathbb{Dsj}$, $(\exists y \leq z)$, and $(\mu y \leq z)$.

T13.37. For any friendly recursive function $\mathsf{f}(\mathsf{x})$ PA defines a coordinate function $\mathbb{f}(\vec{x})$. And for any recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ as defined in Chapter 12, PA defines a coordinate relation $\mathbb{R}(\vec{x})$.

T13.38–T13.39. T13.38 extension of T13.37 by equivalences for $\mathbb{Imp}$, $\mathbb{Cnj}$, $(\exists y < z)$, $(\forall y \leq z)$, $(\forall y < z)$, $\mathbb{Fctr}$ and $\mathbb{Prime}$. T13.39 is some sample applications leading to $\mathbb{Prfpa}$.

T13.40–T13.43. T13.40 results for m^a. T13.41 results for $\mathbb{fact}$. T13.42 results for $\mathbb{pi}$. T13.43 results for $\mathbb{exp}$.

T13.44–T13.46. T13.44 results for $\mathbb{len}$. T13.45 results for $\mathbb{val}$. T13.46 results for $m * n$.

T13.47–T13.50 T13.47 results for $\mathbb{Termseq}$. T13.48 results for $\mathbb{Formseq}$. T13.49 results for $\mathbb{Tsubseq}$. T13.50 results for $\mathbb{Fsubseq}$.

T13.51–T13.52. T13.51 on unique readability for terms. T13.52 on unique readability for formulas.

T13.53. PA $\vdash Prvpa(\mathbb{cnd}(p, q)) \rightarrow (Prvpa(p) \rightarrow Prvpa(q))$.

corollary (D2): PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$.

13.5 The Third Condition: $\Box\mathscr{P} \to \Box\Box\mathscr{P}$

To show the third condition, that PA $\vdash \Box\mathscr{P} \to \Box\Box\mathscr{P}$, it is sufficient to show PA $\vdash$ $\mathscr{Q} \to \Box\mathscr{Q}$. For when $\mathscr{Q}$ is $\Box\mathscr{P}$, the result is immediate. Further, $\Box\mathscr{P}$ is $Prvpa(\overline{\ulcorner\mathscr{P}\urcorner})$ and $Prvpa(\overline{\ulcorner\mathscr{P}\urcorner})$ is a Σ_1 sentence. So it is sufficient to show that for any Σ_1 sentence $\mathscr{Q}$, PA $\vdash \mathscr{Q} \to \Box\mathscr{Q}$. That is what we do. Of course we have already seen from D1 that if PA $\vdash \mathscr{Q}$ then PA $\vdash \Box\mathscr{Q}$. So we need to push the conditional from the metalanguage into the theory. We begin with some additional applications, especially with respect to $\mathscr{f}ormsub$ (section 13.5.1). Then we focus what needs to be shown by an alternate characterization of Σ_1 formulas (13.5.2). Then some results that apply *Prvpa* to special forms substituting numerals into places for free variables (13.5.3). Finally we will be in a position to show the third condition (13.5.4).

13.5.1 More Applications

Recall that where $\mathsf{p} = \ulcorner\mathscr{P}\urcorner$, $\mathsf{v} = \ulcorner v\urcorner$, and $\mathsf{s} = \ulcorner s\urcorner$, $\mathsf{formsub(p, v, s)}$ returns the Gödel number of $\mathscr{P}^{v}_{s}$. Let $\mathsf{gvar(n)} = \hat{2}^{\hat{2}\hat{3}+\hat{2}\mathsf{n}}$ be the Gödel number of variable x_n. In addition, as from page 624, $\mathsf{num(n)}$ returns the Gödel number of the standard numeral for n. So $\mathsf{formsub(p, gvar(n), num(y))}$ numbers the formula replacing free instances of x_n by a numeral for the value assigned to y. So, for example, if y is assigned the value of $\mathsf{2}$, then $\mathsf{formsub(p, gvar(n), num(y))}$ returns $\ulcorner\mathscr{P}^{x_n}_{\overline{2}}\urcorner$. And, of course, PA defines coordinate $\mathscr{f}ormsub(p, \mathscr{g}var(n), \mathscr{n}um(y))$. We require some results for these notions.

First, a pair of theorems with some results for substitutions into terms and then into formulas. As on pages 705–706, I–N and O–S are subformulas of $\mathbb{T}subseq$ and $\mathbb{F}subseq$ respectively.

*T13.54. The following are theorems of PA.

(a) PA $\vdash \mathbb{F}ree_t(t, v) \leftrightarrow \sim\mathbb{T}ermsub(t, v, v \times \overline{4}, t)$

(b) PA $\vdash \mathscr{e}xp(m, k) = \overline{\ulcorner\emptyset\urcorner} \to$
$\sim[J(v, m, n, k) \vee K(v, s, m, n, k) \vee L(m, n, k) \vee M(m, n, k) \vee N(m, n, k)]$

*(c) PA $\vdash [\mathbb{V}ar(\mathscr{e}xp(m, k)) \wedge \mathscr{e}xp(m, k) \neq v] \to$
$\sim[I(m, n, k) \vee K(v, s, m, n, k) \vee L(m, n, k) \vee M(m, n, k) \vee N(m, n, k)]$

(d) PA $\vdash [\mathbb{V}ar(\mathscr{e}xp(m, k)) \wedge \mathscr{e}xp(m, k) = v] \to$
$\sim[I(m, n, k) \vee J(v, m, n, k) \vee L(m, n, k) \vee M(m, n, k) \vee N(m, n, k)]$

(e) PA $\vdash \mathscr{e}xp(m, k) = \overline{\ulcorner S\urcorner} * a \to$
$\sim[I(m, n, k) \vee J(v, m, n, k) \vee K(v, s, m, n, k) \vee M(m, n, k) \vee N(m, n, k)]$

(f) PA $\vdash \mathscr{e}xp(m, k) = \overline{\ulcorner +\urcorner} * a \to$
$\sim[I(m, n, k) \vee J(v, m, n, k) \vee K(v, s, m, n, k) \vee L(m, n, k) \vee N(m, n, k)]$

(g) PA $\vdash$ $\mathbb{exp}(m,k) = \overline{\ulcorner \times \urcorner} * a \rightarrow$
$\sim[I(m,n,k) \vee J(v,m,n,k) \vee K(v,s,m,n,k) \vee L(m,n,k) \vee M(m,n,k)]$

*(h) PA $\vdash [\mathbb{T}ermsub(t,v,s,q) \wedge \mathbb{T}ermsub(t,v,s,r)] \rightarrow q = r$

(i) PA $\vdash [\mathbb{V}ar(w) \wedge \sim\mathbb{F}ree_t(w,v)] \rightarrow v \neq w$

(j) PA $\vdash [\mathbb{V}ar(w) \wedge \sim\mathbb{F}ree_t(\overline{\ulcorner S \urcorner} * w, v)] \rightarrow v \neq w$

*(k) PA $\vdash [\mathbb{T}erm(t) \wedge \mathbb{T}erm(s) \wedge \mathbb{V}ar(v)] \rightarrow [\sim\mathbb{F}ree_t(t,v) \rightarrow \mathbb{T}ermsub(t,v,s,t)]$

*(l) PA $\vdash [\mathbb{T}erm(t) \wedge \mathbb{V}ar(v)] \rightarrow [(\mathbb{F}ree_t(t,v) \wedge \mathbb{T}ermsub(t,v,s,u)) \rightarrow s \leq u]$

Given that $\mathbb{T}ermsub$ depends on $\mathbb{T}subseq$, and given the disjunctive nature of $\mathbb{T}subseq$, reasoning as in (h) with both $\mathbb{T}ermsub(t,v,s,q)$ and $\mathbb{T}ermsub(t,v,s,r)$ results in an extended $\vee$E inside each subderivation of an extended $\vee$E. Theorems like (b)–(g) let us "pick off" disjuncts in a reasonable way. And similarly for (c)–(g) in the theorem that follows.

*T13.55. The following are theorems of PA.

(a) PA $\vdash \mathbb{F}ree_f(p,v) \leftrightarrow \sim\mathbb{F}ormsub(p,v,v \times \overline{4}, p)$

(b) PA $\vdash [\mathbb{V}ar(w) \wedge \sim\mathbb{F}ree_f(p,v)] \rightarrow \sim\mathbb{F}ree_f(\mathbb{unv}(w,p),v)$

(c) PA $\vdash \mathbb{A}tomic(\mathbb{exp}(m,k)) \rightarrow$
$\sim[P(m,n,k) \vee Q(m,n,k) \vee R(v,p,m,n,k) \vee S(v,p,m,n,k)]$

(d) PA $\vdash \mathbb{exp}(m,k) = \overline{\ulcorner \sim \urcorner} * a \rightarrow$
$\sim[O(v,s,m,n,k) \vee Q(m,n,k) \vee R(v,p,m,n,k) \vee S(v,p,m,n,k)]$

*(e) PA $\vdash \mathbb{exp}(m,k) = \overline{\ulcorner (\urcorner} * a \rightarrow$
$\sim[O(v,s,m,n,k) \vee P(m,n,k) \vee R(v,p,m,n,k) \vee S(v,p,m,n,k)]$

(f) PA $\vdash [\mathbb{V}ar(j) \wedge j \neq v \wedge \mathbb{exp}(m,k) = \overline{\ulcorner \forall \urcorner} * j * a] \rightarrow$
$\sim[O(v,s,m,n,k) \vee P(m,n,k) \vee Q(m,n,k) \vee S(v,p,m,n,k)]$

(g) PA $\vdash [\mathbb{V}ar(j) \wedge j = v \wedge \mathbb{exp}(m,k) = \overline{\ulcorner \forall \urcorner} * j * a] \rightarrow$
$\sim[O(v,s,m,n,k) \vee P(m,n,k) \vee Q(m,n,k) \vee R(v,p,m,n,k)]$

(h) PA $\vdash [\mathbb{A}tomsub(p,v,s,q) \wedge \mathbb{A}tomsub(p,v,s,r)] \rightarrow q = r$

*(i) PA $\vdash [\mathbb{F}ormsub(p,v,s,q) \wedge \mathbb{F}ormsub(p,v,s,r)] \rightarrow q = r$

(j) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{T}erm(s)] \rightarrow$
$[\mathbb{F}ormsub(p,v,s,q) \rightarrow \mathbb{f}ormsub(p,v,s) = q]$

(k) PA $\vdash [\mathbb{T}erm(s) \wedge \mathbb{V}ar(v)] \rightarrow$
$[\mathbb{A}tomsub(p,v,v \times \overline{4}, p) \rightarrow \mathbb{A}tomsub(p,v,s,p)]$

(l) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{T}erm(s) \wedge \mathbb{V}ar(v)] \rightarrow$
$[\sim\mathbb{F}ree_f(p, v) \rightarrow \mathbb{f}ormsub(p, v, s) = p]$

corollary: if variable x is not free in formula $\mathcal{P}$ and PA $\vdash \mathbb{T}erm(s)$, then PA $\vdash \mathbb{f}ormsub(\overline{\ulcorner\mathcal{P}\urcorner}, \overline{\ulcorner x\urcorner}, s) = \overline{\ulcorner\mathcal{P}\urcorner}$

*(m) PA $\vdash \mathbb{V}ar(v) \rightarrow$
$[(\sim\mathbb{A}tomsub(p, v, v \times \overline{4}, p) \wedge \mathbb{A}tomsub(p, v, s, q)) \rightarrow s \leq q]$

*(n) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{T}erm(s) \wedge \mathbb{V}ar(v)] \rightarrow [\mathbb{F}ree_f(p, v) \rightarrow s \leq \mathbb{f}ormsub(p, v, s)]$

The corollary to (l) is immediate by capture.

Now some results for $\mathbb{F}fseq$ and $\mathbb{F}reefor$. For a function like FFSEQ(m, s, v, u) let,

$$
\begin{array}{lcl}
T(m,k) & = & \mathbb{A}tomic(\mathbb{e}xp(m,k)) \\
U(m,k) & = & (\exists j < k)[\mathbb{e}xp(m,k) = \mathbb{t}il(\mathbb{e}xp(m,j))] \\
V(m,k) & = & (\exists i < k)(\exists j < k)[\mathbb{e}xp(m,k) = \mathbb{c}nd(\mathbb{e}xp(m,i), \mathbb{e}xp(m,j))] \\
W(u,v,m,k) & = & (\exists p \leq u)(\exists j \leq u)[\mathbb{V}ar(j) \wedge j = v \wedge \mathbb{W}\!f\!f(p) \wedge \mathbb{e}xp(m,k) = \mathbb{u}nv(j,p)] \\
X(u,v,s,m,k) & = & (\exists i < k)(\exists j \leq u)[\mathbb{V}ar(j) \wedge j \neq v \wedge \\
 & & \quad (\sim\mathbb{F}ree_t(s,j) \vee \sim\mathbb{F}ree_f(\mathbb{e}xp(m,i),v)) \wedge \\
 & & \quad \mathbb{e}xp(m,k) = \mathbb{u}nv(j, \mathbb{e}xp(m,i))]
\end{array}
$$

*T13.56. The following are theorems of PA.

(a) PA $\vdash \mathbb{F}fseq(m, s, v, u) \leftrightarrow \{\mathbb{e}xp(m, \mathbb{l}en(m) \dot{-} \overline{1}) = u \wedge \overline{1} < m \wedge$
$(\forall k < \mathbb{l}en(m))[T(m,k) \vee U(m,k) \vee V(m,k) \vee W(u,v,m,k) \vee$
$X(u,v,s,m,k)]\}$

(b) (i) PA $\vdash \mathbb{F}reefor(s, v, u) \leftrightarrow (\exists x \leq B_u)\mathbb{F}fseq(x, s, v, u)$
(ii) PA $\vdash B_u = [\mathbb{p}i(\mathbb{l}en(u))^u]^{\mathbb{l}en(u)}$

(c) PA $\vdash \mathbb{F}fseq(m, s, v, p) \rightarrow (\forall k < \mathbb{l}en(m))(\overline{1} < \mathbb{e}xp(m,k))$

*(d) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{V}ar(v)] \rightarrow \mathbb{F}reefor(v, v, p)$

(e) PA $\vdash \mathbb{A}tomic(p) \rightarrow \mathbb{F}fseq(\overline{2}^p, s, v, p)$

(f) PA $\vdash \mathbb{F}fseq(m, s, v, p) \rightarrow \mathbb{F}fseq(m * \overline{2}^{\mathbb{t}il(p)}, s, v, \mathbb{t}il(p))$

(g) PA $\vdash [\mathbb{F}fseq(m, s, v, p) \wedge \mathbb{F}fseq(n, s, v, q)] \rightarrow$
$\mathbb{F}fseq(m * n * \overline{2}^{\mathbb{c}nd(p,q)}, s, v, \mathbb{c}nd(p,q))$

(h) PA $\vdash [\mathbb{W}\!f\!f(p) \wedge \mathbb{V}ar(v)] \rightarrow \mathbb{F}fseq(\overline{2}^{\mathbb{u}nv(v,p)}, s, v, \mathbb{u}nv(v, p))$

(i) PA $\vdash [\mathbb{F}fseq(m, s, v, p) \wedge \mathbb{V}ar(w) \wedge w \neq v \wedge (\sim\mathbb{F}ree_t(s, w) \vee \sim\mathbb{F}ree_f(p, v))]$
$\rightarrow \mathbb{F}fseq(m * \overline{2}^{\mathbb{u}nv(w,p)}, s, v, \mathbb{u}nv(w, p))$

(j) PA $\vdash \mathbb{e}xp(m,k) = \overline{\ulcorner\sim\urcorner} * a \rightarrow$
$\sim[T(m,k) \vee V(m,k) \vee W(u,v,m,k) \vee X(u,v,s,m,k)]$

(k) PA $\vdash \mathbb{e}xp(m,k) = \overline{\ulcorner (\urcorner} * a \rightarrow$
$\sim[T(m,k) \vee U(m,k) \vee W(u,v,m,k) \vee X(u,v,s,m,k)]$

(l) PA $\vdash [\mathbb{V}ar(w) \wedge w \neq v \wedge \mathbb{e}xp(m,k) = \overline{\ulcorner \forall \urcorner} * w * a] \rightarrow$
$\sim[T(m,k) \vee U(m,k) \vee V(m,k) \vee W(p,v,m,k)]$

(m) PA $\vdash \mathbb{F}fseq(m,s,v,p) \rightarrow (\forall i < \mathbb{l}en(m))\mathbb{W}ff(\mathbb{e}xp(m,i))$
corollary: PA $\vdash \mathbb{F}fseq(m,s,v,p) \rightarrow \mathbb{W}ff(p)$

*(n) PA $\vdash \mathbb{F}fseq(m,s,v,p) \rightarrow (\forall i < \mathbb{l}en(m))\mathbb{F}reefor(s,v,\mathbb{e}xp(m,i))$
corollary: PA $\vdash \mathbb{F}fseq(m,s,v,p) \rightarrow \mathbb{F}reefor(s,v,p)$

*(o) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{F}reefor(s,v,\mathbb{t}il(p)] \rightarrow \mathbb{F}reefor(s,v,p)$

(p) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{W}ff(q) \wedge \mathbb{F}reefor(s,v,\mathbb{c}nd(p,q))] \rightarrow$
$[\mathbb{F}reefor(s,v,p) \wedge \mathbb{F}reefor(s,v,q)]$

(q) PA $\vdash [\mathbb{V}ar(u) \wedge u \neq v \wedge \mathbb{W}ff(p) \wedge \mathbb{F}reefor(s,v,\mathbb{u}nv(u,p))] \rightarrow$
$[\mathbb{F}reefor(s,v,p) \wedge (\sim\mathbb{F}ree_t(s,u) \vee \sim\mathbb{F}ree_f(p,v))]$

(a)–(b) are from the definitions. Reasoning for others works like that for results we have seen before.

We are now positioned for a series of results related to numerals, to the rule Gen, and to axiom A4. For reasoning about numerals let $\mathbb{n}umseq(n)$ be as follows:

PA $\vdash \mathbb{n}umseq(\emptyset) = \mathbb{p}i(\emptyset)^{\mathbb{n}um(\emptyset)}$
PA $\vdash \mathbb{n}umseq(Sy) = \mathbb{n}umseq(y) \times \mathbb{p}i(Sy)^{\mathbb{n}um(Sy)}$

The exponents of $\mathbb{n}umseq(n)$ are the Gödel numbers of $\overline{\mathsf{0}} \ldots \overline{\mathsf{n}}$. We shall be able to show that $\mathbb{n}umseq(n)$ numbers a term sequence for $\mathbb{n}um(n)$ and so that $\mathbb{T}erm(\mathbb{n}um(n))$.

*T13.57. The following are theorems of PA.

(a) (i) PA $\vdash \mathbb{n}um(\emptyset) = \overline{\ulcorner \emptyset \urcorner}$
(ii) PA $\vdash \mathbb{n}um(Sy) = \overline{\ulcorner S \urcorner} * \mathbb{n}um(y)$

(b) PA $\vdash \mathbb{g}var(n) = \overline{\mathsf{2}}^{\overline{\mathsf{23}}+\overline{\mathsf{2}}\times n}$

(c) PA $\vdash \mathbb{V}ar(\mathbb{g}var(n))$

(d) PA $\vdash \mathbb{g}var(m) = \mathbb{g}var(n) \rightarrow m = n$

*(e) PA $\vdash [\mathbb{P}rvpa(p) \wedge \mathbb{V}ar(v)] \rightarrow \mathbb{P}rvpa(\mathbb{u}nv(v,p))$

(f) PA $\vdash \mathbb{A}xiompa(n) \rightarrow \mathbb{P}rvpa(n)$

*(g) PA $\vdash \mathbb{A}xiomad4(n) \leftrightarrow \exists s(\exists p \leq n)(\exists v \leq n)[\mathbb{W}ff(p) \wedge \mathbb{V}ar(v) \wedge$
$\mathbb{T}erm(s) \wedge \mathbb{F}reefor(s,v,p) \wedge n = \mathbb{c}nd(\mathbb{u}nv(v,p), \mathbb{f}ormsub(p,v,s))]$

(h) PA $\vdash \emptyset < \mathbb{n}um(x)$

(i) PA $\vdash \overline{1} < \mathbb{numseq}(x)$

(j) PA $\vdash \mathbb{len}(\mathbb{num}(x)) = Sx$

*(k) PA $\vdash \mathbb{len}(\mathbb{numseq}(x)) = Sx$

*(l) PA $\vdash (\forall y \leq x)\mathbb{exp}(\mathbb{numseq}(x), y) = \mathbb{num}(y)$

*(m) PA $\vdash \mathbb{Var}(v) \to v \neq \mathbb{num}(y)$

*(n) PA $\vdash \mathbb{Termseq}(\mathbb{numseq}(x), \mathbb{num}(x))$
corollary: PA $\vdash \mathbb{Term}(\mathbb{num}(x))$

*(o) PA $\vdash \mathbb{Tsubseq}(\mathbb{numseq}(n), \mathbb{numseq}(n), \mathbb{num}(n), v, s, \mathbb{num}(n))$
corollary: PA $\vdash \mathbb{Termsub}(\mathbb{num}(n), v, s, \mathbb{num}(n))$
corollary: PA $\vdash \sim\mathbb{Free}_t(\mathbb{num}(n), v))$

*(p) PA $\vdash [\mathbb{Wff}(p) \wedge \mathbb{Var}(v)] \to \mathbb{Freefor}(\mathbb{num}(x), v, p)$

(q) PA $\vdash \mathbb{Wff}(p) \to$
$\mathbb{Prvpa}(\mathbb{cnd}(\mathbb{unv}(\mathbb{gvar}(n), p), \mathbb{formsub}(p, \mathbb{gvar}(n), \mathbb{num}(x))))$

(a) and (b) are from the definitions. Effectively, (e) is like Gen. (g) is like the intuitive version of A4 from page 621; it follows from the original version resulting from T13.39h (and E13.31). Then (q) results with (g) when the substituted term is a numeral (so that associated restrictions are automatically met).

Finally, a theorem with results first for distribution of substitution over a conditional, and then for substitution into other substitutions. Each of the latter include matched results for $\mathbb{Termsub}$, $\mathbb{Atomsub}$, and then $\mathbb{Formsub}$.

*T13.58. The following are theorems of PA.

(a) PA $\vdash [\mathbb{Wff}(p) \wedge \mathbb{Wff}(q) \wedge \mathbb{Term}(s)] \to$
$\mathbb{formsub}(\mathbb{cnd}(p, q), v, s) = \mathbb{cnd}(\mathbb{formsub}(p, v, s), \mathbb{formsub}(q, v, s))$

*(b) PA $\vdash [\mathbb{Term}(p) \wedge \mathbb{Term}(t)] \to$
$\exists q[\mathbb{Termsub}(p, v, \mathbb{num}(y), q) \wedge \mathbb{Termsub}(q, v, t, q)]$

*(c) PA $\vdash [\mathbb{Atomic}(p) \wedge \mathbb{Term}(t)] \to$
$\exists q[\mathbb{Atomsub}(p, v, \mathbb{num}(y), q) \wedge \mathbb{Atomsub}(q, v, t, q)]$

*(d) PA $\vdash [\mathbb{Wff}(p) \wedge \mathbb{Term}(t)] \to$
$\mathbb{formsub}(p, v, \mathbb{num}(y)) = \mathbb{formsub}(\mathbb{formsub}(p, v, \mathbb{num}(y)), v, t)$

*(e) PA $\vdash [\mathbb{Term}(p) \wedge v \neq w] \to \exists q \exists t \exists t'[$
$\mathbb{Termsub}(p, v, \mathbb{num}(y), t) \wedge \mathbb{Termsub}(p, w, \mathbb{num}(z), t') \wedge$
$\mathbb{Termsub}(t, w, \mathbb{num}(z), q) \wedge \mathbb{Termsub}(t', v, \mathbb{num}(y), q)]$

*(f) PA $\vdash [\mathbb{A}tomic(p) \wedge v \neq w] \rightarrow \exists q \exists t \exists t'[$
$\mathbb{A}tomsub(p, v, \mathbb{num}(y), t) \wedge \mathbb{A}tomsub(p, w, \mathbb{num}(z), t') \wedge$
$\mathbb{A}tomsub(t, w, \mathbb{num}(z), q) \wedge \mathbb{A}tomsub(t', v, \mathbb{num}(y), q)]$

*(g) PA $\vdash [\mathbb{W}ff(p) \wedge v \neq w] \rightarrow$
$formsub(formsub(p, v, \mathbb{num}(y)), w, \mathbb{num}(z)) =$
$formsub(formsub(p, w, \mathbb{num}(z)), v, \mathbb{num}(y))$

(h) PA $\vdash [\mathbb{T}erm(p) \wedge \mathbb{V}ar(w)] \rightarrow \exists q \exists t \exists t'[$
$\mathbb{T}ermsub(p, v, w, t) \wedge \mathbb{T}ermsub(p, v, \mathbb{num}(y), t') \wedge$
$\mathbb{T}ermsub(t, w, \mathbb{num}(y), q) \wedge \mathbb{T}ermsub(t', w, \mathbb{num}(y), q)]$

(i) PA $\vdash [\mathbb{A}tomic(p) \wedge \mathbb{V}ar(w)] \rightarrow \exists q \exists t \exists t'[$
$\mathbb{A}tomsub(p, v, w, t) \wedge \mathbb{A}tomsub(p, v, \mathbb{num}(y), t') \wedge$
$\mathbb{A}tomsub(t, w, \mathbb{num}(y), q) \wedge \mathbb{A}tomsub(t', w, \mathbb{num}(y), q)]$

(j) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{V}ar(v) \wedge \mathbb{V}ar(w) \wedge \mathbb{F}reefor(w, v, p)] \rightarrow$
$formsub(formsub(p, v, w), w, \mathbb{num}(y)) =$
$formsub(formsub(p, v, \mathbb{num}(y)), w, \mathbb{num}(y))$

*(k) PA $\vdash [\mathbb{T}erm(p) \wedge \mathbb{V}ar(w)] \rightarrow \exists q \exists t \exists t'[$
$\mathbb{T}ermsub(p, v, \overline{\ulcorner S \urcorner} * w, t) \wedge \mathbb{T}ermsub(p, v, \mathbb{num}(Sy), t') \wedge$
$\mathbb{T}ermsub(t, w, \mathbb{num}(y), q) \wedge \mathbb{T}ermsub(t', w, \mathbb{num}(y), q)]$

*(l) PA $\vdash [\mathbb{A}tomic(p) \wedge \mathbb{V}ar(w)] \rightarrow \exists q \exists t \exists t'[$
$\mathbb{A}tomsub(p, v, \overline{\ulcorner S \urcorner} * w, t) \wedge \mathbb{A}tomsub(p, v, \mathbb{num}(Sy), t') \wedge$
$\mathbb{A}tomsub(t, w, \mathbb{num}(y), q) \wedge \mathbb{A}tomsub(t', w, \mathbb{num}(y), q)]$

*(m) PA $\vdash [\mathbb{W}ff(p) \wedge \mathbb{V}ar(v) \wedge \mathbb{V}ar(w) \wedge \mathbb{F}reefor(\overline{\ulcorner S \urcorner} * w, v, p)] \rightarrow$
$formsub(formsub(p, v, \overline{\ulcorner S \urcorner} * w), w, \mathbb{num}(y)) =$
$formsub(formsub(p, v, \mathbb{num}(Sy)), w, \mathbb{num}(y))$

Speaking loosely: From (a), $(\mathcal{P} \rightarrow \mathcal{Q})^v_s = \mathcal{P}^v_s \rightarrow \mathcal{Q}^v_s$. From theorems leading up to (d), $\mathcal{P}^v_{\mathbb{num}(y)} = (\mathcal{P}^v_{\mathbb{num}(y)})^v_t$. From theorems leading up to (g), if $v \neq w$ then $(\mathcal{P}^v_{\mathbb{num}(y)})^w_{\mathbb{num}(z)} = (\mathcal{P}^w_{\mathbb{num}(z)})^v_{\mathbb{num}(y)}$. From ones leading to (j), if w is free for v in $\mathcal{P}$, then $(\mathcal{P}^v_w)^w_{\mathbb{num}(y)} = (\mathcal{P}^v_{\mathbb{num}(y)})^w_{\mathbb{num}(y)}$. And from ones leading to (m), if Sw is free for v in $\mathcal{P}$, then $(\mathcal{P}^v_{Sw})^w_{\mathbb{num}(y)} = (\mathcal{P}^v_{\mathbb{num}(Sy)})^w_{\mathbb{num}(y)}$. For these results it is important that $\mathbb{num}(y)$ is a numeral and so has no variables to be replaced.

*E13.43. Set up the argument for T13.55l including assertion of the main proposition to be shown by induction; then set up the show part working just the P case. Hard-core: finish each of the results in T13.54 and T13.55.

Hints for T13.54. (h): Under assumptions for $\rightarrow$I and ($\exists$E) you have both $\mathbb{T}subseq(m, n, t, v, s, q)$ and $\mathbb{T}subseq(m', n', t, v, s, r)$; with this show $\forall k[k < \mathbb{len}(m) \rightarrow (\forall x < \mathbb{len}(m'))(\mathbb{exp}(m, k) = \mathbb{exp}(m', x) \rightarrow \mathbb{exp}(n, k) = \mathbb{exp}(n', x))]$

by strong induction; the result follows easily from this. (k): Under assumptions for $\rightarrow$I, with T13.54a and T13.49k you can obtain and exploit existentials to get both $\mathbb{T}subseq(m, n, t, v, v \times \overline{4}, t)$ and $\mathbb{T}subseq(m', n', t, v, s, u)$ with goal $t = u$; by strong induction show $\forall k[k < len(m) \rightarrow (\forall x < len(m'))(exp(m, k) = exp(m', x) \rightarrow (exp(m, k) = exp(n, k) \rightarrow exp(m', x) = exp(n', x)))]$; then the result follows easily. (l): Under assumptions for $\rightarrow$I you can obtain and exploit existentials to get $\mathbb{T}subseq(m, n, t, v, v\times\overline{4}, r)$ and $\mathbb{T}subseq(m', n', t, v, s, u)$ where $r \neq t$ with goal $s \leq u$; by strong induction show $\forall k[k < len(m) \rightarrow (\forall x < len(m'))(exp(m, k) = exp(m', x) \rightarrow (exp(m, k) \neq exp(n, k) \rightarrow s \leq exp(n', x)))]$; the result follows.

Hints for T13.55. See the corresponding members of T13.54.

E13.44. Taking as given that PA $\vdash$ $\mathbb{T}ermsub(t, v, v, t)$, show that PA $\vdash$ $(\mathbb{S}ent(p) \wedge \mathbb{V}ar(v)) \rightarrow \sim\mathbb{F}ree_f(p, v)$. Hint: Under the assumption for $\rightarrow$I you will be able to obtain $formsub(p, v, v) = p$; then the result is easy from $v \leq p \vee p < v$.

*E13.45. Show T13.56m. Hard-core: show the rest of the results from T13.56.

Hints for T13.56. For (d) you will be able to apply T13.48j and T13.45o. (m): Under the assumption for $\rightarrow$I, you can show $\forall x(\forall k < len(m))[len(exp(m, k)) \leq x \rightarrow \exists n\mathbb{F}ormseq(n, exp(m, k))]$ by IN; the result follows easily. (n): Let $\mathcal{P} = \exists n[\mathbb{F}fseq(n, s, v, exp(m, k)) \wedge len(n) \leq len(exp(m, k)) \wedge (\forall i < len(n))exp(n, i) \leq exp(m, k)]$; under the assumption for $\rightarrow$I, show $\forall x(\forall k < len(m))[len(exp(m, k)) \leq x \rightarrow \mathcal{P}]$ by IN; the result follows from this.

*E13.46. Show (q) from T13.57. Hard-core: show the rest of the results from T13.57.

Hints for T13.57. For (e) under the assumption for $\rightarrow$I and then $\mathbb{P}rfpa(m, p)$ for $\exists$E, apply T13.39j to obtain $\mathbb{P}rfpa(m * \overline{2}^{unv(v,p)}, unv(v, p))$. (k) is by IN, where you will be able to show both $SSx \leq len(numseq(Sx))$ and $len(numseq(Sx)) \leq SSx$. For (m) under assumptions for $\rightarrow$I and, with T13.47a, ($\exists$E) the argument is by IN on the value of y; for the show it may help to think about the length of v. For (n) apply T13.47b; and similarly for (o) T13.49a. For (p), you will be able to use T13.48j to set up an application of T13.56a.

*E13.47. Show T13.58a. Hard-core: finish the rest of the results in T13.58.

Hints for T13.58. (b): Under the assumption for $\rightarrow$I, obtain and exploit an existential to get $\mathbb{T}ermseq(m, p)$; then you can show $\forall x(\forall k < len(m))[len(exp(m, k) \leq x \rightarrow \exists q(\mathbb{T}ermsub(exp(m, k), v, num(y), q) \wedge \mathbb{T}ermsub(q, v, t, q))]$ by IN. (c): Under the assumption for $\rightarrow$I, apply T13.48a and with (b) obtain and exploit existentials to get $\mathbb{T}ermsub(a, v, num(y), q) \wedge \mathbb{T}ermsub(q, v, t, q)$ and $\mathbb{T}ermsub(b, v, num(y), r) \wedge \mathbb{T}ermsub(r, v, t, r)$. (d): Under the assumption for $\rightarrow$I, obtain and exploit an existential to get $\mathbb{F}ormseq(m, p)$; then you will be able to show $\forall x(\forall k < \mathbb{l}en(m))[\mathbb{l}en(\mathbb{e}xp(m,k) \leq x \rightarrow \exists q(\mathbb{F}ormsub(\mathbb{e}xp(m,k), v, \mathbb{n}um(y), q) \wedge \mathbb{F}ormsub(q, v, t, q))]$ by IN. (e): Under the assumption for $\rightarrow$I and with $\mathbb{T}ermseq(m, p)$, let $\mathcal{P} = \exists q \exists t \exists t'[\mathbb{T}ermsub(\mathbb{e}xp(m,k), v, \mathbb{n}um(y), t) \wedge \mathbb{T}ermsub(\mathbb{e}xp(m,k), w, \mathbb{n}um(z), t') \wedge \mathbb{T}ermsub(t, w, \mathbb{n}um(z), q) \wedge \mathbb{T}ermsub(t', v, \mathbb{n}um(y), q)]$; go for $\forall x(\forall k < \mathbb{l}en(m))(\mathbb{l}en(\mathbb{e}xp(m,k)) \leq x \rightarrow \mathcal{P})$ by IN. (j): Under the assumption for $\rightarrow$I and with $\mathbb{F}ormseq(m, p)$, let $\mathcal{P} = \mathbb{F}reefor(w, v, \mathbb{e}xp(m,k)) \rightarrow \exists q \exists t \exists t'[\mathbb{F}ormsub(\mathbb{e}xp(m,k), v, w, t) \wedge \mathbb{F}ormusb(\mathbb{e}xp(m,k), v, \mathbb{n}um(y), t') \wedge \mathbb{F}ormsub(t, w, \mathbb{n}um(y), q) \wedge \mathbb{F}ormsub(t', w, \mathbb{n}um(y), q)]$; show $\forall x(\forall k < \mathbb{l}en(m))(\mathbb{l}en(\mathbb{e}xp(m,k)) \leq x \rightarrow \mathcal{P})$.

13.5.2 Sigma Star

Our aim is to show PA $\vdash \mathcal{Q} \rightarrow \Box\mathcal{Q}$ for any Σ_1 sentence $\mathcal{Q}$. The task is simplified by a "minimal" specification of the Σ_1 formulas themselves. Toward this end, we introduce a special class of formulas, the $\Sigma_\star$ formulas; and show that every Σ_1 formula is provably equivalent to a $\Sigma_\star$ formula. $\Sigma_\star$ formulas are as follows:

($\Sigma_\star$) For any distinct variables x, y, and z,

(a) $\emptyset = z$, $y = z$, $Sy = z$, $x + y = z$, and $x \times y = z$ are $\Sigma_\star$.

(b) If $\mathcal{P}$ is $\Sigma_\star$, then so is $\exists x\mathcal{P}$.

(c) If $\mathcal{P}$ and $\mathcal{Q}$ are $\Sigma_\star$, then so are $(\mathcal{P} \vee \mathcal{Q})$ and $(\mathcal{P} \wedge \mathcal{Q})$.

(d) If $\mathcal{P}$ is $\Sigma_\star$, then so is $(\forall x \leq y)\mathcal{P}$ where y does not occur in $\mathcal{P}$.

(CL) Any $\Sigma_\star$ formula may be formed by repeated application of these rules.

Notice the new restriction on a bounded universal, where the bound is a variable y that does not occur in $\mathcal{P}$. Given that the specification of $\Sigma_\star$ formulas is a restriction of that for Σ_1 formulas, it is obvious that every $\Sigma_\star$ formula is Σ_1. We aim to show the other direction: that every Σ_1 formula is provably equivalent to a $\Sigma_\star$ formula. Then results which apply to all the $\Sigma_\star$ formulas immediately transfer to the Σ_1 formulas. We begin showing that there are $\Sigma_\star$ formulas equivalent to atomic equalities of the sort $t = x$. Then (depending on an extended notion of *normal* form and a result according to which Δ_0 formulas always have equivalent normal forms) we show that there are $\Sigma_\star$ formulas equivalent to all the Δ_0 formulas. And from this there are $\Sigma_\star$ formulas equivalent to all the Σ_1 formulas.

First, then, the result for atomic equalities.

T13.59. For any atomic $\mathcal{P}$ of the form $t = x$ where x does not appear in t, there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Suppose $\mathcal{P}$ is of the form $t = x$ where x does not appear in t. By induction on the function symbols in t, *Basis*: If t has no function symbols, then it is the constant $\emptyset$ or a variable y other than x; so $\mathcal{P}$ is of the form $\emptyset = x$ or $y = x$; but these are already $\Sigma_\star$ formulas. So let $\mathcal{P}_{\Sigma_\star}$ be the same as $\mathcal{P}$. Then PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, there is a $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Show: If t has k function symbols, there is a $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

If t has k function symbols, then it is of the form Sr, $r + s$, or $r \times s$ for r and s with $< k$ function symbols.

(S) t is Sr, so that $\mathcal{P}$ is $Sr = x$. For some new variable z, set $\mathcal{P}_{\Sigma_\star} = \exists z[(r = z)_{\Sigma_\star} \wedge Sz = x]$. Then $\mathcal{P}_{\Sigma_\star}$ is $\Sigma_\star$. By assumption, PA $\vdash r = z \leftrightarrow (r = z)_{\Sigma_\star}$. So reason as follows:

1.	$r = z \leftrightarrow (r = z)_{\Sigma_\star}$	by assp
2.	$\quad Sr = x$	A (g, $\leftrightarrow$I)
3.	$\quad r = r \wedge Sr = x$	from 2
4.	$\quad \exists z[r = z \wedge Sz = x]$	3 $\exists$I
5.	$\quad \exists z[(r = z)_{\Sigma_\star} \wedge Sz = x]$	1,4 with T9.9
6.	$\quad \exists z[(r = z)_{\Sigma_\star} \wedge Sz = x]$	A (g, $\leftrightarrow$I)
7.	$\quad\quad (r = z)_{\Sigma_\star} \wedge Sz = x$	A (g, 6$\exists$E)
8.	$\quad\quad r = z$	1,7 $\leftrightarrow$E
9.	$\quad\quad Sr = x$	from 7,8
10.	$\quad Sr = x$	6,7-9 $\exists$E
11.	$Sr = x \leftrightarrow \exists z[(r = z)_{\Sigma_\star} \wedge Sz = x]$	2-5,6-10 $\leftrightarrow$I

Since z does not appear in r and is not x, the restriction on $\exists$E is met. So PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

(+) $t = s + r$, so that $\mathcal{P}$ is $s + r = x$. For some new variables u and v, set $\mathcal{P}_{\Sigma_\star} = \exists u \exists v[((s = u)_{\Sigma_\star} \wedge (r = v)_{\Sigma_\star}) \wedge u + v = x]$. Then $\mathcal{P}_{\Sigma_\star}$ is $\Sigma_\star$; and PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

($\times$) Similarly.

Indct: For any $\mathcal{P}$ of the form $t = x$, there is a $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Now generalize some operations from T8.1. There we said a formula is in *normal* form iff its only operators are $\vee$, $\wedge$, and $\sim$, and the only instances of $\sim$ are immediately prefixed to atomics. Now a formula is in (*extended*) *normal* form iff its only operators are $\vee$, $\wedge$, $\sim$, or a bounded quantifier, and the only instances of $\sim$ are immediately prefixed to atomics (which may include inequalities). Again, generalizing from before, where $\mathcal{P}$ is a normal form, let $\mathcal{P}'$ be like $\mathcal{P}$ except that $\vee$ and $\wedge$, universal and existential quantifiers and, for an atomic $\mathcal{A}$, $\mathcal{A}$ and $\sim\mathcal{A}$ are interchanged. So, for example, $(\exists x \leq p)(x = p \vee p \not< x)' = (\forall x \leq p)(x \neq p \wedge p < x)$. Still generalizing, for any Δ_0 formula whose operators are $\sim$, $\rightarrow$, and the bounded quantifiers, for atomic $\mathscr{A}$, let $\mathscr{A}_N = \mathscr{A}$; and $[\sim\mathscr{P}]_N = [\mathscr{P}_N]'$; $(\mathscr{P} \rightarrow \mathscr{Q})_N = ([\mathscr{P}_N]' \vee \mathscr{Q}_N)$; $[(\exists x \leq t)\mathscr{P}]_N = (\exists x \leq t)\mathscr{P}_N$ and $[(\forall x \leq t)\mathscr{P}]_N = (\forall x \leq t)\mathscr{P}_N$ and similarly for $(\exists x < t)\mathscr{P}$ and $(\forall x < t)\mathscr{P}$. Then as a simple extension to the result from E8.10,

T13.60. For any Δ_0 formula $\mathscr{P}_{\Delta_0}$, there is a normal formula $\mathscr{P}_N$ such that $\vdash \mathscr{P}_{\Delta_0} \leftrightarrow \mathscr{P}_N$.

The demonstration is a straightforward extension of the reasoning from E8.9 and E8.10.

So each Δ_0 formula is a provably equivalent to a normal form.

Now we show that each Δ_0 formula is a provably equivalent to a $\Sigma_\star$ formula. With the previous theorem, it is sufficient to show that normal forms are provably equivalent to $\Sigma_\star$ formulas. Thus,

*T13.61. For any Δ_0 formula $\mathscr{P}_{\Delta_0}$ there is a $\Sigma_\star$ formula $\mathscr{P}_{\Sigma_\star}$ such that PA $\vdash \mathscr{P}_{\Delta_0} \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

From T13.60, for any Δ_0 formula $\mathscr{P}_{\Delta_0}$, there is a normal $\mathscr{P}_N$ such that $\vdash \mathscr{P}_{\Delta_0} \leftrightarrow \mathscr{P}_N$. By induction on the number of operators in $\mathscr{P}_N$, we show there is a $\mathscr{P}_{\Sigma_\star}$ such that PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$. The result is immediate.

Basis: If $\mathscr{P}_N$ has no operators, then it is an atomic of the sort $s = t$, $s \leq t$, or $s < t$.

($=$) $\mathscr{P}_N$ is $s = t$. For some new variable z, set $\mathscr{P}_{\Sigma_\star} = \exists z[(s = z)_{\Sigma_\star} \wedge (t = z)_{\Sigma_\star}]$. By T13.59, PA $\vdash s = z \leftrightarrow (s = z)_{\Sigma_\star}$ and PA $\vdash t = z \leftrightarrow (t = z)_{\Sigma_\star}$; so PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

($\leq$) $\mathscr{P}_N$ is $s \leq t$, which is to say $\exists z(z + s = t)$ for z not in s or t. By the case immediately above, PA $\vdash (z + s = t) \leftrightarrow (z + s = t)_{\Sigma_\star}$. Set $\mathscr{P}_{\Sigma_\star} = \exists z(z + s = t)_{\Sigma_\star}$. Then PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$. And similarly for $<$.

Assp: For any i, $0 \leq i < k$, if a normal $\mathscr{P}_N$ has i operator symbols, then there is a $\Sigma_\star$ formula $\mathscr{P}_{\Sigma_\star}$ such that PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

Show: If a normal $\mathscr{P}_N$ has k operator symbols, then there is a $\Sigma_\star$ formula $\mathscr{P}_{\Sigma_\star}$ such that PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

If $\mathscr{P}_N$ has k operator symbols, then it is of the form $\sim\mathscr{A}$, $\mathscr{B} \wedge \mathscr{C}$, $\mathscr{B} \vee \mathscr{C}$, $(\exists x \leq t)\mathscr{B}$, $(\exists x < t)\mathscr{B}$, $(\forall x \leq t)\mathscr{B}$, or $(\forall x < t)\mathscr{B}$, where $\mathscr{A}$ is atomic and $\mathscr{B}$ and $\mathscr{C}$ are normal with $< k$ operator symbols.

($\sim$) $\mathscr{P}_N$ is $\sim\mathscr{A}$ for atomic $\mathscr{A}$. There are three cases:

(i) $\mathscr{P}_N$ is $s \neq t$. Set $\mathscr{P}_{\Sigma_\star} = (s < t)_{\Sigma_\star} \vee (t < s)_{\Sigma_\star}$; then by assumption, PA $\vdash s < t \leftrightarrow (s < t)_{\Sigma_\star}$ and PA $\vdash t < s \leftrightarrow (t < s)_{\Sigma_\star}$; and with T13.11r,t, PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

(ii) $\mathscr{P}_N$ is $s \not< t$; set $\mathscr{P}_{\Sigma_\star} = (t \leq s)_{\Sigma_\star}$; then by assumption, PA $\vdash t \leq s \leftrightarrow (t \leq s)_{\Sigma_\star}$; and with T13.11u, PA $\vdash \mathscr{P}_N \leftrightarrow \mathscr{P}_{\Sigma_\star}$.

(iii) And similarly for $\mathscr{P}_{\Sigma_\star} = s \not\leq t$.

($\wedge$) $\mathcal{P}_{\mathrm{N}}$ is $\mathcal{B} \wedge \mathcal{C}$. Set $\mathcal{P}_{\Sigma_\star} = \mathcal{B}_{\Sigma_\star} \wedge \mathcal{C}_{\Sigma_\star}$; since $\mathcal{B}$ and $\mathcal{C}$ are normal, by assumption PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}_{\Sigma_\star}$ and PA $\vdash \mathcal{C} \leftrightarrow \mathcal{C}_{\Sigma_\star}$; so PA $\vdash \mathcal{P}_{\mathrm{N}} \leftrightarrow \mathcal{P}_{\Sigma_\star}$. And similarly for $\vee$.

($\forall$) $\mathcal{P}_{\mathrm{N}}$ is $(\forall x \leq t)\mathcal{B}$. For some new variable z set $\mathcal{P}_{\Sigma_\star} = \exists z[(t = z)_{\Sigma_\star} \wedge (\forall x \leq z)\mathcal{B}_{\Sigma_\star}]$; by assumption PA $\vdash t = z \leftrightarrow (t = z)_{\Sigma_\star}$ and PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}_{\Sigma_\star}$ so PA $\vdash \mathcal{P}_{\mathrm{N}} \leftrightarrow \mathcal{P}_{\Sigma_\star}$. And, by a related construction, similarly for $(\forall x < t)\mathcal{B}$.

($\exists$) $\mathcal{P}_{\mathrm{N}}$ is $(\exists x \leq t)\mathcal{B}$. Set $\mathcal{P}_{\Sigma_\star} = \exists x[(x \leq t)_{\Sigma_\star} \wedge \mathcal{B}_{\Sigma_\star}]$; then by assumption PA $\vdash x \leq t \leftrightarrow (x \leq t)_{\Sigma_\star}$ and PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}_{\Sigma_\star}$; so PA $\vdash \mathcal{P}_{\mathrm{N}} \leftrightarrow \mathcal{P}_{\Sigma_\star}$. And similarly for $(\exists x < t)\mathcal{B}$.

Indct: For any normal $\mathcal{P}_{\mathrm{N}}$ there is a $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P}_{\mathrm{N}} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

From T13.60 for any Δ_0 formula $\mathcal{P}_{\Delta_0}$, there is a normal $\mathcal{P}_{\mathrm{N}}$ such that $\vdash \mathcal{P}_{\Delta_0} \leftrightarrow \mathcal{P}_{\mathrm{N}}$ and now PA $\vdash \mathcal{P}_{\mathrm{N}} \leftrightarrow \mathcal{P}_{\Sigma_\star}$. So PA $\vdash \mathcal{P}_{\Delta_0} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Finally, with this, you can show that for any Σ_1 formula $\mathcal{P}_{\Sigma_1}$ there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P}_{\Sigma_1} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

*T13.62. For any Σ_1 formula $\mathcal{P}_{\Sigma_1}$ there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P}_{\Sigma_1} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

Treating Δ_0 formulas as atomic, the argument is by induction on the number of operators in $\mathcal{P}_{\Sigma_1}$. Homework.

So every Σ_1 formula is provably equivalent to a $\Sigma_\star$ formula. So a result for all $\Sigma_\star$ formulas transfers to all the Σ_1 formulas. And that is what we set out to show in this section.

*E13.48. By the method of T13.59, T13.60, and T13.61 find a $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash (\forall x \leq y)[z < Sy \rightarrow x = z] \leftrightarrow \mathcal{P}_{\Sigma_\star}$; then show that PA proves the biconditional in at least one direction. Hard-core: show the biconditional in both directions. Hints: Begin converting $(\forall x \leq y)[z < Sy \rightarrow x = z]$ to normal form; then by the methods of T13.61 eliminate negated atomics and inequalities. Consider a tree for the resultant expression; working through the tree, for each term t generate a $\Sigma_\star$ formula equivalent to $t = z$ by the methods of T13.59; and use T13.61 again for the formulas.

E13.49. Povide a demonstration to show T13.60.

E13.50. Fill in the parts of T13.59 and T13.61 that are left as "similarly" to show that PA $\vdash \mathcal{P}_{\Sigma_1} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

*E13.51. Demonstrate T13.62.

13.5.3 Substitutions

The demonstration that for any $\Sigma_\star$ (and so Σ_1) sentence $\mathcal{Q}$, PA $\vdash \mathcal{Q} \rightarrow \Box\mathcal{Q}$ is by induction on the number of operators in $\mathcal{Q}$—and so moves from the parts of $\mathcal{Q}$ to the whole. As it turns out, we shall find it easier to work with whole *sentences* than than with open subformulas. With this in mind, we introduce a $\mathbb{sub}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{t}\,)$ which substitutes numerals for variables free in $\mathcal{P}$. The substitutions result in (numbers of) sentences about which we shall be able to obtain results. And when $\mathcal{P}$ itself is a sentence, there are no variables to replace, so that our results apply directly to the original $\mathcal{P}$.

For this, let $\vec{y}$ be a (possibly empty) sequence of n variables, and $\vec{t} = t_1 \dots t_\mathsf{n}$ a sequence of n terms. Consider an enumeration $\mathsf{enum}(\vec{y}, \mathsf{i})$ of variable subscripts in $\vec{y}$ so that $\mathsf{enum}(\vec{y}, \mathsf{i}) = \mathsf{y_i}$ is the subscript of the i^{th} variable and $\overline{\mathsf{y}}_\mathsf{i}$ the numeral corresponding to that subscript; so if $\vec{y}$ is $x_3x_6x_2$, $\mathsf{enum}(\vec{y}, 1) = \mathsf{y}_1 = 3$, $\mathsf{enum}(\vec{y}, 2) = \mathsf{y}_2 = 6$, and $\mathsf{enum}(\vec{y}, 3) = \mathsf{y}_3 = 2$; and generally the variables of $\vec{y}$ are $x_{\mathsf{y}_1} \dots x_{\mathsf{y}_\mathsf{n}}$, the variables of $\vec{z}$ are $x_{\mathsf{z}_1} \dots x_{\mathsf{z}_\mathsf{n}}$, and so forth. Then for $\mathsf{i} < \mathsf{n}$,

$$\text{PA} \vdash \mathbb{sub}^{0}_{\vec{y}}(p, \vec{t}\,) = p$$

$$\text{PA} \vdash \mathbb{sub}^{\mathsf{Si}}_{\vec{y}}(p, \vec{t}\,) = \mathbb{formsub}(\mathbb{sub}^{\mathsf{i}}_{\vec{y}}(p, \vec{t}\,), \mathbb{gvar}(\overline{\mathsf{y}}_{\mathsf{Si}}), \mathbb{num}(t_{\mathsf{Si}}))$$

And PA $\vdash \mathbb{sub}_{\vec{y}}(p, \vec{t}\,) = \mathbb{sub}^{\mathsf{n}}_{\vec{y}}(p, \vec{t}\,)$.

So each $\mathbb{sub}^{\mathsf{i}}_{\vec{y}}$ substitutes a numeral for the value assigned to t_i into the place of variable $x_{\mathsf{y_i}}$, and $\mathbb{sub}$ replaces them all. In the ordinary case, p is the number of a formula $\mathcal{P}$, and $\vec{y}$ includes all the variables free in $\mathcal{P}$. So $\mathbb{sub}$ substitutes numerals for values assigned to $\vec{t}$ for all the variables $\vec{y}$ that are free in the formula numbered p. For a one-variable case, $\mathsf{enum}(y_\mathsf{a}, 1)$ is just a and $\mathbb{sub}_{y_\mathsf{a}}(\overline{\ulcorner\mathcal{P}(y_\mathsf{a})\urcorner}, t) = \mathbb{sub}^{1}_{y_a}(\overline{\ulcorner\mathcal{P}(y_\mathsf{a})\urcorner}, t) = \mathbb{formsub}(\overline{\ulcorner\mathcal{P}(y_\mathsf{a})\urcorner}, \mathbb{gvar}(\overline{\mathsf{a}}), \mathbb{num}(t))$. Observe that if t includes free variables, $\mathbb{sub}_{y_\mathsf{a}}(\overline{\ulcorner\mathcal{P}(y_\mathsf{a})\urcorner}, t)$ has those variables free, but returns the number of the *sentence* that substitutes a numeral for the value assigned to t into the y_a-place of $\mathcal{P}$. Also we have not defined a function by recursion, but rather recursively specified a sequence of functions. Thus the super- and subscript notations do not indicate variables of $\mathbb{sub}^{\mathsf{i}}_{\vec{y}}(p, \vec{t}\,)$ and a correlate to enum does not appear in the $\mathcal{L}_{\text{NT}}$ expression; rather we use enum to make the specification in which there appears a certain numeral (in this case $\overline{\mathsf{a}}$).

For $\mathbb{sub}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{t}\,)$, we shall often be concerned with the case where $\vec{t}$ just is $\vec{y}$. For this we have,

T13.63. For any i and formula $\mathcal{P}$, PA $\vdash \mathbb{Wff}(\mathbb{sub}^{\mathsf{i}}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}))$.

By an easy induction with T13.50q.

So PA $\vdash \mathbb{Wff}(\mathbb{sub}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}))$. And from a few quick theorems (collected in the substitution vectors box on page 729), so long as $\vec{x}$ and $\vec{y}$ each include all the variables free in $\mathcal{P}$, PA $\vdash \mathbb{sub}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = \mathbb{sub}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$. Given this, we shall not usually worry about details of the vectors.

Now, introducing double brackets as a special notation: Where $\vec{x}$ includes all the variables free in $\mathcal{P}$,

$$Prvpa[\![\mathcal{P}(\vec{x})]\!] = Prvpa(\mathbb{sub}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$$

Suppose the free variables of $\mathcal{P}$ just are the members of $\vec{x}$. Then $Prvpa(\overline{\ulcorner\mathcal{P}\urcorner})$ asserts the provability of the open formula $\mathcal{P}(\vec{x})$. But $Prvpa[\![\mathcal{P}]\!]$ itself has all the free variables of $\mathcal{P}$ and asserts the provability of whatever *sentences* have numerals for the variables free in $\mathcal{P}$. Thus by $\forall$E it follows from $\forall x Prvpa[\![\mathcal{P}(x)]\!]$ that $Prvpa[\![\mathcal{P}(x)]\!]^x_{\emptyset}$; but this is $Prvpa(\mathbb{sub}_x(\overline{\ulcorner\mathcal{P}(x)\urcorner}, x))^x_{\emptyset}$, which is $Prvpa(\mathbb{sub}_x(\overline{\ulcorner\mathcal{P}(x)\urcorner}, \emptyset))$; which, as we shall see, is equivalent to $Prvpa(\overline{\ulcorner\mathcal{P}^x_{\emptyset}\urcorner})$. And, more generally, from $\forall x Prvpa[\![\mathcal{P}(x)]\!]$ follow each of the sentences $Prvpa(\overline{\ulcorner\mathcal{P}^x_{\emptyset}\urcorner})$, $Prvpa(\overline{\ulcorner\mathcal{P}^x_{S\emptyset}\urcorner})$, and so forth. When $\mathcal{P}$ is itself a sentence, there are no substitutions to be made and $Prvpa[\![\mathcal{P}]\!]$ is the same as $Prvpa(\overline{\ulcorner\mathcal{P}\urcorner})$. Thus we set out to show PA $\vdash \mathcal{P} \to Prvpa[\![\mathcal{P}]\!]$ for $\Sigma_\star$ formulas. When $\mathcal{P}$ is a sentence, this gives PA $\vdash \mathcal{P} \to Prvpa(\overline{\ulcorner\mathcal{P}\urcorner})$, which is to be shown.

In order to do this we shall require some quick theorems in order to manipulate this new notion. There are analogs to D1 and D2, and results for substitution. Each is by a short induction. Again, given their equivalence, we apply results for $\mathbb{Prvpa}$ directly to *Prvpa*. First, the results like D1 and D2:

T13.71. If PA $\vdash \mathcal{P}$, then PA $\vdash Prvpa[\![\mathcal{P}]\!]$.

Suppose PA $\vdash \mathcal{P}$ and $\vec{x}$ includes all the variables free in $\mathcal{P}$. By induction on the value of n, we show PA $\vdash Prvpa(\mathbb{sub}^{\mathsf{n}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$. The result is immediate. We revert to (III) from the Chapter 8 induction schemes reference.

Basis: $\mathbb{sub}^{0}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = \overline{\ulcorner\mathcal{P}\urcorner}$. Since PA $\vdash \mathcal{P}$, by D1, PA $\vdash Prvpa(\overline{\ulcorner\mathcal{P}\urcorner})$; so PA $\vdash Prvpa(\mathbb{sub}^{0}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

Assp: PA $\vdash Prvpa(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

Show: PA $\vdash Prvpa(\mathbb{sub}^{S\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

1.	$Prvpa(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$	by assp
2.	$\mathbb{Wff}(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$	T13.63
3.	$\mathbb{Var}(\mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}))$	T13.57c
4.	$Prvpa[\mathbb{unv}(\mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))]$	1,3 T13.57e
5.	$Prvpa[\mathbb{cnd}(\mathbb{unv}(\mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})),$ $\quad formsub(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{num}(x_{\mathsf{x}_{S\mathsf{i}}})))]$	2 T13.57q
6.	$Prvpa[\mathbb{unv}(\mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))] \to$ $\quad Prvpa[formsub(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{num}(x_{\mathsf{x}_{S\mathsf{i}}}))]$	5 T13.53
7.	$Prvpa[formsub(\mathbb{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{gvar}(\mathsf{x}_{S\mathsf{i}}), \mathbb{num}(x_{\mathsf{x}_{S\mathsf{i}}}))]$	6,4 $\to$E
8.	$Prvpa(\mathbb{sub}^{S\mathsf{i}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$	7 def $\mathbb{sub}$

Indct: For any n, PA $\vdash Prvpa(\mathbb{sub}^{\mathsf{n}}_{\vec{x}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

Substitution Vectors

T13.64. Let $\vec{x}$ be some $x_{\mathsf{x}_1} \ldots x_{\mathsf{x}_\mathsf{a}}$, and for some $\vec{u}$ and $\vec{v}$ let $\vec{y} = \vec{x}, \vec{u}$, and $\vec{z} = \vec{x}, \vec{v}$; then for any $\mathsf{i} \leq \mathsf{a}$, PA $\vdash \mathcal{sub}^{\mathsf{i}}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}) = \mathcal{sub}^{\mathsf{i}}_{\vec{z}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{z})$. By an easy induction.

*T13.65. Let $\vec{y}_0 = x_\mathsf{a}, x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_\mathsf{u}}, x_{\mathsf{y}_{\mathsf{u}+1}}, \ldots x_{\mathsf{y}_\mathsf{n}}$ and in general $\vec{y}_\mathsf{u} = x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_\mathsf{u}}, x_\mathsf{a}, x_{\mathsf{y}_{\mathsf{u}+1}} \ldots x_{\mathsf{y}_\mathsf{n}}$. Then for any formula $\mathcal{P}$ and $\mathsf{u} \leq \mathsf{n}$, PA $\vdash \mathcal{sub}^{\mathsf{Su}}_{\vec{y}_0}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_0) = \mathcal{sub}^{\mathsf{Su}}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u})$. Corollary: PA $\vdash \mathcal{sub}_{\vec{y}_0}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_0) = \mathcal{sub}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u})$.

*T13.66. If the variables of $\vec{x}$ are the same as the variables of $\vec{y}$ then PA $\vdash \mathcal{sub}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}) = \mathcal{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})$.

Suppose the variables of $\vec{x}$ are the same as the variables of $\vec{y}$ but in a possibly different order. To convert $\vec{y}$ to $\vec{x}$, a straightforward approach is to use T13.65 to switch members into the first position in the reverse of their order in $\vec{x}$. Suppose $\vec{y} = \langle x_{\mathsf{x}_4}, x_{\mathsf{x}_1}, x_{\mathsf{x}_6}, x_{\mathsf{x}_5}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3} \rangle$. Then we may sort the variables as follows:

0. $x_{\mathsf{x}_4}, x_{\mathsf{x}_1}, x_{\mathsf{x}_6}, x_{\mathsf{x}_5}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3}$
1. $x_{\mathsf{x}_6}, x_{\mathsf{x}_4}, x_{\mathsf{x}_1}, x_{\mathsf{x}_5}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3}$
2. $x_{\mathsf{x}_5}, x_{\mathsf{x}_6}, x_{\mathsf{x}_4}, x_{\mathsf{x}_1}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3}$
3. $x_{\mathsf{x}_4}, x_{\mathsf{x}_5}, x_{\mathsf{x}_6}, x_{\mathsf{x}_1}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3}$
4. $x_{\mathsf{x}_3}, x_{\mathsf{x}_4}, x_{\mathsf{x}_5}, x_{\mathsf{x}_6}, x_{\mathsf{x}_1}, x_{\mathsf{x}_2}$
5. $x_{\mathsf{x}_2}, x_{\mathsf{x}_3}, x_{\mathsf{x}_4}, x_{\mathsf{x}_5}, x_{\mathsf{x}_6}, x_{\mathsf{x}_1}$
6. $x_{\mathsf{x}_1}, x_{\mathsf{x}_2}, x_{\mathsf{x}_3}, x_{\mathsf{x}_4}, x_{\mathsf{x}_5}, x_{\mathsf{x}_6}$

The reasoning is officially by induction, but simple enough, so left as an exercise.

*T13.67. For some formula $\mathcal{P}$, let $\vec{y}_\mathsf{u} = x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_\mathsf{u}}, x_\mathsf{a}, x_{\mathsf{y}_{\mathsf{u}+1}} \ldots x_{\mathsf{y}_\mathsf{n}}$ for variable x_a not free in $\mathcal{P}$; then PA $\vdash \mathcal{sub}^{\mathsf{u}}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u}) = \mathcal{sub}^{\mathsf{Su}}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u})$.

*T13.68. For some formula $\mathcal{P}$, let $\vec{y}_\mathsf{u} = x_{\mathsf{y}_1} \ldots x_\mathsf{a} \ldots x_{\mathsf{y}_\mathsf{u}}, x_\mathsf{a}, x_{\mathsf{y}_{\mathsf{u}+1}} \ldots x_{\mathsf{y}_\mathsf{n}}$ for variable x_a duplicated in the sequence; then PA $\vdash \mathcal{sub}^{\mathsf{Su}}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u}) = \mathcal{sub}^{\mathsf{SSu}}_{\vec{y}_\mathsf{u}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}_\mathsf{u})$.

*T13.69. If the variables of $\vec{y}$ and $\vec{z}$ are ordered by their subscripts, $\vec{y}$ includes just the free variables of formula $\mathcal{P}$, but $\vec{z}$ includes variables not in $\vec{y}$, then PA $\vdash \mathcal{sub}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}) = \mathcal{sub}_{\vec{z}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{z})$.

T13.70. If $\vec{x}$ and $\vec{y}$ include all the free variables of formula $\mathcal{P}$, then PA $\vdash \mathcal{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \mathcal{sub}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y})$.

Let $\vec{x}'$ and $\vec{y}'$ be like $\vec{x}$ and $\vec{y}$ except that variables are in standard order, and $\vec{z}$ be just the free variables of formula $\mathcal{P}$ in standard order. Then by T13.66, PA $\vdash \mathcal{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \mathcal{sub}_{\vec{x}'}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}')$; by T13.69, PA $\vdash \mathcal{sub}_{\vec{x}'}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}') = \mathcal{sub}_{\vec{z}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{z})$; by T13.69 again, PA $\vdash \mathcal{sub}_{\vec{z}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{z}) = \mathcal{sub}_{\vec{y}'}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}')$; and with T13.66, PA $\vdash \mathcal{sub}_{\vec{y}'}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}') = \mathcal{sub}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y})$. So PA $\vdash \mathcal{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \mathcal{sub}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y})$.

So PA $\vdash \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}))$; which is to say PA $\vdash \mathit{Prvpa}[\![\mathcal{P}]\!]$. So if PA $\vdash \mathcal{P}$, then PA $\vdash \mathit{Prvpa}[\![\mathcal{P}]\!]$.

T13.72. PA $\vdash \mathit{Prvpa}[\![\mathcal{P} \to \mathcal{Q}]\!] \to (\mathit{Prvpa}[\![\mathcal{P}]\!] \to \mathit{Prvpa}[\![\mathcal{Q}]\!])$

Suppose $\vec{x}$ includes all the free variables of $\mathcal{P} \to \mathcal{Q}$. We set out to show PA $\vdash \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$. This leads immediately to the desired result.

Basis: PA $\vdash \mathit{sub}^{\mathsf{0}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$.

1. $\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}) = \mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner})$ — def sub
2. $\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \overline{\ulcorner \mathcal{P} \urcorner}$ — def sub
3. $\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})) = \overline{\ulcorner \mathcal{Q} \urcorner}$ — def sub
4. $\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}) = \mathit{cnd}(\mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{0}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$ — 1,2,3 =E

Assp: PA $\vdash \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$.

Show: PA $\vdash \mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$.

1. $\mathit{Wff}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \wedge \mathit{Wff}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$ — T13.63
2. $\mathit{Term}(\mathit{num}(x_{\mathsf{x_{Si}}}))$ — T13.57n
3. $\mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \mathit{formsub}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}}))$ — def sub
4. $\mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}) = \mathit{formsub}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}), \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}}))$ — def sub
5. $\mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})$
6. $\quad = \mathit{formsub}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}), \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}}))$ — def sub
7. $\quad = \mathit{formsub}(\mathit{cnd}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})),$
   $\qquad \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}}))$ — by assp
8. $\quad = \mathit{cnd}(\mathit{formsub}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}})),$
   $\qquad \mathit{formsub}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}), \mathit{gvar}(\overline{\mathsf{x}}_{\mathsf{Si}}), \mathit{num}(x_{\mathsf{x_{Si}}})))$ — 1,2 T13.58a
9. $\quad = \mathit{cnd}(\mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{Si}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$ — 3,4 =E

Indct: For any i, PA $\vdash \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}^{\mathsf{i}}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$.

So PA $\vdash \mathit{sub}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x})) = \mathit{cnd}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$. Now moving to the desired result,

1. $\quad \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x}))$ — A (g, $\to$I)
2. $\quad \overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner} = \mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner})$ — cap
3. $\quad \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}))$ — 1,2 =E
4. $\quad \mathit{Prvpa}(\mathit{cnd}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), \mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$ — 3 above
5. $\quad \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$ — 4 T13.53
6. $\mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to$
   $\qquad [\mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))]$ — 1-5 $\to$I

So PA $\vdash \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to [\mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to \mathit{Prvpa}(\mathit{sub}_{\vec{x}}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))]$ which is to say, PA $\vdash \mathit{Prvpa}[\![\mathcal{P} \to \mathcal{Q}]\!] \to (\mathit{Prvpa}[\![\mathcal{P}]\!] \to \mathit{Prvpa}[\![\mathcal{Q}]\!])$.

Finally the substitution result. In the simplest case, where x is some x_i, then $Prvpa[\![\mathcal{P}(x)]\!]$ is of the sort $Prvpa(formsub(\overline{\ulcorner\mathcal{P}(x)\urcorner}, gvar(\bar{\mathsf{i}}), num(x)))$. The free x may be replaced by t so that $Prvpa[\![\mathcal{P}(x)]\!]^x_t$ is $Prvpa(formsub(\overline{\ulcorner\mathcal{P}(x)\urcorner}, gvar(\bar{\mathsf{i}}), num(t)))$. In case t is free for x in $\mathcal{P}$, one might think this converts to $Prvpa[\![\mathcal{P}(t)]\!]$. We do not show the general case. However, we do obtain a result that moves certain substitutions across the double bracket.

*T13.73. For distinct variables x and y, where t is one of $\emptyset$, y, Sx, or Sy, and t is free for x in $\mathcal{P}$, then PA $\vdash Prvpa[\![\mathcal{P}^x_t]\!] \leftrightarrow Prvpa[\![\mathcal{P}]\!]^x_t$.

We consider just the case when $t = Sy$. Others are similar and left for homework. Where x is distinct from y, let $t = Sy$ and suppose t is free for x in $\mathcal{P}$. Let $\vec{w}$ be the sequence $x, y, \vec{z}$ where x and y do not appear in $\vec{z}$ and x and y are variables x_i and x_j. We set out to show PA $\vdash sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$. The result follows easily. The equality between these terms first obtains at $sub^{2}_{\vec{w}}$ and continues after. So our sequence for the induction is $sub^{2}_{\vec{w}}$, $sub^{3}_{\vec{w}}$, and so on. Say the indexes on $\vec{z}$ begin at three.

Basis: PA $\vdash sub^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$.

See the derivation on the following page.

Assp: For $2 \le \mathsf{u}$, PA $\vdash sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$.

Show: PA $\vdash sub^{\mathsf{Su}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub^{\mathsf{Su}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$.

1.	$sub^{\mathsf{Su}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w})$	
2.	$= formsub(sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}), gvar(\bar{\mathsf{z}}_{\mathsf{Su}}), num(x_{\mathsf{z}_{\mathsf{Su}}}))$	def sub
3.	$= formsub(sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}, gvar(\bar{\mathsf{z}}_{\mathsf{Su}}), num(x_{\mathsf{z}_{\mathsf{Su}}}))$	by assp
4.	$= formsub(sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w}), gvar(\bar{\mathsf{z}}_{\mathsf{Su}}), num(x_{\mathsf{z}_{\mathsf{Su}}}))^x_{Sy}$	abv
5.	$= sub^{\mathsf{Su}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$	def sub

Indct: For any u, PA $\vdash sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub^{\mathsf{u}}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$.

Line (4) is justified insofar as x does not appear in $\vec{z}$ and the substitution does not interact with $x_{\mathsf{z}_{\mathsf{Su}}}$. So PA $\vdash sub_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w}) = sub_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^x_{Sy}$. So by =E, PA $\vdash Prvpa(sub_{\vec{w}}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, \vec{w})) \leftrightarrow Prvpa(sub_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w}))^x_{Sy}$, where this is to say, PA $\vdash Prvpa[\![\mathcal{P}^x_{Sy}]\!] \leftrightarrow Prvpa[\![\mathcal{P}]\!]^x_{Sy}$.

This completes what we set out to show in this section. And we are positioned for a demonstration of the third derivability condition.

E13.52. Provide a demonstration for T13.63.

*E13.53. Provide a demonstration for T13.65. Hard-core: Show all of T13.64–T13.69.

Hints for T13.65. The argument is an induction on the value of u. For the show you need $\text{PA} \vdash \mathit{sub}^{\mathsf{SSu}}_{\vec{y}_0}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_0) = \mathit{sub}^{\mathsf{SSu}}_{\vec{y}_{\mathsf{u}+1}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_{\mathsf{u}+1})$. The key is that $\mathit{sub}^{\mathsf{SSu}}_{\vec{y}_{\mathsf{Su}}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_{\mathsf{Su}}) = \mathit{formsub}[\mathit{formsub}(\mathit{sub}^{\mathsf{u}}_{\vec{y}_{\mathsf{Su}}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_{\mathsf{Su}}), \mathit{gvar}(\overline{\mathsf{y}}_{\mathsf{Su}}), \mathit{num}(x_{\mathsf{y}_{\mathsf{Su}}})), \mathit{gvar}(\overline{\mathsf{a}}), \mathit{num}(x_{\mathsf{a}})]$—and you will be able to use T13.64. As a preliminary it will be useful to show with T13.58g that for any a, b, and formula $\mathcal{P}$, $\text{PA} \vdash \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{a}}), \mathit{num}(x_{\mathsf{a}})), \mathit{gvar}(\overline{\mathsf{b}}), \mathit{num}(x_{\mathsf{b}})) = \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{b}}), \mathit{num}(x_{\mathsf{b}})), \mathit{gvar}(\overline{\mathsf{a}}), \mathit{num}(x_{\mathsf{a}}))$.

Hints for T13.67. Where $\vec{y}_{\mathsf{u}} = x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_{\mathsf{u}}}, x_{\mathsf{a}}, x_{\mathsf{y}_{\mathsf{u}+1}} \ldots x_{\mathsf{y}_{\mathsf{n}}}$, let $\vec{z} = x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_{\mathsf{u}}}$ and $\vec{z}\,' = x_{\mathsf{y}_{\mathsf{u}+1}} \ldots x_{\mathsf{y}_{\mathsf{n}}}$ so $\vec{y}_{\mathsf{u}} = \vec{z}, x_{\mathsf{a}}, \vec{z}\,'$; then it suffices to show that $(*)$ $\text{PA} \vdash \mathit{sub}^{\mathsf{u}}_{\vec{z}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{z}) = \mathit{sub}^{\mathsf{Su}}_{x_{\mathsf{a}},\vec{z}}(\overline{\ulcorner\mathcal{P}\urcorner}, x_{\mathsf{a}}, \vec{z})$: for by T13.64, PA proves $\mathit{sub}^{\mathsf{u}}_{\vec{y}_{\mathsf{u}}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_{\mathsf{u}})$ is $\mathit{sub}^{\mathsf{u}}_{\vec{z}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{z})$; with $(*)$ this is $\mathit{sub}^{\mathsf{Su}}_{x_{\mathsf{a}},\vec{z}}(\overline{\ulcorner\mathcal{P}\urcorner}, x_{\mathsf{a}}, \vec{z})$; by T13.66, this is $\mathit{sub}^{\mathsf{Su}}_{\vec{z},x_{\mathsf{a}}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{z}, x_{\mathsf{a}})$; and by T13.64 again, this is just $\mathit{sub}^{\mathsf{Su}}_{\vec{y}_{\mathsf{u}}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}_{\mathsf{u}})$.

Hints for T13.69. Where the variables of $\vec{y}$ are $x_{\mathsf{y}_1} \ldots x_{\mathsf{y}_{\mathsf{m}}}$ and of $\vec{z}$ are $x_{\mathsf{z}_1} \ldots x_{\mathsf{z}_{\mathsf{n}}}$, let i.j "count" from 0.0 to m.n so that when $\mathsf{y}_{\mathsf{Si}} = \mathsf{z}_{\mathsf{Sj}}$ then $\mathsf{S(i.j) = Si.Sj}$, and when $\mathsf{y}_{\mathsf{Si}} \neq \mathsf{z}_{\mathsf{Sj}}$ then $\mathsf{S(i.j) = i.Sj}$. Then you will be able to show that for any member of this i.j sequence, $\text{PA} \vdash \mathit{sub}^{\mathsf{i}}_{\vec{y}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y}) = \mathit{sub}^{\mathsf{j}}_{\vec{z}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{z})$.

*E13.54. Complete Ø-case to T13.73. Hard-core, complete all of the remaining cases. Hint: for the Ø- and Sx-cases you need only consider the sequence $\vec{w} = x, \vec{z}$.

T13.73 (basis)

1.	$\mathit{Wff}(\overline{\ulcorner\mathcal{P}\urcorner}) \wedge \mathit{Freefor}(\overline{\ulcorner S\urcorner} * \mathit{gvar}(\overline{\mathsf{j}}), \mathit{gvar}(\overline{\mathsf{i}}), \overline{\ulcorner\mathcal{P}\urcorner})$	cap
2.	$\mathit{Var}(\mathit{gvar}(\overline{\mathsf{i}})) \wedge \mathit{Var}(\mathit{gvar}(\overline{\mathsf{j}})) \wedge \mathit{Term}(\mathit{num}(x))$	T13.57c,n
3.	$\mathit{sub}^{1}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \vec{w})$	
4.	$= \mathit{formsub}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \mathit{num}(x))$	def *sub*
5.	$= \overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}$	2 T13.55l
6.	$\mathit{sub}^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \vec{w})$	
7.	$= \mathit{formsub}(\mathit{sub}^{1}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \vec{w}), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))$	def *sub*
8.	$= \mathit{formsub}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))$	3-5 =E
9.	$= \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \overline{\ulcorner S\urcorner} * \mathit{gvar}(\overline{\mathsf{j}})), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))$	cap
10.	$= \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \mathit{num}(Sy)), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))$	1,2 T13.58m
11.	$\mathit{sub}^{1}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})$	
12.	$= \mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \mathit{num}(x))$	def *sub*
13.	$\mathit{sub}^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^{x}_{Sy}$	
14.	$= \mathit{formsub}(\mathit{sub}^{1}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w}), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))^{x}_{Sy}$	def *sub*
15.	$= \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \mathit{num}(x)), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))^{x}_{Sy}$	11-12 =E
16.	$= \mathit{formsub}(\mathit{formsub}(\overline{\ulcorner\mathcal{P}\urcorner}, \mathit{gvar}(\overline{\mathsf{i}}), \mathit{num}(Sy)), \mathit{gvar}(\overline{\mathsf{j}}), \mathit{num}(y))$	abv
17.	$\mathit{sub}^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}^{x}_{Sy}\urcorner}, \vec{w}) = \mathit{sub}^{2}_{\vec{w}}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{w})^{x}_{Sy}$	6-10,13-16 =E

Line (5) is justified by the corollary to T13.55l insofar as x is not free in $\mathcal{P}^{x}_{Sy}$.

13.5.4 The Condition

We turn now showing that for any $\Sigma_\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$. This is the result we need for D3 and so with D1, D2, and T13.8 to complete the demonstration of Gödel's second ĩncompleteness theorem for PA.

T13.74. For any $\Sigma_\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$.

Let $\mathcal{P}$ be a $\Sigma_\star$ formula. By induction on the number of operators in $\mathcal{P}$,

Basis: If a $\Sigma_\star$ $\mathcal{P}$ has no operator symbols, then for distinct variables x, y, and z, it is an atomic of the sort $\emptyset = z$, $y = z$, $Sy = z$, $x + y = z$, or $x \times y = z$.

($\emptyset$) Suppose $\mathcal{P}$ is $\emptyset = z$. Homework.

(y) Suppose $\mathcal{P}$ is $y = z$. Homework.

(S) Suppose $\mathcal{P}$ is $Sy = z$. Reason as follows:

1.	$Sy = Sy$	=I
2.	$\mathit{Prvpa}[\![Sy = Sy]\!]$	1 T13.71
3.	$\quad Sy = z$	A (g, →I)
4.	$\quad \mathit{Prvpa}[\![(Sy = z)^{z}_{Sy}]\!]$	2 abv
5.	$\quad \mathit{Prvpa}[\![Sy = z]\!]^{z}_{Sy}$	4 T13.73
6.	$\quad \mathit{Prvpa}[\![Sy = z]\!]$	5,3 =E
7.	$Sy = z \to \mathit{Prvpa}[\![Sy = z]\!]$	3-6 →I

Observe that T13.71 applies to theorems, and so not to formulas under the assumption for →I. We thus take care to restrict its application to formulas against the main scope line. Also lines (4)–(6) apply a pattern we shall see repeatedly: First line (5) applies T13.73 to (4); unabbreviated, (5) is $\mathit{Prvpa}(\mathbb{sub}_{y,z}(\overline{\ulcorner Sy = z \urcorner}, y, z))^{z}_{Sy}$; which is, $\mathit{Prvpa}(\mathbb{sub}_{y,z}(\overline{\ulcorner Sy = z \urcorner}, y, Sy))$; and so by =E, $\mathit{Prvpa}(\mathbb{sub}_{y,z}(\overline{\ulcorner Sy = z \urcorner}, y, z))$, which is (6).

(+) Suppose $\mathcal{P}$ is $x + y = z$. Then PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$. The proof in PA requires appeal to IN, with induction on the value of x in $\forall y(x + y = z \to \mathit{Prvpa}[\![x + y = z]\!])$. See the derivation on page 736.

($\times$) Suppose $\mathcal{P}$ is $x \times y = z$. Then PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$. The proof in PA requires appeal to IN on the value of x in $\forall z(x \times y = z \to \mathit{Prvpa}[\![x \times y = z]\!])$. The argument is as on page 737.

Assp: For any i, $0 \le i < k$ if a $\Sigma_\star$ $\mathcal{P}$ has i operator symbols, then PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$.

Show: If a $\Sigma_\star$ $\mathcal{P}$ has k operator symbols, then PA $\vdash \mathcal{P} \to \mathit{Prvpa}[\![\mathcal{P}]\!]$.

If $\Sigma_\star$ $\mathcal{P}$ has k operator symbols, then it is of the form, $\mathcal{A} \wedge \mathcal{B}$, $\mathcal{A} \vee \mathcal{B}$, $\exists x\mathcal{A}$, or $(\forall x \le y)\mathcal{A}$ where y is not in $\mathcal{A}$, for $\Sigma_\star$ $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\wedge$) $\mathscr{P}$ is $\mathscr{A} \wedge \mathscr{B}$. Reason as follows:

1.	$\mathscr{A} \to Prvpa[\![\mathscr{A}]\!]$	by assp
2.	$\mathscr{B} \to Prvpa[\![\mathscr{B}]\!]$	by assp
3.	$\mathscr{A} \to (\mathscr{B} \to (\mathscr{A} \wedge \mathscr{B}))$	T9.4
4.	$Prvpa[\![\mathscr{A} \to (\mathscr{B} \to (\mathscr{A} \wedge \mathscr{B}))]\!]$	3 T13.71
5.	$\quad \mathscr{A} \wedge \mathscr{B}$	A (g, $\to$I)
6.	$\quad Prvpa[\![\mathscr{A}]\!]$	from 1,5
7.	$\quad Prvpa[\![\mathscr{B}]\!]$	from 2,5
8.	$\quad Prvpa[\![\mathscr{A}]\!] \to Prvpa[\![\mathscr{B} \to (\mathscr{A} \wedge \mathscr{B})]\!]$	4 T13.72
9.	$\quad Prvpa[\![\mathscr{B} \to (\mathscr{A} \wedge \mathscr{B})]\!]$	8,6 $\to$E
10.	$\quad Prvpa[\![\mathscr{B}]\!] \to Prvpa[\![\mathscr{A} \wedge \mathscr{B}]\!]$	9 T13.72
11.	$\quad Prvpa[\![\mathscr{A} \wedge \mathscr{B}]\!]$	10,7 $\to$E
12.	$(\mathscr{A} \wedge \mathscr{B}) \to Prvpa[\![\mathscr{A} \wedge \mathscr{B}]\!]$	5-11 $\to$I

($\vee$) Similarly.

($\exists$) $\mathscr{P}$ is $\exists x \mathscr{A}$. Reason as follows:

1.	$\mathscr{A} \to Prvpa[\![\mathscr{A}]\!]$	by assp
2.	$\mathscr{A} \to \exists x \mathscr{A}$	T3.30
3.	$Prvpa[\![\mathscr{A} \to \exists x \mathscr{A}]\!]$	2 T13.71
4.	$\quad \exists x \mathscr{A}$	A (g, $\to$I)
5.	$\quad\quad \mathscr{A}$	A (g, 4$\exists$E)
6.	$\quad\quad Prvpa[\![\mathscr{A}]\!]$	1,5 $\to$E
7.	$\quad\quad Prvpa[\![\mathscr{A}]\!] \to Prvpa[\![\exists x \mathscr{A}]\!]$	3 T13.72
8.	$\quad\quad Prvpa[\![\exists x \mathscr{A}]\!]$	7,6 $\to$E
9.	$\quad Prvpa[\![\exists x \mathscr{A}]\!]$	4,5-8 $\exists$E
10.	$\exists x \mathscr{A} \to Prvpa[\![\exists x \mathscr{A}]\!]$	4-9 $\to$I

Let $Prvpa[\![\exists x \mathscr{A}]\!]$ at (8) have the same free variables as $\exists x \mathscr{A}$; then x is not free in $Prvpa[\![\exists x \mathscr{A}]\!]$, and the restriction is met for $\exists$E at (9).

($\forall$) $\mathscr{P}$ is $(\forall x \leq y)\mathscr{A}$. The argument in PA requires appeal to IN, for induction on the value of y. See the derivation on page 738.

Indct: For any $\Sigma_\star$ formula $\mathscr{P}$, PA $\vdash \mathscr{P} \to \mathbb{P}rvpa[\![\mathscr{P}]\!]$.

Now it is a simple matter to pull together our results into the third derivability condition.

T13.75. For any formula $\mathscr{P}$, PA $\vdash \Box\mathscr{P} \to \Box\Box\mathscr{P}$.

Consider any formula $\mathscr{P}$ and the Σ_1 sentence $(\Box\mathscr{P})_{\Sigma_1}$. By T13.62, there is a $(\Box\mathscr{P})_{\Sigma_\star}$ such that PA $\vdash (\Box\mathscr{P})_{\Sigma_1} \leftrightarrow (\Box\mathscr{P})_{\Sigma_\star}$. By T13.74, PA $\vdash (\Box\mathscr{P})_{\Sigma_\star} \to Prvpa[\![(\Box\mathscr{P})_{\Sigma_\star}]\!]$. Reason as follows:

1.	$(\Box\mathcal{P})_{\Sigma_1} \leftrightarrow (\Box\mathcal{P})_{\Sigma_\star}$	T13.62
2.	$(\Box\mathcal{P})_{\Sigma_\star} \to Prvpa[\![(\Box\mathcal{P})_{\Sigma_\star}]\!]$	T13.74
3.	$Prvpa[\![(\Box\mathcal{P})_{\Sigma_\star} \to (\Box\mathcal{P})_{\Sigma_1}]\!]$	1 T13.71
4.	$Prvpa[\![(\Box\mathcal{P})_{\Sigma_\star}]\!] \to Prvpa[\![(\Box\mathcal{P})_{\Sigma_1}]\!]$	3 T13.72
5.	$(\Box\mathcal{P})_{\Sigma_1} \to Prvpa[\![(\Box\mathcal{P})_{\Sigma_1}]\!]$	1,2,4 HS

So for the Σ_1 sentence $\Box\mathcal{P}$, PA $\vdash \Box\mathcal{P} \to Prvpa[\![\Box\mathcal{P}]\!]$; and since $\Box\mathcal{P}$ is a sentence, this is to say, PA $\vdash \Box\mathcal{P} \to Prvpa(\overline{\ulcorner\Box\mathcal{P}\urcorner})$; which is to say, PA $\vdash \Box\mathcal{P} \to \Box\Box\mathcal{P}$.

So, at long last, we have a demonstration of D3 and so, given demonstration of the other conditions, a complete demonstration that PA does not prove its own consistency!

E13.55. Complete the demonstration of T13.74 by completing the remaining cases.

13.6 Reflections on the Second Theorem

It is worth reflecting a bit on what we have accomplished. Beginning in section 13.2 we saw how the second theorem results for recursively axiomatized theories extending Q that satisfy the derivability conditions. We then set out to show that PA satisfies the derivability conditions. The first is easy, the others not. In section 13.3 we introduced the idea of definition in PA and demonstrated that PA defines functions coordinate to (friendly) recursive functions. 13.4 moves to demonstration of the second condition: Supposing $\Box(\mathcal{P} \to \mathcal{Q})$ and $\Box\mathcal{P}$, the basic idea of combining derivations to obtain $\Box\mathcal{Q}$, and so $\Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$ is straightforward. But considerable effort is expended to show that PA has the resources for the relevant results. And we have just completed discussion of the third condition, in which we simplified the problem by substitution theorems and $\Sigma_\star$ formulas. If you have gotten this far you have seen the theorem proved. Thus you have progressed considerably beyond the initial argument from the derivability conditions. One reason why it is typical to bypass the details is that there are *so many* details—not all themselves mathematically significant. Still, it is interesting to see *how* reasoning from Chapter 12 is reflected in PA for the second theorem.

We conclude this chapter with a couple final reflections and consequences on our results. In particular we say something about alternate characterizations of consistency, and then about the relation between Gödel's second theorem and Löb's theorem.

13.6.1 Consistency sentences

As is common for discussions of Gödel's second theorem, we have let *Conpa* be $\sim Prvpa(\overline{\ulcorner\emptyset = S\emptyset\urcorner})$. But other sentences would do as well. So, where $\mathcal{F}$ is any formula whose negation is a theorem, we might let $Conpa_a$ be $\sim Prvpa(\overline{\ulcorner\mathcal{F}\urcorner})$. In

T13.74(+)

1.	$\emptyset + y = y$	T6.56
2.	$x + Sy = z \leftrightarrow Sx + y = z$	T6.49,T6.58
3.	$Prvpa[\![\emptyset + y = y]\!]$	1 T13.71
4.	$Prvpa[\![x + Sy = z \to Sx + y = z]\!]$	2 T13.71
5.	$\quad (x + y = z)^x_\emptyset$	A (g, →I)
6.	$\quad \emptyset + y = z$	5 abv
7.	$\quad y = z$	1,6 =E
8.	$\quad Prvpa[\![(\emptyset + y = z)^z_y]\!]$	3 abv
9.	$\quad Prvpa[\![\emptyset + y = z]\!]^z_y$	8 T13.73
10.	$\quad Prvpa[\![\emptyset + y = z]\!]$	9,7 =E
11.	$\quad Prvpa[\![(x + y = z)^x_\emptyset]\!]$	10 abv
12.	$\quad Prvpa[\![x + y = z]\!]^x_\emptyset$	11 T13.73
13.	$(x + y = z)^x_\emptyset \to Prvpa[\![x + y = z]\!]^x_\emptyset$	5-12 →I
14.	$(x + y = z \to Prvpa[\![x + y = z]\!])^x_\emptyset$	13 abv
15.	$\forall y(x + y = z \to Prvpa[\![x + y = z]\!])^x_\emptyset$	14 ∀I
16.	$\quad \forall y(x + y = z \to Prvpa[\![x + y = z]\!])$	A (g, →I)
17.	$\quad\quad (x + y = z)^x_{Sx}$	A (g, →I)
18.	$\quad\quad Sx + y = z$	17 abv
19.	$\quad\quad x + Sy = z$	2,18 ↔E
20.	$\quad\quad (x + y = z \to Prvpa[\![x + y = z]\!])^y_{Sy}$	16 ∀E
21.	$\quad\quad x + Sy = z \to Prvpa[\![x + y = z]\!]^y_{Sy}$	20 abv
22.	$\quad\quad Prvpa[\![x + y = z]\!]^y_{Sy}$	21,19 →E
23.	$\quad\quad Prvpa[\![x + Sy = z]\!]$	22 T13.73
24.	$\quad\quad Prvpa[\![x + Sy = z]\!] \to Prvpa[\![Sx + y = z]\!]$	4 T13.72
25.	$\quad\quad Prvpa[\![Sx + y = z]\!]$	24,23 →E
26.	$\quad\quad Prvpa[\![x + y = z]\!]^x_{Sx}$	25 T13.73
27.	$\quad (x + y = z)^x_{Sx} \to Prvpa[\![x + y = z]\!]^x_{Sx}$	17-26 →I
28.	$\quad (x + y = z \to Prvpa[\![x + y = z]\!])^x_{Sx}$	27 abv
29.	$\quad \forall y(x + y = z \to Prvpa[\![x + y = z]\!])^x_{Sx}$	28 ∀I
30.	$\forall y(x + y = z \to Prvpa[\![x + y = z]\!]) \to$ $\forall y(x + y = z \to Prvpa[\![x + y = z]\!])^x_{Sx}$	16-29 →I
31.	$\forall y(x + y = z \to Prvpa[\![x + y = z]\!])$	15,30 IN

Observe that insofar as $Prvpa[\![\mathcal{P}(\vec{x})]\!]$ just is a (complex) formula with $\vec{x}$ free, quantifier rules apply in the usual way; in particular, ∀I at (15) and (29), and ∀E at (20) are as usual.

So PA $\vdash x + y = z \to Prvpa[\![x + y = z]\!]$.

T13.74($\times$)

1.	$\emptyset \times y = \emptyset$	T6.63
2.	$Sx \times y = z \leftrightarrow x \times y + y = z$	T6.65
3.	$x \times y = v \to (v + y = z \to x \times y + y = z)$	T3.38
4.	$\mathcal{P}rvpa[\![\emptyset \times y = \emptyset]\!]$	1 T13.71
5.	$\mathcal{P}rvpa[\![x \times y + y = z \to Sx \times y = z]\!]$	2 T13.71
6.	$\mathcal{P}rvpa[\![x \times y = v \to (v + y = z \to x \times y + y = z)]\!]$	3 T13.71
7.	$(x \times y = z)^{x}_{\emptyset}$	A (g, $\to$I)
8.	$\emptyset \times y = z$	7 abv
9.	$z = \emptyset$	1,8 =E
10.	$\mathcal{P}rvpa[\![(\emptyset \times y = z)^{z}_{\emptyset}]\!]$	4 abv
11.	$\mathcal{P}rvpa[\![\emptyset \times y = z]\!]^{z}_{\emptyset}$	10 T13.73
12.	$\mathcal{P}rvpa[\![\emptyset \times y = z]\!]$	11,9 =E
13.	$\mathcal{P}rvpa[\![(x \times y = z)^{x}_{\emptyset}]\!]$	12 abv
14.	$\mathcal{P}rvpa[\![x \times y = z]\!]^{x}_{\emptyset}$	13 T13.73
15.	$(x \times y = z)^{x}_{\emptyset} \to \mathcal{P}rvpa[\![x \times y = z]\!]^{x}_{\emptyset}$	7-14 $\to$I
16.	$(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])^{x}_{\emptyset}$	15 abv
17.	$\forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])^{x}_{\emptyset}$	16 $\forall$I
18.	$\forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])$	A (g, $\to$I)
19.	$(x \times y = z)^{x}_{Sx}$	A (g, $\to$I)
20.	$Sx \times y = z$	19 abv
21.	$x \times y + y = z$	2,20 $\leftrightarrow$E
22.	$\exists v(x \times y = v)$	=I,$\exists$I
23.	$x \times y = v$	A (g, 22$\exists$E)
24.	$v + y = z$	21,23 =E
25.	$\mathcal{P}rvpa[\![v + y = z]\!]$	24 (+) case
26.	$x \times y = v \to \mathcal{P}rvpa[\![x \times y = z]\!]^{z}_{v}$	18 $\forall$E
27.	$\mathcal{P}rvpa[\![x \times y = z]\!]^{z}_{v}$	26,23 $\to$E
28.	$\mathcal{P}rvpa[\![x \times y = v]\!]$	27 T13.73
29.	$\mathcal{P}rvpa[\![x \times y = v]\!] \to \mathcal{P}rvpa[\![v + y = z \to x \times y + y = z]\!]$	6 T13.72
30.	$\mathcal{P}rvpa[\![v + y = z \to x \times y + y = z]\!]$	29,28 $\to$E
31.	$\mathcal{P}rvpa[\![v + y = z]\!] \to \mathcal{P}rvpa[\![x \times y + y = z]\!]$	30 T13.72
32.	$\mathcal{P}rvpa[\![x \times y + y = z]\!]$	31,25 $\to$E
33.	$\mathcal{P}rvpa[\![x \times y + y = z]\!] \to \mathcal{P}rvpa[\![Sx \times y = z]\!]$	5 T13.72
34.	$\mathcal{P}rvpa[\![Sx \times y = z]\!]$	33,32 $\to$E
35.	$\mathcal{P}rvpa[\![x \times y = z]\!]^{x}_{Sx}$	34 T13.73
36.	$\mathcal{P}rvpa[\![x \times y = z]\!]^{x}_{Sx}$	22,23-35 $\exists$E
37.	$(x \times y = z)^{x}_{Sx} \to \mathcal{P}rvpa[\![x \times y = z]\!]^{x}_{Sx}$	19-36 $\to$I
38.	$(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])^{x}_{Sx}$	37 abv
39.	$\forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])^{x}_{Sx}$	38 $\forall$I
40.	$\forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!]) \to \forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])^{x}_{Sx}$	18-39 $\to$I
41.	$\forall z(x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!])$	17,40 IN

So PA $\vdash x \times y = z \to \mathcal{P}rvpa[\![x \times y = z]\!]$.

T13.74($\forall$)

1.	$\mathcal{A}^x_\varnothing \to Prvpa\llbracket \mathcal{A}^x_\varnothing \rrbracket$	by assp
2.	$(\forall x \le \varnothing)\mathcal{A} \leftrightarrow \mathcal{A}^x_\varnothing$	thrm (with T8.25)
3.	$Prvpa\llbracket \mathcal{A}^x_\varnothing \to (\forall x \le \varnothing)\mathcal{A} \rrbracket$	2 T13.71
4.	$\quad (\forall x \le y)\mathcal{A}^y_\varnothing$	A (g, $\to$I)
5.	$\quad (\forall x \le \varnothing)\mathcal{A}$	4 abv
6.	$\quad \mathcal{A}^x_\varnothing$	2,5 $\leftrightarrow$E
7.	$\quad Prvpa\llbracket \mathcal{A}^x_\varnothing \rrbracket$	1,6 $\to$E
8.	$\quad Prvpa\llbracket \mathcal{A}^x_\varnothing \rrbracket \to Prvpa\llbracket (\forall x \le \varnothing)\mathcal{A} \rrbracket$	3 T13.72
9.	$\quad Prvpa\llbracket (\forall x \le \varnothing)\mathcal{A} \rrbracket$	8,7 $\to$E
10.	$\quad Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket^y_\varnothing$	9 T13.73
11.	$(\forall x \le y)\mathcal{A}^y_\varnothing \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket^y_\varnothing$	4-10 $\to$I
12.	$((\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket)^y_\varnothing$	11 abv
13.	$\mathcal{A}^x_{Sy} \to Prvpa\llbracket \mathcal{A}^x_{Sy} \rrbracket$	by assp
14.	$(\forall x \le Sy)\mathcal{A} \leftrightarrow ((\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy})$	use T13.11q
15.	$Prvpa\llbracket ((\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}) \to (\forall x \le Sy)\mathcal{A} \rrbracket$	14 T13.71
16.	$\quad (\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket$	A (g, $\to$I)
17.	$\quad ((\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}) \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy} \rrbracket$	16,13 ($\wedge$) case
18.	$\quad\quad (\forall x \le y)\mathcal{A}^y_{Sy}$	A (g, $\to$I)
19.	$\quad\quad (\forall x \le Sy)\mathcal{A}$	18 abv
20.	$\quad\quad (\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}$	14,19 $\leftrightarrow$E
21.	$\quad\quad Prvpa\llbracket (\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy} \rrbracket$	17,20 $\to$E
22.	$\quad\quad Prvpa\llbracket (\forall x \le y)\mathcal{A} \wedge \mathcal{A}^x_{Sy} \rrbracket \to Prvpa\llbracket (\forall x \le Sy)\mathcal{A} \rrbracket$	15 T13.72
23.	$\quad\quad Prvpa\llbracket (\forall x \le Sy)\mathcal{A} \rrbracket$	22,21 $\to$E
24.	$\quad\quad Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket^y_{Sy}$	23 T13.73
25.	$\quad (\forall x \le y)\mathcal{A}^y_{Sy} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket^y_{Sy}$	18-24 $\to$I
26.	$\quad ((\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket)^y_{Sy}$	25 abv
27.	$((\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket) \to$ $((\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket)^y_{Sy}$	16-26 $\to$I
28.	$(\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket$	12,27 IN

For (5) and (10) and then (19) and (24) it is important that y in a bounded quantifier of the $\Sigma_\star$ formula does not appear in $\mathcal{A}$.

So PA $\vdash (\forall x \le y)\mathcal{A} \to Prvpa\llbracket (\forall x \le y)\mathcal{A} \rrbracket$.

particular, we might simply consider the case where $\mathscr{F}$ is $\bot$ and set $\mathit{Conpa}_a = {\sim}\mathit{Prvpa}(\overline{\ulcorner \bot \urcorner})$. Then it is easy to see that PA $\vdash \mathit{Conpa} \leftrightarrow \mathit{Conpa}_a$.

> PA $\vdash \emptyset = S\emptyset \leftrightarrow \bot$; so with D1, PA $\vdash \mathit{Prvpa}(\overline{\ulcorner \emptyset = S\emptyset \leftrightarrow \bot \urcorner})$; so with D2, PA $\vdash \mathit{Prvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \leftrightarrow \mathit{Prvpa}(\overline{\ulcorner \bot \urcorner})$; and transposing, PA $\vdash \mathit{Conpa} \leftrightarrow \mathit{Conpa}_a$.

Thus, having shown PA $\nvdash \mathit{Conpa}$ we have PA $\nvdash \mathit{Conpa}_a$ as well.

Again, one might let $\mathit{Conpa}_b = {\sim}\exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathit{til}(x))]$. Then we are saying PA is consistent just in case there is no formula such that PA proves both it and its negation. This might seem a particularly natural consistency sentence. Again, PA $\vdash \mathit{Conpa} \leftrightarrow \mathit{Conpa}_b$. We show PA $\vdash \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \leftrightarrow \exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathit{til}(x))]$ and transpose.

First from left to right: Since PA $\vdash \emptyset \neq S\emptyset$, and a contradiction implies anything, for some formula A, PA $\vdash \emptyset = S\emptyset \to A$ and PA $\vdash \emptyset = S\emptyset \to {\sim}A$. Reason as follows:

1.	$\emptyset = S\emptyset \to A$	thrm
2.	$\emptyset = S\emptyset \to {\sim}A$	thrm
3.	$\mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \to A \urcorner})$	1 D1
4.	$\mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \to {\sim}A \urcorner})$	2 D1
5.	$\quad \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	A (g, $\to$I)
6.	$\quad \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to \mathbb{P}\mathit{rvpa}(\overline{\ulcorner A \urcorner})$	3 D2
7.	$\quad \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to \mathbb{P}\mathit{rvpa}(\overline{\ulcorner {\sim}A \urcorner})$	4 D2
8.	$\quad \mathbb{P}\mathit{rvpa}(\overline{\ulcorner A \urcorner}) \wedge \mathbb{P}\mathit{rvpa}(\overline{\ulcorner {\sim}A \urcorner})$	5,6,7
9.	$\quad \mathbb{P}\mathit{rvpa}(\overline{\ulcorner A \urcorner}) \wedge \mathbb{P}\mathit{rvpa}(\mathit{til}(\overline{\ulcorner A \urcorner}))$	8 cap
10.	$\quad \exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathit{til}(x))]$	9 $\exists$I
11.	$\mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to \exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathit{til}(x))]$	7-10 $\to$I

Now the other direction. First, generalize the notion of a *basic* sentence from Chapter 4 (page 93) so that it applies not just to sentences but to formulas (and forms) of any sentential or quantificational language—again, working up a tree, *basic* expressions are the first expressions that do not have an operator from the sentential language as main operator. So the basic formulas of $Fx \wedge \exists yGxy$ are Fx and $\exists yGxy$. Say $\mathscr{B}_1 \ldots \mathscr{B}_n$ are a (sentential) *basis* for the formulas of some context (say a derivation) just in case $\mathscr{B}_1 \ldots \mathscr{B}_n$ are the basic expressions from all the formulas of that context.

Now where $\mathscr{B}_1 \ldots \mathscr{B}_n$ are a basis for $\mathscr{A}$, consider some variables $b_1 \ldots b_n$; let $\mathscr{B}_i^*$ be b_i; ${\sim}\mathscr{P}^*$ be $\mathit{til}(\mathscr{P}^*)$; and $(\mathscr{P} \to \mathscr{Q})^*$ be $\mathit{cnd}(\mathscr{P}^*, \mathscr{Q}^*)$. Then by an easy induction PA $\vdash (\mathit{Wff}(b_1) \wedge \ldots \wedge \mathit{Wff}(b_n)) \to \mathit{Wff}(\mathscr{A}^*)$. And we shall be able to show that if $\vdash_{ADs} \mathscr{P}$, and $\mathscr{B}_1 \ldots \mathscr{B}_n$ are a basis for formulas of its derivation, then PA $\vdash (\mathit{Wff}(b_1) \wedge \ldots \wedge \mathit{Wff}(b_n)) \to \mathbb{P}\mathit{rvpa}(\mathscr{P}^*)$. Though we put off details to homework, it is simple enough to see how the argument goes: The argument is an induction on the line number of a derivation, like ones we saw in Chapter 9. Consider an *ADs* derivation of $\mathcal{P}$; and assume $\mathbb{W}\mathit{ff}(b_1) \wedge \ldots \wedge \mathbb{W}\mathit{ff}(b_n)$: Corresponding to any axiom $\mathcal{A}$, we may use T13.39h,i to get $\mathbb{A}\mathit{xiompa}(\mathcal{A}^*)$ and then T13.57f for $\mathbb{P}\mathit{rvpa}(\mathcal{A}^*)$. Corresponding to an application of MP to some $\mathcal{A}$ and $\mathcal{A} \to \mathcal{B}$, use T13.53 to convert $\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}(\mathcal{A}^*, \mathcal{B}^*))$ to $\mathbb{P}\mathit{rvpa}(\mathcal{A}^*) \to \mathbb{P}\mathit{rvpa}(\mathcal{B}^*)$ and apply MP. As an example, consider the following lines of a sort we might have obtained in Chapter 3:

1.	$A \to (B \to A)$	A1
2.	$[A \to (B \to A)] \to [(A \to B) \to (A \to A)]$	A2
3.	$(A \to B) \to (A \to A)$	1,2 MP

Then A and B are a basis. And we may reason,

0.	$\mathbb{W}\mathit{ff}(a) \wedge \mathbb{W}\mathit{ff}(b)$	A (g, DT)
1.1.	$\mathbb{A}\mathit{xiompa}(\mathbb{c}\mathit{nd}(a, \mathbb{c}\mathit{nd}(b, a)))$	0 T13.39h,i
1.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}(a, \mathbb{c}\mathit{nd}(b, a)))$	1.1 T13.57f
2.1.	$\mathbb{A}\mathit{xiompa}(\mathbb{c}\mathit{nd}(\mathbb{c}\mathit{nd}[a, \mathbb{c}\mathit{nd}(b, a)], \mathbb{c}\mathit{nd}[\mathbb{c}\mathit{nd}(a, b), \mathbb{c}\mathit{nd}(a, a)]))$	0 T13.39h,i
2.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}(\mathbb{c}\mathit{nd}[a, \mathbb{c}\mathit{nd}(b, a)], \mathbb{c}\mathit{nd}[\mathbb{c}\mathit{nd}(a, b), \mathbb{c}\mathit{nd}(a, a)]))$	2.1 T13.57f
3.1.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[a, \mathbb{c}\mathit{nd}(b, a)]) \to \mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[\mathbb{c}\mathit{nd}(a, b), \mathbb{c}\mathit{nd}(a, a)])$	2 T13.53
3.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[\mathbb{c}\mathit{nd}(a, b), \mathbb{c}\mathit{nd}(a, a)])$	3.1, 1 MP
4.	$(\mathbb{W}\mathit{ff}(a) \wedge \mathbb{W}\mathit{ff}(b)) \to \mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[\mathbb{c}\mathit{nd}(a, b), \mathbb{c}\mathit{nd}(a, a)])$	0-3 DT

And similarly we might show the correlate to T3.9, $\vdash \sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$, which we record as a theorem.

T13.76. PA $\vdash (\mathbb{W}\mathit{ff}(a) \wedge \mathbb{W}\mathit{ff}(b)) \to \mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[\mathbb{t}\mathit{il}(a), \mathbb{c}\mathit{nd}(a, b)])$

But then we may reason as follows:

1.	$\mathbb{W}\mathit{ff}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	cap
2.	$\exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathbb{t}\mathit{il}(x))]$	A (g, $\to$I)
3.	$\mathbb{P}\mathit{rvpa}(j) \wedge \mathbb{P}\mathit{rvpa}(\mathbb{t}\mathit{il}(j))$	A (g, 2$\exists$E)
4.	$\mathbb{W}\mathit{ff}(j)$	3 T13.52l
5.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}[\mathbb{t}\mathit{il}(j), \mathbb{c}\mathit{nd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner})])$	1,4 T13.76
6.	$\mathbb{P}\mathit{rvpa}(\mathbb{t}\mathit{il}(j)) \to \mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner}))$	5 T13.53
7.	$\mathbb{P}\mathit{rvpa}(\mathbb{c}\mathit{nd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner}))$	from 3,6
8.	$\mathbb{P}\mathit{rvpa}(j) \to \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	7 T13.53
9.	$\mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	from 3,8
10.	$\mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2,3-9 $\exists$E
11.	$\exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathbb{t}\mathit{il}(x))] \to \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2-10 $\to$I

Note that we reason with free variables under the assumption for $\exists$E. Thus it is important that theorems 13.52l, 13.76, and 13.53 have application not merely to numerals, but to free variables.

Putting the different parts together, PA $\vdash \mathbb{P}\mathit{rvpa}(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \leftrightarrow \exists x[\mathbb{P}\mathit{rvpa}(x) \wedge \mathbb{P}\mathit{rvpa}(\mathbb{t}\mathit{il}(x))]$ and, transposing, PA $\vdash$ *Conpa* $\leftrightarrow$ *Conpa*$_b$. So, to this extent, it does not matter which version of the consistency statement we select. Underlying the point that these different statements are equivalent is that anything follows from a contradiction—so that the one follows from the others.[13] Having shown PA $\nvdash$ *Conpa*, we therefore have PA $\nvdash$ $Conpa_a$ and PA $\nvdash$ $Conpa_b$. These are particular sentences which, like $\mathscr{G}$, are unprovable. They have special interest because they are true just in case PA is consistent.

Still, it is worth asking whether there is some different sentence to express the consistency of PA such that *it* would be provable. Consider for example a trick related to the Rosser sentence. Let,

$$Prfpa_c(x, y) = Prfpa(x, y) \wedge (\forall v \leq x){\sim}Prfpa(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$$

So $Prfpa_c(x, y)$ requires a measure of consistency: it says x numbers a proof of the formula numbered y and no proof numbered less than or equal to x demonstrates inconsistency ($\emptyset = S\emptyset$). Then so long as T is a recursively axiomatized theory extending Q and T *is* consistent, $Prfpa_c(x, y)$ continues to capture $\textsc{prfpa}(\mathsf{x}, \mathsf{y})$.

(i) Suppose T is a recursively axiomatized theory extending Q and T is consistent. Suppose $\langle \mathsf{m}, \mathsf{n} \rangle \in \textsc{prfpa}$. (a) By capture, $T \vdash Prfpa(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. And (b), since T is consistent, there is no proof of a contradiction in T and again by capture, $T \vdash {\sim}Prfpa(\overline{0}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$; $T \vdash {\sim}Prfpa(\overline{1}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$ and ... and $T \vdash {\sim}Prfpa(\overline{\mathsf{m}}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$; so with T8.25, $T \vdash (\forall v \leq \overline{\mathsf{m}}){\sim}Prfpa(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$. So $T \vdash Prfpa_c(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

(ii) Suppose $\langle \mathsf{m}, \mathsf{n} \rangle \notin \textsc{prfpa}$; by capture, $T \vdash {\sim}Prfpa(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; so $T \vdash {\sim}[Prfpa(\overline{\mathsf{m}}, \overline{\mathsf{n}}) \wedge (\forall v \leq \overline{\mathsf{m}}){\sim}Prfpa(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$, and this is just to say $T \vdash {\sim}Prfpa_c(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

Given this, set $Prvpa_c(y) = \exists x Prfpa_c(x, y)$, and $Conpa_c = {\sim}Prvpa_c(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$. The idea, then, is that $Conpa_c$ just in case there is no proof, in the sense of $Prfpa_c$, of a contradiction.

But $Prvpa_c$ is designed so that $Prvpa_c(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$ is impossible—by its definition, $Prvpa_c(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$ requires an x that numbers a proof of $\emptyset = S\emptyset$ such that no $v \leq x$ numbers a proof of $\emptyset = S\emptyset$. This is impossible; thus it is nearly immediate that PA $\vdash {\sim}\exists x[Prfpa(x, \overline{\ulcorner \emptyset = S\emptyset \urcorner}) \wedge (\forall v \leq x){\sim}Prfpa(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$, and so that PA $\vdash Conpa_c$. This works because $Prfpa_c$ builds in from the start that nothing numbers a proof of $\emptyset = S\emptyset$.

Intuitively, so long as PA is consistent, $Prfpa_c$ works just fine. But if PA is not consistent, then $Prfpa_c$ no longer tracks with proof. If PA is not consistent, then there may be an m such that $Prfpa(\overline{\mathsf{m}}, \overline{\mathsf{p}})$ though there is no n such that $Prfpa_c(\overline{\mathsf{n}}, \overline{\mathsf{p}})$—just because m is greater than the number of the proof of $\overline{0} = \overline{1}$. Similarly, if PA is

[13]This equivalence breaks down in a non-classical logic which blocks *ex falso quodlibet*, the principle that from a contradiction anything follows. So, for example, in relevant logic, it might be that there is some $\mathcal{A}$ such that $T \vdash \mathcal{A} \wedge {\sim}\mathcal{A}$ but $T \nvdash \emptyset = S\emptyset$. See Priest, *Non-Classical Logics* for an introduction to these matters.

consistent, $Conpa_c$ is true, as it should be. But if PA is inconsistent then it no longer tracks with consistency—so it is not the case that T is consistent iff $Conpa_c$. So its provability is, in this sense, uninteresting.

Insofar as $Conpa_c$ is provable it must be that $Prvpa_c$ fails one or more of the derivability conditions. To see how this might be, suppose PA is inconsistent and proofs are ordered so that,

$$\mathsf{PRFPA}(\mathsf{p}, \ulcorner \mathcal{A} \to \mathcal{B} \urcorner) \qquad \mathsf{PRFPA}(\mathsf{q}, \ulcorner \mathcal{A} \urcorner) \qquad \mathsf{PRFPA}(\mathsf{r}, \ulcorner \emptyset = S\emptyset \urcorner) \qquad \mathsf{PRFPA}(\mathsf{s}, \ulcorner \mathcal{B} \urcorner)$$

where $\mathsf{p} < \mathsf{q} < \mathsf{r} < \mathsf{s}$, r is the least number for a proof of $\emptyset = S\emptyset$, and s is the least number for a proof of $\mathcal{B}$. Then PA $\vdash$ $Prvpa(\overline{\ulcorner \mathcal{A} \to \mathcal{B} \urcorner})$ and PA $\vdash$ $Prvpa(\overline{\ulcorner \mathcal{A} \urcorner})$, so that PA $\vdash$ $Prvpa(\overline{\ulcorner \mathcal{B} \urcorner})$. However both PA $\vdash$ $Prvpa_c(\overline{\ulcorner \mathcal{A} \to \mathcal{B} \urcorner})$ and PA $\vdash$ $Prvpa_c(\overline{\ulcorner \mathcal{A} \urcorner})$ but, insofar as proofs of $\mathcal{B}$ are numbered greater than the proof of $\emptyset = S\emptyset$, PA $\nvdash$ $Prvpa_c(\overline{\ulcorner \mathcal{B} \urcorner})$. In this case, D2 fails, so that our main argument to show PA $\nvdash$ *Conpa* does not apply to $Conpa_c$.

Of course, one might suggest that yet a different expression, or perhaps some change to the numbering or derivation systems would yield the provability of consisentcy. With respect to expressions, we have seen some equivalent to our original formulation and so not provable—and another that, while provable, does not yield consistency. Thus we have a template for thinking about alternatives to our demonstration that a natural consistency sentence is not provable with our standard derivation and numbering system.[14]

*E13.56. Provide the argument to show that if $\vdash_{ADs} \mathcal{P}$ and $\mathcal{B}_1 \ldots \mathcal{B}_n$ are a basis for its derivation, then PA $\vdash$ $(\mathbb{W}\!f\!f(b_1) \wedge \ldots \wedge \mathbb{W}\!f\!f(b_n)) \to \mathbb{P}rvpa(\mathcal{P}^*)$. You may take as given that for any $\mathcal{Q}$ for which $\mathcal{B}_1 \ldots \mathcal{B}_n$ are a basis, PA $\vdash$ $(\mathbb{W}\!f\!f(b_1) \wedge \ldots \wedge \mathbb{W}\!f\!f(b_n)) \to \mathbb{W}\!f\!f(\mathcal{Q})$.

E13.57. Provide the argument to show that PA $\vdash$ $Conpa_c$.

13.6.2 Löb's Theorem

There is an analogy between Gödel's first ĩncompleteness theorem and the "paradox of the liar." On the face of it, 'This sentence is not true' ('I am lying now') cannot consistently be assumed to be either true or not true (think about it—and for an accessible introduction, see Chapter 6 of Read, *Thinking About Logic*). And there is a related puzzle about the "truth teller." So, 'This sentence is true' can sensibly be assumed to be either true or not true. And there are corresponding questions about provability. By the diagonal lemma there is a sentence $\mathcal{H}$ such that $\mathcal{H} \leftrightarrow \sim\Box\mathcal{H}$, analogous to the liar, which says of itself that it is not provable; we have seen that

[14] For discussion and further references see note 9 on page 647 along with Grabmayr, "On the Invariance of Gödel's Second Theorem."

such an $\mathcal{H}$ is not provable. Similarly by the diagonal lemma there is an $\mathcal{H}$ such that $\mathcal{H} \leftrightarrow \Box\mathcal{H}$, analogous to the truth teller, which says of itself that it *is* provable. In a brief note, "A Problem Concerning Provability" L. Henkin asks whether this latter $\mathcal{H}$ is provable. The answer has interesting ramifications. An answer to Henkin's question follows immediately from Löb's theorem.

T13.77. Suppose T is a recursively axiomatized theory extending Q for which the derivability conditions D1–D3 hold and for some sentence $\mathcal{P}$, $T \vdash \Box\mathcal{P} \to \mathcal{P}$, then $T \vdash \mathcal{P}$. *Löb's Theorem.*

Suppose T is a recursively axiomatized theory extending Q for which the derivability conditions hold and for some sentence $\mathcal{P}$, $T \vdash \Box\mathcal{P} \to \mathcal{P}$. Since T is a recursively axiomatized theory extending Q, the diagonal lemma obtains. Consider $Prvt(y) \to \mathcal{P}$; this is an expression of the sort $\mathcal{F}(y)$ to which the diagonal lemma applies; so by the diagonal lemma there is some $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow (Prvt(\overline{\ulcorner\mathcal{H}\urcorner}) \to \mathcal{P})$—that is, $T \vdash \mathcal{H} \leftrightarrow (\Box\mathcal{H} \to \mathcal{P})$. Reason as in the upper box on page 745.

Now return to our original question. Suppose $T \vdash \mathcal{P} \leftrightarrow \Box\mathcal{P}$; then $T \vdash \Box\mathcal{P} \to \mathcal{P}$; so by Löb's theorem, $T \vdash \mathcal{P}$. So if T proves $\mathcal{P} \leftrightarrow \Box\mathcal{P}$, then T proves $\mathcal{P}$.

Löb's theorem is at least surprising! From s̃oundness, *if* $\mathcal{P}$ is provable then $\mathcal{P}$, so that $\Box\mathcal{P} \to \mathcal{P}$ is true. One might think that PA would "believe" in its s̃oundness so that any such sentence would be provable. But from the theorem, if PA $\nvdash \mathcal{P}$, then PA $\nvdash \Box\mathcal{P} \to \mathcal{P}$. So in any case when PA $\nvdash \mathcal{P}$, PA does not "know" about its own s̃oundness with respect to $\mathcal{P}$. Observe that insofar as $\Box\mathcal{P} \to \mathcal{P}$ is true, for any case where PA $\nvdash \mathcal{P}$ we have here another sentence true but not provable.

And the theorem permits some interesting observations. First, an application to the logic of provability. We have thought of $\Box$ as an abbreviation in $\mathcal{L}_{NT}$, applied in forms whose operators are $\sim$, $\to$, and $\Box$. By obtaining the derivability conditions, we have shown that K4 is s̊ound in the sense that, at the level of such forms, if $\vdash_{K4} \mathcal{P}$ then PA $\vdash \mathcal{P}$—if expressions of form $\mathcal{P}$ are theorems of K4, then (unabbreviated) expressions of that same form are theorems of PA. It is natural to ask if the converse is true, whether K4 is c̊omplete so that if PA $\vdash \mathcal{P}$, then $\vdash_{K4} \mathcal{P}$. But K4 is not so c̊omplete. To see this let K4LR be like K4 but with the addition of the *Löb rule*,

LR If $T \vdash \Box\mathcal{P} \to \mathcal{P}$ then $T \vdash \mathcal{P}$.

By Löb's theorem, K4LR is s̊ound, so that if $\vdash_{K4LR} \mathcal{P}$, then PA $\vdash \mathcal{P}$. But by its appeal to the diagonal lemma, the proof of Löb's theorem is not entirely contained within K4. And, in fact, K4LR has theorems that are not theorems of K4. In particular, $\vdash_{K4LR} \Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$,

1.	$\Box[\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}] \to [\Box\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\Box\mathcal{P}]$	D2
2.	$\Box(\Box\mathcal{P} \to \mathcal{P}) \to (\Box\Box\mathcal{P} \to \Box\mathcal{P})$	D2
3.	$\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\Box(\Box\mathcal{P} \to \mathcal{P})$	D3
4.	$\Box[\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}] \to [\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}]$	1,2,3 T6.4
5.	$\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$	4 LR

From this, PA $\vdash \Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$. But from E13.60 just below, $\nvDash_{K4} \Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$ so that from the soundness of K4 on its (worlds) semantics, $\nvdash_{K4} \Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$. So PA proves something that K4 does not. So K4 is not complete in the sense that if PA $\vdash \mathcal{P}$ then $\vdash_{K4} \mathcal{P}$.

It is worth observing that K4LR is equivalent to a logic GL that drops the Löb rule and is like K4 with D3 replaced by $\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$. Since K4LR proves $\Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$, K4LR proves anything proved by GL. And GL proves anything proved by K4LR: By E13.59 below, the Löb rule is derived in GL. And though D3 is replaced by the new axiom, it remains a theorem of GL. For this, see the lower box on the next page. Since they are equivalent, together with K4LR, GL is sound in the sense that if $\vdash_{GL} \mathcal{P}$ then PA $\vdash \mathcal{P}$. In fact GL (K4LR) is also complete so that if PA $\vdash \mathcal{P}$ then $\vdash_{GL} \mathcal{P}$. So GL (K4LR) represents the *logic of provability*. But discussion of its completeness is a matter for another place (see Boolos, *The Logic of Provability*).

Finally, given that Löb's theorem depends upon the derivability conditions, it is perhaps not surprising that Löb's theorem both results in and results from Gödel's second theorem: First, the second theorem follows from Löb's result.

> For some recursively axiomatized T including PA, suppose Löb's theorem but not Gödel's second theorem. With the latter, T is consistent and $T \vdash \sim\Box(\overline{0} = \overline{1})$; from the second of these, with $\vee$I and Impl, $T \vdash \Box(\overline{0} = \overline{1}) \to \overline{0} = \overline{1}$; so by Löb's theorem $T \vdash \overline{0} = \overline{1}$: but $T \vdash \overline{0} \neq \overline{1}$; so T is inconsistent. Reject the assumption: Gödel's second theorem obtains.

And Löb's theorem follows from Gödel's second theorem.[15] For this we shall need a couple of preliminary results. First, a result analogous to the deduction theorem (T9.3). For some theory T extending PA, let T' be $T \cup \{\mathcal{P}\}$. Then T' demonstrates that if $\mathcal{A}$ is provable in T', then $\mathcal{P} \to \mathcal{A}$ is provable in T. Thus, noticing the distinction between *Prvt* and *Prvt'*,

T13.78. For a recursively axiomatized T including PA and sentences $\mathcal{P}$ and $\mathcal{A}$, let T' be $T \cup \{\mathcal{P}\}$; then $T' \vdash \mathit{Prvt}'(\overline{\ulcorner\mathcal{A}\urcorner}) \to \mathit{Prvt}(\overline{\ulcorner\mathcal{P} \to \mathcal{A}\urcorner})$.

> Consider a recursively axiomatized theory T including PA and sentences $\mathcal{P}$ and $\mathcal{A}$ with Gödel numbers p and a. Let T' be $T \cup \{\mathcal{P}\}$. Reasoning in the case where T just *is* PA, the basic structure of the argument is as from the box on page 747. Reasoning reflects that for the deduction theorem.

[15]This argument originates from a lecture by Saul Kripke. See Kripke, "On Two Paradoxes of Knowledge," pages 47–48; and Smith, *An Introduction to Gödel's Theorems*, page 257.

T13.77

1.	$\Box\mathcal{P} \to \mathcal{P}$	prem
2.	$\mathcal{H} \leftrightarrow (\Box\mathcal{H} \to \mathcal{P})$	diag lemma
3.	$[\mathcal{H} \to (\Box\mathcal{H} \to \mathcal{P})] \wedge [(\Box\mathcal{H} \to \mathcal{P}) \to \mathcal{H}]$	2 abv
4.	$\mathcal{H} \to (\Box\mathcal{H} \to \mathcal{P})$	3 with T3.21
5.	$\Box[\mathcal{H} \to (\Box\mathcal{H} \to \mathcal{P})]$	4 D1
6.	$\Box[\mathcal{H} \to (\Box\mathcal{H} \to \mathcal{P})] \to [\Box\mathcal{H} \to \Box(\Box\mathcal{H} \to \mathcal{P})]$	D2
7.	$\Box\mathcal{H} \to \Box(\Box\mathcal{H} \to \mathcal{P})$	6,5 MP
8.	$\Box(\Box\mathcal{H} \to \mathcal{P}) \to (\Box\Box\mathcal{H} \to \Box\mathcal{P})$	D2
9.	$\Box\mathcal{H} \to (\Box\Box\mathcal{H} \to \Box\mathcal{P})$	7,8 T3.2
10.	$[\Box\mathcal{H} \to (\Box\Box\mathcal{H} \to \Box\mathcal{P})] \to [(\Box\mathcal{H} \to \Box\Box\mathcal{H}) \to (\Box\mathcal{H} \to \Box\mathcal{P})]$	A2
11.	$(\Box\mathcal{H} \to \Box\Box\mathcal{H}) \to (\Box\mathcal{H} \to \Box\mathcal{P})$	10,9 MP
12.	$\Box\mathcal{H} \to \Box\Box\mathcal{H}$	D3
13.	$\Box\mathcal{H} \to \Box\mathcal{P}$	11,12 MP
14.	$\Box\mathcal{H} \to \mathcal{P}$	13,1 T3.2
15.	$(\Box\mathcal{H} \to \mathcal{P}) \to \mathcal{H}$	3 with T3.20
16.	$\mathcal{H}$	15,14 MP
17.	$\Box\mathcal{H}$	16 D1
18.	$\mathcal{P}$	14,17 MP

So if $T \vdash \Box\mathcal{P} \to \mathcal{P}$, then $T \vdash \mathcal{P}$

D3 in GL

1.	$\Box\mathcal{P} \to [\mathcal{P} \to (\Box\mathcal{P} \wedge \mathcal{P})]$	T9.4
2.	$(\Box\Box\mathcal{P} \wedge \Box\mathcal{P}) \to \Box\mathcal{P}$	T3.20
3.	$(\Box\Box\mathcal{P} \wedge \Box\mathcal{P}) \to [\mathcal{P} \to (\Box\mathcal{P} \wedge \mathcal{P})]$	2,1 T3.2
4.	$\mathcal{P} \to [(\Box\Box\mathcal{P} \wedge \Box\mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})]$	3 T3.3
5.	$\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\Box\mathcal{P} \wedge \Box\mathcal{P})$	E13.8
6.	$[(\Box\Box\mathcal{P} \wedge \Box\mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})] \to [\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})]$	5 T3.5
7.	$\mathcal{P} \to [\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})]$	4,6 T3.2
8.	$\Box(\mathcal{P} \to [\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})])$	7 D1
9.	$\Box(\mathcal{P} \to [\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})]) \to$ $(\Box\mathcal{P} \to \Box[\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})])$	D2
10.	$\Box\mathcal{P} \to \Box[\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})]$	9,8 MP
11.	$\Box[\Box(\Box\mathcal{P} \wedge \mathcal{P}) \to (\Box\mathcal{P} \wedge \mathcal{P})] \to \Box(\Box\mathcal{P} \wedge \mathcal{P})$	GL
12.	$\Box\mathcal{P} \to \Box(\Box\mathcal{P} \wedge \mathcal{P})$	10,11 T3.2
13.	$\Box\mathcal{P} \to (\Box\Box\mathcal{P} \wedge \Box\mathcal{P})$	12,5 T3.2
14.	$(\Box\Box\mathcal{P} \wedge \Box\mathcal{P}) \to \Box\Box\mathcal{P}$	T3.21
15.	$\Box\mathcal{P} \to \Box\Box\mathcal{P}$	13,14 T3.2

Now let $T' = T \cup \{\sim\mathcal{P}\}$. Suppose $T \vdash \mathcal{P}$; then both $T' \vdash \mathcal{P}$ and $T' \vdash \sim\mathcal{P}$, so that T' is inconsistent. And with T10.6, if $T \nvdash \mathcal{P}$, then T' is consistent. So T' is consistent iff $T \nvdash \mathcal{P}$. Thus one might treat $\sim Prvt(\overline{\ulcorner\mathcal{P}\urcorner})$ as yet another consistency sentence for T', true iff T' is consistent—and provably equivalent to $Cont'$. We shall require just one side of this equivalence, that if $T' \vdash \sim Prvt(\overline{\ulcorner\mathcal{P}\urcorner})$ then $T' \vdash Cont'$. For this, distinguish $\Box_T$ and $\Box_{T'}$ corresponding to $Prvt$ and $Prvt'$. Then by application of T13.78, $T' \vdash \Box_{T'}\mathcal{A} \to \Box_T(\sim\mathcal{P} \to \mathcal{A})$. And,

T13.79. For a recursively axiomatized T including PA, let T' be $T \cup \{\sim\mathcal{P}\}$; then if $T' \vdash \sim\Box_T\mathcal{P}$, then $T' \vdash Cont'$.

Let T be a recursively axiomatized theory including PA, and T' be $T \cup \{\sim\mathcal{P}\}$, and suppose $T' \vdash \sim\Box_T\mathcal{P}$. Since T includes PA, $T \vdash (\sim\mathcal{P} \to \overline{0} = \overline{1}) \to \mathcal{P}$; so with D1 and D2, $T \vdash \Box_T(\sim\mathcal{P} \to \overline{0} = \overline{1}) \to \Box_T\mathcal{P}$; and because T' extends T, $T' \vdash \Box_T(\sim\mathcal{P} \to \overline{0} = \overline{1}) \to \Box_T\mathcal{P}$. Now reasoning in T',

1.	$\sim\Box_T\mathcal{P}$	from T'
2.	$\Box_T(\sim\mathcal{P} \to \overline{0} = \overline{1}) \to \Box_T\mathcal{P}$	as above
3.	$\Box_{T'}(\overline{0} = \overline{1}) \to \Box_T(\sim\mathcal{P} \to \overline{0} = \overline{1})$	T13.78
4.	$\quad\sim Cont'$	A (c, $\sim$E)
5.	$\quad\Box_{T'}(\overline{0} = \overline{1})$	4 abv
6.	$\quad\Box_T(\sim\mathcal{P} \to \overline{0} = \overline{1})$	5,3 $\to$E
7.	$\quad\Box_T\mathcal{P}$	2,6 $\to$E
8.	$\quad\bot$	1,7 $\bot$I
9.	$Cont'$	4-8 $\sim$E

So if $T' \vdash \sim\Box_T\mathcal{P}$, then $T' \vdash Cont'$.

And if desired, it is not hard to show the other direction, that if $T' \vdash Cont'$ then $T' \vdash \sim\Box_T\mathcal{P}$.[16]

Now to show that Gödel's second theorem implies Löb's theorem we may reason as follows:

For some recursively axiomatized T including PA, suppose Gödel's second theorem but not Löb's theorem. From the latter, for some $\mathcal{P}$, $T \vdash \Box_T\mathcal{P} \to \mathcal{P}$ but $T \nvdash \mathcal{P}$. If T is inconsistent, then $T \vdash \mathcal{P}$; but $T \nvdash \mathcal{P}$; so T is consistent. Since T is consistent and $T \nvdash \mathcal{P}$ by T10.6, $T \cup \{\sim\mathcal{P}\}$ is consistent. Let T' be $T \cup \{\sim\mathcal{P}\}$; so T' is consistent. Since T' extends T, $T' \vdash \Box_T\mathcal{P} \to \mathcal{P}$, and since it has $\sim\mathcal{P}$ as an axiom, $T' \vdash \sim\mathcal{P}$; so by MT, $T' \vdash \sim\Box_T\mathcal{P}$. But since T' is consistent, by Gödel's second theorem, $T' \nvdash Cont'$; so by T13.79, $T' \nvdash \sim\Box_T\mathcal{P}$. This is impossible: Löb's theorem obtains.

[16] Within T' suppose $Cont'$; then, as from the previous section, $\sim\exists x[\mathbb{P}rvt'(x) \wedge \mathbb{P}rvt'(\mathfrak{til}(x))]$. Assume for contradiction that $Prvt(\overline{\ulcorner\mathcal{P}\urcorner})$; then since T' includes T, $Prvt'(\overline{\ulcorner\mathcal{P}\urcorner})$; and since $\sim\mathcal{P}$ is an axiom of T', $\mathbb{P}rvt'(\mathfrak{til}(\overline{\ulcorner\mathcal{P}\urcorner}))$; so, generalizing, $\exists x[\mathbb{P}rvt'(x) \wedge \mathbb{P}rvt'(\mathfrak{til}(x))]$; reject the assumption, $\sim\mathbb{P}rvt(\overline{\ulcorner\mathcal{P}\urcorner})$.

T13.78

1.	$\mathit{Sent}(\overline{\mathsf{p}})$	cap
2.	$\mathit{Wff}(\overline{\mathsf{p}})$	cap
3.	$\mathit{Prvt}'(\overline{\mathsf{a}})$	A (g, $\rightarrow$I)
4.	$\exists x\, \mathit{Prft}'(x, \overline{\mathsf{a}})$	3 abv
5.	$\mathit{Prft}'(m, \overline{\mathsf{a}})$	A (g, 4$\exists$E)
6.	$\mathit{exp}(m, \mathit{len}(m \dot{-} \overline{1})) = \overline{\mathsf{a}}$	5 T13.39j
7.	$\overline{1} < m$	5 T13.39j
8.	$(\forall k < \mathit{len}(m))[\mathit{Axiomt}(\mathit{exp}(m,k)) \vee \overline{\mathsf{p}} = \mathit{exp}(m,k) \vee$ $(\exists i < k)(\exists j < k)\mathit{Icon}(\mathit{exp}(m,i), \mathit{exp}(m,j), \mathit{exp}(m,k))]$	5 T13.39j
9.	$(\forall z < x)[z < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,z)))]$	A (g, $\rightarrow$I)
10.	$x < \mathit{len}(m)$	A (g, $\rightarrow$I)
11.	$\mathit{Axiomt}(\mathit{exp}(m,x)) \vee \overline{\mathsf{p}} = \mathit{exp}(m,x) \vee$ $(\exists i < x)(\exists j < x)\mathit{Icon}(\mathit{exp}(m,i), \mathit{exp}(m,j), \mathit{exp}(m,x))$	8,10 ($\forall$E)
12.	$\mathit{Axiomt}(\mathit{exp}(m,x))$	A (g, 11$\vee$E)
13.	$\mathit{Wff}(\mathit{exp}(m,x))$	12 T13.52k
14.	$\mathit{Prvt}(\mathit{exp}(m,x))$	12 T13.57f
15.	$\mathit{Axiomt}(\mathit{cnd}(\mathit{exp}(m,x), \mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x))))$	2,13 T13.39h,i
16.	$\mathit{Prvt}(\mathit{cnd}(\mathit{exp}(m,x), \mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x))))$	15 T13.57f
17.	$\mathit{Prvt}(\mathit{exp}(m,x)) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	16 T13.53
18.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	17,14 $\rightarrow$E
19.	$\overline{\mathsf{p}} = \mathit{exp}(m,x)$	A (g, 11$\vee$E)
20.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \overline{\mathsf{p}}))$	2 as T13.76
21.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	20,19 =E
22.	$(\exists i < x)(\exists j < x)\mathit{Icon}(\mathit{exp}(m,i), \mathit{exp}(m,j), \mathit{exp}(m,x))$	A (g, 11$\vee$E)
	$\vdots$	
23.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	homework
24.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	11,12-18,19-21,22-23 $\vee$E
25.	$x < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$	10-24 $\rightarrow$I
26.	$(\forall z < x)[z < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,z)))] \rightarrow$ $[x < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))]$	9-25 $\rightarrow$I
27.	$\forall x\{(\forall z < x)[z < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,z)))] \rightarrow$ $[x < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))]\}$	26 $\forall$I
28.	$\forall x[x < \mathit{len}(m) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))]$	27 T13.11ah
29.	$\emptyset < \mathit{len}(m)$	7 T13.44h
30.	$\mathit{len}(m) \dot{-} \overline{1} < m$	29 T13.21g
31.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m, \mathit{len}(m) \dot{-} \overline{1})))$	from 28,30
32.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \overline{\mathsf{a}}))$	31,6 =E
33.	$\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \overline{\mathsf{a}}))$	4,5-32 $\exists$E
34.	$\mathit{Prvt}'(\overline{\mathsf{a}}) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \overline{\mathsf{a}}))$	3-33 $\rightarrow$I

So $T' \vdash \mathit{Prvt}'(\overline{\ulcorner \mathcal{A} \urcorner}) \rightarrow \mathit{Prvt}(\mathit{cnd}(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{A} \urcorner}))$, and by capture, $T' \vdash \mathit{Pprvt}'(\overline{\ulcorner \mathcal{A} \urcorner}) \rightarrow \mathit{Prvt}(\overline{\ulcorner \mathcal{P} \rightarrow \mathcal{A} \urcorner})$. From $\mathit{Prvt}'(\overline{\mathsf{a}})$ there is a sequence numbered m with last member $\mathcal{A}$ such that each member is an axiom of T, $\mathcal{P}$ itself, or arises from previous members by a rule. Then from the inductive assumption (9), and $x < \mathit{len}(m)$ at (10), the main task is to show $\mathit{Prvt}(\mathit{cnd}(\overline{\mathsf{p}}, \mathit{exp}(m,x)))$. In this case where T is PA, we freely appeal to theorems from before. It will be clear how to extend the result to recursively axiomatized theories extending PA.

This argument depends upon T13.79, which in turn depends upon D1 and D2. But by reasoning from the second-to-last sentence above, if T' is consistent, then $T' \nvdash \sim\Box_T \mathcal{P}$. Treating $\sim\Box_T \mathcal{P}$ itself as a consistency sentence for T', this result appears as an *instance* of Gödel's second theorem. Then Löb's theorem follows from (this instance of) the second theorem without separate appeal to T13.79. At any rate, the force of Löb's theorem is closely related to that of Gödel's second theorem. We might have expected something of the sort insofar as our K4 derivation of Löb's theorem for T13.77 requires all three of the derivability conditions no less than the K4 derivation of the key result T13.7 for Gödel's second theorem.

E13.58. In the middle of a restless night dreaming about PA you bolt out of bed. "Eureka!" you cry, "I have discovered a simple means for proving the consistency of arithmetic in a consistent theory." Supposing PA is consistent, your idea is to show PA $\vdash \Box(\overline{0} = \overline{1}) \to \overline{0} = \overline{1}$; then from PA $\vdash \overline{0} \neq \overline{1}$ it follows that PA $\vdash \sim\Box(\overline{0} = \overline{1})$ and so that PA $\vdash$ *Conpa*. Explain why this is one of those ideas that seems better at night than in the cold light of day.

E13.59. Show that if $\vdash_{GL} \Box\mathcal{P} \to \mathcal{P}$ then $\vdash_{GL} \mathcal{P}$, and so that the Löb rule is derived in GL.

*E13.60. For those with some knowledge of worlds semantics for modal logic: K4 is the normal modal logic with a transitive access relation. (i) Find a K4 interpretation to show $\nvDash_{K4} \Box(\Box\mathcal{P} \to \mathcal{P}) \to \Box\mathcal{P}$ by a case where $\mathcal{P}$ is atomic. Hint: You can do this on an interpretation with just one world. (ii) Where the worlds are $\mathsf{a}, \mathsf{b}, \mathsf{c}$ and $\mathsf{aRb}, \mathsf{bRc}, \mathsf{aRc}$, show that the axiom is true at a. Remark: The axiom is valid on interpretations which are such that R is transitive and every non-empty set Z of worlds has a member $\mathsf{x} \in \mathsf{Z}$ with no $\mathsf{y} \in \mathsf{Z}$ such that xRy (so R is transitive and R^{-1} is *well-founded*). Observe that (ii) meets this condition, but in your answer to (i) it must fail. Challenge: show that the axiom is valid on interpretations which meet the condition.

E13.61. Complete the demonstration for T13.78 in the case where T is PA. You may find the result from E13.44, that PA $\vdash (\mathbb{S}ent(p) \wedge \mathbb{V}ar(v)) \to \sim\mathbb{F}ree_f(p, v)$ useful.

*E13.62. Reasoning for Löb's theorem is closely related to *Curry's paradox*. For this read $\Box\mathcal{P}$ to say that $\mathcal{P}$ is *true* rather than that it is provable. Consider some false sentence $\mathcal{F}$, as 'I have no head'. Let $\mathcal{C}$ be the sentence, "If this sentence is true then $\mathcal{F}$"—that is, "If '$\mathcal{C}$' is true then $\mathcal{F}$." Take as given,

D1′.	if $\mathcal{P}$, then $\Box\mathcal{P}$	truth analog to D1
D2′.	$\Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$	truth analog to D2
D3′.	$\Box\mathcal{P} \to \Box\Box\mathcal{P}$	truth analog to D3

And as premises,

1′.	$\Box\mathcal{F} \to \mathcal{F}$	from nature of truth (Tarski's schema T)
2′.	$\mathcal{C} \leftrightarrow (\Box\mathcal{C} \to \mathcal{F})$	from the definition of $\mathcal{C}$

Use these principles to show that you have no head. Reflect on this result (if, indeed, you can without a head): When $\Box$ indicates provability, we are in a position to deny (1) that PA $\vdash \Box\mathcal{P} \to \mathcal{P}$ whenever PA $\vdash \sim\mathcal{P}$. But it may seem less plausible to deny (1′) in a context where $\sim\mathcal{F}$. Supposing you do have a head, what do you think is wrong? For discussion, see Chapter 6 of Read, *Thinking About Logic*.

E13.63. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text. The last three might be based on working, or maybe just paging, through the relevant sections.

a. The essential elements contributing to the proof (from this chapter) of the ĩncompleteness of arithmetic.

b. With special focus on section 13.2, the essential elements contributing to the demonstration that PA does not prove its own consistency.

c. The essential elements contributing to the demonstration that PA defines friendly recursive functions.

d. The essential elements contributing to the demonstration of the second derivability condition.

e. The essential elements contributing to the demonstration of the third derivability condition.

Final Theorems of Chapter 13

T13.54. Further results for $\mathbb{T}ermsub$.

T13.55. Further results for $\mathbb{F}ormsub$.

T13.56. Results for $\mathbb{F}reefor$.

T13.57. Results for $\mathbb{n}um$, Gen, and A4.

T13.58. Results for iterated substitutions.

T13.59. For any atomic $\mathcal{P}$ of the form $t = x$ where x does not appear in t, there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

T13.60. For any Δ_0 formula $\mathcal{P}_{\Delta_0}$, there is a normal formula $\mathcal{P}_{\text{N}}$ such that $\vdash \mathcal{P}_{\Delta_0} \leftrightarrow \mathcal{P}_{\text{N}}$.

T13.61. For any Δ_0 formula $\mathcal{P}_{\Delta_0}$ there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P}_{\Delta_0} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

T13.62 For any Σ_1 formula $\mathcal{P}_{\Sigma_1}$ there is a $\Sigma_\star$ formula $\mathcal{P}_{\Sigma_\star}$ such that PA $\vdash \mathcal{P}_{\Sigma_1} \leftrightarrow \mathcal{P}_{\Sigma_\star}$.

T13.63 For any i and formula $\mathcal{P}$, PA $\vdash \mathbb{W}f\!f(\mathbb{s}ub^{\mathsf{i}}_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}))$.

corollary: PA $\vdash \mathbb{W}f\!f(\mathbb{s}ub_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y}))$.

T13.64–T13.70 Theorems leading to T13.70: If $\vec{x}$ and $\vec{y}$ include all the free variables of formula $\mathcal{P}$, then PA $\vdash \mathbb{s}ub_{\vec{x}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = \mathbb{s}ub_{\vec{y}}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{y})$.

T13.71 If PA $\vdash \mathcal{P}$, then PA $\vdash Prvt[\![\mathcal{P}]\!]$. (analog to D1)

T13.72 PA $\vdash Prvt[\![\mathcal{P} \to \mathcal{Q}]\!] \to (Prvt[\![\mathcal{P}]\!] \to Prvt[\![\mathcal{Q}]\!])$. (analog to D2)

T13.73 For distinct variables x and y, where t is one of $\emptyset$, y, Sx, or Sy, and t is free for x in $\mathcal{P}$, then PA $\vdash Prvpa[\![\mathcal{P}^x_t]\!] \leftrightarrow Prvpa[\![\mathcal{P}]\!]^x_t$.

T13.74 For any $\Sigma_\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \to Prvt[\![\mathcal{P}]\!]$.

T13.75 For any formula $\mathcal{P}$, PA $\vdash \Box\mathcal{P} \to \Box\Box\mathcal{P}$. (D3)

T13.76 PA $\vdash (\mathbb{W}f\!f(a) \wedge \mathbb{W}f\!f(b)) \to \mathbb{P}rvt(\mathbb{c}nd[\mathbb{t}il(a), \mathbb{c}nd(a, b)])$.

T13.77 Suppose T is a recursively axiomatized theory extending Q for which the derivability conditions D1–D3 hold and for some sentence $\mathcal{P}$, $T \vdash \Box\mathcal{P} \to \mathcal{P}$, then $T \vdash \mathcal{P}$. *Löb's Theorem.*

T13.78 For a recursively axiomatized T including PA and sentences $\mathcal{P}$ and $\mathcal{A}$, let T' be $T \cup \{\mathcal{P}\}$; then $T' \vdash Prvt'(\overline{\ulcorner \mathcal{A} \urcorner}) \to Prvt(\overline{\ulcorner \mathcal{P} \to \mathcal{A} \urcorner})$.

T13.79 For a recursively axiomatized T including PA, let T' be $T \cup \{\sim\mathcal{P}\}$; then if $T' \vdash \sim\Box_T\mathcal{P}$, then $T' \vdash Cont'$.

Chapter 14

Logic and Computability

In the introduction to Part IV we raised the question whether there is an effective method to decide if a given sentence is a theorem. In this chapter we take up that question, along with some topics in computability more generally. We begin with the notion of a Turing machine and a Turing computable function; then we shall be able to show that the Turing computable functions are the same as the recursive functions (section 14.1). Once we have seen this, it is a short step from a problem about computability—the *halting problem*—to another demonstration of essential results (section 14.2). Further, according to Church's thesis, the Turing computable functions, and so the recursive functions, are *all* the algorithmically computable functions. This converts results like T12.23 according to which no recursive relation is true just of (numbers for) theorems of predicate logic, into ones according to which no algorithmically decidable relation is true just of theorems of predicate logic—where this result is much more than a curiosity about an obscure class of functions (section 14.3).

14.1 Turing Computable Functions

We begin saying what a Turing machine, and the Turing computable functions are. Then we turn to demonstrations that Turing computable functions are recursive, and recursive functions are Turing computable.

14.1.1 Turing Machines

A Turing machine is a simple device which, despite its simplicity, is capable of computing any recursive function—and capable of computing whatever is computable by the more sophisticated computers with which we are familiar.[1]

[1]So called after Alan Turing, who originally proposed them hypothetically, prior to the existence of modern computing devices, for purposes much like our own. Turing went on to develop electro-mechanical machines for code breaking during World War II, and was involved in development of early

We may think of a Turing machine as consisting of a *tape*, *machine head*, and a finite set of *instruction quadruples*.

(A)

		1	0	1		
		▲				

The tape is a sequence of cells, infinite in two directions, where the cells may be empty or filled with 0 or 1. The machine head, indicated by arrow, reads or writes the contents of a given cell, and moves left or right, one cell at a time. The head is capable of five actions: (L) move left one cell; (R) move right one cell; (B) write a blank; (0) write a zero; (1) write a one. When the head is over a cell it is capable of reading or writing the contents of that cell.

Instruction quadruples are of the sort, $\langle q_1, C, A, q_2\rangle$ and constitute a function in the sense that no two quadruples have $\langle q_1, C\rangle$ the same but $\langle A, q_2\rangle$ different. For an instruction quadruple: (q_1) labels the quadruple; (C) is a possible state or content of the scanned cell; (A) is one of the five actions; (q_2) is a label for some (other) quadruples. In effect, an instruction quadruple q_1 says, "if the current cell has content C, perform action A and go to instruction q_2." The machine begins at an instruction with label $q_1 = 1$, and stops after executing an instruction with $q_2 = 0$.[2]

For a simple example, consider the following quadruples, along with the tape (A) from above.

(B)

$\langle 1, 0, R, 1\rangle$ 1: if 0, move right and return to instruction 1
$\langle 1, 1, 0, 1\rangle$ 1: if 1, write 0 and return to instruction 1
$\langle 1, B, L, 2\rangle$ 1: if blank, move left and go to instruction 2

$\langle 2, 0, L, 2\rangle$ 2: if 0, move left and return to instruction 2
$\langle 2, B, R, 0\rangle$ 2: if blank, move right and stop

The machine begins at label 1. In this case, the head is over a cell with content 1; so from the second instruction the machine writes 0 in that cell and returns to instruction label 1. Because the cell now contains 0, the machine reads 0; so, from instruction 1, the head moves right one space and returns to instruction 1 again. Now the machine reads 0; so it moves right again and returns to instruction 1. Because it reads 1, again the machine writes 0 and goes to instruction 1 where it moves right and goes to 1. Now the head is over a blank; so it moves left one cell, and goes to 2. At instruction 2, the head moves left so long as the tape reads 0. When the head reaches a blank, it moves right one space, back over the word, and stops. So the result is,

(C)

		0	0	0		
		▲				

In the standard case, we begin with a blank tape except for one or more binary "words" where the words are separated by single blank cells, and the machine head is

stored-program computers after the war. *The Imitation Game* (film, 2014) is a dramatization of his life.

[2]Specifications of Turing machines differ somewhat. So, for example, some versions allow instruction quintuples, and allow different symbols on the tape. Nothing about what is computable changes on the different accounts.

over the leftmost cell of the leftmost block. The above example is a simple case of this sort, but also,

(D)

		1	0	1	1		1	1	0	1	0		1		
		▲													

And in the usual case the program halts with the head over the leftmost cell of a single word on the tape. A total function $\mathsf{f}(\vec{\mathsf{x}})$ is *Turing computable* when, beginning with $\vec{\mathsf{x}}$ on the tape in binary digits, the result is $\mathsf{f}(\vec{\mathsf{x}})$. (A Turing machine might calculate the values of a function that is *partial* in the sense that it does not return a value for every input string; we are particularly interested in total functions.) Thus our little program computes $\mathsf{zero(x)}$, beginning with any x and returning the value $\mathsf{0}$.

It will be convenient to require that programs are *dextral* (right-handed), in the sense that (a) in executing a program we never write in a cell to the left of the initial cell, or scan a cell more than one to the left of the initial cell; and (b) when the program halts, the head is over the initial cell and the final result begins in the same cell as the

Binary Numbers

With the proliferation of electronic devices, we are surrounded by applications of the binary number system. Still, it is possible to interact with such devices without understanding how they work! Given their importance, most of us will have been introduced to binary numbers at one time or another. In any case, here is a quick primer.

A standard number system has some *base* whose powers are the places of numbers in that system. So, for example, in the usual base 10,

...	10000	1000	100	10	1
	10^4	10^3	10^2	10^1	10^0

A numeral tells us *how many* of the powers there are in each place. So, for example, $1234 = (1 \times 1000) + (2 \times 100) + (3 \times 10) + (4 \times 1)$. Notice that for base b, the digits are 0 through $b - 1$. And similarly for the binary system whose base is 2 with just digits 0 and 1.

...	16	8	4	2	1
	2^4	2^3	2^2	2^1	2^0

Then, for example, $1101 = (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$—that is, 13. The binary system is particularly convenient in the context of electronic devices (and Turing Machines!) whose "mechanism" conveniently records just the "on/off" states corresponding to 1 and 0.

initial scanned cell. This does not affect what can be computed, but aids in predicting results when Turing programs are combined. Our little program is dextral.

A program to compute $\mathsf{suc}(x)$ is not much more difficult. Let us begin by thinking about what we want the program to do. With a three-digit input word, the desired outputs are,

000	$\Longrightarrow$	001	100	$\Longrightarrow$	101
001	$\Longrightarrow$	010	101	$\Longrightarrow$	110
010	$\Longrightarrow$	011	110	$\Longrightarrow$	111
011	$\Longrightarrow$	100	111	$\Longrightarrow$	1000

Moving from the right of the input word, we want to turn any one to a zero until we can turn a zero (or a blank) to a one. Here is a way to do that:

(F)

$\langle 1, 0, R, 1\rangle$ move to end of word
$\langle 1, 1, R, 1\rangle$
$\langle 1, B, L, 5\rangle$

$\langle 5, 0, 1, 7\rangle$ flip 1 to 0 then 0 or blank to 1
$\langle 5, 1, 0, 6\rangle$
$\langle 5, B, 1, 7\rangle$

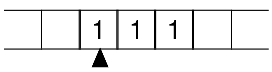
$\langle 6, 0, L, 5\rangle$

$\langle 7, 0, L, 7\rangle$ return to start
$\langle 7, 1, L, 7\rangle$
$\langle 7, B, R, 0\rangle$

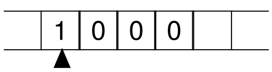
Do not worry about the gap in instruction labels. Nothing so far requires instruction labels be sequential. This program moves the head to the right end of the word; from the right, flips one to zero until it finds a zero or blank; once it has acted on a zero or blank, it returns to the start.

So far, so good. But there is a problem with this program: In the case when the input is, say,

(G)



the output is,



with the first symbol one to the left of the initial position. We turn the first blank to the left of the initial position to a one. So the program is not dextral. The problem is solved by "shifting" the word in the case when it is all ones:

(H)

if solid ones shift right	flip 1 to 0 then 0 to 1
$\langle 1, 0, R, 4\rangle$	$\langle 5, 0, 1, 7\rangle$
$\langle 1, 1, R, 1\rangle$	$\langle 5, 1, 0, 6\rangle$
$\langle 1, B, 1, 2\rangle$	$\langle 5, B, 1, 7\rangle$
$\langle 2, 1, L, 2\rangle$	$\langle 6, 0, L, 5\rangle$
$\langle 2, B, R, 3\rangle$	
	return to start
$\langle 3, 1, B, 3\rangle$	$\langle 7, 0, L, 7\rangle$
$\langle 3, B, R, 4\rangle$	$\langle 7, 1, L, 7\rangle$
	$\langle 7, B, R, 0\rangle$
$\langle 4, 0, R, 4\rangle$	
$\langle 4, 1, R, 4\rangle$	
$\langle 4, B, L, 5\rangle$	

Stages 5, 6, and 7 are as before. This time we test to see if the word is all ones. If not, the program jumps to 4 where it goes to the end, and to the routine from before. If it gets to the end without encountering a zero, it writes a one, returns to the beginning and deletes the initial symbol—so that the entire word is shifted one to the right. Then it goes to instruction 4 so that it goes to the right and works entirely as before. This time the output from tape (G) is,

		1	0	0	0	

as it should be. It is worthwhile to follow the actual operation of this and the previous program on one of the many Turing simulators available on the web (see E14.1).

More complex is a copy program to take an input x and return x.x. This program has four basic elements:

(1) A sort of control section which says what to do, depending on what sort of character we have in the original word. If the character is 0 or 1, write a blank to "mark the spot" and jump to the appropriate copy program. If the character is a blank, jump to the finish.

(2) A program to copy 0; beginning from blank in the original word, move right to the second blank (across the blank between words, and to the blank to be filled); write a 0; move left to the original position, and replace the 0.

(3) Similarly a program to copy 1; beginning from blank in the original word, move right to the second blank; write a 1; move left to the original position, and replace the 1.

(4) And a program to move the head back to the original position when we are done.

Here is a program to do the job:

(I)

(1) *Control*	(2) *Copy* 0	(3) *Copy* 1
$\langle 1, 0, B, 10\rangle$	move from blank	move from blank
$\langle 1, 1, B, 20\rangle$	$\langle 10, B, R, 11\rangle$	$\langle 20, B, R, 21\rangle$
$\langle 1, B, L, 30\rangle$		
	right 2 blanks: 0	right 2 blanks: 1
(4) *Finish*	$\langle 11, 0, R, 11\rangle$	$\langle 21, 0, R, 21\rangle$
start of word	$\langle 11, 1, R, 11\rangle$	$\langle 21, 1, R, 21\rangle$
$\langle 30, 0, L, 30\rangle$	$\langle 11, B, R, 12\rangle$	$\langle 21, B, R, 22\rangle$
$\langle 30, 1, L, 30\rangle$		
$\langle 30, B, R, 0\rangle$	$\langle 12, 0, R, 12\rangle$	$\langle 22, 0, R, 22\rangle$
	$\langle 12, 1, R, 12\rangle$	$\langle 22, 1, R, 22\rangle$
	$\langle 12, B, 0, 13\rangle$	$\langle 22, B, 1, 23\rangle$
	left 2 blanks: 0	left 2 blanks: 1
	$\langle 13, 0, L, 13\rangle$	$\langle 23, 0, L, 23\rangle$
	$\langle 13, 1, L, 13\rangle$	$\langle 23, 1, L, 23\rangle$
	$\langle 13, B, L, 14\rangle$	$\langle 23, B, L, 24\rangle$
	$\langle 14, 0, L, 14\rangle$	$\langle 24, 0, L, 24\rangle$
	$\langle 14, 1, L, 14\rangle$	$\langle 24, 1, L, 24\rangle$
	$\langle 14, B, 0, 15\rangle$	$\langle 24, B, 1, 25\rangle$
	next char: return	next char: return
	$\langle 15, 0, R, 1\rangle$	$\langle 25, 1, R, 1\rangle$

You should be able to follow each stage.

*E14.1. If you have not already done so, install some convenient version of Ruby on your computing platform (compare E12.3). Then obtain the Turing machine simulator from the text website, `https://tonyroyphilosophy.net/symbolic-logic/`. (See the files "running ruby" and "running the simulator" for help.) Study the copy program from the text along with the sample file `suc.rb` from the website. Then, starting with `blank.rb`, create Turing programs to compute the following. It will be best to submit your programs electronically.

a. copy(m). Takes input m and returns m.m. This is a simple implementation of the program from the text.

b. zero(). Hint: Since zero() has no input, it operates only on a blank portion of the tape. This will be the easiest Turing program you ever write.

c. pred(n). Hint: For later applications, it will be helpful to give your function two separate exit paths: One when the input is a string of 0s, returning with the input. In any other case, subtract one. The method simply flips that for successor: From the right, change 0 to 1 until some 1 can be flipped to 0. There is no need to worry about the addition of a possible leading 0 to your result.

d. $\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{z})$. For $\mathsf{x.y.z}$ observe that z might be longer than x and y put together. Here is a way to proceed: z is not longer than x, y, and z put together. So move to the start of the third word; use copy to generate $\mathsf{x.y.z.z}$ then plug spaces so that you have one long first word, $\mathsf{xoyoz.z}$; you can mark the first position of the long word with a blank; then it is a simple matter of running a basic copy routine from right to left, and erasing junk when you are done.

e. Combine your $\mathsf{zero}()$ and $\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{z})$ to form $\mathsf{zero}^2(\mathsf{x}, \mathsf{y}) = \mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{zero}())$. You will want to move past the first two words, run $\mathsf{zero}()$, return to the start, and run idnt^3_3.

Hint: Be sure to comment liberally, or you will never be able to unscramble what you have done! Also, "outline-style" indentation of code can help clarify subordination.

14.1.2 Turing Computable Functions are Recursive

We turn now to showing that the (dextral) Turing computable functions are the same as the recursive functions. This divides into showing that every Turing computable function is recursive, and then that every recursive function is Turing computable. In this section, we show the first. But we begin with the simpler result that there is a recursive enumeration of Turing machines. We shall need this as we go forward, and it will let us compile some important preliminary results along the way.

The method is by now familiar. It will require some work, but we can do it in the same way as we approached formulas and theorems before. Begin by assigning to each symbol a *Gödel Number*.

a.	$g[\mathsf{B}] = 3$	d.	$g[\mathsf{L}] = 9$
b.	$g[\mathsf{0}] = 5$	e.	$g[\mathsf{R}] = 11$
c.	$g[\mathsf{1}] = 7$	f.	$g[\mathsf{q_i}] = 13 + 2i$

For a quadruple, say, $\langle \mathsf{q_1}, \mathsf{B}, \mathsf{L}, \mathsf{q_1} \rangle$, set $g = 2^{15} \times 3^3 \times 5^9 \times 7^{15}$. And for a sequence of quadruples with numbers $g_0, g_1, \ldots, g_n$ the super Gödel number $g_s = 2^{g_0} \times 3^{g_1} \times \cdots \times p_n^{g_n}$. Again, for convenience we frequently refer to the individual symbol codes with angle quotes around the symbol so $\langle \mathsf{B} \rangle = 3$, and to expressions by corner quotes so $\ulcorner \mathsf{B} \urcorner = 2^3$.

Now we define a recursive function and some simple recursive relations,

$$\mathsf{lb}(\mathsf{v}) = \hat{1}\hat{3} + \hat{2}\mathsf{v}$$

$$\textsc{lb}(\mathsf{n}) = (\exists \mathsf{v} \leq \mathsf{n})(\mathsf{n} = \mathsf{lb}(\mathsf{v}))$$

$$\textsc{sym}(\mathsf{n}) = \mathsf{n} = \langle \widehat{\mathsf{B}} \rangle \lor \mathsf{n} = \langle \hat{0} \rangle \lor \mathsf{n} = \langle \hat{1} \rangle$$

$$\textsc{act}(\mathsf{n}) = \textsc{sym}(\mathsf{n}) \lor \mathsf{n} = \langle \widehat{\mathsf{L}} \rangle \lor \mathsf{n} = \langle \widehat{\mathsf{R}} \rangle$$

$$\text{QUAD}(n) = \text{len}(n) = \hat{4} \wedge \text{LB}(\text{exp}(n,\hat{0})) \wedge \text{SYM}(\text{exp}(n,\hat{1})) \wedge \text{ACT}(\text{exp}(n,\hat{2})) \wedge \text{LB}(\text{exp}(n,\hat{3}))$$

$\text{lb}(v)$ is the Gödel number of instruction label v. Then the relations are true when n is the number for an instruction label, a symbol, an action, and a quadruple. In particular, a code for a quadruple numbers a sequence of four symbols of the appropriate sort.

We are now ready to number the Turing machines. For this, adopt a simple modification of our original specification: We have so-far supposed that a Turing machine might lack any given quadruple, say $\langle 3, 1, x, y\rangle$. In case it lacks this quadruple, if the machine reads 1 and is sent to state 3 it simply "hangs" with no place to go. Where q is the largest label in the machine, we now suppose that for any $p \leq q$, if no $\langle p, C, x, y\rangle$ is a member of the machine, the machine is simply supplemented with $\langle p, C, C, p\rangle$. The effect is as before: In this case, there is a place for the machine to go; but if the machine goes to $\langle p, C, C, p\rangle$, it remains in that state, repeating it over and over. In the case of label 0, the states are added to the machine, but serve no function, as the zero label forces halt. Further, we suppose that the quadruples in a Turing machine are taken in order, $\langle 0, 0, x, y\rangle$, $\langle 0, 1, x, y\rangle$, $\langle 0, B, x, y\rangle$, $\langle 1, 0, x, y\rangle$, $\langle 1, 1, x, y\rangle$, $\langle 1, B, x, y\rangle$, $\ldots$, $\langle q, 0, x, y\rangle$, $\langle q, 1, x, y\rangle$, $\langle q, B, x, y\rangle$. So each Turing machine has a unique specification. On this account, a Turing machine halts when it reaches a state of the sort $\langle 0, x, y, z\rangle$. And the ordered specification itself guarantees the functional requirement—that there are no two quadruples with the first values the same and the latter different. So for $\text{TMACH}(n)$,

$$(\exists w < \text{len}(n))(\text{len}(n) = \hat{3} \times (w + \hat{2})) \wedge (\forall v : \hat{3} \times v + \hat{2} < \text{len}(n))(\forall x \leq n)\{$$
$$[x = \text{exp}(n, \hat{3} \times v) \rightarrow (\text{QUAD}(x) \wedge \text{exp}(x,\hat{0}) = \text{lb}(v) \wedge \text{exp}(x,\hat{1}) = \langle\hat{0}\rangle)] \wedge$$
$$[x = \text{exp}(n, \hat{3} \times v + \hat{1}) \rightarrow (\text{QUAD}(x) \wedge \text{exp}(x,\hat{0}) = \text{lb}(v) \wedge \text{exp}(x,\hat{1}) = \langle\hat{1}\rangle)] \wedge$$
$$[x = \text{exp}(n, \hat{3} \times v + \hat{2}) \rightarrow (\text{QUAD}(x) \wedge \text{exp}(x,\hat{0}) = \text{lb}(v) \wedge \text{exp}(x,\hat{1}) = \langle\hat{B}\rangle)]\}$$

Given our modifications, the length of a Turing machine must be a non-zero multiple of three including at least the initial labels zero and one. So for some w, $\text{len}(n) = 3\times(w{+}2)$. Then for each initial label v there are three quadruples; so there are quadruples $3 \times v$, $3 \times v + 1$, and $3 \times v + 2$ taken in the standard order, each with initial label v. Since n is a super Gödel number and each x the number of a quadruple, it is the exponents of x that reveal the instruction label and cell content.

But now it is easy to see,

T14.1. There is a recursive enumeration of the Turing machines. Set,

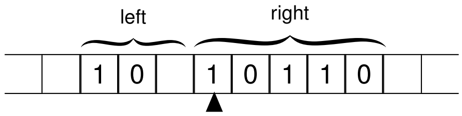
$$\text{mach}(0) = \mu z[\text{TMACH}(z)]$$

$$\text{mach}(Sn) = \mu z[\text{mach}(n) < z \wedge \text{TMACH}(z)]$$

Since $\text{mach}(n)$ is a recursive function from the natural numbers onto the Turing machines, they are recursively enumerable. While this enumeration is recursive, it is not primitive recursive.

Now, as we work toward a demonstration that Turing computable functions are recursive, let us pause for some key ideas. Consider a tape divided as follows:

(J)



We shall code the tape with a pair of numbers—where at any stage the head divides the tape into left and right parts, first a standard code for the right hand side $\ulcorner\mathsf{10110}\urcorner$, and second a code for the left side read from the inside out $\ulcorner\mathsf{B01}\urcorner$. Taken as a pair, these numbers record at once contents of the tape, and the position of the head—which is always over the first digit of the coded right number.

Say a dextral Turing machine computes a total function $\mathsf{f(n) = m}$. Let us suppose that we have recursive functions $\mathsf{encode(a) = b}$ and $\mathsf{decode(b) = a}$ to move between a natural number a and the code b for its binary representation—so when $\mathsf{f(n) = m}$, $\mathsf{encode(n)}$ takes natural number n and returns the code for the initial tape value; and decode takes the code for the final tape value and returns natural number m; so running encode followed by (a coding of) the Turing machine then decode computes the function. Thus we concentrate on the machine itself, and wish to track the status of the Turing machine i given input n for each step j of its operation. In order to track the status of the machine, we shall require functions $\mathsf{left(i, n, j)}$, $\mathsf{right(i, n, j)}$ to record codes of the left and right portions of the tape, and $\mathsf{state(i, n, j)}$ for the current quadruple state of the machine.

First, as we have observed, for any Turing machine, there is a unique quadruple for any instruction label $\mathsf{q_1}$ and content C. Thus $\mathsf{machs(i, k, c)}$ numbers a quadruple as a function of the number of the machine in the enumeration, and Gödel numbers for an initial label and a cell content. Let $\mathsf{machs(i, k, c)}$ be,

$$(\mu \mathsf{y} \leq \mathsf{mach(i)})(\exists \mathsf{v} < \mathsf{len(mach(i))})[\mathsf{y} = \mathsf{exp(mach(i), v)} \wedge \mathsf{exp(y, \widehat{0})} = \mathsf{k} \wedge \mathsf{exp(y, \widehat{1})} = \mathsf{c}]$$

So $\mathsf{machs(i, k, c)}$ returns the number of that quadruple in machine i whose initial label has number k and cell content number c. Since the machine is a total function, there must be a unique state with those values (when k is not an initial label or c not a content the function simply defaults to $\mathsf{mach(i)}$).

In addition, let us adopt a sort of converse to concatenation, $\mathsf{lop(n, a)}$ that "lops" an initial portion of length a off from n.

$$\mathsf{lop(n, a)} = (\mu \mathsf{x} \leq \mathsf{n})(\forall \mathsf{i} < \mathsf{len(n)} \dot{-} \mathsf{a})(\mathsf{exp(x, i)} = \mathsf{exp(n, a + i)})$$

So we want the least x such that its length is the length of n less a, and the exponents of x at any position i are the same as those of n at $\mathsf{a + i}$.

Recall that our Turing machine is to calculate a function $\mathsf{f(n) = m}$. Initial values of $\mathsf{left(i, n, j)}$, $\mathsf{right(i, n, j)}$, and $\mathsf{state(i, n, j)}$ are straightforward.

$$\mathsf{left(i, n, 0)} = \ulcorner\widehat{\mathsf{BB}}\urcorner$$

$$\mathsf{right(i, n, 0)} = \mathsf{encode(n)}$$

$$\mathsf{state(i, n, 0)} = \mathsf{machs(i, lb(\widehat{1}), exp(right(i, n, \widehat{0}), \widehat{0}))}$$

On a dextral machine, the machine never writes to the left of its initial position, and the head never moves more than one position to the left of its initial position; so we simply set the value of the left portion to a couple of blanks. This ensures that there is enough "space" on the left for the machine to operate (and that, for any position of the machine head, there is always a left portion of the tape). The starting right number is just the code of the input to the function. And the initial state value is determined by the instruction label $\mathsf{1}$ and the first value on the tape which is coded by the first exponent of $\mathsf{right(i, n, 0)}$.

For the successor values,

$$\mathsf{left(i, n, Sj)} = \begin{cases} \mathsf{left(i, n, j)} & \text{if} \quad \textsc{sym}(\mathsf{exp(state(i, n, j), \widehat{2})}) \\ \widehat{2}^{\mathsf{exp(right(i,n,j),\widehat{0})}} \star \mathsf{left(i, n, j)} & \text{if} \quad \mathsf{exp(state(i, n, j), \widehat{2})} = \langle\widehat{\mathsf{R}}\rangle \\ \mathsf{lop(left(i, n, j), \widehat{1})} & \text{if} \quad \mathsf{exp(state(i, n, j), \widehat{2})} = \langle\widehat{\mathsf{L}}\rangle \end{cases}$$

If a symbol is written in the current cell, there is no change in the left number. If the head moves to the left or the right, the first value is either appended or deleted, depending on direction. And similarly for $\mathsf{right(i, n, Sj)}$ but with separate clauses for each of the symbols that may be written onto the first position. And now the successor value for state is determined by the Turing machine together with the new label and the value under the head after the current action has been performed.

$$\mathsf{state(i, n, Sj) = machs(i, exp(state(i, n, j), \widehat{3}), exp(right(i, n, Sj), \widehat{0}))}$$

The machine jumps to a new state depending on the label and value on the tape. Observe that we are here proceeding by *simultaneous* recursion, defining multiple functions together. It should be clear enough how this works (see E12.26, page 611).

Let $\mathsf{stop(i, n, j)}$ return the instruction label to which machine i on input n moves after step j. If the machine enters a zero state then it halts; so $\mathsf{stop(i, n, j)}$ takes the value $\mathsf{0}$ just in case machine i with input n stops after step j. Thus,

$$\mathsf{stop(i, n, j)} = (\mu \mathsf{y} \leq \mathsf{len(mach(i))})(\mathsf{exp(state(i, n, j), \widehat{3})} = \mathsf{lb(y)})$$

$\mathsf{exp(state(i, n, j), \widehat{3})}$ is the Gödel number of the new instruction label; $\mathsf{lb(y)}$ is the Gödel number of label y; so $\mathsf{exp(state(i, n, j), \widehat{3}) = lb(y)}$ when y is the new label. So $\mathsf{stop(i, n, j)}$ takes the value $\mathsf{0}$ in case the label y is zero, and the machine halts.[3]

T14.2. Every Turing computable function is a recursive function. Supposing Turing machine i computes a function $\mathsf{f(n)}$,

$$\mathsf{f(n) = decode(right(i, n, \mu j[stop(i, n, j) = \widehat{0}]))}$$

[3]These recursive functions are *defined* for any value of j; but their values after $\mathsf{stop(i, n, j)}$ hits zero do not matter. Supposing that the zero states are filled with the particular instructions $\langle 0, 0, 0, 0\rangle$, $\langle 0, 1, 1, 0\rangle$, $\langle 0, \mathsf{B}, \mathsf{B}, 0\rangle$, the recursive functions will continue to reflect the state of the halted machine.

When a dextral Turing machine stops, the value of right is just the code of its output value m; so if we decode $\mathsf{right(i, n, j)}$ at that stage, we have the value of the function calculated by the Turing machine. Since the Turing computable function is total, there must be some j where the machine is stopped; so the minimization operates on a regular function. Since this function is recursive, the function calculated by Turing machine i is a recursive function.

*E14.2. Find a recursive function to calculate $\mathsf{right(i, n, Sj)}$. Hint: You might find a combination of $\star$ and lop useful for the case when a symbol is written into the first cell.

*E14.3. Find recursive functions to calculate $\mathsf{encode(n)}$ and $\mathsf{decode(m)}$. Hint: You may find it helpful to start with codes reversed so that they read from right to left; then you can find a recursive $\mathsf{rcode(n)}$ that returns the code for n, and flip the result; decode results easily from encode.

E14.4. Suppose a "dual" Turing machine has two tapes, with a machine head for each. Instructions are of the sort $\langle \mathsf{q_i, C_a, C_b, A_a, A_b, q_j} \rangle$ where a and b indicate the relevant tape. Show that every function $\mathsf{f(m, n)}$ that is dual Turing computable is recursive. You may take encode and decode as given, and assume the machine starts with values m and n on tapes a and b and ends with the value on tape a. Hint: Once you set up your sextuples, each label will be associated with nine different possible input combinations.

14.1.3 Recursive Functions are Turing Computable

To complete the demonstration that the recursive functions are identical to the Turing computable functions, we now show that all recursive functions are Turing computable.

T14.3. Every recursive function is Turing computable.

Suppose $\mathsf{f(\vec{x})}$ is a recursive function. Then there is a sequence of recursive functions $\mathsf{f_0, f_1, \ldots, f_n}$ such that $\mathsf{f_n = f}$, where each member is either an initial function or arises from previous members by composition, recursion, or regular minimization. The argument is by induction on this sequence.

Basis: We have already seen that the initial function $\mathsf{suc(x)}$ is Turing computable; and similarly for $\mathsf{zero()}$, and $\mathsf{idnt^j_k}$, as illustrated in E14.1.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, $\mathsf{f_i(\vec{x})}$ is Turing computable.

Show: $\mathsf{f_k}(\vec{\mathsf{x}})$ is Turing computable.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion, or regular minimization. If it is an initial function, then reason as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ are Turing computable. For the simplest case, consider $\mathsf{h}(\mathsf{g}(\mathsf{y}))$: Chain together Turing programs to calculate $\mathsf{g}(\mathsf{y})$ and then $\mathsf{h}(\mathsf{w})$—so the first program operates upon y to calculate $\mathsf{g}(\mathsf{y})$ and the second begins where the first leaves off, operating on the result to calculate $\mathsf{h}(\mathsf{g}(\mathsf{y}))$. A case like $\mathsf{h}(\mathsf{x}, \mathsf{g}(\mathsf{y}), \mathsf{z})$ is more complex insofar as $\mathsf{g}(\mathsf{y})$ may take up a different number of cells from y: it is sufficient to run a copy to get $\mathsf{x.y.z.y}$; then $\mathsf{g}(\mathsf{y})$ to get $\mathsf{x.y.z.g(y)}$; then copy for $\mathsf{x.y.z.g(y).z}$ and a copy that uses the last two numbers to get $\mathsf{x.g(y).z}$. Then you can run h. And similarly in other cases.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ are Turing computable. Recall our little programs from Chapter 12 which begin by using $\mathsf{g}(\vec{\mathsf{x}})$ to find $\mathsf{f}(0)$ and then use $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ repeatedly for y in 0 to $\mathsf{b}-1$ to find the value of $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{b})$ (see, for example, page 571). We shall reason similarly. For a representative case, consider $\mathsf{f}(\mathsf{m}, \mathsf{b})$.

a. Produce a sequence,

$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.m.0.m}$$

This requires a $\mathsf{copypair}(\mathsf{x}, \mathsf{y})$ that takes $\mathsf{m.n}$ and returns $\mathsf{m.n.m.n}$ and $\mathsf{pred}(\mathsf{x})$. Given $\mathsf{m.b}$ on the tape, run $\mathsf{copypair}$ to get $\mathsf{m.b.m.b}$ (and mark the first cell of the first m with a blank). Then loop as follows: if the final b is 0, delete it, go to the previous m, and move on to step (b); otherwise run pred on the final b, move to previous m, run $\mathsf{copypair}$, and loop.

b. Run g on the last block of digits m. This gives,

$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.m.0.f(m,0)}$$

c. Back up to the previous m and run h on the concluding three blocks $\mathsf{m.0.f(m,0)}$. This gives,

$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.f(m,1)}$$

And so forth. Stop when you reach the m with an extra blank (with two blanks in a row). At that stage, we have, $\mathsf{m^*.b.f(m,b)}$. Fill the first blank, run idnt^3_3, and you are done.

Observe that the original $\mathsf{m.b}$ plays no role in the calculation other to serve as the initial template for the series, and then as an end marker on your way back up—there is never a need to apply h to any value greater than $\mathsf{b}-1$ in the calculation of $\mathsf{f}(\mathsf{m}, \mathsf{b})$.

(m) $f_k(\vec{x})$ arises by regular minimization from $g(\vec{x}, y)$. By assumption, $g(\vec{x}, y)$ is Turing computable. For a representative case, suppose we are given m and want $\mu y[g(m, y) = 0]$.

a. Given m, produce m.0.m.0.

b. From a tape of the form m.y.m.y loop as follows: Move to the second m; run g on m.y; this gives m.y.g(m, y); check to see if the result is zero; if it is, run idnt^3_2 and you are done (this is the same as deleting the last zero and running idnt^2_2); if the result is not zero, delete g(m, y) to get m.y; run suc on y; and then a copier to get m.y′.m.y′, and loop. The loop halts when it reaches the value of y for which g has output 0—and there must be some such value if g is regular.

Indct: Any recursive function $f(\vec{x})$ is Turing computable.

And from T14.2 together with T14.3, the Turing computable functions are identical to the recursive functions. It is perhaps an "amazing" coincidence that functions independently defined in these ways should turn out to be identical. And we have here the beginnings of an idea behind Church's thesis which we shall explore in section 14.3.

*E14.5. From exercise E14.1 you should already have Turing programs for suc(x), pred(x), copy(x), and $\mathsf{idnt}^3_3(x, y, z)$. Now produce each of the following, in order, leading up to the recursive addition function. When you require one as part of another simply copy it into the larger file.

a. The function, hplus(x, y, u) = suc(idnt^3_3(x, y, u)). For addition, g(x) = idnt^1_1(x) = x, which requires no action; so we will not worry about that.

b. The function, copypair. Take a.b and return a.b.a.b. One approach is to produce a simple modification of copy that takes a.b and produces a.b.a. Run this program starting at a, and then another copy of it starting at b.

c. The function, cascade. This is the program to produce m.n.m.n – 1.m.n – 2 . . . m.0.m. The key elements are copypair and pred. To prepare for the next stage, you should begin by running copypair and then "damage" the very first m by putting a blank in its first cell. Let the program finish with the head on m at the end.

d. The function, plus(m, n). For this, g is trivial. So from plus(m, 0) = m at the far right of the sequence, back up two words; check to see if there is an extra blank; if so, run idnt^3_3 and you are done; if not, run h(x, y, u) and loop.

There are easier ways to do addition on a Turing machine. The obvious strategy is to put m in a location x and n in a location y; run suc on the value in location x

and then pred on the value in location y; the result appears in x when pred hits zero. The advantage of our approach is that it illustrates (an important case of) the demonstration that a Turing machine can compute any recursive function.

E14.6. Produce each of the following, leading up to a Turing program for the function $\mu\mathsf{y}[\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y})) = \widehat{0}]$, that is the function which returns the least y such that x equals the predecessor of y—such that the characteristic function of $\mathsf{x} = \mathsf{pred}(\mathsf{y})$ returns 0.

a. The function $\mathsf{idnt}^2_2(\mathsf{x}, \mathsf{y})$. This can be a simple modification of idnt^3_3.

b. The function $\mathsf{ch}(\mathsf{x} = \mathsf{y})$, which returns 0 when $\mathsf{x} = \mathsf{y}$ and otherwise 1. This is, of course, a recursive function. But you can get it more efficiently and more directly. To compare numbers, you have to worry about leading zeros that might make equivalent numbers physically distinct. One approach is to check whether one or both of x and y is all zeros: if both, they are equivalent; if one, they are not; otherwise, run pred on both, and loop.

c. The function $\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y}))$. This is a simple case of composition.

d. The function $\mu\mathsf{y}[\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y})) = \widehat{0}]$, by the routine discussed in the text.

Of course, for any number except 0, this is nothing but a long-winded equivalent to $\mathsf{suc}(\mathsf{x})$. The point, however, is to apply the algorithm for regular minimization, and so to work through the last stage of the demonstration that recursive functions are Turing computable.

14.2 Essential Results

In Chapter 12 essential results were built on Carnap's equivalence and the diagonal lemma. This time, we depend on a *halting problem* with special application to Turing machines. Once we have established the halting problem, results like ones from before follow in short order.

14.2.1 Halting

A Turing machine is a set of quadruples. Things are arranged so that Turing machines do not "hang" in the sense that they reach a state with no applicable instruction. But a Turing machine may go into a loop or routine from which it never emerges. That is, a Turing machine may or may not *halt* in a finite number of steps. So for example, this machine never stops.

$\langle 1, 0, 0, 1\rangle$
$\langle 1, 1, 1, 1\rangle$
$\langle 1, \mathsf{B}, \mathsf{B}, 1\rangle$

For any input it simply repeats forever. This raises the question whether there is a general way to *tell* whether Turing machines halt when started on a given input. This is an issue of significance for computing theory. And, as we shall see, the answer has consequences beyond computing.

The problem divides into narrower "self-halting" and broader "general-halting" versions. First, the self-halting problem: By T14.1 there is an enumeration of the Turing machines. Consider an enumeration of Turing machines, $\Pi_0, \Pi_1, \ldots$ and an array as follows:

$$
\text{(K)} \quad \begin{array}{c|cccc}
 & 0 & 1 & 2 & \cdots \\
\hline
\Pi_0 & \boldsymbol{\Pi}_0(0) & \Pi_0(1) & \Pi_0(2) & \\
\Pi_1 & \Pi_1(0) & \boldsymbol{\Pi}_1(1) & \Pi_1(2) & \\
\Pi_2 & \Pi_2(0) & \Pi_2(1) & \boldsymbol{\Pi}_2(2) & \\
\vdots & & & &
\end{array}
$$

We run Π_0 on inputs $0, 1, \ldots$; Π_1 on $0, 1, \ldots$; and so forth. Now ask whether there is a Turing program (that is, a recursive function) to decide in general whether Π_i halts when applied to its own number in the enumeration—a program $H(i)$ such that $H(i) = 0$ if $\Pi_i(i)$ halts, and $H(i) = 1$ if $\Pi_i(i)$ does not halt.

T14.4. There is no Turing machine $H(i)$ such that $H(i) = 0$ if $\Pi_i(i)$ halts and $H(i) = 1$ if it does not.

Suppose otherwise. That is, suppose there is a halting machine $H(i)$ where for any $\Pi_i(i)$, $H(i) = 0$ if $\Pi_i(i)$ halts and $H(i) = 1$ if it does not. Chain this program into a simple looping machine $\Lambda(j)$ defined as follows:

$\langle q, 0, 0, q\rangle$
$\langle q, 1, 1, 0\rangle$

So when $j = 0$, Λ goes into an infinite loop, remaining in state q forever; when $j = 1$, Λ halts gracefully with output 1. Let the combination of H and Λ be $\Delta(i)$; so $\Delta(i)$ calculates $\Lambda(H(i))$. On our assumption that there is a Turing machine $H(i)$, the machine Δ must appear in the enumeration of Turing machines with some number d.

But this is impossible. Consider $\Delta(d)$ and suppose $\Delta(d)$ halts; since Δ halts on input d, the halting machine, $H(d) = 0$; and with this input, Λ goes into the infinite loop; so the composition $\Lambda(H(d))$ does not halt; and this is just to say $\Delta(d)$ does not halt. Reject the assumption: $\Delta(d)$ does not halt. But since $\Delta(d)$ does not halt, the halting machine $H(d) = 1$; and with this input, Λ halts gracefully with output 1; so the composition $\Lambda(H(d))$ halts; and this is just to say $\Delta(d)$ halts. Reject the original assumption, there is no machine $H(i)$ which says whether an arbitrary $\Pi_i(i)$ halts.

For this argument, it is important that H is a component of Δ. Information about whether Δ halts gives information about the behavior of H, and information about the behavior of H, about whether Δ halts.

The more general question is whether there is a machine to decide for any Π_{i} and n whether $\Pi_{\mathsf{i}}(\mathsf{n})$ halts. But it is immediate that if there is no Turing machine to decide the more narrow self-halting problem, there is no Turing machine to decide this more general version.

T14.5. There is no Turing machine $\mathsf{H(i,n)}$ such that $\mathsf{H(i,n) = 0}$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H(i,n) = 1}$ if it does not.

Suppose otherwise. That is, suppose there is a halting machine $\mathsf{H(i,n)}$ where for any $\Pi_{\mathsf{i}}(\mathsf{n})$, $\mathsf{H(i,n) = 0}$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H(i,n) = 1}$ if it does not. Chain this program after a copier $\mathsf{K(n)}$ which takes input n and gives $\mathsf{n.n}$. The combination $\mathsf{H(K(i))}$ decides whether $\Pi_{\mathsf{i}}(\mathsf{i})$ halts. This is impossible; reject the assumption: There is no such Turing machine $\mathsf{H(i,n)}$.

And when combined with T14.3 according to which every recursive function is Turing computable, these theorems which tell us that no Turing program is sufficient to solve the halting problem, yield the result that no recursive function solves the halting problem: If a function is recursive, then it is Turing computable; and since it is Turing computable, it does not solve the halting problem. Observe that we may be able to decide in particular cases whether a program halts. No doubt you have been able to do so in exercises! What we have shown is that there is no perfectly general recursive method to decide whether $\Pi_{\mathsf{i}}(\mathsf{n})$ halts.

E14.7. Say a function is *μ-recursive* just in case it satisfies the conditions for the recursive functions but without the regularity requirement for minimization; so $\mu\mathsf{y}[\mathsf{g}(\vec{\mathsf{x}},\mathsf{y}) = \widehat{0}]$ returns the least y such that both $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y}) = 0$ and for every $\mathsf{z} < \mathsf{y}$, $\mathsf{g}(\vec{\mathsf{x}},\mathsf{z}) > 0$ *if there is one* and otherwise is undefined. Where every recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is *total* in the sense that it returns a value for every $\vec{\mathsf{x}}$, some μ-recursive functions are *partial* insofar as there may be values of $\vec{\mathsf{x}}$ for which they return no value (as occurs when minimization is applied to a $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$ that never evaluates to zero); so all the recursive functions are μ-recursive, but some μ-recursive functions are not recursive. Suppose that the μ-recursive functions can be numbered and that there is a μ-recursive function $\mathsf{emrfnc(i)}$ to enumerate them; so $\mathsf{emrfnc(i)}$ returns the Gödel number of the i^{th} function in the enumeration. (You will have occasion to produce this function in a later exercise.) Show that there is no μ-recursive function $\mathsf{def(i)}$ such that $\mathsf{def(i)} = 0$ if $\mathsf{f_i(i)}$ is defined and $\mathsf{def(i)} = 1$ if $\mathsf{f_i(i)}$ is undefined. Hint: Consider $\mathsf{delta(i)} = \mu\mathsf{y}[\mathsf{def(i)} = \mathsf{y} \wedge \mathsf{y} = \widehat{1}]$ as applied to its own number in the enumeration. We might think of this as the *definition problem*.

14.2.2 The Decision Problem

Recall our demonstration from section 12.5.3 that if Q is consistent and $\mathscr{L}$ extends $\mathscr{L}_{NT}$ then no recursive relation identifies the $\mathscr{L}$-theorems of predicate logic. With the identity between the recursive functions and the Turing computable functions, this is the same as the result that no Turing computable function identifies the $\mathscr{L}$-theorems of predicate logic. We are now in a position to obtain a related result directly, by means of the halting problem. Recall from section 12.5.2 that a theory T is *ω-inconsistent* iff for some $\mathscr{P}(x)$, T proves each $\mathscr{P}(\overline{\mathsf{m}})$ but also proves $\sim\forall x\mathscr{P}(x)$. Equivalently, T is ω-inconsistent iff T proves each $\sim\mathscr{P}(\overline{\mathsf{m}})$ but also proves $\exists x\mathscr{P}(x)$. We show,

T14.6. If Q is ω-consistent and $\mathscr{L}$ includes $\mathscr{L}_{NT}$, then no Turing computable function $\mathsf{prvpl(n)}$ is such that $\mathsf{prvpl(n)} = 0$ just in case n numbers an $\mathscr{L}$-theorem of predicate logic.

Suppose otherwise, that Q is ω-consistent, $\mathscr{L}$ includes $\mathscr{L}_{NT}$, and some Turing computable $\mathsf{prvpl(n)} = 0$ just in case n numbers an $\mathscr{L}$-theorem of predicate logic. Consider our recursive function $\mathsf{stop(i, n, j)}$ which takes the value $\mathsf{0}$ if $\Pi_\mathsf{i}(\mathsf{n})$ halts after step j. Since it is recursive, stop is captured by some $Stop(i, n, j, z)$ so that,

(i) If $\Pi_\mathsf{i}(\mathsf{i})$ halts after step j, Q $\vdash Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$

(ii) If $\Pi_\mathsf{i}(\mathsf{i})$ never halts, Q $\vdash \sim Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$ for any j

Let $\mathscr{H}(i) = \exists z Stop(i, i, z, \emptyset)$. Then if $\Pi_\mathsf{i}(\mathsf{i})$ halts, there is some j such that Q $\vdash Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$; so Q $\vdash \mathscr{H}(\overline{\mathsf{i}})$. And if $\Pi_\mathsf{i}(\mathsf{i})$ never halts, for every j, Q $\vdash$ $\sim Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$; and since Q is ω-consistent, Q $\nvdash \mathscr{H}(\overline{\mathsf{i}})$. So $\Pi_\mathsf{i}(\mathsf{i})$ halts iff Q $\vdash \mathscr{H}(\overline{\mathsf{i}})$.

The axioms Q1–Q7 of Q are equivalent to their universal closures; with the axioms in this form, let $\mathscr{Q}$ be the conjunction of Q1–Q7; since Q1–Q7 are particular sentences, $\mathscr{Q}$ is a particular sentence; so Q $\vdash \mathscr{H}(\overline{\mathsf{i}})$ iff $\mathscr{Q} \vdash \mathscr{H}(\overline{\mathsf{i}})$; by DT iff $\vdash \mathscr{Q} \to \mathscr{H}(\overline{\mathsf{i}})$. So,

$$\vdash \mathscr{Q} \to \mathscr{H}(\overline{\mathsf{i}}) \quad \text{iff} \quad \Pi_\mathsf{i}(\mathsf{i}) \text{ halts}$$

Let $\mathsf{q} = \ulcorner\mathscr{Q}\urcorner$ and $\mathsf{h(i)} = \mathsf{formsub}(\widehat{\ulcorner\mathscr{H}(i)\urcorner}, \widehat{\ulcorner i\urcorner}, \mathsf{num(i)})$—so $\mathsf{h(i)}$ is the number of $\mathscr{H}(\overline{\mathsf{i}})$. Then $\mathsf{prvpl(cnd(q, h(i)))}$ takes the value $\mathsf{0}$ iff $\mathscr{Q} \to \mathscr{H}(\overline{\mathsf{i}})$ is a theorem, iff $\Pi_\mathsf{i}(\mathsf{i})$ halts. So prvpl solves the halting problem. This is impossible; reject the assumption: If Q is ω-consistent, then there is no Turing computable function that returns the value zero just for numbers of $\mathscr{L}$-theorems of predicate logic.

Further, as observed on page 632, Q is ω-consistent; it follows that no Turing computable function $\mathsf{prvpl(n)}$ takes the value $\mathsf{0}$ just in case n numbers an $\mathscr{L}$-theorem of predicate logic. And, of course, this is equivalent to the result that no recursive function returns zero just for $\mathscr{L}$-theorems of predicate logic.[4]

[4]This argument, and the parallel one in Chapter 12 have the advantage of simplicity. However, this result that no recursive function returns zero just for $\mathscr{L}$-theorems of predicate logic need not be

E14.8. Return again to the μ-recursive functions from E14.7. Take as given that (i) relative to the enumeration of μ-recursive functions, there is a μ-recursive $\mathsf{murec(i, n)}$ that returns the value a iff $\mathsf{f_i(n) = a}$ and (ii) for any μ-recursive function f there is a Σ_1 formula $\mathscr{F}$ so that if $\langle\langle \mathsf{m_1 \ldots m_n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$ then $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$ and if $\langle\langle \mathsf{m_1 \ldots m_n}\rangle, \mathsf{a}\rangle \notin \mathsf{f}$ then $\mathsf{N}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$. Given this, use the definition problem from E14.7 to show that if Q is sound, then no μ-recursive function $\mathsf{muprvpl(n)}$ is such that $\mathsf{muprvpl(n) = 0}$ just in case n numbers a theorem of predicate logic.

Hint: Given $\mathsf{murec(i, n)}$ consider the $Murec(i, n, y)$ such that if $\mathsf{f_i(n) = a}$ then $\mathsf{N}[Murec(\overline{\mathsf{i}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$, and if $\mathsf{f_i(n) \neq a}$ then $\mathsf{N}[\sim Murec(\overline{\mathsf{i}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$; as an analog to $\mathscr{H}(i)$, let $Defined(i) = \exists z Murec(i, i, z)$; show that $\mathsf{f_i(i)}$ is defined iff $\mathsf{Q} \vdash Defined(\overline{\mathsf{i}})$.

14.2.3 Ĩncompleteness Again

In Chapter 12 and Chapter 13 we saw ĩncompleteness results in different forms: one from consistency and capture, and another from s̃oundness and expression. We are positioned to see the result again in both forms.

Semantic Version

In T12.19 we showed that the theorems of a recursively axiomatized formal theory are recursively enumerable, and used this to show that PRVT is recursive for consistent and negation c̃omplete theories. This contrasts with the corollary to T12.21 according to which PRVT is not recursive for consistent theories extending Q. An ĩncompleteness result follows. This time we shall be able to contrast the enumeration of theorems with an enumeration of *truths*. The idea is to show that a Turing machine $\Pi_t(\mathsf{i})$ to enumerate the truths of $\mathscr{L}_{NT}$ solves the halting problem, and so that there is no such Turing machine. Thus the enumeration of theorems is not an enumeration of all truths.

T14.7. The set of all truths in a language including $\mathscr{L}_{NT}$ is not recursively enumerable.

Consider a language $\mathscr{L}$ including $\mathscr{L}_{NT}$ and again our recursive function $\mathsf{stop(i, n, j)}$; since it is recursive, it is expressed by some $Stop(i, n, j, z)$; set $\mathscr{H}(i) = \exists z Stop(i, i, z, \emptyset)$ and let $\mathsf{h(i)} = \mathsf{formsub}(\ulcorner\widehat{\mathscr{H}(i)}\urcorner, \ulcorner\widehat{i}\urcorner, \mathsf{num(i)})$—so $\mathsf{h(i)}$ is the number of $\mathscr{H}(\overline{\mathsf{i}})$.

Suppose $\mathsf{N}[\mathscr{H}(\overline{\mathsf{i}})] = \mathsf{T}$; then for some m, $\mathsf{N}[Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{m}}, \emptyset)] = \mathsf{T}$; so by expression, $\langle\langle \mathsf{i, i, m}\rangle, \mathsf{0}\rangle \in \mathsf{stop}$; so $\Pi_\mathsf{i}(\mathsf{i})$ stops. Suppose $\mathsf{N}[\mathscr{H}(\overline{\mathsf{i}})] \neq \mathsf{T}$; then for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{N}[Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{m}}, \emptyset)] \neq \mathsf{T}$; so by expression, $\langle\langle \mathsf{i, i, m}\rangle, \mathsf{0}\rangle \notin \mathsf{stop}$; so $\Pi_\mathsf{i}(\mathsf{i})$ never stops. So (*a*) $\mathsf{N}[\mathscr{H}(\overline{\mathsf{i}})] = \mathsf{T}$ iff $\Pi_\mathsf{i}(\mathsf{i})$ halts.

---

conditional on the consistency (or ω-consistency) of Q. For an illuminating and direct demonstration that no Turing machine solves the decision problem, see Chapter 11 of Boolos, Burgess, and Jeffrey, *Computability and Logic*. See also page 633, note 21.

Now suppose some $\Pi_t(i)$ enumerates the truths of $\mathcal{L}$, halting with output 0 if $h(i)$ appears in the enumeration—if $N[\mathcal{H}(\bar{i})] = T$, and halting with output 1 if $til(h(i))$ appears in the enumeration—if $N[\sim\mathcal{H}(\bar{i})] = T$. Exactly one of $\mathcal{H}(\bar{i})$ or $\sim\mathcal{H}(\bar{i})$ is true; so the number for one of them will eventually turn up insofar as Π_t enumerates all the truths of $\mathcal{L}_{NT}$. So (*b*) $\Pi_t(i)$ halts with output 0 iff $N[\mathcal{H}(\bar{i})] = T$.

By (*a*) and (*b*) $\Pi_t(i)$ halts with output 0 iff $\Pi_i(i)$ halts; so $\Pi_t(i)$ solves the halting problem. This is impossible; reject the assumption: there is no such Turing machine. And since no Turing machine enumerates the truths of $\mathcal{L}$, no recursive function enumerates the truths of $\mathcal{L}$.

This theorem, together with T12.19 which tells us that if T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable, puts us in a position to obtain an ĩncompleteness result mirroring T12.17 and T13.3.

T14.8. If T is a recursively axiomatized s̃ound theory whose language includes $\mathcal{L}_{NT}$, then there is a sentence $\mathcal{P}$ such that $T \nvdash \mathcal{P}$ and $T \nvdash \sim\mathcal{P}$.

Suppose T is a recursively axiomatized s̃ound theory whose language $\mathcal{L}$ includes $\mathcal{L}_{NT}$. By T12.19, there is a recursive enumeration of the theorems of T, and since T is s̃ound, all of the theorems in the enumeration are true. But by T14.7, there is no recursive enumeration of all the truths of $\mathcal{L}$; so the enumeration of theorems is not an enumeration of all truths; so some true $\mathcal{P}$ is not among the theorems of T; from this, $\mathcal{P}^u$ is true but not among the theorems of T. And since $\mathcal{P}^u$ is true, $\sim\mathcal{P}^u$ is not true; and since T is s̃ound, neither is $\sim\mathcal{P}^u$ among the theorems of T. So $T \nvdash \mathcal{P}^u$ and $T \nvdash \sim\mathcal{P}^u$.

This ĩncompleteness result requires the *s̃oundness* of T, where s̃oundness is more than mere consistency. But it requires only that the language include $\mathcal{L}_{NT}$ and so have the power to *express* recursive functions—where this leaves to the side a requirement that T extends Q, and so be able to capture recursive functions.

Syntactic Version

By related reasoning, we can obtain the other sort of ĩncompleteness result as well. Thus we have a theorem like T12.18 and T13.4. Let T be a recursively axiomatized theory extending Q, and once again consider $stop(i, n, j)$; since $stop$ is recursive and T extends Q, $stop$ is captured in T by some $Stop(i, n, j, z)$; let $\mathcal{H}(i) = \exists z Stop(i, i, z, \emptyset)$ and $h(i) = formsub(\widehat{\ulcorner\mathcal{H}(i)\urcorner}, \widehat{\ulcorner i \urcorner}, num(i))$; so $h(i)$ is the number of $\mathcal{H}(\bar{i})$. Consider a Turing machine $\Pi_s(i)$ which tests whether successive values of m number a proof of $\sim\mathcal{H}(\bar{i})$, halting if some m numbers a proof and otherwise continuing forever—so $\Pi_s(i)$ evaluates $\textsc{prft}(m, til(h(i)))$ for successive values of m; since T is a recursively axiomatized theory, this is a recursive relation so that there must be some such Turing machine. We can think of $\Pi_s(i)$ as seeking a proof that $\Pi_i(i)$ does not halt.

T14.9. Suppose T is a recursively axiomatized theory extending Q; then if T is consistent $T \nvdash \mathscr{H}(\overline{\mathsf{s}})$, and if T is ω-consistent, $T \nvdash \sim\mathscr{H}(\overline{\mathsf{s}})$.

Suppose T is a recursively axiomatized theory extending Q, and let $\mathscr{H}(i)$ and $\Pi_{\mathsf{s}}(\mathsf{i})$ be as above.

Suppose T is consistent and $\Pi_{\mathsf{s}}(\mathsf{s})$ halts. By its definition, $\Pi_{\mathsf{s}}(\mathsf{i})$ halts just in case some m numbers a proof of $\sim\mathscr{H}(\overline{\mathsf{i}})$; since $\Pi_{\mathsf{s}}(\mathsf{s})$ halts, then, there is some m such that $\textsc{prft}(\mathsf{m}, \mathsf{til}(\mathsf{h}(\mathsf{s})))$; so $T \vdash \sim\mathscr{H}(\overline{\mathsf{s}})$. But if $\Pi_{\mathsf{s}}(\mathsf{s})$ halts, for some m, $\langle\langle \mathsf{s}, \mathsf{s}, \mathsf{m}\rangle, 0\rangle \in \mathsf{stop}$; so by capture, $T \vdash \mathit{Stop}(\overline{\mathsf{s}}, \overline{\mathsf{s}}, \overline{\mathsf{m}}, \emptyset)$; so $T \vdash \exists z \mathit{Stop}(\overline{\mathsf{s}}, \overline{\mathsf{s}}, z, \emptyset)$, which is to say, $T \vdash \mathscr{H}(\overline{\mathsf{s}})$; and since T is consistent, $T \nvdash \sim\mathscr{H}(\overline{\mathsf{s}})$. This is impossible; reject the assumption: $(*)$ if T is consistent, then $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt.

(i) Suppose T is consistent and $T \vdash \sim\mathscr{H}(\overline{\mathsf{s}})$; then for some m, $\textsc{prft}(\mathsf{m}, \mathsf{til}(\mathsf{h}(\mathsf{s})))$; so by its definition, $\Pi_{\mathsf{s}}(\mathsf{s})$ halts. But since T is consistent, by $(*)$ $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt. Reject the assumption: $T \nvdash \sim\mathscr{H}(\overline{\mathsf{s}})$.

(ii) Suppose T is ω-consistent and $T \vdash \mathscr{H}(\overline{\mathsf{s}})$; then $T \vdash \exists z \mathit{Stop}(\overline{\mathsf{s}}, \overline{\mathsf{s}}, z, \emptyset)$. But since T is ω-consistent, it is consistent; so by $(*)$ $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt; so for any m, $\langle\langle \mathsf{s}, \mathsf{s}, \mathsf{m}\rangle, 0\rangle \notin \mathsf{stop}$; and by capture, for any m, $T \vdash \sim\mathit{Stop}(\overline{\mathsf{s}}, \overline{\mathsf{s}}, \overline{\mathsf{m}}, \emptyset)$; so by ω-consistency, $T \nvdash \exists z \mathit{Stop}(\overline{\mathsf{s}}, \overline{\mathsf{s}}, z, \emptyset)$. This is impossible, $T \nvdash \mathscr{H}(\overline{\mathsf{s}})$

Observe that T14.8 is an existential result—there is some $\mathscr{P}$ such that neither it nor its negation is provable—while T14.9 finds a particular sentence $\mathscr{H}(\overline{\mathsf{s}})$ such that $T \nvdash \mathscr{H}(\overline{\mathsf{s}})$ and $T \nvdash \sim\mathscr{H}(\overline{\mathsf{s}})$. So these theorems reflect a difference between ĩncompleteness results of Chapter 12 and Chapter 13. Given the difference between sentences $\mathscr{G}$ and $\mathscr{H}(\overline{\mathsf{s}})$, again, T14.9 is roughly the form in which Gödel first proved the ĩncompleteness of arithmetic. We shall leave the matter here—although, as we saw from chapters 12 and 13, it is possible to drop the requirement of ω-consistency for the simple result that no consistent, recursively axiomatized theory extending Q is negation c̃omplete.

E14.9. Use the definition problem for μ-recursive functions to show that there is no recursive enumeration of the set of truths of $\mathscr{L}_{NT}$. Use this result to show that if T is a recursively axiomatized sound theory whose language includes $\mathscr{L}_{NT}$, then T is negation ĩncomplete.

Hint: Return to $\mathsf{murec}(\mathsf{i}, \mathsf{n})$, $\mathit{Murec}(i, n, y)$, and $\mathit{Defined}(i)$ from E14.8, along with $\mathsf{defined}(\mathsf{i}) = \mathsf{formsub}(\ulcorner\overbrace{\mathit{Defined}(i)}\urcorner, \ulcorner\widehat{i}\urcorner, \mathsf{num}(\mathsf{i}))$. Suppose there is an enumeration $\mathsf{entruth}(\mathsf{n})$ of the truths of $\mathscr{L}_{NT}$; and let $\textsc{enumdef}(\mathsf{i})$ be the equality $\mathsf{entruth}[\mu \mathsf{y}(\mathsf{entruth}(\mathsf{y}) = \mathsf{defined}(\mathsf{i}) \vee \mathsf{entruth}(\mathsf{y}) = \mathsf{til}(\mathsf{defined}(\mathsf{i}))] = \mathsf{defined}(\mathsf{i})$. Then its characteristic function, $\mathsf{chenumdef}$ is 0 just in case $\mathsf{N}[\mathit{Defined}(\overline{\mathsf{i}})] = \mathsf{T}$.

14.3 Church's Thesis

We have attained a number of negative results, as T14.6 that if Q is ω-consistent then no Turing computable function $\mathsf{prvpl}(\mathsf{n})$ returns zero just for numbers of theorems of predicate logic, and from T14.7 that no Turing machine enumerates the truths of $\mathscr{L}_{\text{NT}}$. These are interesting. But, one might very well think, if no Turing machine computes a function, then we ought simply to compute the function some *other* way. So the significance of our negative results is magnified if the Turing computable functions are, in some sense, the *only* computable functions. If in some important sense the Turing computable functions are the only computable functions, and no Turing machine computes a function, then in the relevant sense the function is not computable. Thus Church's Thesis:

CT The total numerical functions that are effectively computable by some algorithmic method are just the recursive functions.

We want to be clear first, on the *content* of this thesis, and once we know what it says, on reasons for thinking that it is true.

14.3.1 The Content of Church's thesis

Church's thesis makes a claim about "total numerical functions that are effectively computable by an algorithmic method." Original motivations are from the simple routines we learn in grade school for addition, multiplication, and the like. These effectively compute total numerical functions by an algorithmic method. By themselves, such methods are of interest. However, we mean to include the sorts of methods contemporary computing devices can execute. These are of considerable interest as well. Let us take up the different elements of the proposal in turn.

First, as always, a numerical function is *total* iff it is defined on the entire numerical domain. Arbitrary functions on a finite domain may be finitely specified by listing their members, and then computed by simple lookup. This was our approach with simple, but arbitrary, functions from Chapter 4. The question of computability becomes interesting when domains are not finite (and from methods like those in the Chapter 2 countability reference, a function on an infinite domain is always comparable to one that is total). So Church's thesis is a thesis about the computability of total functions.

A function is *effectively computable* iff there is a method for finding its output for any given input value(s). Correspondingly, a property or relation is *effectively decidable* iff its characteristic function is effectively computable. So methods for addition and multiplication are adequate to calculate the value of the function for any inputs. Or consider a Turing machine programmed to enumerate the theorems of T, stopping with output $\mathsf{0}$ if it reaches (the number for) $\mathcal{P}$, and output 1 if it reaches $\sim\mathcal{P}$ (for the sake of this example, restrict attention just to theorems that are sentences). Then if T is a consistent recursively axiomatized and negation c̃omplete theory, this is an effective method for deciding the theorems of T. If $\mathscr{P}$ is a theorem, it eventually shows up in the enumeration, and the Turing machine stops with output 0. If $\mathscr{P}$ is not a theorem, $\sim\mathscr{P}$ is a theorem, so $\sim\mathscr{P}$ eventually shows up in the enumeration, and the machine stops with output 1. This was the idea behind T12.20. But if T is not negation c̃omplete, this is not an effective method for deciding theorems of T. If $\mathscr{P}$ is a theorem, it eventually shows up in the enumeration, and the machine stops with output 0. But if $\mathscr{P}$ is not a theorem and T is not negation c̃omplete, $\sim\mathscr{P}$ might also fail to be a theorem. In this case, the machine continues forever, and does not stop with output 1; so for some inputs, this method does not find the value of the characteristic function, and we have not described an *effective* method for deciding the theorems of this T.

From the start, we may agree that there is some uncertainty about the notion of an *algorithmic* method; so, for example, different texts offer somewhat different definitions. However, as we did for logical validity and soundness in Chapter 1, we shall take a particular account as a technical definition—partly as clarified in the definition (AC) and examples that follow. Difficulties to the side, there does seem to be a relevant core notion: For our purposes an *algorithmic* method is a finitely constrained rule-based procedure (rote, if you will).[5]

There is some vagueness in how much "processing" is allowed in following a rule. So, "write down the value of $\mathsf{f(n)}$" will not do a as a rule for arbitrary $\mathsf{f(n)}$; and, less dramatically, an algorithm for multiplication does not typically include instructions for required additions. However, we may take it that if a function is Turing computable, then the function is algorithmically computable. A Turing machine operates by a finitely constrained rule-based method. Again, standard methods for addition and multiplication are examples of algorithmic procedures. Truth table construction is another example of a method that proceeds by rote in this way. Given the basic tables for the operators, one simply follows the rules to complete the tables and determine validity—and one could program a Turing machine to perform the same task. Thus validity in sentential logic is effectively decidable by an algorithmic method. In contrast, derivations are not an algorithmic method. The strategies are helpful! But, at least in complex cases, there may come a stage where insight or something like lucky guessing is required. And at such a stage, you are not following any rules by rote, and so not following any specific algorithm to reach your result.

And algorithmic methods operate under finite constraints. In general, we shall not worry about how large these constraints may be, so long as they remain finite. So, for example, searching (numbers for) the theorems of a negation c̃omplete theory is likely to involve massive integers, to consume huge amounts of memory, and enormous amounts of time. All the same, the search terminates while the integers, memory, and time remain finite. Or consider truth table construction. If this is to be

[5]We have no intention of engaging Wittgenstenian concerns about following a rule. See, for example, Kripke, *Wittgenstein on Rules and Private Language*.

an effective method for determining validity, it should return a result for any sentence. But for any $n > 0$ there are sentences with that many atomic sentences (for example, $A_1 \wedge A_2 \wedge \ldots \wedge A_n$), so the corresponding table requires 2^n rows. This number may be arbitrarily large—and a table may require more paper or memory than are in the entire universe. But, in every case, the limit is finite. So, for our purposes, the methods qualify as algorithmic. Contrast a device, which we may call "god's mind," that stores all the theorems of an ĩncomplete theory sorted in order of their Gödel numbers. Since the theory is ĩncomplete, a search to find one of $\mathcal{P}$ or $\sim\mathcal{P}$ among the theorems is not an effective method to decide if $\mathcal{P}$ is a theorem. But, in this case, it is sufficient to search up to the Gödel number of $\mathcal{P}$ to see if that sentence is in the database: if it is in the database then $\mathcal{P}$ is a theorem, if it is not in the database then $\mathcal{P}$ is not a theorem. It is not our intent to deny the existence of god, or that one might very well solve mathematical problems by prayer (though this might not go over very well on examinations which require that you show your work)! But, insofar as a device requires infinite memory or the like, it will not for our purposes count as an algorithmic method.

Or consider again a Turing machine programmed to enumerate the theorems of T, stopping with output 0 if it reaches (the number for) $\mathcal{P}$, but continuing forever if $\mathcal{P}$ does not appear. One might suppose the information that $\mathcal{P}$ is not a theorem is contained already in the fact that *the machine never halts*, and that god or some being with an infinite perspective might very well extract this information from the machine. Perhaps so. But this method is not algorithmic just because it requires the infinite perspective. Still, there are interesting attempts to attain the effect of this latter machine without appeals to god. Consider, first, "Zeno's machine." As before, the machine enumerates theorems, this time flashing a light if $\mathcal{P}$ appears in the list. However, for some finite time t (say 60 seconds), this machine takes its first step in $t/2$ seconds, its second step in $t/4$ seconds, and for any n, step n in $t/2^n$ seconds. But the sum $t/2 + t/4 + \cdots = t$, and the Turing machine runs through all of infinitely many steps in time t.[6] So start the machine. If the light flashes before t seconds elapse, $\mathcal{P}$ is a theorem. If t elapses, the machine has run through all of infinitely many steps, so if the light does not flash, $\mathcal{P}$ is not a theorem.

One might object that this proposal reduces to a tautology of the sort, "If such-and-such (impossible) circumstances obtain, then the theorems are decidable." Great, but who cares? However we should not reject the general strategy out of hand. From even a very basic introduction to special relativity, one is exposed to time dilation effects (for a simple case see the time dilation reference on page 775). General relativity allows a related effect. Where special relativity applies just to reference frames moving at constant velocity relative to one another, general relativity allows accelerated frames. And it is at least consistent with the laws of general relativity for one frame to have

[6] An infinite sum is defined to be the *limit* of its partial sums. Compare E8.16 and, for discussion of Zeno, Sainsbury, *Paradoxes*, Chapter 1.

an infinite elapsed time, while another's time is finite.[7] So, for a Malament-Hogarth (MH) machine, put a Turing machine in the one frame and an observer in the other. The Turing machine operates in the usual way in its frame enumerating the theorems forever. If $\mathscr{P}$ is a theorem, it sends a signal back to the observer's frame that is received within the finite interval. From the observer's perspective, this machine runs through infinitely many operations. So if a signal is received in the finite interval, $\mathscr{P}$ is a theorem. If no signal is received in the finite interval, then $\mathscr{P}$ is not a theorem. There is considerable room for debate about whether such a machine is physically possible. But, even if physically realized, it is not *algorithmic*. For we require that an algorithmic method terminates in a finite number of steps.

Church's thesis is thus that the total numerical functions that are effectively computable by some algorithmic method are the same as the recursive functions. Suppose we obtain a negative result that some function is not algorithmically computable. The result does not apply to god's mind or Zeno's machine. Still, even as contemporary devices increase speed, memory, and efficiency, their capacities remain finite. Thus the negative result remains of considerable interest: So long as a routine follows definite rules, no (finite) amount of parallel processing, high-speed memory, nanotechnology, and so forth is going to make a difference—the function remains uncomputable.

14.3.2 The Basis for Church's thesis

It is widely accepted that Church's thesis is true, but also that it is not susceptible to *proof*. We shall return to the question of proof. There are perhaps three sorts of reasons that have led philosophers, computer scientists, and logicians to think it is true. (i) A number of independently defined notions plausibly associated with computability converge on the recursive functions. (ii) No plausible counterexamples—algorithmically computable functions not recursive—have come to light. And (iii) there is a sort of rationale from the nature of an algorithm. This last may verge on, or amount to, demonstration of Church's thesis.

Independent Definitions

Historically, Church's thesis arose out of early attempts to specify the computable functions. The thesis emerged as it was recognized that a number of such attempts, independently proposed, converge on the recursive functions. So, for example, the recursive functions and the Turing computable functions are independently defined. We have seen that the Turing computable functions are the same as the recursive functions.

[7] Students with the requisite math and physics background might be interested in Hogarth, "Does General Relativity Allow an Observer To View an Eternity In a Finite Time?" See also Earman and Norton, "Forever is a Day," and for the same content, Chapter 4 of Earman, *Bangs, Crunches, Whimpers, and Shrieks* (but with additional, though still difficult, setup in earlier chapters of the text).

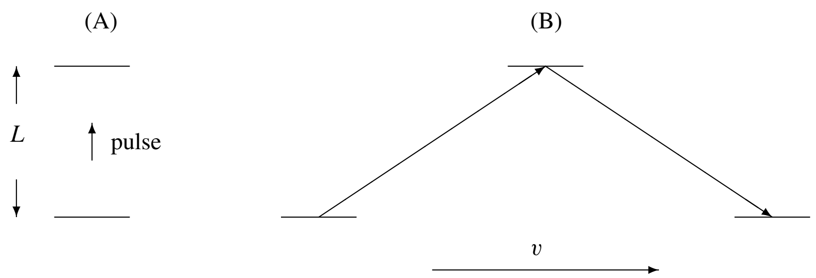
Simple Time Dilation

It is natural to think that, just as a wave in water approaches a boat faster when the boat is moving toward it than when the boat is moving away, so light would approach an observer faster when she is moving toward it, and more slowly when she is moving away. But this is not so. The 1887 Michelson-Morley experiment (and many others) verify that the speed of light has the *same* value for all observers. Special relativity takes as foundational:

1. The laws of physics may be expressed in equations having the same form in all frames of reference moving at constant velocity with respect to one another.

2. The speed of light in free space has the same value for all observers, regardless of their state of motion.

These principles have many counterintuitive consequences. Here is one: Consider a clock which consists of a pulse of light bouncing between two mirrors separated by distance L as in (A) below. Where c is the constant speed of light, the time between ticks is the distance traveled by the pulse divided by its speed L/c.



Now consider the same clock as observed from a reference frame relative to which it is in motion, as in (B). The speed of light remains c (instead of being increased, as one might expect, by the addition of the horizontal component to its velocity). But the distance traveled between ticks is greater than L, so the time between ticks is greater than L/c—which is to say the clock ticks more slowly from the perspective of the second frame.

One might wonder what happens if this clock is rotated 90 degrees so that the pulse is bouncing parallel to the direction of motion, or what would happen if time were measured by a pendulum clock. But within a frame, everything is coordinated according to the usual laws: On special relativity, there are coordinated changes to length, mass, and the like so that the effect is robust. As observed from a reference frame relative to which the frame is in motion, time, mass, and length are distorted together. For further discussion, consult any textbook on introductory modern physics.

And we are in a position to close another loop. From T12.14, the recursive functions are captured by recursively axiomatized theories extending Q. But the recursive functions are total functions; and consistent recursively axiomatized theories extending Q are among the recursively axiomatized theories extending Q. So the recursive functions are total functions captured by consistent recursively axiomatized theories extending Q. But also,

T14.10. Every total function that can be captured by a consistent recursively axiomatized theory extending Q is recursive.

Suppose a total function $\mathsf{f(m)}$ can be captured in a consistent recursively axiomatized theory T extending Q; then there is some $\mathcal{F}(x, y)$ such that if $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{f}$, then $T \vdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ and with T12.6 if $\langle \mathsf{m}, \mathsf{n} \rangle \notin \mathsf{f}$ then $T \vdash \sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. Suppose $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{f}$; since f is a function, any $\mathsf{k} \neq \mathsf{n}$ is such that $\langle \mathsf{m}, \mathsf{k} \rangle \notin \mathsf{f}$; so $T \vdash \sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{k}})$; and since T is consistent, $T \nvdash \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{k}})$. So for any m, (i) there are some n and a such that $\langle \mathsf{m}, \mathsf{n} \rangle \in \mathsf{f}$ and $\textsc{prft}(\mathsf{a}, \ulcorner \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}) \urcorner)$; and (ii) for $\mathsf{k} \neq \mathsf{n}$ there is no a such that $\textsc{prft}(\mathsf{a}, \ulcorner \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{k}}) \urcorner)$.

Intuitively, we can find the value of $\mathsf{f(m)}$ by searching the theorems until we find one of the sort $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; and from this extract the value n. More formally: First, for the number of $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$,

$$\mathsf{numf(m, n) = formsub[formsub(}\widehat{\ulcorner \mathcal{F}(x, y) \urcorner}, \widehat{\ulcorner x \urcorner}, \mathsf{num(m)}), \widehat{\ulcorner y \urcorner}, \mathsf{num(n)}]$$

So $\mathsf{numf(m, n)}$ gives the Gödel number of $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ as a function of m and n. By (loose) analogy with code from Chapter 12 (page 629),

$$\mathsf{codef(m)} = \mu \mathsf{z}[\mathsf{len(z)} = \widehat{2} \wedge \textsc{prft}(\mathsf{exp(z}, \widehat{0}), \mathsf{numf(m, exp(z}, \widehat{1})))]$$

So $\mathsf{codef(m)}$ is of the sort $2^{\mathsf{a}} \times 3^{\mathsf{n}}$, where a numbers a proof of $\mathsf{numf(m, n)}$, that is, of $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. The minimization is well-defined since there always are such an a and n. And there is only one n for which there is a proof of $\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. So,

$$\mathsf{f(m) = exp(codef(m)}, \widehat{1})$$

n is easily recovered from codef: The exponents of $\mathsf{codef(m)}$ are a and n; and $\mathsf{exp(codef(m)}, \widehat{1})$ returns the n. And since $\mathsf{exp(codef(m)}, \widehat{1})$ is a recursive function, $\mathsf{f(m)}$ is a recursive function.

We use the $\mathcal{F}(x, y)$ that captures $\mathsf{f(m)}$ to generate a recursive specification for $\mathsf{f(m)}$. So every total function that can be captured by a consistent recursively axiomatized theory extending Q is recursive. So a total function is captured in a consistent recursively axiomatized theory extending Q iff it is recursive. And increasing the power of a deductive system from Q to PA and beyond does not extend the range of captured functions. So the recursive functions, Turing computable functions, and total functions captured by a consistent recursively axiomatized theory extending Q are the same.[8]

E14.10. Given that $Plus(x, y, z)$ captures $\mathsf{plus}(\mathsf{m}, \mathsf{n})$ in a consistent recursively axiomatized theory extending Q, apply the method of T14.10 to show that plus is recursive.

Failure of Counterexamples

Another reason for accepting Church's thesis is the failure to find counterexamples. This may be very much the same point as before: When we set out to define a notion of computability, or compute a function, what we end up with are recursive functions, rather than something other. We have seen that many standard computable functions are in fact recursive. Of course, god's mind, Zeno's machine, an MH machine, or the like might compute a non-recursive function. Perhaps there are such devices. However, on our account, they are not algorithmic. What we do not seem to have are algorithmic methods for computing non-recursive functions.

But also in this category of reasons to accept Church's thesis is the failure of a natural strategy for showing that Church's thesis is false. Suppose one were to propose that the *primitive* recursive functions are all the recursive functions, and so that regular minimization is redundant (perhaps you have had this very idea). Here is a way to see this hypothesis false:

Observe that, as in the recursive enumeration reference on the following page, the primitive recursive functions are recursively enumerable. Consider an enumeration of the primitive recursive functions of one free variable and an array as follows:

$$\text{(L)}\qquad \begin{array}{c|cccc} & 0 & 1 & 2 & \cdots \\ \hline \mathsf{f}_0 & \mathbf{f_0(0)} & \mathsf{f}_0(1) & \mathsf{f}_0(2) & \\ \mathsf{f}_1 & \mathsf{f}_1(0) & \mathbf{f_1(1)} & \mathsf{f}_1(2) & \\ \mathsf{f}_2 & \mathsf{f}_2(0) & \mathsf{f}_2(1) & \mathbf{f_2(2)} & \\ \vdots & & & & \end{array}$$

And consider the function $\mathsf{d}(\mathsf{n}) = \mathsf{f}_\mathsf{n}(\mathsf{n}) + \hat{1}$. This function is recursive. For any n: (i) run the enumeration to find f_n; (ii) run f_n to find $\mathsf{f}_\mathsf{n}(\mathsf{n})$; (iii) add one. Since each step is recursive, the whole is recursive. But $\mathsf{d}(\mathsf{n})$ is not primitive recursive: $\mathsf{d}(0) \neq \mathsf{f}_0(0)$; $\mathsf{d}(1) \neq \mathsf{f}_1(1)$; and in general, $\mathsf{d}(\mathsf{n}) \neq \mathsf{f}_\mathsf{n}(\mathsf{n})$; so d is not identical to any of the primitive recursive functions. So there are recursive functions that are not primitive recursive. And since recursive functions are (Turing-)computable, there are computable functions that are not primitive recursive.

[8] And there are more. Church himself was originally impressed by an equivalence between his *lambda-definable* functions and the recursive functions. As additional examples, Markov algorithms are discussed in Mendelson, *Introduction to Mathematical Logic*, §5.5; abacus machines in Boolos, Burgess, and Jeffrey, *Computability and Logic*, §5; see below for discussion of the Kolmogorov-Uspenskii machine.

Enumerating Primitive Recusive Functions

Introduce a language $\mathcal{L}_{\text{R}}$ for an alternative representation of the recursive functions. The syntax of this language is developed in the usual way. Symbols are Z^0, S^1, I_i^n, $Comp^n$, and Rec^n with parentheses and comma. Then,

RL (b) If $\mathcal{P}^n$ is Z^0, S^1, or I_i^n (for $1 \le i \le n$) then $\mathcal{P}^n$ is a *formula*.

(c) If $\mathcal{P}^m$ and $\mathcal{Q}_1^n \dots \mathcal{Q}_m^n$ are formulas, then $Comp^n(\mathcal{P}^m, \mathcal{Q}_1^n \dots \mathcal{Q}_m^n)$ is a *formula*.

(r) If $\mathcal{G}^n$ and $\mathcal{H}^{n+2}$ are formulas, then $Rec^{n+1}(\mathcal{G}^n, \mathcal{H}^{n+2})$ is a *formula*.

(CL) Any formula can be formed by repeated application of these rules.

For (c) we allow the superscript on a $\mathcal{Q}_i$ to be 0 so long as at least some are n. These expressions may be exhibited on trees in the usual way. So, for example, you should be able to see that $Rec^2(I_1^1, Comp^3(S^1, I_3^3))$ is a formula.

And expressions of this language may be interpreted so that each $\mathcal{P}^n$ represents a recursive function that applies to n objects. Say $\vec{\mathsf{x}}$ is $\mathsf{x}_1 \dots \mathsf{x}_n$.

IR (z) $\mathsf{I}[Z^0] = \mathsf{zero}()$

(s) $\mathsf{I}[S^1](\mathsf{x}) = \mathsf{suc}(\mathsf{x})$

(i) $\mathsf{I}[I_i^n](\vec{\mathsf{x}}) = \mathsf{idnt}_i^n(\vec{\mathsf{x}})$

(c) $\mathsf{I}[Comp^n(\mathcal{P}^m, \mathcal{Q}_1^n \dots \mathcal{Q}_m^n)](\vec{\mathsf{x}}) = \mathsf{I}[\mathcal{P}^m](\mathsf{I}[\mathcal{Q}_1^n](\vec{\mathsf{x}}) \dots \mathsf{I}[\mathcal{Q}_m^n](\vec{\mathsf{x}}))$

(r) $\mathsf{I}[Rec^{n+1}(\mathcal{G}^n, \mathcal{H}^{n+2})](\vec{\mathsf{x}}, 0) = \mathsf{I}[\mathcal{G}^n](\vec{\mathsf{x}})$
$\mathsf{I}[Rec^{n+1}(\mathcal{G}^n, \mathcal{H}^{n+2})](\vec{\mathsf{x}}, \mathsf{Sy}) = \mathsf{I}[\mathcal{H}^{n+2}](\vec{\mathsf{x}}, \mathsf{y}, \mathsf{I}[Rec^{n+1}(\mathcal{G}^n, \mathcal{H}^{n+2})](\vec{\mathsf{x}}, \mathsf{y}))$

You should be able to construct a tree parallel to one that shows $\mathcal{P}$ is a formula, to show its interpretation. Thus, for example, $\mathsf{I}[Rec^2(I_1^1, Comp^3(S^1, I_3^3))](\mathsf{x}, \mathsf{y}) = \mathsf{plus}(\mathsf{x}, \mathsf{y})$, where $\mathsf{I}[Rec^2(I_1^1, Comp^3(S^1, I_3^3))](\mathsf{x}, 0) = \mathsf{idnt}_1^1(\mathsf{x})$ and $\mathsf{I}[Rec^2(I_1^1, Comp^3(S^1, I_3^3))](\mathsf{x}, \mathsf{Sy}) = \mathsf{suc}(\mathsf{idnt}_3^3(\mathsf{x}, \mathsf{y}, \mathsf{plus}(\mathsf{x}, \mathsf{y})))$. Again, for case (c), $\mathcal{Q}(\vec{\mathsf{x}})$ may have $\vec{x}$ empty when the superscript on $\mathcal{Q}$ is 0.*

Now a recursive enumeration of the primitive recursive functions is straightforward. From their interpretation, an enumeration of the formulas is an enumeration of the primitive recursive functions: Assign numbers to the symbols and formulas of $\mathcal{L}_{\text{R}}$; find a recursive $\mathsf{RLWFF}(\mathsf{n})$ true of numbers for formulas; and enumerate,

$$\mathsf{eprfnc}(0) = \mu\mathsf{z}[\mathsf{RLWFF}(\mathsf{z})]$$
$$\mathsf{eprfnc}(\mathsf{Sn}) = \mu\mathsf{z}[\mathsf{eprfnc}(\mathsf{n}) < \mathsf{z} \wedge \mathsf{RLWFF}(\mathsf{z})]$$

So there is a recursive enumeration of the primitive recursive functions. Observe that the enumeration of the primitive recursive functions is based entirely on *syntactical* considerations from the formulas of our language.

---

*Observe that we apply a *generalized* version of composition on which $\mathsf{I}[\mathcal{Q}_1^n](\vec{\mathsf{x}}) \dots \mathsf{I}[\mathcal{Q}_m^n](\vec{\mathsf{x}})$ are substituted respectively for the variables of $\mathsf{I}[\mathcal{P}^m]$. Clearly, a generalized composition results from multiple applications of our familiar singular form. And singular composition can be seen as an instance of the generalized form: Say we have $\mathcal{P}^m(\vec{x}, w, \vec{z})$ and $\mathcal{Q}^b(\vec{y})$ and want $\mathcal{R}^n(\vec{x}, \vec{y}, \vec{z}) = \mathcal{P}(\vec{x}, \mathcal{Q}(\vec{y}), \vec{z})$. Suppose indexes on the n variables of $\mathcal{R}$ are $x_1 \dots x_a$, $y_1 \dots y_b$, $z_1 \dots z_c$ where y-indexes may or may not overlap x- and z-indexes. If $b = 0$ so that $\vec{y}$ is empty, then $Comp^n(\mathcal{P}^m, I_{x_1}^n \dots I_{x_a}^n, \mathcal{Q}^0, I_{z_1}^n \dots I_{z_c}^n)$ will do. Otherwise take, $Comp^n(\mathcal{P}^m, I_{x_1}^n \dots I_{x_a}^n, Comp^n(\mathcal{Q}^b, I_{y_1}^n \dots I_{y_b}^n), I_{z_1}^n \dots I_{z_c}^n)$.

It is natural to think that a related argument would show that not all computable functions are recursive: Enumerate the recursive functions, and consider the diagonal function $\mathsf{d(n)} = \mathsf{f_n(n)} + 1$; if the enumeration is computable, then this function is computable but not among the recursive functions; so there are computable functions not recursive. Observe, however, that this enumeration of the recursive functions cannot itself be recursive. As described in the recursive enumeration box, it is an entirely "grammatical" matter to enumerate the primitive recursive functions. But there is no parallel method for the recursive fuctions. This is clear already by the halting and definition problems (for the latter see E14.7)—there is no recursive way to say in general whether a function is regular, and so to identify functions as recursive. But we may make the point by another diagonal argument (here applied to Turing machines).

> Suppose there is a recursive enumeration of Turing machines to compute recursive functions (of one free variable) and consider an array as follows:
>
> $$\text{(M)}\quad \begin{array}{c|cccc} & 0 & 1 & 2 & \cdots \\ \hline \Pi_0 & \boldsymbol{\Pi_0(0)} & \Pi_0(1) & \Pi_0(2) & \\ \Pi_1 & \Pi_1(0) & \boldsymbol{\Pi_1(1)} & \Pi_1(2) & \\ \Pi_2 & \Pi_2(0) & \Pi_2(1) & \boldsymbol{\Pi_2(2)} & \\ \vdots & & & & \end{array}$$
>
> With modifications appropriate to this enumeration, by reasoning from T14.2 each $\Pi_{\mathsf{n}}(\mathsf{n})$ computes a recursive $\mathsf{f(n)} = \mathsf{decode(right(n, n, \mu j[stop(n, n, j)} \boldsymbol{=} \widehat{\mathsf{0}}]))$; since $\mathsf{f(n)}$ is recursive $\mathsf{f(n)} \boldsymbol{+} \widehat{\mathsf{1}}$ is recursive and computed by some $\Delta(\mathsf{n})$; $\Delta(\mathsf{n})$ is a Turing program of one free variable; so $\Delta(\mathsf{n})$ appears in the enumeration of Turing programs. But this is impossible: $\Delta(0) \neq \Pi_0(0)$; $\Delta(1) \neq \Pi_1(1)$; and in general $\Delta(\mathsf{n}) \neq \Pi_{\mathsf{n}}(\mathsf{n})$. Reject the assumption: there is no recursive enumeration of Turing machines to compute recursive functions.

And since there is no recursive enumeration of Turing machines to compute recursive functions, there is no recursive enumeration of the recursive functions.[9]

Even so, we could "diagonalize out" of the recursive functions given a *computable* enumeration of the recursive functions—a computable enumeration would let us compute a function not on the list, and so show that the recursive functions are not the same as the computable functions. Our demonstration that there is no recursive enumeration of the recursive functions does not show that there is no computable enumeration. But it does show that our strategy for finding a counterexample to Church's thesis requires that which it is attempting to show: a computable function (to

[9] From T14.1 there is a recursive enumeration of all the Turing machines; but not every Turing machine computes a total function—and the recursive enumeration of Turing machines does not convert to a recursive enumeration of the recursive functions.

do the enumeration) that is not recursive. So we are blocked from the proposed means of finding a computable function that is not a recursive function. So this attempt to find a counterexample to Church's thesis fails.[10]

*E14.11. From the recursive enumeration box, (i) Construct a tree to show that $Rec^2(I^1_1, Comp^3(S^1, I^3_3))$ is a formula of $\mathscr{L}_R$. Then for each $\mathscr{P}^i$ in your tree, give its interpretation as applied to i variables. At the bottom you may let $Plus^2$ abbreviate $Rec^2(I^1_1, Comp^3(S^1, I^3_3))$, and plus^2 abbreviate $\mathsf{I}[Plus^2]$. (ii) And similarly to show that $Rec^2(Comp^1(I^2_2, I^1_1, Z^0), Comp^3(Plus^2, I^3_3, I^3_1))$ is a formula of $\mathscr{L}_R$ whose interpretation is times.

*E14.12. (i) Assign numbers to expressions of $\mathscr{L}_R$ and produce the relation RLWFF to complete the demonstration that there is an enumeration of primitive recursive functions. (ii) Extend the demonstration that there is an enumeration of primitive recursive functions to an enumeration emrfnc of μ-recursive functions (as from E14.7). Hints: Take section 10.3.2 as a model for assigning numbers to symbols with superscripts and/or subscripts. For an RLSEQ like FORMSEQ, you will need to know the number of PLACES for members of the sequence (always contained in the first symbol). Also, for *Comp*, you will want a number for a sequence of m prior formulas each of which has $\mathsf{0}$ or n places.

The Nature of an Algorithm

There are also reasons for Church's thesis from the very nature of an algorithm.[11] Perhaps the "received wisdom" with respect to Church's thesis is as follows:

> The reason why Church's [Thesis] is called a *thesis* is that it has not been rigorously proved and, in this sense, it is something like a "working hypothesis." Its plausibility can be attested inductively—this time not in the sense of mathematical induction, but "on the basis of particular confirming cases." The Thesis is corroborated by the number of intuitively computable functions commonly used by mathematicians, which can be defined within recursion theory. But Church's Thesis is believed by many to be destined to *remain* a thesis. The reason lies, again, in the fact that the notion of effectively computable function is a merely intuitive

[10] S. Kleene, who was among the founders of recursion theory (and author of the classic text, *Introduction to Metamathematics*), reports that "When Church proposed this thesis [around 1933], I sat down to disprove it by diagonalizing out of the class of the λ-definable functions. But, quickly realizing that the diagonalization cannot be done effectively, I became overnight a supporter of the thesis." ("Origins of Recursive Function Theory," page 59). Again, the λ-definable functons are equivalent to the recursive functions (see note 8 on page 777).

[11] Material in this section is developed from Smith, *An Introduction to Gödel's Theorems*, Chapter 45; and Smith, "Squeezing Arguments"; along with Kolmogorov and Uspenskii, "On the Definition of an Algorithm." See also Black, "Proving Church's Thesis."

> and somewhat fuzzy one. It is quite difficult to produce a completely rigorous proof of the equivalence between intuitively computable and recursive functions, precisely because one of the sides of the equivalence is not well-defined.[12]

There are a couple of themes in this passage. First, that Church's thesis is typically accepted on grounds of the sort we have already considered. Fair enough. But second that it is not, and perhaps cannot, be proved. The idea seems to be that the recursive functions are a precise mathematically defined class, while the algorithmically computable functions are not. Thus there is no hope of a demonstrable equivalence between the two.

But we should be careful. Granted: If we start with an inchoate notion of computable function that includes, at once, calculations with pencil and paper, calculations on the latest and greatest supercomputer, and calculations on Zeno's machine, there will be no saying whether the computable functions definitely are, or are not, identical to the Turing computable functions. But this is not the notion with which we are working. We have a relatively refined technical account of algorithmic computability. Of course, it is not yet a *mathematical* definition. But neither are our Chapter 1 accounts of logical validity and soundness; yet we have been able to show in T9.1 that any argument that is quantificationally valid (in our mathematical sense) is logically valid. And similarly, the whole translation project of Chapter 5 assumes the possibility of moving between ordinary and mathematical notions. It is at least possible that an informally defined predicate might pick out a precise object. The question is whether we can "translate" the notion of an algorithm to formal terms.

So let us turn to the hard work of considering whether there is an argument for accepting Church's thesis. A natural first suggestion is that the step-by-step and finite nature of any algorithm is always within the reach of, or reflected by, some Turing program or recursive function, so that the algorithmically computable functions are inevitably recursively computable.[13] Already, this may amount to a consideration or reason in favor of accepting the thesis. In Chapter 45 of his *An Introduction to Gödel's Theorems*, Peter Smith advances a proposal according to which such considerations amount to proof.

Smith's overall strategy involves "squeezing" algorithmic computability between a pair of mathematically precise notions. Even if a condition C (say, "being a tall person") is informally defined, it might remain that there is some completely precise sufficient condition S (being over seven feet tall), such that anything that is S is C, and perfectly precise necessary condition N (being over five feet tall) such that anything that is C is N. So,

$$S \implies C \implies N$$

[12] Berto, *There's Something About Gödel*, pages 76–77. Compare Rogers, *Mathematical Logic and Formalized Theories*, page 216, and Shoenfield, *Mathematical Logic*, pages 119-121.

[13] This idea is contained already in the foundational papers of Church, "An Unsolvable Problem," and Turing, "On Computable Numbers."

If it should also happen that N implies S, then the loop is closed, so that,

$$S \iff C \iff N$$

And the target condition C is equivalent to (squeezed between) the precise necessary and sufficient conditions. Of course, in our simple example, N does not imply S: being over five feet tall does not imply being over seven feet tall.

For Church's thesis, we already have that Turing computability is sufficient for algorithmic computability. So what is required is some necessary condition so that,

$$T \implies A \implies N$$

Turing computability implies algorithmic computability and algorithmic computability implies the necessary condition. Church's thesis follows if, in addition, N implies Turing computability. As it turns out, we shall be able to specify a condition N which (mathematically) implies T. Then translation connects T to A and A to N.

We take as given that Turing computability implies algorithmic computability. Given this, the argument has three stages: (i) there are some necessary features of an algorithm, such that any algorithm has those features; (ii) any routine with those features is embodied in a modified Kolmogorov-Uspenskii (MKU) machine; (iii) every function that is MKU computable is recursive, and so Turing computable.

$$\boxed{\textit{Necessary features}} \longrightarrow \boxed{\textit{MKU computability}} \longrightarrow \boxed{\textit{Turing computability}}$$

The result is that MKU computability works as the precise condition N in the squeezing argument: T implies A, A implies N, and N implies T. So T iff A iff N, and Church's thesis is established.

Perhaps the following are necessary conditions on any algorithm, so that any algorithm satisfies the conditions. If, additionally, we hold that any routine which satisfies the constraints is an algorithm, then the conditions are necessary and sufficient—so we may see them as an extension or sharpening of our initial more sketchy account. At this stage, though, the important requirement is that any algorithm satisfies the conditions.

AC (1) There is some *space* consisting of a finite array of "cells" which may stand in some relations $R_0, R_1, \ldots, R_a$ and contain some entities $s_0, s_1, \ldots, s_b$.

(2) At every stage, there is some finite "active" portion of the space upon which the algorithm operates.

(3) The body of the algorithm includes finitely many instructions for (rote) modification of the active space depending on its character, and for jumping to the next set of instructions.

(4) There is some finite initial setup, and result after a finite number of steps.

So this sets up an algorithm abstractly described.[14] So far, an excellent manual might specify an algorithm for installing a computer or assembling a piece of IKEA furniture. But we are specially interested in algorithms to compute arithmetical functions. Thus the calculation of an arithmetical function $\mathsf{f}(\vec{\mathsf{x}}) = \mathsf{y}$ somehow takes $\vec{\mathsf{x}}$ as an input, and gives a way to read off the value of y after a finite number of steps.

Observe that the finite constraints on the space, relations, and objects in AC(1) are a consequence of the other conditions: Beginning with a finite initial setup, including finitely many cells standing in finitely many relations and filled with finitely many objects, then modifying finite portions of the space finitely many times, all we are going to get are finitely many cells, standing in finitely many relations, filled with finitely many objects.

Just as there are infinitely many integers, each individually finite, so our account of an algorithm permits infinitely many different active areas, each individually finite. Continuing with the analogy to integers, it is natural to think that an algorithm could contain instructions of the sort "if a is odd write 1, and otherwise 0." This instruction has application to infinitely many integers, and might be represented as a set (function),

$$\{\langle 0, 0\rangle, \langle 1, 1\rangle, \langle 2, 0\rangle, \langle 3, 1\rangle, \ldots\}$$

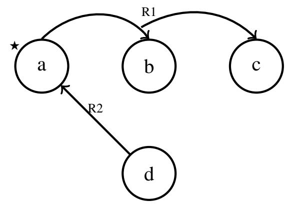
with infinitely many members. One might therefore think that an algorithm should permit infinitely many instructions (one for each pair) to accommodate the infinitely many inputs. But consider how the instruction to write 1 iff a is odd is actually implemented: Given an input, we do not apply an infinite "lookup table" to find the result; rather we apply a rule, dividing by 2 to see if there is a remainder. As for a Turing machine, such a rule is implemented by finitely many instructions. If an instruction does require an infinite lookup table, then it is not algorithmic just because it requires the infinite table. Notice that this example illustrates the point that finitely many instructions cannot include specifications for all the infinitely many possible arrangements a space may have.

All the same, algorithms have wide compass. On the face of it, given their extreme simplicity, it is not obvious that Turing machines compute every algorithmically computable function. But a related device, the MKU machine (modified from Kolmogorov and Uspenskii, "On the Definition of an Algorithm") purports to implement conditions along these lines.

MKU (1) There is a *dataspace* consisting of some cells $c_0, c_1, \ldots, c_a$ which may stand in relations $R_0, R_1, \ldots, R_b$ and contain symbols $s_0, s_1, \ldots, s_c$. In simple cases, we may think of such arrangements graphically as follows:

[14] Smith seems to grant that some such conditions are necessary, even though some method may satisfy the conditions yet fail to count as an algorithm. Perhaps this is because he is impressed by the initial examples of routines implemented by human agents with relatively limited computing power. This is not a problem for his squeezing argument, since the corresponding recursive function may yet be computable by some other method which satisfies more narrow constraints—for example, by a Turing machine.

(N)

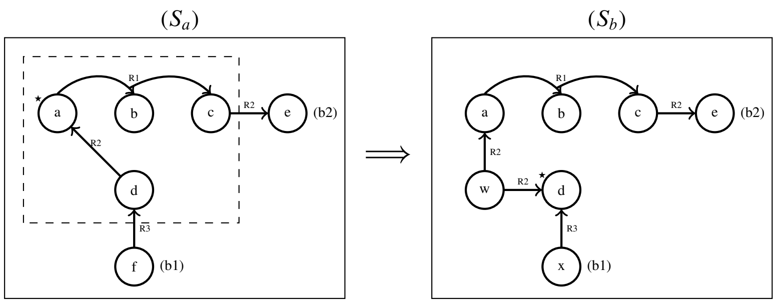


In this case there are four cells with contents a, b, c, d—though there is no requirement that a cell contain just a single symbol. There are relations R1 and R2; R2 is a binary relation and R1 tertiary; each such relation constitutes an *edge*.

(2) Among the one-place relations is an *origin* property O such that exactly one cell has it—as indicated by $\star$ above. Then, depending on the instructions, for some n the active area includes all cells on paths $\leq n$ edges from the origin. From (N), cells other than the origin are all one edge from the origin cell.

(3) Instructions are finitely many quadruples of the sort $\langle q_i, S_a, S_b, q_j \rangle$ where q_i and q_j are instruction labels; S_a describes an active area; and S_b a state with which the active area is to be replaced. Associate each cell in S_a with the least number of edges between it and the origin; let n be the greatest such integer in S_a; this n remains the same in every quadruple with label q_i, though the value of n may vary as q_i varies. Again, instructions are a function in the sense that no instruction has $\langle q_i, S_a \rangle$ the same but $\langle S_b, q_j \rangle$ different.[15] We may see S_a and S_b as follows:

(O)



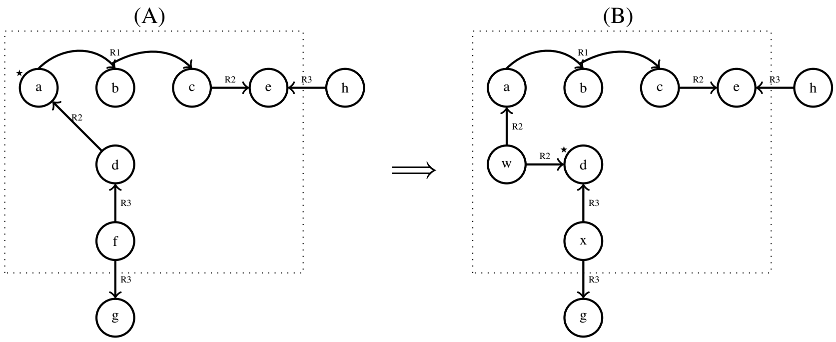
In this case $n = 2$. The active area S_a is replaced by the configuration S_b. The concentric rectangles indicate "boundary" cells, n edges from the origin, which may themselves be related to cells not part of the active area; the replacing area retains boundary cells, and the substitution of one area for another their relations from and to cells outside the active area.

[15]States are the *same* when they map onto the same dataspace. Observe that some S_a and S'_a might both map to a given dataspace in case one is included in the other. But the consistency requirement is satisfied when n is constant: for consistency, it is sufficient to require that so long as $n(q_i, S_a)$ is a constant, there are no instructions with $\langle q_i, S_a \rangle$ the same but $\langle S_b, q_j \rangle$ different.

(4) There is some finite initial setup, and some means of reading off the final value of the function (for different relation and symbol sets, these may be different). We think of the origin cell as the "machine head," where an algorithm always begins with an instruction label $q_i = 1$ and terminates when $q_i = 0$.

Thus an MKU machine is a significant generalization of a Turing machine. We allow arbitrarily many symbols. And the dataspace is no longer a tape with cells in a fixed linear relation, but a space with cells in arbitrary relations which may themselves be modified by the program. Instructions respond to, and modify, not just individual cells, but arbitrarily large areas of the dataspace. Still, it remains that an instruction q_i is of the sort, if S_a perform action A and go to instruction q_j. So, the instruction (O) might be applied to get,

(P)



As indicated by the dotted line, the dataspace (A) has an active area of the sort required in instruction (O); so the active area is replaced according to the instruction for the resultant space (B). The example is arbitrary. But that is the point: The machine allows arbitrary rote modifications of a dataspace.

Insofar as the MKU machine is a generalization of a Turing machine, it is clear enough that Turing computable functions are MKU computable. But for us the important point is that every MKU computable function is recursive and so Turing computable.

T14.11. Every MKU computable function is a recursive function.

We have been through this sort of thing before. And there are different ways to proceed. I indicate only some natural first steps. Begin assigning numbers to labels, symbols, cells, and relations in some reasonable way.

a. $g[q_i] = 3 + 8i$ c. $g[c_i] = 7 + 8i$
b. $g[s_i] = 5 + 8i$ d. $g[r^i_j] = 9 + 8(2^i \times 3^j)$

Then the number for a *page* is $p_0^{\langle c_i \rangle} \times p_1^{\langle s_a \rangle} \times \cdots \times p_n^{\langle s_b \rangle}$, and for an *edge* $p_0^{\langle r^i_j \rangle} \times p_1^{\langle c_{a_1} \rangle} \times \cdots \times p_i^{\langle c_{a_i} \rangle}$. So a page is a cell with some sequence of symbols, and an edge is an

i-place relation applied to i cells. Some *data* is a sequence of pages with distinct cell numbers, and a *structure* is a sequence of distinct edges. Cells are (*immediately*) *connected* on a structure when the structure has an edge of which both are members, and *connected* on a structure when there is a sequence of cells from the structure, beginning with the one, ending with the other such that each is immediately connected to the next. A *space* is a structure with exactly one origin and every cell connected to all the others. A *dataspace* is of the sort $\pi_0^{\mathsf{m}} \times \pi_1^{\mathsf{n}}$ where m numbers some data, n a space, and every cell from m appears in n.

After that, with considerable work, $\textsc{mkumachine}(\mathsf{m})$ when m numbers an MKU machine. So, as above, the MKU machines are enumerable. Then $\mathsf{mkumachs}(\mathsf{u}, \mathsf{a}, \mathsf{d})$ numbers an instruction as a function of the index for the machine, initial label, and dataspace.[16] For machine u with input n, $\mathsf{mkudataspace}(\mathsf{u}, \mathsf{n}, \mathsf{j})$ and $\mathsf{mkustate}(\mathsf{u}, \mathsf{n}, \mathsf{j})$ give the current number of the dataspace and state. And $\mathsf{mkustop}(\mathsf{u}, \mathsf{n}, \mathsf{j})$ takes the value zero when the machine is stopped. Then,

$$\mathsf{f}(\mathsf{n}) = \mathsf{mkudecode}(\mathsf{mkudataspace}(\mathsf{u}, \mathsf{n}, \mu\mathsf{j}[\mathsf{mkustop}(\mathsf{u}, \mathsf{n}, \mathsf{j}) = \hat{0}]))$$

It is a chore to work this out (and you have an opportunity to do so in exercises). But it should be clear that it can be done. Then any MKU computable function is recursive, and therefore every MKU computable function is Turing computable.

Given this, the squeezing argument is complete: Turing computability implies algorithmic computability; algorithmic computability implies MKU computability; and MKU computability implies Turing computability. So the algorithmically computable functions are the same as the Turing computable functions. Church's thesis!

Insofar as MKU computability mathematically implies Turing computability, this argument is just as strong as the premises that Turing computability implies algorithmic computability, and then that algorithmic computability implies MKU computability. For these, we translate between the formal and informal notions. Insofar as translation is not itself a formal procedure, the result is not formal proof of Church's thesis. Still, translation may play a role in proof: We take as given that Turing computable functions are algorithmically computable. For the other direction, note how the four parts of definition MKU reflect the parts of AC. Insofar as an algorithm just *is* a method for rote modifications of a (data) space and MKU computability accommodates arbitrary rote modifications of a data space, our idea is that MKU computability is inevitably sufficient to embody the arbitrary algorithm.

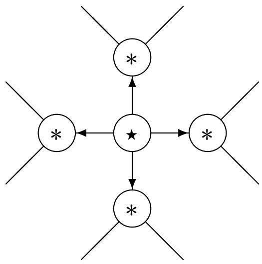
It is clear enough that Turing computability implies algorithmic computability. However the inference from algorithmic computability to MKU computability is more contentious. Perhaps it is difficult to imagine an algorithmic method that does not conform to AC and then MKU. But failure of imagination is not the same as proof.

[16]Given the potential for infinitely many different active areas, rather than supplementing the machine with repeating commands for every missing instruction, include a single label that loops on the origin such that the machine defaults to it.

So there is space for different objections: First, one might worry that the account AC of an algorithm is insufficient in some respect. AC is offered as a further exposition or sharpening of our initial notion of a "finitely constrained rule-based procedure." It is abstract and generic enough to encompas a wide variety of procedures we care about. Let us take it as a *specification* of the methods to which our version of Church's thesis applies. Take this way, AC is definitional.

Still, one might worry that the MKU machine does not compute every algorithm from AC. Against this, there are a couple of replies. First, careful about what the MKU machine can do. Say we are interested in parallel computing, whether by persons following instructions or by computing devices. An MKU machine has but a single origin; this might seem to be a problem. Still, an active area might have many "shapes"—and things might be set up as follows:

(Q)



with "satellite" centers to achieve the effect of parallel computing. Similarly, with a bit of thought, one can see how the MKU machine might achieve the effect of absolute addressing or the like. And there might be more or less sophisticated ways of mapping instructions onto a dataspace; for example, there might be some "filtering" such that only certain features of a space are relevant to a match. So it is important to recognize the generality already built into the MKU machine.

Perhaps, though, the objection goes through and some algorithmic method really is beyond the reach of the MKU machine. So for example some algorithm might require physical actions other than symbol manipulation. Consider a method for truth table construction with the instruction, "whack yourself in the head three times and write a T in the first row of the first column." An MKU machine does not have a head, and so cannot perform this action. More seriously, we might consider actions as applied to, say, a physical abacus—as "move the bead on the second wire to the leftmost available position." The MKU machine does not move physical beads on a wire, so it does not perform addition on an abacus. Still, it should be possible to *number* the states of an abacus, and to represent the successive states so as to calculate any function that can be worked on the physical device. In this case, the claim is not that the MKU machine *implements* every algorithm, but rather that it *models* every algorithm. Supposing this is sustained, the argument for Church's thesis stands.

So we are not left with a formal proof of Church's thesis. Rather we have reasons from the independent definitions, the failure of counterexamples, and the nature of an algorithm. For the latter, we translated an informal notion into a formal one—and our argument is as strong as that translation. Taken together, these amount to a (compelling) case for Church's thesis. Not all knowledge is mathematical. And similarly we might know Church's thesis, without a mathematical proof of it.

To the extent that Church's thesis is either plausible or established, our limiting results become full-fledged *incomputability* results with applications to logic and computing more generally. So, for example, by the decision problem, no Turing machine computes numbers for theorems of predicate logic. So by Church's thesis, no algorithmic method computes numbers for theorems of predicate logic. And the result does not apply just to numbers: Suppose *some* algorithmic method identifies the theorems of predicate logic; this method is naturally extended to one that calculates numbers for theorems of predicate logic—but there is no such method; so no algorithmic method identifies the theorems of predicate logic. Thus there is, say, no extension of our proof strategies from Chapter 6 to an algorithmic method that determines, for arbitrary $\mathcal{P}$, whether $\mathcal{P}$ is a theorem. Insofar as there is no such method, provability is not a decidable relation.

In addition, from Church's thesis, the *computability* of a function implies that it is recursive. Having attained Church's thesis only at the very end, we have not applied the thesis in this way. But one might move from the observation that some function is computable, through the thesis, to the result that the function is recursive. In many cases, this shortcuts elaborate demonstrations that a function can be built up from the initial functions. So, for example, from the existence of computerized proof-checking programs, one might move to the conclusion that there is a recursive $\mathsf{PRFT}(\mathsf{m},\mathsf{n})$ to say whether the sequence numbered m is a proof of the expression numbered n. We already know that there is this recursive relation. But this sort of thing is frequently done.

E14.13. Write down MKU instructions to mimic the effect of the Turing machine from example (B) computing $\mathsf{zero}(\mathsf{x})$. You may assume that cells are arranged as in a Turing machine, with each standing in the "to the left of" relation to the next. Given this simple arrangement, it will be convenient to present your program in table form—so, for example, the following is sufficient to move the origin cell back along a string of zeros to the start position and stop.

q_i			S_a					S_b			q_j
a	B	$\rightarrow$	$0^\star$	$\rightarrow$	B	B	$\rightarrow$	$0^\star$	$\rightarrow$	B	0
a	B	$\rightarrow$	$0^\star$	$\rightarrow$	0	B	$\rightarrow$	$0^\star$	$\rightarrow$	0	0
a	0	$\rightarrow$	$0^\star$	$\rightarrow$	B	$0^\star$	$\rightarrow$	0	$\rightarrow$	B	a
a	0	$\rightarrow$	$0^\star$	$\rightarrow$	0	$0^\star$	$\rightarrow$	0	$\rightarrow$	0	a

In this case the S_a areas have depth 1 edge. This exercise illustrates the point that Turing computable functions are MKU computable.

*E14.14. Work out codes for the MKU machine through dataspace. Hard-core: Assuming functions mkuencode(a) and mkudecode(b), complete the demonstration that any MKU computable function f(n) is recursive. For this, you may assume a straightforward picture on which dataspaces "match" when there is a one-to-one relation from the cells of one onto cells of the other that preserves both contents and relations among matching cells.

E14.15. For each of the following concepts, explain in an essay of about two pages, so that (college freshman) Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The Turing computable functions, and their relation to the recursive functions.

b. The essential elements from the chapter contributing to a demonstration of the decision problem, along with the significance of Church's thesis for this result.

c. The essential elements from this chapter contributing to a demonstration of (the semantic version of) the ĩncompleteness of arithmetic.

d. Church's thesis, along with reasons for thinking it is true, including the possibility of demonstrating its truth.

Theorems of Chapter 14

T14.1 There is a recursive enumeration of the Turing machines.

T14.2 Every Turing computable function is a recursive function.

T14.3 Every recursive function is Turing computable.

T14.4 There is no Turing machine $\mathsf{H}(\mathsf{i})$ such that $\mathsf{H}(\mathsf{i}) = 0$ if $\Pi_{\mathsf{i}}(\mathsf{i})$ halts and $\mathsf{H}(\mathsf{i}) = 1$ if it does not.

T14.5 There is no Turing machine $\mathsf{H}(\mathsf{i}, \mathsf{n})$ such that $\mathsf{H}(\mathsf{i}, \mathsf{n}) = 0$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H}(\mathsf{i}, \mathsf{n}) = 1$ if it does not.

T14.6 If Q is ω-consistent and $\mathscr{L}$ includes $\mathscr{L}_{NT}$, then no Turing computable function $\mathsf{prvpl}(\mathsf{n})$ is such that $\mathsf{prvpl}(\mathsf{n}) = 0$ just in case n numbers an $\mathscr{L}$-theorem of predicate logic.

T14.7 The set of all truths in a language including $\mathscr{L}_{NT}$ is not recursively enumerable.

T14.8 If T is a recursively axiomatized s̃ound theory whose language includes $\mathscr{L}_{NT}$, then there is a sentence $\mathcal{P}$ such that $T \nvdash \mathcal{P}$ and $T \nvdash \sim\mathcal{P}$.

T14.9 Suppose T is a recursively axiomatized theory extending Q; then if T is consistent $T \nvdash \mathcal{H}(\bar{\mathsf{s}})$, and if T is ω-consistent, $T \nvdash \sim\mathcal{H}(\bar{\mathsf{s}})$.

T14.10 Every total function that can be captured by a consistent recursively axiomatized theory extending Q is recursive.

T14.11 Every MKU computable function is a recursive function.

And we mention,

CT *Church's Thesis:* The total numerical functions that are effectively computable by some algorithmic method are just the recursive functions.

Concluding Remarks

Looking Forward and Back

We began this text in Part I setting up the elements of classical symbolic logic. Thus we began with four notions of validity: logical validity, validity in *AD*, validity in *ND*, and semantic quantificational validity. After a parenthesis in Part II to think about techniques for reasoning about logic, we began to put those techniques to work. The main burden of Part III was to show s̊oundness and c̊ompleteness of our classical logic, that $\Gamma \vdash \mathcal{P}$ iff $\Gamma \vDash \mathcal{P}$. This is the good news. In Part IV we established some limiting results. These include Gödel's first and second theorems, that no consistent recursively axiomatized extension of Q is negation c̃omplete, and that no consistent recursively axiomatized theory extending PA proves its own consistency. Results about derivations are associated with computations, and the significance of this association extended by means of Church's thesis. This much constitutes a solid introduction to classical logic, and should position you make progress in logic and philosophy along with related areas of mathematics and computer science.

Excellent texts which mostly overlap the content of this one, but extend it in different ways are Mendelson, *Introduction to Mathematical Logic*; Enderton, *A Mathematical Introduction to Logic*; and Boolos, Burgess, and Jeffrey, *Computability and Logic*; these put increased demands on the reader (and such demands are one motivation for our text), but should be accessible to you now; Shoenfield, *Mathematical Logic* is excellent yet still more difficult. Smith, *An Introduction to Gödel's Theorems* extends the material of Part IV; Cooper, *Computability Theory* develops it especially from the perspective of Chapter 14. Manzano, *Model Theory* and, more advanced, Hodges, *A Shorter Model Theory* extend the material of section 11.4. Much of what we have done presumes some set theory as Enderton, *Elements of Set Theory*.

In places, we have touched on logics alternative to classical logic, including free logic, multi-valued logic, modal logic, and logics with alternative accounts of the conditional. A good place to start is Priest, *Non-Classical Logics*, which is profitably read with Roy, "Natural Derivations for Priest" which introduces derivations in a style much like our own. Our logic is *first-order* insofar as quantifiers bind just variables for objects. Second-order logic lets quantifiers bind variables for classes or properties as well (so $\forall x \forall y[x = y \rightarrow \forall F(Fx \leftrightarrow Fy)]$ expresses the *indiscernibility of identicals*). Second-order logic has important applications in mathematics, and raises important issues in metalogic. For this, see Shapiro, *Foundations Without Foundationalism*, and Manzano, *Extensions of First Order Logic*. Plural logic adds to our symbols quantifiers $\forall xx$ and $\exists yy$, read *for any things xx*, and *there are some things yy*, along with relations of the sort $t \prec T$ to say that t is among the T's. This permits some (but not all) of the powers of second-order logic without apparent quantification over classes or properties. Oliver and Smiley, *Plural Logic* is a good introduction. Though our languages have infinitely many symbols, *formulas* are always finitely long. Infinitary logic drops this constraint and allows expressions that are infinitely long. As for plural and second-order logic, the expressive power of infinitary logic exceeds that of our own. Discussions of infinitary logic presuppose significant background in set theory though Bell, "Infinitary Logic" and Nadel, "$\mathcal{L}_{\omega_1\omega}$ and Admissible Fragments" are a reasonable place to start.

Philosophy of logic and mathematics is a subject matter of its own. Shapiro, "Philosophy of Mathematics and Its Logic" (along with the rest of the articles in the same *Oxford Handbook*), and Shapiro, *Thinking About Mathematics* are a good place to start. Benacerraf and Putnam, *Philosophy of Mathematics* and Marcus and McEvoy, *Philosophy of Mathematics* are collections of classic articles.

Smith's online, "Beginning Mathematical Logic: A Study Guide" is an excellent comprehensive guide to further resources.

Have fun!

Bibliography

Appel, and Haken. "The Solution of the Four-Color Map Problem." *Scientific American* 237 (1976): 108–121.

Bell, J. "Infinitary Logic." In *The Stanford Encyclopedeia of Philosophy*, edited by Zalta, Metaphysics Research Lab, Stanford University, 2016. URL `https://plato.stanford.edu/archives/win2016/entries/logic-infinitary/`.

Benacerraf, and Putnam, editors. *Philosophy of Mathematics: Selected Readings*. Cambridge: Cambridge University Press, 1983, 2nd edition.

Bergmann, Moor, and Nelson. *The Logic Book*. New York: McGraw-Hill, 2004, 4th edition.

Berto, F. *There's Something About Gödel: The Complete Guide to the Incompleteness Theorem*. Oxford: Wiley-Blackwell, 2009.

Black, R. "Proving Church's Thesis." *Philosophia Mathematica* 8 (2000): 244–258.

Boolos, Burgess, and Jeffrey. *Computability and Logic*. Cambridge: Cambridge University Press, 2002, 4th edition.

Boolos, G. *The Logic of Provability*. Cambridge: Cambridge University Press, 1993.

Button, and Walsh. *Philosophy and Model Theory*. Oxford: Oxford University Press, 2018.

Carnap, R. *Logical Syntax of Language*. New York: Routledge, 1937.

Cederblom, and Paulsen. *Critical Reasoning*. Belmont: Wadsworth, 2005, 6th edition.

Church, A. "An Unsolvable Problem of Elementary Number Theory." *American Journal of Mathematics* 58 (1936): 345–363.

Cohen, P. "The Independence of the Continuum Hypothesis." *Proceedings of the National Academy of Sciences of the USA* 50 (1963): 1143–1148.

Cooper, B. *Computability Theory*. Boca Raton: Chapman & Hall/CRC Mathematics, 2004.

Dennett, editor. *The Philosopher's Lexicon*. 1987. URL https://dl.tufts.edu/concern/pdfs/jw827p68f.

Detlefsen, M. "On Interpreting Gödel's Second Theorem." *Journal of Philosophical Logic* 8 (1979): 297–313.

———. *Hilbert's Program*. Dordrecht: D. Reidel, 1986.

Drake, and Singh. *Intermediate Set Theory*. Chichester, England: John Wiley & Sons, 1996.

Earman, and Norton. "Forever is a Day: Supertasks in Pitowsky and Malament-Hogarth Spacetimes." *Philosophy of Science* 60 (1993): 22–42.

Earman, J. *Bangs, Crunches, Whimpers, and Shrieks: Singularities and Acausalities in Relativistic Spacetimes*. New York: Oxford University Press, 1995.

Enderton, H. *A Mathematical Introduction to Logic*. San Diego: Academic Press, 1972.

———. *Elements of Set Theory*. Boston: Academic Press, Inc., 1977.

Feferman, et al., editors. *Gödel's Collected Works: Volume I*. New York: Oxford University Press, 1986.

Feferman, S. "Does Mathematics Need New Axioms?" *The American Mathematical Monthly* 106 (1999): 99–111.

Fine, K. "The Role of Variables." *The Journal of Philosophy* 100 (2003): 605–631.

Gentzen, G. "The Consistency of Elementary Number Theory." In *The Collected Papers of Gehard Gentzen*, edited by Szabo, Amsterdam: North Holland, 1969a.

———. "New Version of the Consistency Proof for Elementary Number Theory." In *The Collected Papers of Gehard Gentzen*, edited by Szabo, Amsterdam: North Holland, 1969b.

Gödel, K. "Die Vollständigkeit der Axiome des Logischen Funktionenkalküls." *Monatshefte für Mathematik und Physik* 37 (1930): 349–360.

———. *The Consistency of the Continuum Hypothesis*. Princeton: Princeton University Press, 1940.

———. "On Formally Undecidable Propositions of *Principia Mathematica* and Related Systems I." In *Gödel's Collected Works: Volume I*, edited by Feferman, et al., Oxford: Oxford University Press, 1986, 144–95.

Grabmayr, B. "On the Invariance of Gödel's Second Theorem With Regard to Numberings." *The Review of Symbolic Logic* 14 (2021): 51–84.

van Heijenoort, editor. *From Frege to Gödel*. Cambridge: Harvard University Press, 1967.

Henkin, L. "The Completeness of the First-Order Functional Calculus." *Journal of Symbolic Logic* 14 (1949): 159–166.

———. "A Problem Concerning Provability." *Journal of Symbolic Logic* 17 (1952): 160.

Hilbert, and Bernays. *Grundlagen der Mathematik, Vol II*. Berlin: Springer, 1939.

Hilbert, D. "On the Infinite." In *Philosophy of Mathematics: Selected Readings*, edited by Benacerraf, and Putnam, Cambridge University Press, 1983, 183–201. Second edition.

Hodges, W. *A Shorter Model Theory*. Cambridge: Cambridge University Press, 1997.

Hogarth, M. "Does General Relativity Allow an Observer To View an Eternity In a Finite Time?" *Foundations of Physics Letters* 173–181.

Hurley, P. *A Concise Introduction to Logic*. Belmont: Wadsworth, 2000, seventh edition.

Kleene, S. *Introduction to Metamathematics*. Bronx: Ishi Press International, 1952.

———. "Origins of Recursive Function Theory." *Annals of the History of Computing* 3 (1981): 52–67.

Kolmogorov, and Uspenskii. "On the Definition of an Algorithm." *American Mathematical Society Translations* 29 (1963): 217–245.

Kripke, S. *Wittgenstein on Rules and Private Language: An Elementary Exposition*. Cambridge, Mass.: Harvard University Press, 1982.

———. "On Two Paradoxes of Knowledge." In *Philosophical Troubles: Collected Papers, Volume I*, edited by Kripke, Oxford: Oxford University Press, 2011, 27–51.

Manzano, M. *Extensions of First Order Logic*. Cambridge: Cambridge University Press, 1996.

———. *Model Theory*. Oxford: Clarendon Press, 1999.

Marcus, and McEvoy, editors. *An Historical Introduction to the Philosophy of Mathematics*. London: Bloomsbury Publishing, 2016.

Marker, D. *Model Theory: An Introduction*. New York: Springer, 2002.

Mendelson, E. *Introduction to Mathematical Logic*. New York: Chapman and Hall, 1997, 4th edition.

Nadel, M. "$\mathscr{L}_{\omega_1\omega}$ and Admissible Fragments." In *Model-Theoretic Logics*, edited by Barwise, and Feferman, New York: Springer-Verlag, 1985, 271–316.

Oliver, and Smiley. *Plural Logic*. Oxford: Oxford University Press, 2013, second edition.

Plantinga, A. *God, Freedom, and Evil*. Grand Rapids: Eerdmans, 1977.

Priest, G. *Non-Classical Logics*. Cambridge: Cambridge University Press, 2001.

Putnam, H. *Reason, Truth and History*. Cambridge: Cambridge University Pres, 1981.

Read, S. *Thinking About Logic*. Oxford: Oxford University Press, 1995.

Robinson, A. *Non-Standard Analysis*. Princeton: Princeton University Press, 1966.

Robinson, R. "An Essentially Undecidable Axiom System." *Proceedings of the International Congress of Mathematics* 1 (1950): 729–730.

Robinson, T. "A New Formalization of Anselm's Ontological Argument." *Philosophy Faculty Publications* 6. URL `http://digitalcommons.csbsju.edu/philosophy_pubs/6`.

Rogers, R. *Mathematical Logic and Formalized Theories*. Amsterdam: North-Holland Publishing Company, 1971.

Rosser, B. "Extensions of Some Theorems of Gödel and Church." *Journal of Symbolic Logic* 1 (1936): 230–235.

Rowling, J.K. *Harry Potter and the Prisoner of Azkaban*. New York: Scholastic Inc., 1999.

Roy, T. "Natural Derivations for Priest, *An Introduction to Non-Classical Logic*." *The Australasian Journal of Logic* 47–192. URL `http://ojs.victoria.ac.nz/ajl/article/view/1779`. See also Roy and Fry, "More Natural Derivations for Priest," at `https://tonyroyphilosophy.net/research-projects/`.

———. "Modality." In *The Continuum Companion to Metaphysics*, edited by Manson, and Barnard, London: Continuum Publishing Group, 2012, 46–66.

Russell, and Whitehead. *Principia Mathematica*. Cambridge: Cambridge University Press, 1910–13.

Russell, B. "On Denoting." *Mind* 14 (1905): 479–493.

Sainsbury, R. *Paradoxes*. Cambridge: Cambridge University Press, 2009, third edition.

Shapiro, S. *Foundations Without Foundationalism: A Case for Second Order Logic*. Oxford: Clarendon Press, 1991.

———. *Thinking About Mathematics: The Philosophy of Mathematics*. Oxford: Oxford University Press, 2000.

———. "Philosophy of Mathematics and Its Logic: Introduction." In *The Oxford Handbook of Philosophy of Mathematics and Logic*, edited by S. Shapiro, Oxford: Oxford University Press, 2005, 3–28.

Shoenfield, J. *Mathematical Logic*. New York: CRC Press, 1967.

Smith, P. "Squeezing Arguments." *Analysis* 71 (2011): 22–30.

———. *An Introduction to Gödel's Theorems*. Cambridge: Cambridge University Press, 2013, second edition.

———. "Beginning Mathematical Logic: A Study Guide.", 2022. URL https://www.logicmatters.net/resources/pdfs/LogicStudyGuide.pdf.

Tarski, A. "The Concept of Truth in Formalized Languages." In *Logic, Semantics, Metamathematics: Papers from 1923 to 1938*, edited by Corcoran, Indianapolis: Hackett, 1983, 152–278.

Turing, A. "On Computable Numbers, With an Application to the *Entscheidungsproblem*." *Proceedings of the London Mathematical Society* 42 (1936): 230–265.

Wang, H. "The axiomatization of Arithmetic." *Journal of Symbolic Logic* 22 (1957): 145–158.

Index

$\sim$ tilde, 33, 47
$\to$ arrow, 33, 47
$\vee$ wedge, 40, 57
$\wedge$ caret, 40, 57
$\leftrightarrow$ double arrow, 40, 57
$=$ equals, 47, 62, 85, 300
$\forall$ for all, 47
$\forall x$ universal quantifier, 49
$\exists$ exists, 57
$\exists x$ existential quantifier, 57
Ø zero (object language), 62, 85, 300
S successor, 62, 85, 300
$+$ plus, 62, 85, 300
$\times$ times, 62, 85, 300
$<$ less than, 62, 85, 300
$\vdash$ single turnstile, 67, 201
$\mathcal{A}^x_t$ x replaced by t, 78, 263
$\leq$ less than or equal, 85, 300
$\mathsf{I}[\mathcal{A}]$ truth valuation, 94, 122, 337
T true, 94
F false, 94
$\nvDash$, $\neq$ (etc.), slash notation, 101, 312
$\vDash$ double turnstile, 102, 125
$\in$ set membership, 112
$\{\,\}$ curley brace, 112
$\langle\,\rangle$ angle brace, 112
$\subseteq$, $\subset$ subset, proper subset, 112
$\cap$, $\bigcap$ intersection, 112
$\cup$, $\bigcup$ union, 112
$\varnothing$ empty set, 112
$\mathbb{N}$ natural numbers, 113
$\bar{\mathsf{N}}$ interpretation $\mathcal{L}^{<}_{\mathrm{NT}}$, 113
$\mathsf{d}(x|\mathsf{o})$ variable assignment, 115
$\mathsf{I}_{\mathsf{d}}[t]$ term assignment, 115
$\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ satisfaction assignment, 118
S satisfied, 118
N not satisfied, 118
I interpretation function, 135
I_ω intended interpretation, 135
$\perp$ bottom, 217, 245, 275
$\mathcal{P}^t/_s$ replace one or more, 290
$(\forall x : \mathcal{B})$ restricted universal, 296
$(\exists x : \mathcal{B})$ restricted existential, 296
$(\forall x < t)$ bounded universal, 296, 300
$(\exists x < t)$ bounded existential, 296, 300
N interpretation $\mathcal{L}_{\mathrm{NT}}$, 300
$\uparrow$ up arrow, 320
$\Rightarrow$ metalinguistic conditional, 321
$\Leftrightarrow$ metalinguistic biconditional, 321
$\neg$ metalinguistic negation, 321
$\vartriangle$ metalinguistic conjunction, 321
$\triangledown$ metalinguistic disjunction, 321
$\underline{\perp}$ metalinguistic contradiction, 321
S metalinguistic existential, 324
A metalinguistic universal, 337
$\bar{\mathsf{n}}$ numeral of $\mathcal{L}_{\mathrm{NT}}$, 391
$\mathcal{T}^r/\!\!/_s$ replace at most one, 427
$\mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$ replace at most one, 430
$\mathcal{P}^u$ universal closure, 465
$\simeq$ equivalence relation, 481, 532
$[\mathsf{a}]$ equivalence class, 482, 532
$\mathfrak{Md}(\Sigma)$ models of Σ, 503
$|\mathfrak{M}|$ formulas true on $\mathfrak{M}$, 505
$\overset{\iota}{\cong}$ ι-isomorphism, 507
$\cong$ isomorphism, 508
$\mathbb{P}$ positive integers, 509
$\equiv$ elementary equivalence, 509
$\upharpoonright$ restriction, 513
$2\mathbb{N}$ even natural numbers, 513
$\sqsubseteq$ submodel, 515
$\preceq$ elementary submodel, 516
$\overset{\iota}{\underset{\approx}{\sqsubset}}$ ι-embedding, 518

$\precapprox$ embedding, 518
$\mathbb{Z}$ integers, 518
ran(f) range of function, 519
$\precsim^{\iota}$ ι-elementary embedding, 520
$\precsim$ elementary embedding, 520
$\bigvee$ generalized disjunction, 524, 527
$\bigwedge$ generalized conjunction, 524, 527
$\mathcal{M}_n$ at most n, 524
$\mathcal{L}_n$ at least n, 524
$\approx$ cardinal same size, 535
$\prec$ cardinal less than, 535
$\preccurlyeq$ cardinal less or equal, 535
$\aleph$ cardinal number, 535
card(M) cardinality of M, 535
$\mathbb{R}$ real numbers, 535
$\wp$ powerset, 537
$\mathcal{P}^e$ existential closure, 544
$\vec{\mathsf{x}}/\vec{x}$ vector, 568, 576
$\overline{\mathsf{n}}$ numeral of $\mathcal{L}_{NT}$, 574
$\exists\vec{v}/\forall\vec{v}$ quantifier block, 589, 590
$\ulcorner\mathcal{P}\urcorner$ Gödel number, 613, 757
$\overline{\ulcorner\mathcal{P}\urcorner}$ numeral, 613
$\widehat{\ulcorner\mathcal{P}\urcorner}$ recursive function, 613
$\langle\ \rangle$ symbol code, 613, 757
$\exists!$ exactly one, 650
$\mathcal{A}^{\mathcal{B}}_{\mathcal{C}}$ replace all, 651
$[\![\mathcal{P}]\!]$ number after *sub*, 728

AB abbreviation, quantificational, 57
AB abbreviation, sentential, 40
AC algorithmic computability, 782
AD axiomatic derivation, full, 82
ADq axiomatic derivation, quantifier, 79
ADs axiomatic derivation sentential, 70
AI term assignment on interpretation, 395
AP axiomatic derivation preliminary, 67
AR argument, 5
AS atomic subformula, sentential, 39
AV axiomatic consequence, 67
AX axiomatization, 522
CA categorical, 522
CF characteristic function, 604
Cf coordinate functions, 683
CG criterion of goodness, 135
CM composition, 568
Cm coordinate minimization, 688
Con consistency, 454, 464
CP capture, 588
Cr coordinate relations, 688
CS compound and simple, 142
CT Church's thesis, 771
DC declarative sentence, 142
Δ_0 formulas, 589
EE elementary equivalence, 509
EL elementary embedding, 520
EM embedding, 518
ES elementary submodel, 516
EXf expression for function, 575
EXr expression for relation, 574
FA accessible formula, 206
FR formula, quantificational, 52
FR formula, sentential, 37
FR$'$ quantificational formula abbreviated, 58
FR$'$ sentential formula abbreviated, 41
IR interpretation recursive language, 778
IS isomorphism, 507
IS immediate subformula, sentential, 39
IT invalidiey test, 13
LS logical soundness, 12
LV logical validity, 12
Max maximality, 459, 467
MKU machine, 783
MO main operator (informal), 142
MO main operator, sentential, 39
NC (negation) $\tilde{\text{c}}$omplete, 522
ND natural derivations, full, 291
ND+ natural derivation plus, 298
NDs natural derivation sentential, 225
NDs+ natural derivation sentential plus, 259
NP natural derivation preliminary, 199
PA Peano arithmetic, 85, 313
Π_1 formulas, 589
PR primitive recursive, 572
Q Robinson arithmetic, 301, 313
QI quantificational interpretation, 111
QV quantificational validity, 125, 337
RC relatively complete, 522
RD recursion, 568
RF recursive function, 572
RL recursive language, 778
RM regular minimization, 572
RQ restricted quantifiers, 296

RS relatively sound, 522
RT recursion theorem, 570
SA accessible subderivation, 206
SB subformula, sentential, 39
SC strategies for a contradiction (quantificational), 278
SC strategies for a contradiction (sentential), 243
Scgt scapegoat set, 469
SD subderivation, 206
SF satisfaction, 118, 335, 341, 344
SF′ satisfaction abbreviations, 132, 335, 345
SG strategies for a goal (quantificational), 278
SG strategies for a goal (sentential), 231
SI s̃ound (on intended models), 522
SM submodel, 515
SO sentential operator, 142
Σ_1 Sigma one formulas, 589
$\Sigma_\star$ Sigma star formulas, 723
ST sentential truth, 94, 321
ST′ sentential truth abbreviations, 109, 329
SV sentential validity, 101, 324
TA term assignment, 115, 339
TF truth functional operator, 142
TI truth on an interpretation, 122, 337
TP translation procedure, 143
TR term, 50
VC vocabulary, quantificational, 47
VC vocabulary, sentential, 33
VT validity test, 16

abb, unpack abbreviation (meta), 329
abv, unpack abbreviation (obj), 70, 300
alternate semantics of variables, 353
alternatives to classical logic
- free logic, 114
- infinitary logic, 527
- modal logic, 413
  - GL, 744
  - K4, 645, 748
  - K4LR, 743
- multi-valued logic, 151
- relevant logic, 26, 741
- second-order logic, 527

argument AR, 5
- standard form, 5

axiomatic derivation
- A^*, 76, 377
- $A^\star$, 420, 427, 431, 440, 444, 452, 475
- $A^\#$, 488
- AD, 82
  - deduction theorem, 421–424
  - equivalent to ND, 414–444
  - independence of axioms, 377, 499–503
  - Peano arithmetic in, 85–91
  - quantificational ADq, 79
  - sentential ADs, 70
- generic account AL, 66
  - axiom, 66
  - axiom schema, 66
  - consequence AV, 67
  - cumulative nature, 68, 71–72, 86
  - derivation, 68
  - preliminary system AP, 67
  - proof, 67
  - rule, 66
  - theorem, 67
  - validity, 67

beta function
- *see* recursive functions

biconditional
- formal expression, 41
- metalinguistic expression, 321
- ordinary operator
  - if and only if, 160
  - just in case, 160

Carnap's Equivalence, 625, 638
Chinese remainder theorem, 582
- proved in *PA*, 673

Church's thesis
- *see* computability

completeness, 408
- c̊ompleteness, 446
  - quantificational, 464–488
  - sentential, 453–463
- c̊ompleteness, definitions for
  - consistency Con, 454, 464, 486
  - core thesis $*$, 455
  - core thesis $\star$, 465
  - core thesis $\star\star$, 476

maximality Max, 459, 467, 486
scapegoat set Scgt, 469, 486
universal closure, 465
c̆ompleteness, 512, 561, 564
by quantifier elimination, 546
finite domain, 542
of theory L, 557
of theory S, 551
ĩncompleteness of arithmetic, 627–632, 637–643, 768–770
relative completeness, 506
c̊ompleteness and, 507
c̆ompleteness and, 512
computability
algorithmic computability, 772–774
definition AC, 782
Church's thesis CT, 771–774
basis for, 774–788
effective computability, 771
effective decidability, 561, 563, 771
halting problem, 764
MKU computable function
same as recursive, 785–786
MKU machine, 783–785
Turing computable function, 753
same as recursive, 757–763
Turing machine, 751–756
conditional
and cause, 159
formal expression, 33
antecedent, 38
consequent, 38
material, 38
metalinguistic expression, 321
ordinary operator
if then, 157
only if, 158
subjunctive, 159
conjunction
formal expression, 40
conjunct, 40
metalinguistic expression, 321
ordinary operator
and, 151
both and, 154
but, 151
consistency
defined Con, 454, 464, 486, 561
formal consistency sentences
alternatives to *Cont*, 735–742
Cont, 644
not provable in *PA*, 643–735
derivability conditions, 645
semantic argument for, 452
contradiction
formal expression, 217
metalinguistic expression, 321
ordinary, 20
countability, 35, 48

decidability, effective, *see* computability
decision problem, 633, 767
diagonal lemma, 625, 638
disjunction
formal expression, 40
disjunct, 40
metalinguistic expression, 321
ordinary operator
either or, 154
or, exclusive, 153
or, inclusive, 153
unless, 153

enumeration, 456, 629
equality
reflexivity
theorem of *AD*, 82
theorem of *ND*, 292
symmetry
theorem of *AD*, 83
theorem of *ND*, 292
transitivity
theorem of *AD*, 83
theorem of *ND*, 292
equivalence class, 482, 532
equivalence relation, 482, 532
expressive completeness, 491–494

fixed point, 626
font conventions, 694
formal language
basic sentence, 93, 739
defined expressions, 650–656
diagonalization, 624, 637–638
expression of, 34

expressive completeness, 491–494
characteristic sentence, 492
$\mathscr{L}_{\text{NT}}^{<}$, 62
$\mathscr{L}_{\text{NT}}$, 85, 300
$\mathscr{L}_q$, 47
$\mathscr{L}_s$, 33
normal form, 372
disjunctive normal form, 379, 548
extended normal form, 724
object language, 34
operator
binary, 38, 54
scope, 54
unary, 38, 54
unique readability, 495–498
see also formal language, sentential; formal language, quantificational
formal language, quantificational, 46–64
Δ_0 formula, 589
Π_1 formula, 589
Σ_1 formulas, 589
$\Sigma_\star$ formula, 723
abbreviation AB, 57
abbreviations, other, 59, 62
atomic formula, 54, 58
bound variable, 54
bounded quantifiers, 296, 300
closure
existential, 544
universal, 465
first-order, 527
formula FR, FR′, 52, 58
free for, 78, 263
free term, 290, 427
free variable, 54
immediate subformula, 54, 58
main operator, 53, 58
numeral, 391, 574
open formula, 54
operator, 53, 58
restricted quantifiers, 296, 300
sentence, 56
subformula, 53, 58
term TR, 50
variable-free term, 79, 264, 472
vocabulary VC, 47
see also formal language
formal language, sentential, 33–45
abbreviation AB, 40
abbreviations, other, 42
atomic formula AS, 39, 41
formula FR, FR′, 37, 41
immediate subformula IS, 39, 41
main operator MO, 39, 41
sentence, 37
subformula SB, 39, 41
unabbreviation, 43–45
vocabulary VC, 33
see also formal language
function
capture of CPf, 588
expression of EXf, 575
see also set theory; recursive function
fundamental theorem of arithmetic, 458, 468

generalization
derivation rule, 79
existential
formal expression, 57
universal
formal expression, 46
Gödel number
formula, 456, 465, 612
super, 612
Turing machine, 757
Gödel's first ĩncompleteness theorem
see completeness
Gödel's second ĩncompleteness theorem
see consistency
Gödel, K., 454, 562, 581, 639

halting problem
see computability
Henkin, L., 454, 743
Hilbert, D., 562, 646

iff, if and only if, 12
ĩncompleteness of arithmetic
see completeness
interpretation
quantificational, 111
domain of, 111
universe of, 111
sentential, 94

- standard
  - for $\mathcal{L}^{<}_{NT}$ $\bar{\mathsf{N}}$, 113
  - for $\mathcal{L}_{NT}$ N, 300
- invalidity
  - in derivation systems
    - semantic argument for, 452
  - logical
    - test for IT, 13
  - quantificational
    - and tree diagram, 126
    - in metalanguage, 354–358
  - sentential
    - and truth table, 103
  - *see also* validity

- Löb's theorem, 742–748

- mathematical induction
  - formal
    - axiom schema, 85, 307
    - rule, 85, 307
  - metalinguistic scheme, 363, 368
- mathematics
  - philosophy of, 419, 563, 647
- metalanguage
  - and form, 34
  - derivations in, 322
  - formal, 321
  - natural, 34–36, 49, 66
  - rules for reasoning in, 333, 352
    - biconditional rules bcnd, 328
    - bottom introduction bot, 321
    - commutation com, 321
    - conditional rules cnd, 328
    - conjunctive rules cnj, 321
    - definition def, 348
    - DeMorgan dem, 321
    - disjunctive rules dsj, 321
    - distribution dst, 330
    - equality rules eq, 339
    - existential rules exs, 324
    - idempotence idm, 321
    - inspection ins, 325
    - negation rules neg, 321
    - quantifier negation qn, 337
    - universal rules unv, 337
- model theory
  - axiomatization, 503
    - finite, 503
  - categorical set, 512
  - compactness, 523
  - completeness, relative, 506
  - c̃ompleteness, 512, 561, 564
  - elementary equivalence EE, 509
  - embedding EM, 518
  - embedding, elementary EL, 520
  - isomorphism IS, 507
  - Löwenheim-Skolem theorems, 534
  - model, 452
    - nonstandard for arithmetic, 531
    - size of, 535
  - quantifier elimination, 547
    - and theory L, 553
    - and theory S, 549
  - satisfiability, 523
    - finite, 523
  - Skolem's paradox, 539
  - soundness, relative, 505
  - s̃oundness, 512, 561, 564
  - submodel SM, 515
  - submodel, elementary ES, 516

- natural derivation
  - N^*, 440
  - $N^\star$, 444
  - accessible formula FA, 206
  - accessible subderivation SA, 206
  - auxiliary assumption, 203
    - discharge of, 205
  - condition on rules, 201
  - definition def, 659
  - *ND* equivalent to *AD*, 414–444
  - preliminary system *NP*, 199
  - proof, 200
  - scope line, 202
  - subderivation SD, 206
    - exit strategy for, 208, 216, 227
  - theorem, 201
  - validity, 201
  - without premises, 227
  - *see also* natural derivation, sentential; natural derivation, quantificational
- natural derivation, quantificational
  - includes rules of *NDs*, 263

instantiation of variable, 264
Peano arithmetic in, 306–312
Robinson arithmetic in, 300–306
rules of *ND* (*NDs* and), 264–292
$\forall$ exploitation $\forall$E, 264
$\forall$ introduction $\forall$I, 269
$\exists$ exploitation $\exists$E, 271
$\exists$ introduction $\exists$I, 265
$=$ exploitation $=$E, 290
$=$ introduction $=$I, 290
rules of *ND+* (*ND*, *NDs+* and), 294–297
bounded $\forall$ exploitation ($\forall$E), 296
bounded $\forall$ introduction ($\forall$I), 296
bounded $\exists$ exploitation ($\exists$E), 296
bounded $\exists$ introduction ($\exists$I), 296
quantifier distribution QD, 295
quantifier negation QN, 295
quantifier placement QP, 295
quantifier switch QS, 294
restricted quantifier negation RQN, 296
symmetry Sym, 294
strategies SC, SG, 278
SC1, 282
SC2, 282
SC3, 284
SC4, 285
SG1, 279
SG2, 279
SG3, 280
SG4, 281
SG5, 281
with equality, 292
see also natural derivation
natural derivation, sentential
rules of *NDs*, 207–228
reiteration R, 205
$\to$ exploitation $\to$E, 207
$\to$ introduction $\to$I, 208
$\wedge$ exploitation $\wedge$E, 211
$\wedge$ introduction $\wedge$I, 212
$\sim$ exploitation $\sim$E, 216, 217
$\sim$ introduction $\sim$I, 216, 217
$\vee$ exploitation $\vee$E, 219
$\vee$ introduction $\vee$I, 218
$\leftrightarrow$ exploitation $\leftrightarrow$E, 224
$\leftrightarrow$ introduction $\leftrightarrow$I, 224
rules of *NDs+* (*NDs* and), 254–261
$\perp$ introduction $\perp$I, 217
$\perp$ exploitation $\perp$E, 254
association Assoc, 257
commutation Com, 257
DeMorgan DeM, 257
disjunctive syllogism DS, 255
distribution Dist, 258
double negation DN, 256
equivalence Equiv, 258
exportation Exp, 258
hypothetical syllogism HS, 255
idempotence Idem, 257
implication Impl, 257
modus tollens MT, 254
negated biconditional NB, 255
transposition Trans, 257
strategies for a contradiction SC, 243–251
SC1, 243
SC2, 244
SC3, 245
SC4, 247
strategies for a goal SG, 231–239
SG1, 232
SG2, 233
SG3, 233
SG4, 235
SG5, 236
strategy for main operator $\vee$, 250
in *NDs+*, 258
see also natural derivation
negation
and quantity, 21
formal expression, 33, 47, 312
metalinguistic expression, 101, 321
ordinary operator
neither nor, 154, 155
not both, 154, 155
not the case that, 151
number
even natural number $2\mathbb{N}$, 513
integer $\mathbb{Z}$, 518
natural number $\mathbb{N}$, 113
positive integer $\mathbb{P}$, 509
real number $\mathbb{R}$, 535

Peano arithmetic
- and Gödel's second theorem, 643
- axioms of PA, 85, 307
- extends Q_s, 594
- in *AD*, 85–91
- in *ND*+, 306–312
- theorem of, 85, 307

predicate logic, 46

proof
- formal relation
  - *Prvt*, 644
  - *Prvpa*, 708
- *see also* axiomatic derivation; natural derivation; recursive relations, proof

quotation symbols, 613, 757

recursive definition, 37
- basis for mathematical induction, 363, 365
- closure clause, 37
- recursion theorem RT, 570
- simultaneous recursion, 611
- *see also* recursive function; recursive relation

recursive function
- captured in Q, Q_s, 587–602
  - canonical formula, 602
  - original formula, 595
- coordinate definitions, 683, 688
- defined, 572
  - composition CM, 568
  - *cmp*, 656
  - initial functions, 567
  - recursion RD, 568
  - regular minimization RM, 572
  - $\mu y \mathcal{Q}(\vec{x}, y)$, 657
- defined complements
  - friendly recursive, 662
  - μ-recursive, 766, 768, 770, 780
  - primitive recursive PR, 572
- expressed in $\mathcal{L}_{NT}$, 573–585
  - canonical formula, 602
  - original formula, 580
- represented in RL, 778
- same as captured functions, 776
- *see also* recursive functions

recursive functions
- arithmetic
  - absval $|x - y|$, 603
  - *absval*, 685
  - fact !, 569
  - *fact*, 697
  - *lcm*, 672
  - *maxp*, 673
  - *maxs*, 673
  - $\hat{n}$, 568
  - $\bar{\bar{n}}$, 685
  - plus $+$, 513, 569
  - *plus*, 684, 685
  - power x^y, 573
  - *power*, 684, 696
  - pred, 603
  - *pred*, 685
  - qt, 611
  - *qt*, 661
  - rm, 611
  - *rm*, 661
  - subc $\dot{-}$, 603
  - *subc*, 685
  - $x \dot{-} y$, 668
  - times $\times$, 515, 569
  - *times*, 684, 685
- β-function, 582
  - capture of, 595
  - definition in *PA*, 661
  - expression of, 584
- enumeration of theorems
  - code, 629
  - ethrmt, 629
  - pos, 630
- formula substitution
  - formsub, 620
  - *formsub*, 706
- Gödel numbers
  - caret, 618
  - *caret*, 692
  - cncat $\star$, 613
  - *cncat* $*$, 691, 701
  - cnd, 616
  - *cnd*, 692
  - diag, 624
  - gvar, 716
  - *gvar*, 719

num, 624
num, 719
numseq, 719
sub, 727
til, 616
til, 692
unv, 616
unv, 692
wedge, 618
initial functions
idnt^j_k, 567
idnt^j_k, 662
suc S, 513, 567
suc, 662
zero, 567
zero, 662
prime
exp, 610
exp, 698
exc, 700
len, 610
len, 699
pi π, 609
pi, 697
val, 700
val^*, 700
selection by cases
f($\vec{x}$) = {, 608
sign
csg $\overline{\text{sg}}$, 604
csg, 687
sg, 603
sg, 687
Turing machine
decode, 759
encode, 759
lb, 757
left, 759
mach, 758
machs, 759
right, 759
state, 759
stop, 760
recursive operators
bounded quantification
$(\forall y \leq z)$, 607
$(\forall y \leq z)$, 691
$(\forall y < z)$, 607
$(\forall y < z)$, 691
$(\exists y \leq z)$, 607
$(\exists y \leq z)$, 688
$(\exists y < z)$, 607
$(\exists y < z)$, 691
least under bound
$(\mu y \leq z)$, 607
$(\mu y \leq z)$, 688
$(\mu y \leq z)$, 658
truth functions
CNJ $\wedge$, 607
Cnj, 691
DSJ $\vee$, 607
Dsj, 688
IMP $\rightarrow$, 607
Imp, 691
NEG $\sim$, 607
Neg, 688
recursive relation, 604
captured in Q, 605
coordinate, 688
expressed in $\mathcal{L}_{\text{NT}}$, 604
friendly, 686
defined in *PA*, 686
see also recursive relations
recursive relations
arithmetic
EQ =, 605
Eq, 688
FCTR |, 609
Fctr, 691
$m|n$, 669
LEQ $\leq$, 605
Leq, 688
LESS <, 605
Less, 688
PRIME, 609
Prime, 691
Pr, 670
Rp, 670
formula
ATOMIC, 615
Atomic, 704
ATOMSUB, 619
Atomsub, 706
FFSEQ, 620

Ffseq, 718
FORMSEQ, 616
Formseq, 704
FORMSUB, 619
Formsub, 706
FREEf, 620
Free_f, 717
FREEFOR, 621
Freefor, 718
FSUBSEQ, 619
Fsubseq, 706
SENT, 620
WFF, 615
Wff, 704

proof
- AXIOMAD, 623
- AXIOMADS, 617
- AXIOMPA, 623
- *Axiompa*, 692
- AXIOMQ, 623
- GEN, 622
- *Gen*, 692
- ICON, 622
- *Icon*, 692
- MP, 617
- *Mp*, 692
- PRFAD, 623
- PRFADS, 617
- PRFPA, 623
- *Prfpa*, 692
- PRFQ, 623
- PRFT, 627
- PRVT (negation complete theory), 630

term
- FREEt, 620
- Free_t, 716
- TERM, 615
- *Term*, 703
- TERMSEQ, 615
- *Termseq*, 702
- TERMSUB, 619
- *Termsub*, 705
- TSUBSEQ, 618
- *Tsubseq*, 705
- VAR, 614
- *Var*, 702

Turing machine
- ACT, 757
- LB, 757
- QUAD, 758
- SYM, 757
- TMACH, 758

relation
- capture of CPr, 588
- characteristic function of CF, 604
- expression of EXr, 574
- *see also* set theory; recursive relation

Robinson arithmetic
- axioms of Q, 301
- decides atomics of $\mathscr{L}_{NT}$, 390–397
- decides Δ_0 sentences, 590
- in *ND+*, 301–306
- ĩncompleteness and, 565
- proves true Σ_1 sentences, 592
- strengthened Q_s, 594
- theorem of, 302
- trichotomy in, 397–403

Rosser, B., 640

Ruby language, 571, 573, 756

semantic reasoning
- derivations for, 322
- introductory, 317–320
- quantificational, 335–358
- sentential, 320–332
- *see also* metalanguage; semantics sentential; semantics quantificational

semantics, quantificational, 110–131
- abbreviations, 129–131
- entailment QV
  - formal, 337
  - natural, 125
- interpretation QI, 111
  - domain of, 111
  - universe of, 111
- satisfaction SF SF′
  - formal, 335, 341, 344, 345
  - natural, 118, 132
- tautology, 125
- term assignment TA
  - formal, 339
  - natural, 115
- term on interpretation AI, 395
- tree branch conditions

        - B(s), 119
        - B(r), 119
        - B(∼), 119
        - B(→), 119
        - B(∀), 119
        - B′(∧), 129
        - B′(∨), 129
        - B′(↔), 129
        - B′(∃), 129
    - trees
        - for satisfaction, 118–121, 130
        - for term assignment, 116
        - invalidity, 126–128, 130
        - truth, 122
    - truth on interpretation TI
        - formal, 337
        - natural, 122
    - validity QV
        - formal, 337
        - naural, 125
    - variable assignment, 114
    - variable, semantics of, 353
- semantics, sentential, 93–108
    - abbreviations, 106–108
    - characteristic table
        - T(∼), 94
        - T(→), 94
        - T′(∨), 106
        - T′(∧), 106
        - T′(↔), 106
        - T(↑), 320
    - entailment SV
        - formal, 324
        - natural, 101
    - interpretation, 94
    - tautology, 102
    - truth on interpretation ST, ST′
        - formal, 321, 329
        - natural, 94, 109, 320
    - truth table
        - and validity, 102–104, 107
        - arbitrary interpretations, 96–97
        - constructing values, 97–100, 107
        - short method, 104–105, 108
    - truth tree, 94, 106
    - validity SV
        - formal, 324
        - natural, 101
- set theory
    - Cantor’s theorem, 537
    - cardinality, 535
        - cardinal comparison, 535
        - cardinal number, 535
    - closure
        - under constants, 515
        - under functions, 515
    - continuum hypothesis, 501, 535
    - function, 112
        - domain, 570
        - into, 507
        - one-to-one, 507
        - onto, 507
        - partial, 570
        - range of, 519
        - regular, 571
        - restriction of, 513
        - total, 112
    - n-tuple, 112
    - orderings
        - linear order, 528
        - partial order, 527
        - well-order, 528
    - ordinal number, 535
    - powerset, 537
    - relation, 112
        - restriction of, 513
    - set, 112
        - countable, 33, 35, 48
        - enumeration of, 629
        - intersection, 112
        - uncountable, 48
        - union, 112
    - subset, 112
    - ZFC, 529, 535, 540
    - *see also* function; recursive function; relation; recursive relation
- Smith, P., 781
- soundness, 408
    - logical LS, 12
        - and truth, 12, 23–24
    - relative soundness, 505
        - s̊oundness and, 506
        - s̃oundness and, 512
    - s̊oundness, 446–452

    - strong, 377
    - weak, 376
  - s̃oundness, 512, 561, 564
- stories, and logical validity, 6–9
  - consistent, 8
  - maximal, 7
- successor, *see* recursive functions, 113

- Tarski's theorem, 633–634
- Tarski, A., 353
- theory, 561
  - ω-complete, 628
  - ω-consistent, 628
  - nicely specified, 562, 627
  - recursively axiomatized, 627
  - theorem of, 561
  - theories
    - L, 553
    - PA, 85–91, 306–312
    - Q, 301–306
    - Q_s, 594
    - S, 549
- translation
  - basis for evaluation of validity, 161, 410–414
  - criterion of goodness CG, 135
  - intended interpretation, 135
  - interpretation function, 135
  - natural, 150
  - of argument, 136, 161
  - *see also* translation, sentential; translation, quantificational
- translation, quantificational, 164–194
  - elementary sentences, 166–169
    - quantifier switching, 168
  - equality, 190–194
    - definite descriptions, 193
    - quantity expressions, 191–193
  - exception clauses, 176
    - equality in, 193
  - overlapping quantifiers, 179–188
    - quantifier placement, 184–188
  - quantification of complexes, 171–177
    - quantifier switching, 175
  - sentences without quantifiers, 165–166
  - variables as placeholders, 169
  - *see also* tranlsation
- translation, sentential, 136–162
  - compound and simple CS, 137, 142
  - declarative sentence DC, 136, 142
  - main operator MO, 137, 142
  - sentential operator SO, 136, 142
    - and underlines, 136–138
  - translation procedure TP, 143
    - equivalent sentences, 143
    - equvalent operators, 146
    - parallel tree, 146
    - parse tree, 143
    - results in good translation, 148
    - $\sim$, 151
    - $\wedge$, 151–152
    - $\vee$, 153–154
    - $\rightarrow$, 157–158
    - $\leftrightarrow$, 160
  - truth functional operator TF, 138–142
    - test not truth functional, 138
    - test truth functional, 139
  - *see also* translation; negation; conjunction; disjunction; conditional; biconditional
- tree diagram
  - consequence in *AL*, 67
  - formula, 37, 42, 52, 58, 63
  - invalidity, quantificational, 126, 130
  - parallel tree, translation, 146
  - parse tree, translation, 143
  - recursive function, 573
  - satisfaction, 119–121, 130
    - application to translation, 166–192
  - sentential truth, 94, 106
  - term, 50
  - term assignment, 116
  - unabbreviation, 45, 58, 63
  - with language RL, 778
- truth table, *see* semantics, sentential
- Turing machine
  - *see* computability
- Turing, A., 751

- unique readability, 495–498

- validity
  - in axiomatic derivation, 67
  - in natural derivation, 201
  - logical LV, 12

alternate formulations, 28
and form, 24–25
and relevance, 25–27
and truth, 12, 23
test for VT, 16
quantificational QV
formal, 337
implies logical validity, 409–410
natural, 125
sentential SV
formal, 324
implies logical validity, 409–410
natural, 101
see also invalidity
vector, 568