The delarray package

David Carlisle
2014/10/28

1 Examples

The addition to array.sty added in delarray.sty is a system of implicit \left \right pairs. If you want an array surrounded by parentheses, you can enter:
\begin{array}{{cc}} \ldots \\
\begin{pmatrix} a & b \\ c & d \end{pmatrix}
\end{array}

Similarly if an environment equivalent to Plain TeX’s \cases could be defined by:
\begin{array}{{ll}} \ldots \\
f(x) = \begin{cases} 0 & \text{if } x = 0 \\
\sin(x)/x & \text{otherwise} \end{cases}
\end{array}

Here L is supposed to denote a column of left aligned L-R text. It may be defined via:\newcolumntype{L}{>{$}l<{$}}, as discussed in array.sty. Note that as the delimiters must always be used in pairs, the ‘.’ must be used to denote a ‘null delimiter’.

This feature is especially useful if the [t] or [b] arguments are also used. In these cases the result is not equivalent to surrounding the environment by \left...\right, as can be seen from the following example:

\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
not \begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array}

\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
\text{not} \begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array}

\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
\begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array} \\
\quad \text{not} \quad \begin{array}{{tc}} 1 \\
\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
\end{array}

*This file has version number v1.01, last revised 2014/10/28.
\left(\begin{array}{c} 1
2
3 \end{array}\right)
\left(\begin{array}{c} 1
2
3 \end{array}\right)
\left(\begin{array}{c} 1
2
3 \end{array}\right)

\section{The Macros}

\begin{enumerate}
\item (*package)
\item \texttt{\\@tabarray} This macro tests for an optional bracket and then calls up \texttt{\@\@array} or \texttt{\@\@array[c]} (as default).
\item \texttt{\\@tabarray{\@ifnextchar[\{\@\@array\}{\@\@array[c]}}
\item \texttt{\@\@array} This macro tests for an optional delimiter before the left brace of the main preamble argument. If there is no delimiter, \texttt{\@arrayleft} and \texttt{\@arrayright} are made a no-ops, and \texttt{\@array} is called with the positional argument. Otherwise call \texttt{\@del@array}.
\item \texttt{\@del@array[#1]{\@ifnextchar[\{\@\@array\}{\@\@array[c]}}}
\item \texttt{\@\@array[#1]}\texttt{\{\@ifnextchar[\{\@\@array\}{\@\@array[c]}}
\item \texttt{\{\\let\@arrayleft\relax\let\@arrayright\relax\arrayright\arrayleft\array[#1]\}%}
\item \texttt{\\@del@array[#1]}
\end{enumerate}

\@\@array We now know that we have an \texttt{array} (or \texttt{tabular}) with delimiters.

\begin{enumerate}
\item \texttt{\\@del@array[#1]2\#3\#4{}}
\item The following line is completely redundant but it does catch errors involving delimiters before the processing of the alignment begins. A common error is likely to be omitting the "." in a \texttt{cases}-type construction. This causes the first token of the alignment to be gobbled, possibly causing lots of spurious errors before the cause of the error, the missing delimiter, is discovered as \texttt{\@arrayright} puts the alignment and the delimiters together.
\item \texttt{\setbox\z@\hbox{$\left#2\right#4$}}%
\item In the case of a \texttt{c} argument we do not need to rebox the alignment, so we can define \texttt{\@arrayleft} and \texttt{\@arrayright} just to insert the delimiters.
\item \texttt{\if#1c\def\@arrayleft{\left#2}\def\@arrayright{\right#4}\%}
\item Otherwise we (should) have a \texttt{[t]} or \texttt{[b]} argument, so first we store the alignment, without delimiters in box0.
\item \texttt{\else\def\@arrayleft{\setbox\z@\%}
\item Then after the alignment is finished:
\item \texttt{\def\@arrayright{}}
\item Calculate the amount the box needs to be lowered (this will be negative in the case of \texttt{[b]}). A little bit of arithmetic cf. the \texttt{\TeX}Book, Appendix G, rule 8. We calculate the amount this way, rather than just taking the difference between the depth of box0 and the depth of the box defined below, as the depth of that box may be affected by the delimiters if \texttt{\delimitershortfall} or \texttt{\delimiterfactor} have non-standard values.
\item \texttt{\dimen@=\\ht\z@}
\item \texttt{\advance\dimen@-\\ht\z@}
\item \texttt{\divide\dimen@by\tw@}
\item \texttt{\advance\dimen@by\fontdimen22\textfont\tw@}
\end{enumerate}
Now lower the alignment and the delimiters into place.

\lower\dimen0\hbox{$\left\{\vcenter{\unvbox\z@}\right\}$}\%  
End the \if#1c
\fi
Now that we have defined \@arrayleft and \@arrayright, call \@array.
\@array[#1]{#3}

(//package)

2.1 newarray.sty

All the features of the old newarray style option have been merged into the array or delarray options.