

Advanced topics in L^AT_EX

Jamie Findlay
February 22, 2017

1 Bits and bobs

1.1 Housekeeping

- As you have no doubt noticed, running `pdflatex` doesn't just produce the PDF output file, but also some other files (including `.aux` and `.log`). (L^AT_EX uses these to keep track of various things like counters, etc.)
- This can make it rather confusing if you have more than one project in the same folder, so it may be worth thinking about the structure of your filing system before it all gets too out of hand ...

1.2 The local tex directory

- Various files, including e.g. packages, are installed in your 'local TeX directory'. If you ever want to add your own packages (`.sty` files) or document classes (`.cls` files), and make them available anywhere on your system (they are always available to files in the same folder), then this is where you have to put them.
- Location on Mac OS: `/Users/<user name>/Library/texmf/tex/latex/`
- Location on Windows: `C:\Users\<user name>\texmf\tex\latex\`

1.3 Unicode

- If you make frequent use of more 'exotic' characters (e.g. you write in a language that uses a lot of accents or other non-English characters), it might be worth investigating X_ƎT_EX (XeTeX) or LuaT_EX (LuaTeX), which allow for native unicode recognition (so you can write in e.g. Arabic script).

2 Defining your own commands

- It can be convenient to define your own commands. These can be as complex as you like, but we'll stick to pretty simple examples here. As ever, check out the Wikibook and elsewhere for more.

Syntax

```
\newcommand{name}[number of arguments]{definition}
```

- The simplest use of this is just to make a shortcut for something you have to write frequently:

Example

```
\newcommand{\ds}{description theory of names}
```

- Now, wherever you put `\ds`, you will get 'description theory of names'.
- **Note:** Macros like this swallow up whitespace after them, so that if you write

```
The \ds is wrong.
```

you will get

```
The description theory of names is wrong.
```

- To fix this, you need to add manual whitespace, either with `\` , or `{ }`. Thus, either

```
The \ds\ is wrong.
```

or

The `\ds{}` is wrong.

will correctly give

The description theory of names is wrong.

- To include arguments in your commands, you simply use tags of the form `#n` to identify where each argument should go. For example:

Example

```
\newcommand{\op}[2]{\ensuremath{\langle #1, #2 \rangle}}
```

- Now we can more easily write out ordered pairs: `\op{x}{y}` gives $\langle x, y \rangle$.
- **Note:** `\newcommand` will not allow you to overwrite an existing command (so it is quite safe in that respect). If you explicitly want this behaviour, use `\renewcommand`, which has the same syntax.

3 Tables

- Tables are, sadly, comparatively hard work in \LaTeX . The basics are easy, but as soon as you want to do something a bit more complicated things can very quickly get out of hand.

Syntax

```
\begin{tabular}[position]{columns and separators}
column1 & column2 & \\\
column1 & column2 & \\
\end{tabular}
```

Example

```
\begin{tabular}{| l | c | r |}
\hline
1 & 2 & 333 \\\ \hline
444 & 5 & 6 \\\ \hline
7 & 888 & 9 \\\
\hline
\end{tabular}
```

Example output

1	2	333
444	5	6
7	888	9

- By default, \LaTeX won't wrap text in `tabular` environments, which means that cells with a lot of writing in will spill over, even going off the page.

Example

```
\begin{tabular}{| l |}
\hline
Number & Long text \\\
\hline
1 & This is some really long text which
    is going to end up going off the page if we're not careful---OH GOD! \\\
\hline
\end{tabular}
```

Example output

Number	Long text
1	This is some really long text which is going to end up going off the page if we're not careful—OH

- To fix this, we use the `p` attribute and specify the desired width of the column in question:

Example

```
\begin{tabular}{|l|p{8cm}|}  
\hline  
Number & Long text\\  
\hline  
1 & This is some really long text which  
    is going to end up going off the page if we're not careful---OH PHEW! \\  
\hline  
\end{tabular}
```

Example output

Number	Long text
1	This is some really long text which is going to end up going off the page if we're not careful—OH PHEW!

- How to span multiple columns:

Syntax

```
\multicolumn{number of columns}{alignment}{contents}
```

Example

```
\begin{tabular}{llrr}  
\hline  
    & & & \multicolumn{2}{c}{Style} \\  
    & & & \cline{3-4}  
Class & Sex & FS & CS \\  
\hline  
MMC   & M   & 4   & 31 \\  
      & F   & 0   & 0  \\  
LMC   & M   & 27  & 17 \\  
      & F   & 3   & 67 \\  
\hline  
\end{tabular}
```

Example output

Class	Sex	Style	
		FS	CS
MMC	M	4	31
	F	0	0
LMC	M	27	17
	F	3	67

- The package `multirow` provides an equivalent command for spanning multiple rows:

```
\multirow{number of rows}{width}{contents}
```

(Use `*` for the value of `width` to just use the natural width of the cell's contents.)

- Worth checking out the `booktabs` package, which makes your tables look a lot more professional:

Class	Sex	Style	
		FS	CS
MMC	M	4	31
	F	0	0
LMC	M	27	17
	F	3	67

- Note that the `table` environment is different: this marks off a table that will be numbered, have a caption, appear in a table of contents, etc. Tables and figures are ‘floats’, which means \LaTeX will display them where it thinks is best (the default preference is for the top of the page).

Example

```
\begin{table}[h]
\centering

\begin{tabular}{ll}
Foo & Bar \\ \hline
1   & 0.5 \\
2   & 274.2
\end{tabular}

\caption{An example of a table}
\label{table example}
\end{table}
```

Example output

Foo	Bar
1	0.5
2	274.2

Table 1: An example of a table

- Note the `\label{...}` command. This can be used to label all sorts of things, e.g. sections, figures, example sentences. Now that it has a label, we can refer to it by using the `\ref{...}` command, which will return the number associated with that object:

Example

Now we can talk about Table `\ref{table example}`,
and even mention that it is on p. `\pageref{table example}`.

Example output

Now we can talk about Table 1, and even mention that it is on p. 4.

- For more on tables, see the Wikibook, or one of the many online guides. This tutorial is quite thorough, for example: <http://www.texnology.com/teachingSamp.pdf>. The `booktabs` documentation (<http://bit.ly/2lp9QgT>) is also very good on table layout in general.

4 Graphics

- Use the `graphicx` package.
- Lots of powerful features (see documentation), but we’ll look at two frequently useful ones for now.
- Importing graphics:

Syntax

```
\includegraphics[attr1=val1, attr2=val2, ...]{filename}
```

Example

```
\includegraphics[scale=0.5]{bod}
```

Example output



- **Beware:** If you are compiling via `latex` rather than `pdflatex`, the image file *must* be an EPS (Encapsulated PostScript) file. If you are using `pdflatex` it can be a JPG, PNG, or PDF.
- Resizing diagrams, etc.:

Syntax

```
\scalebox{horizontal scale}[vertical scale]{object}
```

Example

```
\scalebox{0.5}{\includegraphics{bod}}
```

Example output



- Using a negative value for one of the scales allows you to flip the object.

Example

```
\scalebox{0.5}[-0.5]{\includegraphics{bod}}
```

Example output



- There is also the `resizebox` command for fitting to a particular size:

Syntax

```
\resizebox{horizontal length}{vertical length}{object}
```

- Use `!` as the horizontal or vertical length to have it scale with the other one.
- E.g. use `\resizebox{\linewidth}{!}{object}` to fit the object in question to the width of the line (great for making sure figures, trees, diagrams fit on the page).

5 Example numbers and glossing

- Various packages available (`gb4e`, `linguex`, `expex`, ...). Try them out and see what you like.
- `linguex` is very straightforward:

Example

Here is some text.

```
\ex. Here is an example.
```

```
\ex.
```

```
\a. Here is a sub-example.
```

```
\b. And another.
```

```
\b. *And a other.
```

```
\bg. Et voici un exemple avec une glose. \\
and here.is a.\textsc{masc} example with a.\textsc{fem} gloss \\
‘And here is one with a gloss.’
```

And here is some text which follows.

Example output

Here is some text.

(1) Here is an example.

(2) a. Here is a sub-example.
b. And another.
c. *And a other.
d. Et voici un exemple avec une glose.
and here.is a.MASC example with a.FEM gloss
‘And here is one with a gloss.’

And here is some text which follows.

- As well as the usual `\label{}` and `\ref{}` way of referring to examples, `linguex` gives you the handy commands `\Next` and `\Last` to refer to the next example coming up and the immediately preceding one. (Note that these automatically put parentheses around the example number, unlike `\ref{}`, which just returns the number.)

6 IPA

- Use the `tipa` package.
- This provides a bunch of macros and, more usefully, a lot of shortcuts that can be used inside a set of commands or environments:

1. `\textipa{...}`
2. `{\tipaencoding ...}`
3. `\begin{IPA}`
`...`
`\end{IPA}`

- According to the authors, “about 92% of the symbols can be inputted within three keystrokes” (!)
- A lot of the shortcuts are guessable, but check out the documentation for a full list (a good list is found in the appendix here: <http://bit.ly/217ZzTP>).

Example

```
\textipa{["f@nEtIks Iz ku:l]}
```

Example output

```
[fə'netɪks ɪz ku:l]
```

- Some special macros:

- `\;` is used for small capitals.

Example

```
\textipa{\;B \;E \;A \;H \;L \;R}
```

Example output

```
B E A H L R
```

- `\:` is used for retroflex symbols.

Example

```
\textipa{\:d \:l \:n \:r \:s \:z}
```

Example output

```
ɖ ɭ ɳ ʀ ʂ ʐ
```

- `\!` is used for implosives and clicks.

Example

```
\textipa{\!b \!d \!g \!j \!G \!o}
```

Example output

```
ɓ ɗ ɠ ɟ ʘ
```

- `*` is used for turned characters, miscellaneous special characters, and to escape the IPA coding for individual characters.

Example

```
\textipa{\*f \*r \*n \*l \*I \*E}
```

Example output

```
ɸ ɹ ɳ ɭ ɪ E
```

- **Note:** `linguex` doesn't play nicely with `tipa` and has to be loaded after it.

7 Syntactic trees

- Lots of different options here, but one easy-to-use package is `qtree`.

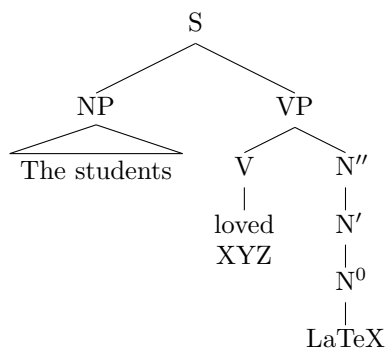
Example

```
\qtreecenterfalse
```

```
\Tree [.S \qroof{The students}.NP  

      [.VP [.V ] .V [.N\2 [.N\1 [ LaTeX ] .N\0 ] ] ] ]
```

Example output



- The default setting is for trees to be centred, and so if you want them not to be, you must declare `\qtreecenterfalse`. This will stay until the end of whatever environment you are in. If you want your trees to never be centred, load the package with the `[nocenter]` option:

```
\usepackage[nocenter]{qtree}
```

8 AVMs

- Use the `avm` package.
- This provides the `avm` environment, which allows you to draw AVMs fairly intuitively.

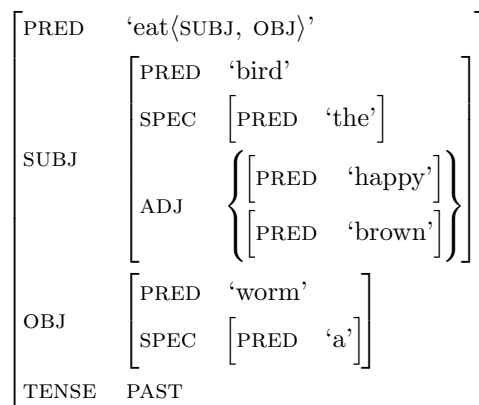
Syntax

```
\[ attr1 & val1 \\  
  attr2 & val2 \]
```

Example

```
\avmfont{\sc}  
\avmvalfont{\rm}  
  
\begin{avm}  
  
\[  
pred & 'eat\sc$\langle$subj, obj$\rangle$' \\  
subj & \[ pred & 'bird' \\  
      spec & \[pred & 'the' \]\\  
      adj  & \{ \[ pred & 'happy' \] \\  
              \[ pred & 'brown' \] \}  
      \] \\  
obj    & \[ pred & 'worm' \\  
      spec & \[ pred & 'a' \]  
      \] \\  
tense & \sc past  
\  
  
\end{avm}
```


Example output



9 Drawing lines

- Use `pstricks` (need to include `pst-node` as well).
- **Note:** Anything that relies on PostScript functionality cannot be produced by simply compiling with `pdflatex`. Instead, you have to use regular `latex`, which produces a DVI file, followed by `dvips`, which produces a PS file, then `ps2pdf`, to generate the PDF.
- TeXShop can do this for you in one step, under Typeset/TeX and DVI.
- (There are workarounds if this is a major issue for you for whatever reason, e.g. `tikz` and `tikz-qtree` allow you to do a lot of this with `pdflatex`.)
- The `\rnode` command creates nodes:

Syntax

`\rnode{name}{contents}`

- These are then joined by the various ‘node connection’ commands, which all start with `nc`, and all have the same syntax:

Syntax

`<nccommand>[options]{arrows}{node A}{node B}`

- See the `pstricks` documentation for all the details. Here are a few useful kinds:

Example

```
\rnode{wh}{What$_i$} did Mary see \rnode{trace}{$_t_i$}?
\ncbar[nodesep=3pt,angle=-90]{| ->}{trace}{wh}
```

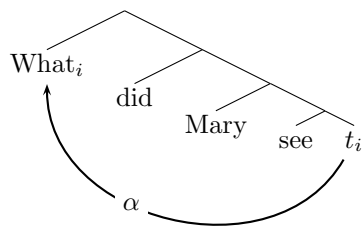
Example output

What_{*i*} did Mary see *t_i*?


Example

```
\qtreecenterfalse
\Tree [ \rnode{wh-tree}{What$_i$} [ did
      [ Mary [ see \rnode{trace-tree}{$_t_i$} ] ] ] ]
\nccurve[angleA=-120,angleB=-90,nodesep=3pt,ncurv=0.9]{->}{trace-tree}{wh-tree}
\ncput*[npos=0.6]{$_\alpha$}
```

Example output

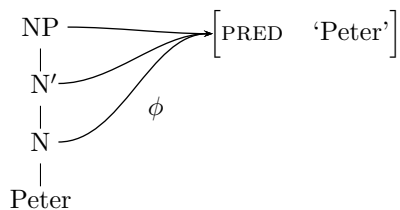


Example

```
\qtreetcenterfalse
\Tree [.\rnode{np}{NP} [.\rnode{nbar}{N\1} [.\rnode{n}{N} Peter ] ] ]
%
\hskip 5em
%
\begin{avm}
\rnode{fstruc}{\[\ \sc pred & 'Peter'\]}
\end{avm}

\nccurve[linewidth=0.5pt,nodesepA=3pt,angleB=-180]{->}{np}{fstruc}
\nccurve[linewidth=0.5pt,nodesepA=3pt,angleB=-180]{->}{nbar}{fstruc}
\nccurve[linewidth=0.5pt,nodesepA=3pt,angleB=-180]{->}{n}{fstruc}
\nbput*{\$\phi$}
```

Example output



Example

```
\begin{avm}
\[
pred & 'see'\\
focus & \rnode{focus}{\[\ pred & \sc'pro'\\
prontype & \sc wh \]} \\
subj & \[\ pred & 'Mary' \] \\
obj & \rnode{obj}{}
\]
\end{avm}

\ncangles[nodesepA=-1pt,lineararc=0.15,arm=0.5]{focus}{obj}
```

Example output

