

The `mathstyle` package

Morten Høgholm
mh.ctan@gmail.com

2008/07/23 v0.85

User's guide

This package exists for two reasons:

- The primitive operations for creating a super- or subscript in \TeX work almost as if `^` and `_` are macros taking an argument. However, that is not quite the case, and some things that you'd expect to work don't (e.g., `^{\cong}`) whereas others which you'd think shouldn't work actually do (such as `^{\mathsf{s}}`). We do everyone a favor if it behaves consistently, i.e., if the superscript and subscript operations act as if they are macros taking exactly one argument.
- Because the \TeX math typesetting engine uses infix notation for fractions, one has to use `\mathchoice` or `\mathpalette` whenever trying to do anything requiring boxing or measuring math. This creates problems for loading fonts on demand as the font loading mechanism has to load fonts for all styles without even knowing if the font is going to be used. Getting the timing of `\mathchoice` right can be tricky as well. Since \LaTeX does not promote the primitive infix notation, this package keeps track of a current `mathstyle` parameter.

1 Some usage tips

If you want to use this package with `amsmath`, it is important `mathstyle` is loaded *after* `amsmath`.

The current `mathstyle` is stored in the variable `\mathstyle`. The command `\currentmathstyle` can be used to switch to the mode currently active. Below is shown how the macro `\mathrlap` from `mathtools` is implemented without knowing about the current `mathstyle` using `\mathpalette`.

```
\providecommand*\mathrlap[1][]{%  
  \ifx\@empty#1\@empty  
    \expandafter \mathpalette \expandafter \@mathrlap
```

```

\else
  \expandafter \mathrlap \expandafter #1%
\fi}
\providecommand*\mathrlap #1#2{{}\rlap{$\mathrlap{#1#2}$}}

```

The same definition using `\currentmathstyle` from this package.

```

\providecommand*\mathrlap[2][]{%
  #1 {} \rlap{$\mathrlap{\currentmathstyle {#2}$}}
}

```

Implementation

```

1 <*package>
2 \ProvidesPackage{mathstyle}[2008/07/23 v0.85]

\@saveprimitive A straight copy from breqn, see implementation details there. Of course, with a
recent pdfTeX (v1.40+), one can just use \primitive to get the original. We will
implement that some day.

3 \providecommand\@saveprimitive[2]{%
4   \begingroup
5   \edef\@tempa{\string#1}\edef\@tempb{\meaning#1}%
6   \ifx\@tempa\@tempb \global\let#2#1%
7   \else
8     \edef\@tempb{\meaning#2}%
9     \ifx\@tempa\@tempb
10      \else \@saveprimitive@a#1#2%
11      \fi
12    \fi
13  \endgroup
14 }

15 \providecommand\@saveprimitive@a[2]{%
16   \begingroup
17   \def\@tempb##1#1##2{\edef\@tempb{##2}\@car{}}%
18   \@tempb\nullfont{select font nullfont}%
19   \topmark{\string\topmark:}%
20   \firstmark{\string\firstmark:}%
21   \botmark{\string\botmark:}%
22   \splitfirstmark{\string\splitfirstmark:}%
23   \splitbotmark{\string\splitbotmark:}%
24   #1{\string#1}%
25   \edef\@tempa{\expandafter\strip@prefix\meaning\@tempb}%
26   \edef\@tempb{\meaning#1}%
27   \ifx\@tempa\@tempb \global\let#2#1%
28   \else
29     \PackageError{mathstyle}%
30       {Unable to properly define \string#2; primitive

```

```

31      \noexpand#1no longer primitive}\@eha
32      \fi
33    \fi
34  \endgroup
35 }

```

Do initial `\chardef` of `\mathstyle`.

```
36 \chardef\mathstyle=\z@
```

Save the four style changing primitives, `\mathchoice` and the fraction commands.

```

37 \@saveprimitive\displaystyle\@@displaystyle
38 \@saveprimitive\textstyle\@@textstyle
39 \@saveprimitive\scriptstyle\@@scriptstyle
40 \@saveprimitive\scriptscriptstyle\@@scriptscriptstyle
41 \@saveprimitive\mathchoice\@@mathchoice
42 \@saveprimitive\over\@@over
43 \@saveprimitive\atop\@@atop
44 \@saveprimitive\above\@@above
45 \@saveprimitive\overwithdelims\@@overwithdelims
46 \@saveprimitive\atopwithdelims\@@atopwithdelims
47 \@saveprimitive\abovewithdelims\@@abovewithdelims

```

Then we redeclare the four style changing primitives.

```

48 \DeclareRobustCommand{\displaystyle}{%
49   \@@displaystyle \chardef\mathstyle\z@}
50 \DeclareRobustCommand{\textstyle}{%
51   \@@textstyle \chardef\mathstyle\@ne}
52 \DeclareRobustCommand{\scriptstyle}{%
53   \@@scriptstyle \chardef\mathstyle\tw@}
54 \DeclareRobustCommand{\scriptscriptstyle}{%
55   \@@scriptscriptstyle \chardef\mathstyle\thr@@}

```

First we get the primitive operations. These should have been control sequences in \TeX just like operations for `begin math`, `end math`, `begin display`, `end display`.

```

56 \begingroup \catcode'\^=7\relax \catcode'\_ =8\relax % just in case
57 \lowercase{\endgroup}
58 \let\@@superscript=\^ \let\@@subscript=\_
59 }%

```

If we enter a sub- or superscript the `\mathstyle` must be adjusted. Since all is happening in a group, we do not have to worry about resetting.

```

60 \def\subsupstyle{%
61   \ifnum\mathstyle<\tw@ \chardef\mathstyle\tw@
62   \else \chardef\mathstyle\thr@@
63   \fi
64 }

```

Provide commands with meaningful names for the two primitives, cf. `\mathrel`.

```

65 \let\mathsup=\@@superscript
66 \let\mathsub=\@@subscript

```

`\sb` and `\sp` are then defined as macros.

```
67 \def\sb#1{\mathsub{\protect\subsupstyle#1}}%
68 \def\sp#1{\mathsup{\protect\subsupstyle#1}}%
```

`\mathchoice` is now just a switch. Note that this redefinition does not allow the arbitrary *<filler>* of the `TEX` primitive. Very rarely used anyway.

```
69 \def\mathchoice{%
70   \relax\ifcase\mathstyle
71     \expandafter\@firstoffour
72   \or
73     \expandafter\@secondoffour
74   \or
75     \expandafter\@thirdoffour
76   \else
77     \expandafter\@fourthoffour
78   \fi
79 }
```

Helper macros.

```
80 \providecommand\@firstoffour[4]{#1}
81 \providecommand\@secondoffour[4]{#2}
82 \providecommand\@thirdoffour[4]{#3}
83 \providecommand\@fourthoffour[4]{#4}
```

The fractions. Note that this uses the same names as in `amsmath`. Much the same except here they call `\fracstyle`.

```
84 \DeclareRobustCommand\genfrac[6]{%
85   {\fracstyle #1%
86     {\begingroup #5\endgroup
87       \csname @@\ifx\maxdimen#4\maxdimen over\else above\fi
88       \if @#2@\else withdelims\fi\endcsname #2#3#4\relax
89     #6}%
90   }%
91 }
92 \renewcommand{\frac}{\genfrac{}{}{}{}}
93 \providecommand{\dfrac}{}
94 \providecommand{\tfrac}{}
95 \renewcommand{\dfrac}{\genfrac\displaystyle{}{}{}}
96 \renewcommand{\tfrac}{\genfrac\textstyle{}{}{}}
97 \providecommand{\binom}{}
98 \providecommand{\tbinom}{}
99 \providecommand{\dbinom}{}
100 \renewcommand{\binom}{\genfrac{}{}{0pt}}
101 \renewcommand{\dbinom}{\genfrac\displaystyle{}{0pt}}
102 \renewcommand{\tbinom}{\genfrac\textstyle{}{0pt}}
```

The `\fracstyle` command is a switch to go one level down but no further than three.

```
103 \def\fracstyle{\ifcase\mathstyle
104   \chardef\mathstyle=\@ne
105   \or
```

```

106   \chardef\mathstyle=\tw@
107   \else
108     \chardef\mathstyle=\thr@@
109   \fi
110 }

```

The `\currentmathstyle` checks the value of `\mathstyle` and switches to it so it is in essence the opposite of `\displaystyle` and friends.

```

111 \def\currentmathstyle{%
112   \ifcase\mathstyle
113     \@@displaystyle
114   \or
115     \@@textstyle
116   \or
117     \@@scriptstyle
118   \or
119     \@@scriptscripstyle
120   \fi}

```

Finally, we declare the package options.

```

121 \DeclareOption{mathactivechars}{%
122   \catcode'\^=12\relax
123   \catcode'\_ =12\relax}
124 \DeclareOption{activechars}{%
125   \catcode'\^=13\relax
126   \catcode'\_ =13\relax}
127 \DeclareOption{noactivechars}{%
128   \catcode'\^=7\relax
129   \catcode'\_ =8\relax}
130 \ExecuteOptions{mathactivechars}
131 \ProcessOptions\relax

```

WSPR: Set up the active behaviours: (this is set even in the `noactivechars` case but they are never activated. no worries?)

```

132 \ifnum\catcode'\^=13\relax
133   \let^=\sp \let_=\sb
134 \else
135   \mathcode'\^="8000\relax
136   \mathcode'\_ ="8000\relax
137   \begingroup
138     \catcode'\^=\active
139     \catcode'\_ =\active
140     \global\let^=\sp
141     \global\let_=\sb
142   \endgroup
143 \fi
144 \end{package}

```