Typing Derivations in LaTeX

This document explains how to work derivations in LaTeX. There are a number of external packages for working natural derivations in LaTeX.[1] Supposing you already have LaTeX installed (as from "Getting started with LaTeX") I set up and explain the method from *Symbolic Logic*.

Derivations are set up (in a modified "table" envoronment) as follows,

```
\begin{der}{t}{r|l|l|l|l|l|llllllllllllllll}

\end{der}
```

(also {`sder`} and {`tder`} for smaller size fonts)—where vertical spacers are added or deleted from the right so that the total is the maximum depth of the derivation. Then lines at depth N are entered with,

```
\clN[xx]{line id}{math mode expression}{justification}
```

That is, `\clOne`, `\clTwo` and so forth. For this,

- The square brackets are optional and, if included, substitute `xx` for the (automatically generated) period after the line number.
- `line id` labels the line. It may, but need not be a number (my own preference is to us a combination of numbers and letters to track derivation structure). In any case, lines are automatically numbered, where reference to a line is by `\li{id}`.
- It is possible to bypass the automatic numbering with commands `\nclOne`, `\nclTwo` and so forth; then the line is labeled by the `id` itself.
- If a line is a premise or an assumption at depth N, it can be followed by `\caN{}` except when it is at the maximum depth, in which case the command is `\caN{&}`. This adds the "shelf" under the premise or assumption.
- To put a small gap at depth N + 1 between scope lines (as for $\vee$E or $\leftrightarrow$I) use `\ctN`.

[1] Peter Smith's `https://www.logicmatters.net/latex-for-logicians/nd/` lists some.

- `\dsep` may be included in the justification portion of a line command when an assumption is discharged. This keeps the discharged line from coming too close to the terminated scope line.

- In general, special characters and commands begin with a slash, as: `\command`. `$...$` encloses a "math mode" expression (the second component of a derivation line is automatically in math mode). Math mode has an italic font, special spacing, and permits special characters. A starter glossary:

$\sim$	:	`\til`
$\rightarrow$	:	`\imp`
$\vee$	:	`\wdg`
$\wedge$	:	`\crt`
$\leftrightarrow$	:	`\bimp`
$\forall$	:	`\forall`
$\exists$	:	`\exists`
$\bot$	:	`\bot`

  Thus `$(A \imp B)$` appears as $(A \rightarrow B)$. In math mode you can get superscripts A^1, A^{123} (`A^1` and `A^{123}`) and subscripts A_1, A_{123} (`A_1` and `A_{123}`) or both A^1_2 (`A^1_2`). In general, brackets create a "scope" to which a given command applies.

Let's create the following derivation. The derivation is discussed as example (AM) in Chapter 6 of *Symbolic Logic*. However, our aim is not to understand the derivation as such, but rather it's production in LaTeX.

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
3.	A	A (c, 2$\vee$E)
4.	$\sim A$	1 $\wedge$E
5.	$\bot$	3,4 $\bot$I
6.	B	A (c, 2$\vee$E)
7.	$\sim B$	1 $\wedge$E
8.	$\bot$	6,7 $\bot$I
9.	$\bot$	2,3-5,6-8 $\vee$E
10.	$\sim(A \vee B)$	2-9 $\sim$I

Begin setting up the derivation environment:

```
\begin{der}{t}{r|l|l|llllllllllllllllllll}

\nclOne[]{}{}{}

\end{der}
```

There will be three scope lines, so we include the three vertical spacers. It is best to paste the long last part of the first line into your derivation as—plus or minus the spacers—LaTeX expects that very expression. `\nclOne[]{}{}{}` inserts a "dummy" line to separate the top of the derivation from the bottom; in the end, we will delete it. Now enter the premise and conclusion:

```
\begin{der}{t}{r|l|l|llllllllllllllllllll}
\clOne{1}{\til A \crt \til B}{P}
\caOne{}

\nclOne[]{}{}{}

\clOne{a}{\til(A \wdg B)}{}
\end{der}
```

We don't yet know the line number of the conclusion. However, it is enough to give it some label and let LaTeX keep track of numbers. `\caOne{}` adds the shelf under the premise. This much compiles to,

1.	$\sim A \wedge \sim B$	P
2.	$\sim(A \vee B)$	

It is natural to go for the goal by ∼I. So add the assumption and goal for the subderivation:

```
\begin{der}{t}{r|l|l|llllllllllllllllllll}
\clOne{1}{\til A \crt \til B}{P}
\caOne{}
   \clTwo{2}{A \wdg B}{A ($c$, $\til$I)}
   \caTwo{}

   \nclOne[]{}{}{}

   \clTwo{2a}{\bot}{}
\clOne{a}{\til(A \wdg B)}{\li{2}-\li{2a} $\til$I\dsep}
\end{der}
```

Indentation is not required, and is included only to help you see what is going on. Similarly, the space before and after `\nclOne` has no effect, and is included for visual convenience. Observe that as soon as we set up the subderivation, we are positioned to include the justification for the conclusion, using labels for its assumption and goal. The justification portion of a derivation line is not automatically in math mode. Thus math mode expressions in a justification are enclosed by `$ $`. In the exit strategy, 'c' is in math mode simply to take advantage of the italic font. We include `\dsep` in the justification part of the last line so that the conclusion will not come too close to the discharged scope line. This much compiles to,

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
3.	$\bot$	
4.	$\sim(A \vee B)$	2-3 $\sim$I

Now our plan is to reach the contradiction by $\vee$E. So we set up for that.

```
\begin{der}{t}{r|l|l|lllllllllllllllll}
\clOne{1}{\til A \crt \til B}{P}
\caOne{}
   \clTwo{2}{A \wdg B}{A ($c$, $\til$I)}
   \caTwo{}
      \clThree{3a}{A}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}

      \nclOne[]{}{}{}

      \clThree{3b}{\bot}{}
   \ctTwo
      \clThree{3c}{B}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}

      \nclOne[]{}{}{}

      \clThree{3d}{\bot}{}
   \clTwo{2a}{\bot}{\li{2},\li{3a}-\li{3b},\li{3c}-\li{3d} $\wdg$E\dsep}
\clOne{a}{\til(A \wdg B)}{\li{2}-\li{2a} $\til$I \dsep}
\end{der}
```

`\ctTwo` adds the gap between the subderivations. This time we have included spacers for each subderivation. Again, as soon as we set up the subderivations, we are positioned to state the justification for ∨E. This compiles to,

```
1. | ∼A ∧ ∼B          P
   |―
2. | | A ∨ B          A (c, ∼I)
   | |―
3. | | | A            A (c, 2∨E)
   | | |―
   | | |
4. | | | ⊥
5. | | | B            A (c, 2∨E)
   | | |―
   | | |
6. | | | ⊥
7. | | ⊥              2,3-4,5-6 ∨E
8. | ∼(A ∨ B)         2-7 ∼I
```

And these subderivations are easily completed. So we remove the spacers and complete the derivation as follows.

```
\begin{der}{t}{r|l|l|lllllllllllllllll}
\clOne{1}{\til A \crt \til B}{P} \caOne{}
   \clTwo{2}{A \wdg B}{A ($c$, $\til$I)}
   \caTwo{}
      \clThree{3a}{A}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}
      \clThree{4}{\til A}{\li{1} $\crt$E}
      \clThree{3b}{\bot}{\li{3a},\li{4} $\bot$I}
   \ctTwo
      \clThree{3c}{B}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}
      \clThree{5}{\til B}{\li{1} $\crt$E}
      \clThree{3d}{\bot}{\li{3c},\li{5} $\bot$I}
   \clTwo{2a}{\bot}{\li{2},\li{3a}-\li{3b},\li{3c}-\li{3d} $\wdg$E \dsep}
\clOne{a}{\til(A \wdg B)}{\li{2}-\li{2a} $\til$I \dsep}
\end{der}
```

It is worth observing that brackets for `\li` (and other commands) are required only when the command applies to more than one character; so `\li 2` is fine, but only `\li{2a}`.

Note: (i) Once you get used to it, this can be more convenient than pencil and paper. You get beautiful output. And, especially for longer derivations, the ability to insert and delete lines, and to state justifications both from the top down and from the bottom up—letting LaTeX keep track of numbers—is very nice. (ii) Internally LaTeX uses a counter to number lines; this means it takes *two* compiles for line references to work: the first pass sets the anchors, and the second matches references to the anchors. And (iii), while this setup permits derivations of arbitrary complexity (answers in Chapter 13 go up to 16 scope lines, and over 200 lines long!), it can produce a raft of error messages in response to certain mistakes; this happens especially when brackets `{...}` are mismatched; every bracket must be paired with a mate!! I usually compile frequently to (see what I am doing and) catch mistakes.

This much should be enough for derivations through Chapter 6. However, there is much more that you can do. For later chapters, some of the following may be helpful (all in math mode):

$\neg A$	`\neg A`	
$A \Rightarrow B$	`A \mimp B`	
$A \Leftrightarrow B$	`A \mbimp B`	
$A \vartriangle B$	`A \mcrt B`	
$A \triangledown B$	`A \mwdg B`	
$A \uparrow B$	`A \sstroke B`	
$\perp$	`\Bottom`	
$A \neq B$	`A \not = B`	and similarly for other negated relations
$A \vDash B$	`A \vDash B`	
$A \vdash B$	`A \vdash B`	
$A \vDash_{s} B$	`A \doubleS B`	
$A \vdash_{AD} B$	`A \singleAD B`	
$A \vdash_{ADs} B$	`A \singleADs B`	
$A \vdash_{ADq} B$	`A \singleADq B`	
$A \vdash_{ND} B$	`A \singleND B`	
$A \vdash_{ND+} B$	`A \singleNDp B`	
$A \vdash_{NDs} B$	`A \singleNDs B`	
$A \vdash_{NDs+} B$	`A \singleNDsp B`	

$a \times b$	`a \times b`	
$a \leq b$	`a \leq b`	
$\langle a, b, c\rangle$	`\langle a,b,c\rangle`	
$\{a, b, c\}$	`\{a,b,c\}`	force {} to literal
$a \in b$	`a \in b`	
$\emptyset$	`\zeros`	
$\mathcal{ABC}$	`\mc{ABC}`	script, no lowercase in regular LaTeX
$\mathfrak{ABCabc}$	`\mf{ABCabc}`	Fraktur
ABCabc	`\ms{ABCabc}`	sans serif
$\textsc{abc}$	`\prp{ABC}`	small cap
ABCabc	`\mi{ABCabc}`	italic without math spacing
ABCabc	`\mr{ABCabc}`	roman without math spacing
$\mathbb{A}\mathit{bcd}$	`\mdb{Abcd}`	italic with first in letter hollow font
α, β	`\alpha, \beta`	
Γ, Δ	`\Gamma, \Delta`	and similarly for other Greek characters
$\overline{abc}$	`\ol{abc}`	
$\overline{\mathsf{abc}}$	`\os{abc}`	
$\ulcorner ABC \urcorner$	`\Godelnum{ABC}`	
$\overline{\ulcorner ABC \urcorner}$	`\OGodelnum{ABC}`	
$a^{\overline{\ulcorner B \urcorner}}$	`a^{\OSGodelnum{B}}`	small size (for superscript)
A'	`A\pr`	

Addendum on mathematical induction. Arguments by mathematical induction, as from Chapter 8 of *Symbolic Logic*, are set up as follows,

```
\begin{ind}
   \Basis the basis
   \Assp the assumption
   \Show the show
      \item[a.] subitem
      \item[b.] subitem
      \Argline
      \item sub-conclusion
   \argline
   \Indct the main conclusion
\end{ind}
```

Again, indentation is not required, and is included for visual convenience. This compiles to,

Basis: the basis

Assp: the assumption

Show: the show

  a. subitem

  b. subitem

  ---

  sub-conclusion

Indct: the main conclusion

Subitems may be added or deleted (including in the basis) as necessary. And similarly, `\Argline` may be deleted for a case without multiple items in the show section.

In general, there are (many) external "packages" that enhance the capabilities of basic LaTeX. These packages are typically loaded from a preamble file. In addition, LaTeX lets you create and modify commands, again typically in the preamble. The supplied preamble file is sufficient to support commands listed in this document.

This is barely a start with what you can do with LaTeX. It should, however, give you a good start into exercises to *Symbolic Logic*!