Le package delarray*

David Carlisle
carlisle@cs.man.ac.uk

Traduit de l'anglais par :
Jean-Pierre Drucbert

1994/03/14

1 Exemples

Ce package ajoute au package `array` un système de paires implicites `\left`–`\right`. Bien sûr ce package sera chargé si nécessaire.

Si vous désirez un tableau entouré de parenthèses, vous pouvez taper :
`\begin{array}({cc})` ...

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

De la même manière, on peut définir un environnement équivalent au `\cases` du Plain TeX de la façon suivante :
`\begin{array}\{{lL}.` ...

$$f(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ \sin(x)/x & \text{dans les autres cas} \end{array} \right.$$

Ici, `L` désigne une colonne de texte LR[1]. On peut le définir comme suit : `\newcolumntype{L}{>{$}l<{$}}`, comme expliqué dans `array.sty`. Prenez note que les délimiteurs doivent toujours être utilisés par paires, le '`.`' doit être utilisé pour indiquer un 'délimiteur nul'.

Tout délimiteur peut apparaître dans ces positions. Donc un utilisateur du Plain TeX reconnaîtra les environnements suivants :

```
\newenvironment{cases}{\begin{array}\{{lL}.}{\end{array}}
\newenvironment{matrix}{\begin{array}{*{20}{c}}}{\end{array}}
\newenvironment{pmatrix}{\begin{array}({*{20}{c}})}{\end{array}}
```

Ce dispositif est particulièrement utile si les arguments `[t]` ou `[b]` sont aussi utilisés. Dans ces cas le résultat n'est pas équivalent à celui obtenu en entourant l'environnement par `\left`...`\right`, comme vous pouvez le voir dans les exemples suivants :

$$\begin{array}[b]({c})1\\2\\3\end{array} \begin{array}[c]({c})1\\2\\3\end{array} \begin{array}[t]({c})1\\2\\3\end{array} \quad \text{et non} \quad \left(\begin{array}[b]{c}1\\2\\3\end{array}\right) \left(\begin{array}[c]{c}1\\2\\3\end{array}\right) \left(\begin{array}[t]{c}1\\2\\3\end{array}\right)$$

*Ce fichier a le numéro de version v1.01, dernière mise à jour le 1994/03/14.

[1] NdT : Il s'agit d'un mode où le texte est écrit de gauche à droite sur une seule ligne.

```
\begin{array}[t]({c}) 1\\2\\3 \end{array}
\begin{array}[c]({c}) 1\\2\\3 \end{array}
\begin{array}[b]({c}) 1\\2\\3 \end{array}
\quad\mbox{et non}\quad
\left(\begin{array}[t]{c} 1\\2\\3 \end{array}\right)
\left(\begin{array}[c]{c} 1\\2\\3 \end{array}\right)
\left(\begin{array}[b]{c} 1\\2\\3 \end{array}\right)
```

2 Les Macros (section non traduite)

```
⟨*package⟩
```

```
\RequirePackage{array}[1994/02/03]
```

`\@tabarray` This macro tests for an optional bracket and then calls up `\@@array` or `\@@array[c]` (as default).

```
\def\@tabarray{\@ifnextchar[{\@@array}{\@@array[c]}}
```

`\@@array` This macro tests for an optional delimiter before the left brace of the main preamble argument. If there is no delimiter, `\@arrayleft` and `\@arrayright` are made a no-ops, and `\@array` is called with the positional argument. Otherwise call `\@del@array`.

```
\def\@@array[#1]{\@ifnextchar\bgroup
  {\let\@arrayleft\relax\let\@arrayright\relax\@array[#1]}%
  {\@del@array[#1]}}
```

`\@del@array` We now know that we have an `array` (or `tabular`) with delimiters.

```
\def\@del@array[#1]#2#3#4{%
```

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 omiting the '.' in a `\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 `\@arrayright` puts the alignment and the delimiters together.

```
  \setbox\z@\hbox{$\left#2\right#4$}%
```

In the case of a 'c' argument we do not need to rebox the alignment, so we can define `\@arrayleft` and `\@arrayright` just to insert the delimiters.

```
  \if#1c\def\@arrayleft{\left#2}\def\@arrayright{\right#4}%
```

Otherwise we (should) have a `[t]` or `[b]` argument, so first we store the alignment, without delimiters in box0.

```
  \else\def\@arrayleft{\setbox\z@}%
```

Then after the alignment is finished:

```
  \def\@arrayright{%
```

Calculate the amount the box needs to be lowered (this will be negative in the case of `[b]`). A little bit of arithmetic cf. the TEXBook, 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 `\delimitershortfall` or `\delimiterfactor` have non-standard values.

```
    \dimen@=\dp\z@
    \advance\dimen@-\ht\z@
    \divide \dimen@ by \tw@
    \advance\dimen@ by\fontdimen22 \textfont\tw@
```

Now lower the alignment and the delimiters into place.

```
    \lower\dimen@\hbox{$\left#2\vcenter{\unvbox\z@}\right#4$}}%
```

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.