94 TUGboat, Volume 13 (1992), No. 1

Review. Leban requires as input the number of disks and the first tower as a sequence of one-digit numbers. This means that creation of the initial tower is not part of the paper. Furthermore, Leban develops a set of Lisp-like functions for \text{T}_{\text{f}}\text{X}. Apparently Knuth’s list macros, \text{\text{T}_{\text{f}}\text{X}book} Appendix D.2, have been overlooked. Leban’s small list processing system is an example of reinventing the wheel, with the concept of an active list separator absent. As a consequence the printing of the towers is done by recursion which is not necessary when using Knuth’s list separator appropriately to ‘execute’ the list.


\[\text{o Kees van der Laan}
\text{Hunzeweg 57, 9893PB}
\text{Garnwerd (Gr), The Netherlands}
\text{cgl@rug.nl}\]

In any case, the footnote marks are typeset as exponents in math mode so that the math italic font is used by default. If you want the regular roman font, or any other font, you have to request it explicitly:

\[
\text{\texttt{\textbackslash renewcommand\{\textbackslash tempfootnote\}\%}}
\text{\texttt{\{\textbackslash \textup{\textbackslash alph\{mpfootnote\}\}\}}}\]

2. The double spacing scheme suggested by J. Colmenares does not work if the commands—\textbackslash single and \textbackslash double—are issued while in \textbackslash normalsize. In fact, \textbackslash baselinestretch operates only when the size is changed, but \textbackslash normalsize does not actually change size if the size is already normal. In order to have these commands work, you need to type: \texttt{\textbackslash small\textbackslash single or \textbackslash small\textbackslash double}.

By doing this, the \textbackslash normalsize command embedded in \textbackslash single and in \textbackslash double acts as it should with the new value of \textbackslash baselinestretch becoming active.

A better command, that does not require changes to size in advance and operates also when in \textbackslash normalsize, unfortunately requires the category recoding of 9. The macro appears below:

\[
\text{\texttt{\textbackslash newcommand\{\textbackslash single\}\%}}
\text{\texttt{\{\textbackslash baselinestretch\{1.0\}\}}}\]
\[
\text{\texttt{\textbackslash newcommand\{\textbackslash double\}\%}}
\text{\texttt{\{\textbackslash baselinestretch\{1.5\}\}}}\]
\textbackslash makeatletter
\texttt{\textbackslash newcommand\{\textbackslash leadingstretch\}[1]\%}
\texttt{\{\textbackslash let\{\textbackslash tempa\}\textbackslash currsize}
\texttt{\textbackslash let\{\textbackslash currsize\}\textbackslash empty}
\texttt{\textbackslash def\textbackslash baselinestretch\{#1\}\textbackslash tempa}
\texttt{\textbackslash makeatother}\]

Footnotes: Problems and Solutions

Sometimes you need to attach footnotes to entries in tables. If you enclose a \texttt{\textbackslash tabular} environment within a \texttt{\textbackslash minipage} environment within a \texttt{\textbackslash table} environment, your footnotes float around together with your table.

But if you have to tag two or more table entries with the same footnote number (or letter) while you have one footnote text, you cannot use \textbackslash footnotemark\{\texttt{\textbackslash number}\} several times in the table and \textbackslash footnotetext\{\texttt{\textbackslash number}\} \{text\} after the \texttt{\textbackslash end\{\textbackslash tabular\} and before the \texttt{\textbackslash end\{\textbackslash minipage\}\} commands.

In fact if you do so, you will notice that the footnote text will be tagged with a number while the table entries are tagged with letters (default).
There is no simple means to go around this problem (at least I did not find a simple way out) while preserving the feature devised by L. Lamport of separating the footnote marks from the footnote texts, as he explains on pages 99 and 156 of the \LaTeX{} book.

1st solution

Explicitly place multiple footnote marks as math exponents in the table entries. Afterwards typeset the footnote texts using only \verb|[footnotetext{(number)}]| with its optional argument that agrees with the exponents that were set.

2nd solution

- Redefine a new boolean variable, say \verb|tablernote:|
  \verb|\newif {tablernote}|
  % \tablernote is false by default
  \verb|\iftable{tablernote}|
- Redefine \verb|\table| so that it sets \verb|tablernote|.
- Redefine the \verb|\footnotemark| and \verb|\@footnotemark| commands so they operate on the mpfootnote counter, instead of footnote, if \tablernote is true.
- Tag all your table entries that required tagging with the same mark (except the first one, which is marked with the full \verb|\footnote| command) with the \verb|\footnotemark{|(number)}| that makes use of its optional argument.

3rd solution

Define a new environment. Locally redefine \verb|\@footnote| and \verb|\thefootnote| to be equivalent to \verb|\ifmpfootnote| and \verb|\thempfootnote| respectively, using \verb|\let|.

I used the first two solutions. The second one is definitely better, but it requires that you know where you put your hands within the internal \LaTeX{} macros. The third solution seems very simple.

Maybe someone has an even better solution?

Claudio BECCARI
Department of Electronics
Institute of Technology of Turin, Italy
Corso Duca degli Abruzzi 24
110129 – TORINO, Italy
E-MAIL: beccari@itopoli.bitnet

Errata: “See also” indexing with Makeindex

Harold Thimbleby

In TUGboat 12, no. 2 (page 290) I gave the \LaTeX{} definitions to enable an author to obtain ‘see also’ entries in their index. I am grateful to Professor John C. Slattery of Texas A&M University for pointing out that they did not work.

The following correction works for me (using Textures and \LaTeX{} 2.09), but not for Slattery who is using a NeXT, though the same version of \LaTeX{}:

\begin{verbatim}
def\subsee#1#2{\do seen also} #1 \def\nosee#1() % consume the page number 
def\seealso#1#2\index(C#1|zzzzzz)
\def\seealso{Scheme)(LISP}
\end{verbatim}

The intention is, given the definitions as shown above, and supposing index entries for “Scheme” \verb|\index{Scheme}| occur on pages 147 and 401, this is how \verb|\seealso{Scheme}{LISP}| would end up in the index:

\begin{verbatim}
Scheme. 147, 401
see also LISP
\end{verbatim}

If you have the problems reported by Slattery, \verb|\seealso| must written out in full with you manually replacing the parameters \#1 and \#2 with what you want.

I made two errors in the original note: First, I published a fragment of \LaTeX{} without testing it exactly as it appeared in print. The second error was conceptual. I naïvely forgot that a \TeX{} definition is referentially opaque: I had assumed that given \verb|\def\seealso{x},| then \verb|\seealso| can be written for \verb|x| (with the exception of various cases where \verb|x| contains things like \verb|futurelet|). In my case I had checked \verb|x| but not the form \verb|\seealso| that I used in the article. I had been fooled by the innocent appearance of \verb|\index{argument}| — and I had not appreciated the \LaTeX{} manual’s remark that \verb|\index| should not appear inside another command’s argument, as it does here with \verb|\def|.

I apologize for inconvenience caused, and I will look forward to any suggestions for a general solution. Is there any way for macros like \LaTeX{}’s \verb|\index| to detect when they are being used improperly?

○ Jackie Damrau
SSC Laboratory
Mail Stop 1011
2550 Beckleymeade Avenue
Dallas, TX
e-mail: damrau@sscuvx1.ssc.gov

○ Harold Thimbleby
Stirling University
Stirling
Scotland, FK9 4LA