Typing Derivations in LaTeX

This document explains how to work derivations in LaTeX. There are a number of external packages for working natural derivations (`https://www.logicmatters.net/latex-for-logicians/nd/` lists some). I set up and explain the method from *Symbolic Logic*. I assume LaTeX is already installed, as from "getting started with LaTeX."

A straightforward a derivation is set up in the environment,

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

```
\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 enterd 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 begin with numbers from the top down, and letters from the bottom up). In any case, the 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 ∨E or ↔I) use `ctN`.

- `\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, `\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`

Here is an example:

What you see:

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

What you type:

```
\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{3}{A}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}
      \clThree{4}{\til A}{\li{1} $\crt$E}
      \clThree{c}{\bot}{\li{3},\li{4} $\bot$I}
    \ctTwo
      \clThree{5}{B}{A ($c$, \li{2}$\wdg$E)}
      \caThree{&}
      \clThree{6}{\til B}{\li{1} $\crt$E}
      \clThree{d}{\bot}{\li{5},\li{6} $\bot$I}
    \clTwo{b}{\bot}{2,\li{3}-\li{c},\li{5}-\li{d} $\wdg$E \dsep}
  \clOne{a}{\til(A \wdg B)}{\li{2}-\li{b} $\til$I \dsep}
\end{der}
```

(Indentations are not required, and included purely for visual convenience.) 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!!

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.

This much should be enough for derivations through Chapter six. 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}`	no lowercase script in regular LaTeX
$\mathfrak{ABCabc}$	`\mf{ABCabc}`	
ABCabc	`\ms{ABCabc}`	
ABC	`\prp{ABC}`	
ABCabc	`\mi{ABCabc}`	italic without math spacing
ABCabc	`\mr{ABCabc}`	roman without math spacing
$\mathbb{A}bcd$	`\mdb{Abcd}`	italic with first in letter hollow font
α, β	`\alpha, \beta`	and similarly for other Greek characters
Γ, Δ	`\Gamma, \Delta`	

$\overline{abc}$	`\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`	

Arguments by mathematical induction (as from Chaper 8) are as follows.

What you see:

Basis: The basis.

Assp: The assumption.

Show: Main show statement.

a. Case.

b. Case.

Sub-conclusion.

Indct: Conclusion.

What you type:

```
\begin{ind}
   \Basis The basis.
   \Assp The assumption.
   \Show Main show statement.
      \item[a.] Case.
      \item[b.] Case.
      \Argline
      \item Sub-conclusion.
   \argline
   \Indct Conclusion.
\end{ind}
```

Again, indentation is not required. Cases may be added or subtracted (including in the basis) as necessary. And similarly `\Argline` and the sub-conclusion may be omitted in case an argument does not divide into separate cases.

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*!