

Some misunderstood or unknown L^AT_EX 2_ε tricks (II)

Luca Merciadri

1 Introduction

L^AT_EX is written in such a way that even skilled T_EXnicians sometimes learn new tricks, or come to problems or errors that they cannot easily solve, or explain. This time, our article will be divided in two (imaginary) parts: the first (Section 2) will treat

1. Avoiding erroneous references for floats, and the second (the rest) will give, as in my preceding paper (Merciadri, 2010), some ways to achieve special things in L^AT_EX 2_ε. These tricks are often explained on the Internet, but can be difficult to find. Specifically, the second part will speak about

2. Exporting spreadsheets into L^AT_EX,
3. Writing QED symbols as nicely as possible,
4. Counting the number of pages and tables,
5. Writing messages on would-be blank pages,
6. Writing dots in matrices,
7. Drawing logic gates,
8. Writing enumerations with textcircled numbers.

2 Avoiding erroneous references for floats

When writing a paper with L^AT_EX, the authors often let L^AT_EX do the cross-reference work. This results in a notable gain of time, because the work for every reference is automated. Consider a reference r declared using `\label{r}`. If r is cited, L^AT_EX will

1. Know its page number, which can be displayed and linked (if `hyperref` is used) using `\pageref{r}`,
2. Know its reference, meaning that it knows r 's place in the document structure.

But consider now a `tabular` environment placed in a `table` environment. Placing the `tabular` environment centered at the page is a good idea, thereby using

```
\begin{table}
  \begin{center}
    \begin{tabular}{cc}
      Text & Text
    \end{tabular}
  \end{center}
\end{table}
```

or its `\centering` variant. To link this table to a reference, one needs to place a `\label{reference}` in the `table` environment. One thing to remember is that `\label{}` always comes *after* `\caption{}`. That is, you must use neither

```
\begin{table}
\label{reference}
\caption{Name of the table.}
\begin{center}
  \begin{tabular}{cc}
    Text & Text
  \end{tabular}
\end{center}
\end{table}
```

nor

```
\begin{table}
\begin{center}
  \begin{tabular}{cc}
    Text & Text
  \end{tabular}
\end{center}
\label{reference}
\caption{Name of the table.}
\end{table}
```

You also need to *end* the `center` environment *before* using `\caption{}`. That is, you should not use

```
\begin{table}[!h]
\begin{center}
  \begin{tabular}{cc}
    Text & Text
  \end{tabular}
\caption{Name of the table.}
\end{center}
\label{tab:test}
\end{table}
```

but rather use

```
\begin{table}[!h]
\begin{center}
  \begin{tabular}{cc}
    Text & Text
  \end{tabular}
\caption{Name of the table.}
\label{tab:test}
\end{table}
```

Notice also the better reference: `tab:test` is clearer than `reference`. As `\centering` is local to the (most nested) environment which contains it, you can evidently replace the `center` environment by a simple `\centering`:

```
\begin{table}[!h]
\centering
\begin{tabular}{cc}
  Text & Text
\end{tabular}
\caption{Name of the table.}
\label{tab:test}
\end{table}
```

This concept is very important: some classes will not render a reference if the `\caption{}`–`\label{}` order is not respected. Even worse, others will put

unrelated reference numbers, such as `\thesection`, which can be disastrous: writing “thanks to Theorem x , we have [...]” is a good way not to lose the reader, but if x is a theorem number which does not exist, or which has no link with the citation, the whole paper might seem hastily written, or at least not edited, or simply confusing to the reader.

3 Exporting spreadsheets into L^AT_EX

It is sometimes desirable to export spreadsheets into L^AT_EX. It can be useful for many purposes, such as scientific experiments (collected data, for example), or financial reports. This is easily achieved with Calc2LaTeX (calc2latex.sourceforge.net).

4 Writing QED symbols as nicely as possible

When ending an environment, it is often desirable to let the reader know that the environment (property, theorem, etc.) has ended. It is often done using an elegant symbol: a QED symbol. This symbol might be anything you want, but such symbols are often small, and geometric shapes (squares, diamonds, ...). You can define many QED symbols. For example, you might define a QED symbol for each environment of your choice, or use the same one for every environment.

For example, to use \diamond as the QED symbol, you could simply use `\diamond`. The problem with such a simple approach is that, since you will end an environment with it (using `\diamond` or a homemade command such as `\myqedsymbol`), nothing guarantees that it will be placed correctly, *i.e.* that it will not begin a new line, or be placed at a new page.

So, you can use a tricky combination of `\hfill` and other commands to have your QED symbols placed as nicely as you want. Such a combination can be used to define a personal command such as `\qedsymbol`, like this:

```
\def\qedsymbol{%
  \mbox{}%
  \nolinebreak
  \hfill
  $\diamond$\% the qed symbol
  \medbreak
  \par
}
```

where the `\medbreak` is optional. You can then use `\qedsymbol`, or, better, a package which does it for you, such as `ntheorem`.

5 Counting the number of pages or tables

It might be interesting to know the number of pages of the current document. This can be done easily (MrUnix.de, 2010), e.g. by calling

```
\ref{TotPages}
```

which would give as output the number of pages. In an analogous way, one would for example want to know the number of tables of the document. This can be achieved using

```
\AbsTables
```

But before using the latter command, we must declare (in the preamble)

```
\newcommand*\OrigChapter{}
\let\OrigChapter\chapter
\newcounter{abstables}
\renewcommand*\chapter{%
  \addtocounter{abstables}{\value{table}}%
  \OrigChapter%
}
\newcommand*\AbsTables{0}
\makeatletter
\AtBeginDocument{%
  \AtEndDocument{%
    \addtocounter{abstables}{\value{table}}%
    \immediate\write\@mainaux{%
      \string\gdef\string\AbsTables{%
        \number\value{abstables}}%
    }%
  }%
}
```

```
\makeatother
```

For the former command, we need only

```
\usepackage{totpages}
```

in the preamble.

6 Writing messages on would-be blank pages

When reading a book, one sometimes encounters “blank” pages whose only text is some sentence like ‘This page intentionally left blank.’

This allows the reader to know that there has not been any printing issue with the book he is reading, and that the blank pages he sees are normal, and there for editorial reasons.

If you want such a message to appear in a L^AT_EX document whose class is `book`, you might redefine `\cleardoublepage` as follows:

```
\makeatletter
\def\cleardoublepage{\clearpage\if@twoside%
  \ifodd\c@page\else
  \vspace*{\fill}
  \hfill
  \begin{center}
    This page intentionally left blank.
  \end{center}
  \vspace{\fill}
  \thispagestyle{empty}
  \newpage}
```

```
\if@twocolumn\hbox{}\newpage\fi\fi\fi
}
```

```
\makeatother
```

in the preamble. The pages which would otherwise be left blank will now contain this message.

7 Writing dots in matrices

If sometimes happen to write special matrices, such as matrices where elements of one column could be moved to the next column, because other elements could replace them. An example is given by

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & \vdots \\ & & i \end{pmatrix}$$

This can be achieved using

```
\left(
\begin{array}{@{}cc@}{c@}{c@}{c@}{}
```

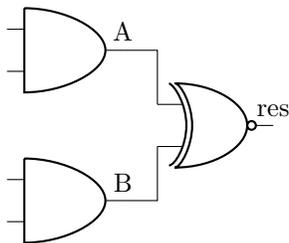
```
a & b & & c \\
d & e & & f \\
g & h & & \makebox[2\arraycolsep]{\smash{\vdots}}
& i
```

```
\end{array}
\right)
```

in a math environment. Thanks to Philipp Stephani for this trick.

8 Drawing logic gates

I found myself disappointed when looking for a simple way to *draw logic gates*. After much research, I found `circuitikz`, which allows you to write “traditional” circuits (that is, circuits with simple resistances, generators, inductors, ...), but also



This part of a circuit is created with

```
\begin{circuitikz} \draw
(0,2) node[and port] (myand1) {}
(0,0) node[and port] (myand2) {}
(2,1) node[xnor port] (myxnor) {}
(myand1.out) node[above] {A} -| (myxnor.in 1)
(myand2.out) node[above] {B} -| (myxnor.in 2)
(myxnor.out) node[above] {res};
\end{circuitikz}
```

and

```
\usepackage{tikz}
\usepackage{circuitikz}
```

in the preamble. There are other features to this package. This example was given to me by Massimo Redaelli. You might check the package’s manual (Redaelli, 2009) for other details.

9 Writing enumerations with textcircled numbers

Using the `enumerate` package, you can write

```
\begin{enumerate}[\textcircled{\arabic{enumi}}]
\item Item 1
\item Item 2
\item \ldots
\item Item $n$
\end{enumerate}
```

for such a result:

- ① Item 1
- ② Item 2
- ③ ...
- ④ Item n

This is simple to achieve, and might improve some enumerated lists. Do not forget to put

```
\usepackage{enumerate}
```

in the preamble.

- ◇ Luca Merciadri
University of Liège
Luca.Merciadri (at) student dot ulg dot
ac dot be
<http://www.student.montefiore.ulg.ac.be/~merciadri/>

References

- Merciadri, Luca. “A Practical Guide to L^AT_EX Tips and Tricks”. 2009.
- Merciadri, Luca. “Some misunderstood or unknown L^AT_EX 2_ε tricks”. *TUGboat* **31**(1), 76–78, 2010. <http://tug.org/TUGboat/31-1/tb97merciadri.pdf>.
- MrUnix.de. “Gesamtanzahl Seiten, Abbildungen usw.—mrunix.de”. 2010. <http://www.mrunix.de/forums/showthread.php?t=56716>.
- Redaelli, Massimo. “CircuiTikZ”. 2009. <http://mirror.ctan.org/graphics/pgf/contrib/circuitikz/doc/latex/circuitikz/circuitikzmanual.pdf>.