

The `mathstyle` package

Authors: Michael J. Downes, Morten Høgholm
Maintained by Morten Høgholm, Will Robertson
Feedback: <https://github.com/wspr/breqn/issues>

2017/01/27 0.98e

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{$\m@th#1{#2}$}}

```

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

```

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

```

1.1 Package options

This package has one set of options affecting the `_` and `^` characters:

- `\usepackage[mathactivechars]{mathstyle}`

This is the default behaviour. Here, `_` and `^` are made into harmless characters in text mode and behave as expected (for entering sub/superscript) when inside math mode. Certain code that assumes the catcodes of these characters may get confused about this; see below for a possible fix.

- `\usepackage[activechars]{mathstyle}`

With this option, `_` and `^` are made into active characters for entering sub/superscript mode in all cases—therefore, in text mode they will produce a regular error (‘Missing \$ inserted’) indicating they are being used out of place.

- `\usepackage[noactivechars]{mathstyle}`

This is the option most like to solve any compatibility problems. Here, `_` and `^` retain their regular catcodes at all times and behave in their default fashion. **However**, certain other features of this package (such as `\currentmathstyle` inside a subscript) will then fail to work, so only use this option as a last resort.

Implementation

```
1 <*package>
```

`\@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.

```
2 \providecommand\@saveprimitive[2]{%
3 \begingroup
4 \edef\@tempa{\string#1}\edef\@tempb{\meaning#1}%

```

```

5 \ifx\@tempa\@tempb \global\let#2#1%
6 \else
7 \edef\@tempb{\meaning#2}%
8 \ifx\@tempa\@tempb
9 \else \@saveprimitive@a#1#2%
10 \fi
11 \fi
12 \endgroup
13 }
14 \providecommand\@saveprimitive@a[2]{%
15 \begingroup
16 \def\@tempb##1##2{\edef\@tempb{##2}\@car{}}%
17 \@tempb\nullfont{select font nullfont}%
18 \topmark{\string\topmark:}%
19 \firstmark{\string\firstmark:}%
20 \botmark{\string\botmark:}%
21 \splitfirstmark{\string\splitfirstmark:}%
22 \splitbotmark{\string\splitbotmark:}%
23 #1{\string#1}%
24 \@nil % for the \@car
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 }

```

`\everydisplay` We need to keep track of whether we're in inline or display maths, and the only way to do that is to add a switch inside `\everydisplay`. We act sensibly and preserve any of the previous contents of that token register before adding our own code here. As we'll see in a second, Lua \TeX provides a native mechanism for this so we don't need any action in that case. (Various other parts of the code also need to have different paths for Lua \TeX use.)

```

36 \begingroup\expandafter\expandafter\expandafter\endgroup
37 \expandafter\ifx\csname directlua\endcsname\relax
38 \everydisplay=\expandafter{\the\everydisplay\chardef\mathstyle\z@}
39 \fi

```

`\mathstyle` A counter for the math style: 0–display, 2–text, 4–script, 6–scriptscript. The logic is that display maths will explicitly set `\mathstyle` to zero (see above), so by default it is set to the 'text' maths style. With Lua \TeX there is a primitive to do the same so it just has to be enabled. Note that in all cases we use Lua \TeX -like numbering for the states.

```

40 \begingroup\expandafter\expandafter\expandafter\endgroup

```

```

41 \expandafter\ifx\csname directlua\endcsname\relax
42   \chardef\mathstyle\@ne
43 \else
44   \directlua{tex.enableprimitives("", {"mathstyle"})}
45 \fi

```

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

```

46 \@saveprimitive\displaystyle\@@displaystyle
47 \@saveprimitive\textstyle\@@textstyle
48 \@saveprimitive\scriptstyle\@@scriptstyle
49 \@saveprimitive\scriptscriptstyle\@@scriptscriptstyle
50 \@saveprimitive\mathchoice\@@mathchoice
51 \@saveprimitive\over\@@over
52 \@saveprimitive\atop\@@atop
53 \@saveprimitive\above\@@above
54 \@saveprimitive\overwithdelims\@@overwithdelims
55 \@saveprimitive\atopwithdelims\@@atopwithdelims
56 \@saveprimitive\abovewithdelims\@@abovewithdelims

```

Then we redeclare the four style changing primitives: set the value of `\mathstyle` if Lua \TeX is not in use.

```

57 \begingroup\expandafter\expandafter\expandafter\endgroup
58 \expandafter\ifx\csname directlua\endcsname\relax
59   \DeclareRobustCommand{\displaystyle}{%
60     \@@displaystyle \chardef\mathstyle\z@}
61   \DeclareRobustCommand{\textstyle}{%
62     \@@textstyle \chardef\mathstyle\tw@}
63   \DeclareRobustCommand{\scriptstyle}{%
64     \@@scriptstyle \chardef\mathstyle4 }
65   \DeclareRobustCommand{\scriptscriptstyle}{%
66     \@@scriptscriptstyle \chardef\mathstyle6 }
67 \fi

```

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`.

```

68 \begingroup \catcode'\^=7\relax \catcode'\_ =8\relax % just in case
69 \lowercase{\endgroup
70 \let\@@superscript=\^ \let\@@subscript=_
71 }%
72 \begingroup \catcode'\^=12\relax \catcode'\_ =12\relax % just in case
73 \lowercase{\endgroup
74 \let\@@superscript@other=\^ \let\@@subscript@other=_
75 }%

```

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. We can't tell the difference between cramped and non-cramped styles unless Lua \TeX is in use, in which case this command is a no-op.

```

76 \begingroup\expandafter\expandafter\expandafter\endgroup

```

```

77 \expandafter\ifx\csname directlua\endcsname\relax
78   \def\subsupstyle{%
79     \ifnum\mathstyle<5 \chardef\mathstyle4 %
80     \else \chardef\mathstyle6 %
81     \fi
82   }
83 \else
84   \def\subsupstyle{}
85 \fi

```

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

```

86 \let\mathsup=\@@superscript
87 \let\mathsub=\@@subscript

```

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

```

88 \def\sb#1{\mathsub{\protect\subsupstyle#1}}%
89 \def\sp#1{\mathsup{\protect\subsupstyle#1}}%

```

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

```

90 \def\mathchoice{%
91   \relax\ifcase\numexpr\mathstyle\relax
92     \expandafter\@firstoffour % Display
93   \or
94     \expandafter\@firstoffour % Cramped display
95   \or
96     \expandafter\@secondoffour % Text
97   \or
98     \expandafter\@secondoffour % Cramped text
99   \or
100    \expandafter\@thirdoffour % Script
101  \or
102    \expandafter\@thirdoffour % Cramped script
103  \else
104    \expandafter\@fourthoffour % (Cramped) Scriptscript
105  \fi
106 }

```

Helper macros.

```

107 \providecommand\@firstoffour[4]{#1}
108 \providecommand\@secondoffour[4]{#2}
109 \providecommand\@thirdoffour[4]{#3}
110 \providecommand\@fourthoffour[4]{#4}

```

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

```

111 \DeclareRobustCommand\genfrac[6]{%
112   {#1\fracstyle
113   {\begingroup #5\endgroup
114     \csname @@\ifx\maxdimen#4\maxdimen over\else above\fi
115     \if @#2@\else withdelims\fi\endcsname #2#3#4\relax

```

```

116     #6}%
117   }%
118 }

119 \renewcommand{\frac}{\genfrac{}{}{}{}}
120 \providecommand{\dfrac}{}
121 \providecommand{\tfrac}{}
122 \renewcommand{\dfrac}{\genfrac\displaystyle{}{}{}}
123 \renewcommand{\tfrac}{\genfrac\textstyle{}{}{}}
124 \providecommand{\binom}{}
125 \providecommand{\tbinom}{}
126 \providecommand{\dbinom}{}
127 \renewcommand{\binom}{\genfrac{}{}{0pt}}
128 \renewcommand{\dbinom}{\genfrac\displaystyle{}{0pt}}
129 \renewcommand{\tbinom}{\genfrac\textstyle{}{0pt}}

```

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

```

130 \begingroup\expandafter\expandafter\expandafter\endgroup
131 \expandafter\ifx\csname directlua\endcsname\relax
132   \def\fracstyle{%
133     \ifcase\numexpr\mathstyle\relax
134       \chardef\mathstyle=\@ne
135     \or
136       \chardef\mathstyle=\@ne
137     \or
138       \chardef\mathstyle=\tw@
139     \or
140       \chardef\mathstyle=\tw@
141     \else
142       \chardef\mathstyle=\thr@@
143     \fi
144   }
145 \else
146   \def\fracstyle{}
147 \fi

```

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

```

148 \def\currentmathstyle{%
149   \ifcase\numexpr\mathstyle\relax
150     \@@displaystyle
151   \or
152     \@@displaystyle
153   \or
154     \@@textstyle
155   \or
156     \@@textstyle
157   \or
158     \@@scriptstyle
159   \or

```

```

160   \@@scriptstyle
161   \else
162   \@@scriptscriptstyle
163   \fi}

Finally, we declare the package options.
164 \DeclareOption{mathactivechars}{%
165 % \catcode'\^=12\relax
166 % \catcode'\_ =12\relax
167 \AtBeginDocument{\catcode'\^=12\relax \catcode'\_ =12\relax}%
168 }
169 \DeclareOption{activechars}{%
170 % \catcode'\^=13\relax
171 % \catcode'\_ =13\relax
172 \AtBeginDocument{\catcode'\^=13\relax \catcode'\_ =13\relax}%
173 }
174 \DeclareOption{noactivechars}{%
175 % \catcode'\^=7\relax
176 % \catcode'\_ =8\relax
177 \AtBeginDocument{\catcode'\^=7\relax \catcode'\_ =8\relax}%
178 }
179 \ExecuteOptions{mathactivechars}
180 \ProcessOptions\relax

WSPR: Set up the active behaviours: (this is set even in the noactivechars case
but they are never activated. no worries?)
181 \ifnum\catcode'\^=13\relax
182   \let^=\sp \let_=\sb
183 \else
184   \mathcode'\^="8000\relax
185   \mathcode'\_ ="8000\relax
186   \begingroup
187     \catcode'\^=\active
188     \catcode'\_ =\active
189     \global\let^=\sp
190     \global\let_=\sb
191   \endgroup
192 \fi
193 \</package>

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in **roman** refer to the code lines where the entry is used.

	Symbols		<code>\@@atop</code>	52
<code>\@@above</code>	53	<code>\@@atopwithdelims</code>	55	
<code>\@@abovewithdelims</code>	56	<code>\@@displaystyle</code>	46, 60, 150, 152	

<code>\@mathchoice</code>	50	<code>\DeclareOption</code>	164, 169, 174
<code>\@over</code>	51	<code>\DeclareRobustCommand</code>	59, 61, 63, 65, 111
<code>\@overwithdelims</code>	54	<code>\def</code> ...	16, 78, 84, 88–90, 132, 146, 148
<code>\@scriptscriptstyle</code>	49, 66, 162	<code>\dfrac</code>	120, 122
<code>\@scriptstyle</code>	48, 64, 158, 160	<code>\directlua</code>	44
<code>\@subscript</code>	70, 87	<code>\displaystyle</code>	46, 59, 122, 128
<code>\@subscript@other</code>	74	E	
<code>\@superscript</code>	70, 86	<code>\edef</code>	4, 7, 16, 25, 26
<code>\@superscript@other</code>	74	<code>\else</code>	6, 9, 28, 43, 80, 83, 103, 114, 115, 141, 145, 161, 183
<code>\@textstyle</code>	47, 62, 154, 156	<code>\endcsname</code> ...	37, 41, 58, 77, 115, 131
<code>\@car</code>	16, 24	<code>\endgroup</code>	12, 34, 36, 40, 57, 69, 73, 76, 113, 130, 191
<code>\@eha</code>	31	<code>\everydisplay</code>	<u>36</u>
<code>\@firstoffour</code>	92, 94, 107	<code>\ExecuteOptions</code>	179
<code>\@fourthoffour</code>	104, 110	<code>\expandafter</code>	25, 36–38, 40, 41, 57, 58, 76, 77, 92, 94, 96, 98, 100, 102, 104, 130, 131
<code>\@one</code>	42, 134, 136	F	
<code>\@nil</code>	24	<code>\fi</code> ..	10, 11, 32, 33, 39, 45, 67, 81, 85, 105, 114, 115, 143, 147, 163, 192
<code>\@saveprimitive</code>	<u>2</u> , 46–56	<code>\firstmark</code>	19
<code>\@saveprimitive@a</code>	9, 14	<code>\frac</code>	119
<code>\@secondoffour</code>	96, 98, 108	<code>\fracstyle</code>	112, 132, 146
<code>\@tempa</code>	4, 5, 8, 25, 27	G	
<code>\@tempb</code>	4, 5, 7, 8, 16, 17, 25–27	<code>\genfrac</code> ..	<u>111</u> , 119, 122, 123, 127–129
<code>\@thirdoffour</code>	100, 102, 109	<code>\global</code>	5, 27, 189, 190
<code>\^</code>	68, 72, 165, 167, 170, 172, 175, 177, 181, 184, 187	I	
<code>_</code>	68, 72, 166, 167, 171, 172, 176, 177, 185, 188	<code>\if</code>	115
A		<code>\ifcase</code>	91, 133, 149
<code>\above</code>	53	<code>\ifnum</code>	79, 181
<code>\abovewithdelims</code>	56	<code>\ifx</code> ...	5, 8, 27, 37, 41, 58, 77, 114, 131
<code>\active</code>	187, 188	L	
<code>\AtBeginDocument</code>	167, 172, 177	<code>\let</code> .	5, 27, 70, 74, 86, 87, 182, 189, 190
<code>\atop</code>	52	<code>\lowercase</code>	69, 73
<code>\atopwithdelims</code>	55	M	
B		<code>\mathchoice</code>	50, <u>90</u>
<code>\begingroup</code>	3, 15, 36, 40, 57, 68, 72, 76, 113, 130, 186	<code>\mathcode</code>	184, 185
<code>\binom</code>	124, 127	<code>\mathstyle</code>	38, <u>40</u> , 60, 62, 64, 66, 79, 80, 91, 133, 134, 136, 138, 140, 142, 149
<code>\botmark</code>	20	<code>\mathsub</code>	87, 88
C		<code>\mathsup</code>	86, 89
<code>\catcode</code>	68, 72, 165–167, 170–172, 175–177, 181, 187, 188	<code>\maxdimen</code>	114
<code>\chardef</code>	38, 42, 60, 62, 64, 66, 79, 80, 134, 136, 138, 140, 142	<code>\meaning</code>	4, 7, 25, 26
<code>\csname</code>	37, 41, 58, 77, 114, 131		
<code>\currentmathstyle</code>	148		
D			
<code>\dbinom</code>	126, 128		

N		S	
<code>\noexpand</code>	31	<code>\sb</code>	88, 182, 190
<code>\nullfont</code>	17	<code>\scriptscriptstyle</code>	49, 65
<code>\numexpr</code>	91, 133, 149	<code>\scriptstyle</code>	48, 63
O		<code>\sp</code>	89, 182, 189
<code>\or</code>	93, 95, 97, 99, 101, 135, 137, 139, 151, 153, 155, 157, 159	<code>\splitbotmark</code>	22
<code>\over</code>	51	<code>\splitfirstmark</code>	21
<code>\overwithdelims</code>	54	<code>\string</code>	4, 18–23, 30
P		<code>\strip@prefix</code>	25
<code>\PackageError</code>	29	<code>\subsupstyle</code>	78, 84, 88, 89
<code>\ProcessOptions</code>	180	T	
<code>\protect</code>	88, 89	<code>\tbinom</code>	125, 129
<code>\providecommand</code>	2, 14, 107–110, 120, 121, 124–126	<code>\textstyle</code>	47, 61, 123, 129
R		<code>\tfrac</code>	121, 123
<code>\relax</code>	37, 41, 58, 68, 72, 77, 91, 115, 131, 133, 149, 165–167, 170– 172, 175–177, 180, 181, 184, 185	<code>\the</code>	38
<code>\renewcommand</code> ..	119, 122, 123, 127–129	<code>\thr@@</code>	142
		<code>\topmark</code>	18
		<code>\tw@</code>	62, 138, 140
		Z	
		<code>\z@</code>	38, 60