The **jmsdelim** package

Jonathan Sterling

September 14, 2019

1 Overview

Sizing delimiters using `\left` and `\right` should be outlawed! The results are nearly always unaesthetic, primarily because the correct size of a mathematical delimiter is a typesetting consideration which does *not* emanate from the physical size of the interior.

Correctly sizing delimiters is very difficult, particularly in well-architected documents: a correctly engineered mathematical document will include macros for all operations, and these macros necessarily will include delimiters (such as parentheses). However, the correct size for the delimiter cannot be chosen ahead of time, because it will depend on the arguments; two options are available:

1. Provide optional arguments to each notation macro for choosing delimiter sizes. This is nearly intractable to do in practice.

2. Ignore delimiter sizes.

With **jmsdelim** we offer an alternative: the correct delimiter sizes can be set at the *leaf nodes* of a mathematical expression, and magically bubble upward through the delimiters.

1.1 Document interface

`\DelimMin`

`\DelimMin{⟨intexpr_{min}⟩}`

This sets the minimum delimiter size to $⟨\textit{intexpr}_{min}⟩$ outside the current location; delimiter sizes are represented as natural numbers, with `0` the smallest size.

`\DelimMin` is the work-horse of **jmsdelim**; let us consider an example of what one might do prior to adopting **jmsdelim**. Suppose we have defined a macro `\Psh` for the free co-completion, following the notation of the French school, and we wish to parenthesize an instance of it:

$$\mathrm{Hom}_{\mathbf{Cat}}(1, \widehat{\mathbb{C}})$$

```
\NewDocumentCommand\Cat{}{\mathbf{Cat}}
\NewDocumentCommand\Psh{m}{\widehat{#1}}
\NewDocumentCommand\Hom{mmm}{
  \operatorname{Hom}_{#1}(#2,#3)
}
\[ \Hom{\Cat}{1}{\Psh{\mathbb{C}}} \]
```

One might have tried to get a better result by using `\left` and `\right`:

$$\operatorname{Hom}_{\mathbf{Cat}}\left(1, \widehat{\mathbb{C}}\right)$$
$$\operatorname{Hom}_{\mathbf{Cat}}\left(1, \widehat{\mathbb{C}}\right)$$

```
\NewDocumentCommand\Cat{}{\mathbf{Cat}}
\NewDocumentCommand\Psh{m}{\widehat{#1}}
\NewDocumentCommand\HomX{mmm}{
  \operatorname{Hom}_{#1}\left(#2,#3\right)
}
\NewDocumentCommand\Hom{mmm}{
  \operatorname{Hom}_{#1}\mleft(#2,#3\mright)
}
\[ \Hom{\Cat}{1}{\Psh{\mathbb{C}}} \]
\[ \HomX{\Cat}{1}{\Psh{\mathbb{C}}} \]
```

The above is hugely worse: the height of the hat does not in any way determine the correct size for the delimiter! The solution using jmsdelim is quite simple, however: first, we change `\Hom` to call `\DelimPrn`, and then we use `\DelimMin` within the `\Psh` notation.

$$\operatorname{Hom}_{\mathbf{Cat}}\left(1, \widehat{\mathbb{C}}\right)$$

```
\NewDocumentCommand\Cat{}{\mathbf{Cat}}
\NewDocumentCommand\Psh{m}{\DelimMin{1}\widehat{#1}}
\NewDocumentCommand\Hom{mmm}{
  \operatorname{Hom}_{#1}\DelimPrn{#2,#3}
}
\[ \Hom{\Cat}{1}{\Psh{\mathbb{C}}} \]
```

`\DelimBump` \DelimBump

This increases the minimum delimiter size by one outside the current location. This can be used to achieve a readable notation for cuts in sequent calculus as in Munch-Maccagnoni [Mun13; Mun17], for instance:

$$\langle t \parallel \tilde{\mu}x.\langle\mu\alpha.\langle u \parallel e\rangle \parallel e'\rangle\rangle$$

```
\NewDocumentCommand\Cut{mm}{%
  \DelimBump%
  \DelimBetweenSurround{\langle}{\Vert}{\rangle}{#1}{#2}%
}
\NewDocumentCommand\Mu{mm}{\mu #1.#2}
\NewDocumentCommand\MuTilde{mm}{\tilde\mu #1.#2}
\[
  \Cut{t}{
    \MuTilde{x}{
      \Cut{\Mu{\alpha}{\Cut{u}{e}}}{e'}
    }
  }
\]
```

1.2 Basic Delimiter commands

Like mleftright [Obe16], jmsdelim ensures the correct amount of space on the outside of the delimiters using `\mathopen` and `\mathclose`.

`\DelimSurround` \DelimSurround{⟨left⟩}{⟨right⟩}{⟨body⟩}

Surrounds ⟨body⟩ with appropriately sized ⟨left⟩ and ⟨right⟩ delimiters respectively.

\DelimBetween	`\DelimSurround{⟨sep⟩}{⟨lbody⟩}{⟨rbody⟩}`

Places an appropriately sized ⟨*sep*⟩ between ⟨*lbody*⟩ and ⟨*rbody*⟩.

\DelimBetweenSurround	`\DelimSurround{⟨left⟩}{⟨sep⟩}{⟨right⟩}{⟨lbody⟩}{⟨rbody⟩}`

Places an appropriately sized ⟨*sep*⟩ between ⟨*lbody*⟩ and ⟨*rbody*⟩, surrounding the result by ⟨*left*⟩ and ⟨*right*⟩ respectively.

1.3 Derived delimiter commands

\DelimPrn	`\DelimPrn{⟨body⟩}`

Surrounds ⟨*body*⟩ in parentheses.

\DelimBrk	`\DelimBrk{⟨body⟩}`

Surrounds ⟨*body*⟩ in square brackets.

\DelimBrc	`\DelimBrc{⟨body⟩}`

Surrounds ⟨*body*⟩ in curly braces.

\DelimGl	`\DelimGl{⟨body⟩}`

Surrounds ⟨*body*⟩ in angle brackets.

\DelimBbrk	`\DelimBbrk{⟨body⟩}`

Surrounds ⟨*body*⟩ in Scott brackets (requires `\llbracket`, `\rrbracket` to be defined).

1.4 Configuration and options

\jmsdelimsetup	`\jmsdelimsetup{⟨options⟩}`

jmsdelim can be customized along a few axes.

size␣commands	The option `size commands` is a comma-separated list which contains a list of sizing commands for delimiters, from smallest to largest.

1.5 Interface for macro authors

The internals of jmsdelim are implemented in expl3.

`jmsdelim_surround:nnn`	`jmsdelim_surround:nnn {⟨left⟩} {⟨right⟩} {⟨body⟩}`

This routine surrounds ⟨*body*⟩ with the delimiters ⟨*left*⟩ and ⟨*right*⟩ of the appropriate size respectively.

`jmsdelim_between:nnn`	`jmsdelim_between:nnn {⟨sep⟩} {⟨lbody⟩} {⟨rbody⟩}`

This routine separates ⟨*lbody*⟩ and ⟨*rbody*⟩ with a separator ⟨*sep*⟩ of the appropriate size.

`jmsdelim_between:nnnnn`	`jmsdelim_between:nnnnn {⟨left⟩} {⟨sep⟩} {⟨right⟩} {⟨lbody⟩} {⟨rbody⟩}`

This routine separates ⟨*lbody*⟩ and ⟨*rbody*⟩ with a separator ⟨*sep*⟩ of the appropriate size, and surrounds the result by ⟨*left*⟩ and ⟨*right*⟩ respectively of the same size.

2 Extended example from `perfectcut`

The following states the idempotency of an adjunction:

$$\langle t \parallel \tilde{\mu}x.\langle \mu\alpha.\langle u \parallel e \rangle \parallel e' \rangle \rangle = \langle \mu\alpha.\langle t \parallel \tilde{\mu}x.\langle u \parallel e \rangle \rangle \parallel e' \rangle$$

The following states the commutativity of a strong monad:

$$\left\langle t \, \middle\| \, \tilde{\mu}x.\langle u \parallel \tilde{\mu}y.\langle v \parallel e \rangle \rangle \right\rangle = \left\langle u \, \middle\| \, \tilde{\mu}y.\langle t \parallel \tilde{\mu}x.\langle v \parallel e \rangle \rangle \right\rangle$$

Using `\underline` to mark redexes:

$$\delta(V, x.y, x.y)$$
$$= \mu\star.\left\langle V \, \middle\| \, [\tilde{\mu}x.\underline{\langle y \parallel \star \rangle} \mid \tilde{\mu}x.\underline{\langle y \parallel \star \rangle}] \right\rangle$$
$$= \mu\star.\left\langle V \, \middle\| \, [\underline{\tilde{\mu}x.\langle \iota_1(x) \parallel \tilde{\mu}z.\langle y \parallel \star \rangle \rangle} \mid \tilde{\mu}x.\langle \iota_2(x) \parallel \tilde{\mu}z.\langle y \parallel \star \rangle \rangle] \right\rangle$$
$$= \mu\star.\left\langle V \, \middle\| \, \underline{\tilde{\mu}z.\langle y \parallel \star \rangle} \right\rangle$$
$$= \mu\star.\underline{\langle y \parallel \star \rangle}$$
$$= y$$

```
\ignoremathstyle
\NewDocumentCommand\Cut{mm}{%
  \DelimBump\DelimBetweenSurround{\langle}{\Vert}{\rangle}{#1}{#2}%
}
\NewDocumentCommand\mt{}{\tilde\mu}
\NewDocumentCommand\Case{mm}{%
  \DelimBetweenSurround{[}{\vert}{]}{#1}{#2}%
}
The following states the idempotency of an adjunction:
\[
\Cut{t}{\mt x.\Cut{\mu\alpha.\Cut{u}{e}}{e'}}=\Cut{\mu\alpha.\Cut{t}{\mt x.\Cut{u}{e}}}{e'}
\]

The following states the commutativity of a strong monad:
```

```
\[
\Cut t{\mt x.\Cut u{\mt y.\Cut ve}}=\Cut u{\mt y.\Cut t{\mt x.\Cut ve}}
\]
Using \cs{underline} to mark redexes:
\begin{align*}
  & \delta(V,x.y,x.y)\\
  & = \mu{\star}.
      \Cut{V}{
        \Case{
          \mt x.\underline{\Cut y{\star}}
        }{
          \mt x.\underline{\Cut y{\star}}
        }
      }\\
  & = \mu{\star}.
      \Cut{V}{
        \underline{
          \Case{
            \mt x.\Cut{\iota_{1}(x)}{\mt z.\Cut{y}{\star}}
          }{
            \mt x.\Cut{\iota_{2}(x)}{\mt z.\Cut{y}{\star}}
          }
        }
      }\\
  & = \mu{\star}.\Cut{V}{\underline{\mt z.}\Cut{y}{\star}}\\
  & = \mu{\star}.\Cut{y}{\star}\\
  & =y
\end{align*}
```

3   jmsdelim implementation

```
1 ⟨*package⟩
2 \RequirePackage{expl3}
3 \RequirePackage{l3keys2e}
4 \RequirePackage{xparse}
5 \RequirePackage{ifluatex}
6 \RequirePackage{scalerel}
7 \ProvidesExplPackage {jmsdelim} {2020/11/02} {0.2.0}
8   {Compositional delimiter sizing}
9 ⟨@@=jmsdelim⟩
```

We first declare the options for the jmsdelim module, together with their default valeus.

```
10 \keys_define:nn { jmsdelim } {
11   size~commands .clist_set:N = \l__jmsdelim_size_cmds,
12 }
13 \keys_set:nn { jmsdelim } {
14   size~commands = {{},\big,\Big,\bigg,\Bigg},
15 }
```

Then, we set up the internal state that will be used by jmsdelim.

```
16 \int_new:N \g__jmsdelim_size
```

```
17  \int_gset:Nn \g__jmsdelim_size {0}
18  \bool_new:N \g__jmsdelim_bump
19  \bool_gset:Nn \g__jmsdelim_bump \c_false_bool
```

__jmsdelim_clist_item:Nn A version of \clist_item:Nn that takes the last item when the index is out of bounds.

```
20  \cs_new:Npn \__jmsdelim_clist_item:Nn #1 #2 {
21    \clist_item:Nn #1 {
22      \int_min:nn { #2 } {\clist_count:N #1}
23    }
24  }
```

(*End definition for* __jmsdelim_clist_item:Nn.)

__jmsdelim_set_delim_size:n Updates the current delimiter size only if it results in an increase.

```
25  \cs_new:Npn \__jmsdelim_set_delim_size:n #1 {
26    \int_gset:Nn \g__jmsdelim_size {
27      \int_max:nn {\g__jmsdelim_size} {#1}
28    }
29  }
```

(*End definition for* __jmsdelim_set_delim_size:n.)

__jmsdelim_incr_delim_size: Increases the current delimiter size by 1.

```
30  \cs_new:Npn \__jmsdelim_incr_delim_size: {
31    \__jmsdelim_set_delim_size:n {\g__jmsdelim_size + 1}
32  }
```

(*End definition for* __jmsdelim_incr_delim_size:.)

3.1 Preservation of math styles

It is fairly complicated and inefficient to preserve math styles across boxes. There is an appropriate way to do so in LuaLaTeX, which we use conditionally if available; otherwise, we make use of \ThisStyle and \SavedStyle from scalerel, which are more inefficient. In fact, it becomes impossible to use jmsdelim in PDFLaTeX when the nesting is sufficiently deep, whereas there is no corresponding blowup in LuaLaTeX. The \ignoremathstyle and \discernmathstyle macros from scalerel can be used to turn off the inefficient preservation of math styles locally, such as in the case where no subscripts are used.

__jmsdelim_luatex_save_mathstyle:N

```
33  \cs_new:Npn \__jmsdelim_luatex_save_mathstyle:N #1 {
34    \ifcase \mathstyle
35      \cs_set_eq:NN #1 \displaystyle
36    \or
37      \cs_set_eq:NN #1 \crampeddisplaystyle
38    \or
39      \cs_set_eq:NN #1 \textstyle
40    \or
41      \cs_set_eq:NN #1 \crampedtextstyle
```

6

```
42    \or
43      \cs_set_eq:NN #1 \scriptstyle
44    \or
45      \cs_set_eq:NN #1 \crampedscriptstyle
46    \or
47      \cs_set_eq:NN #1 \scriptscriptstyle
48    \or
49      \cs_set_eq:NN #1 \crampedscriptscriptstyle
50    \fi
51  }
```

(*End definition for* `__jmsdelim_luatex_save_mathstyle:N`.)

`__jmsdelim_restore_mathstyle:n`

```
52  \cs_new:Npn \__jmsdelim_restore_mathstyle: {
53    \SavedStyle
54  }
```

(*End definition for* `__jmsdelim_restore_mathstyle:n`.)

`__jmsdelim_save_mathstyle:n`

```
55  \cs_new:Npn \__jmsdelim_save_mathstyle:n #1 {
56    \ifluatex
57      \__jmsdelim_luatex_save_mathstyle:N \__jmsdelim_restore_mathstyle:
58      #1
59    \else
60      \ThisStyle{#1}
61    \fi
62  }
```

(*End definition for* `__jmsdelim_save_mathstyle:n`.)

`__jmsdelim_hbox_set:Nn`   Can only be called in a `\__jmsdelim_save_mathstyle:n` scope; sets the contents of an hbox in the saved math (or text) style.

```
63  \cs_new:Npn \__jmsdelim_hbox_set:Nn #1 #2 {
64      \mode_if_math:TF
65        { \hbox_set:Nn #1 {$\__jmsdelim_restore_mathstyle: #2$} }
66        { \hbox_set:Nn #1 { #2 } }
67  }
```

(*End definition for* `__jmsdelim_hbox_set:Nn`.)

3.2   Public interface for macro authors

`__jmsdelim_scope:n`

```
68  \cs_new:Npn \__jmsdelim_scope:n #1 {
69    \group_begin:
70    \__jmsdelim_save_mathstyle:n {#1}
71    \group_end:
72  }
```

7

(*End definition for* `__jmsdelim_scope:n`.)

`__jmsdelim_typeset:Nn`   Used internally to the delimiter primitives, this function typesets its second argument into the box named by its first argument, keeping track of delimiter size states; in particular, this function causes `\jmsdelim_size_cmd:` to be locally set to the correct size command.

```
73 \cs_new:Npn \__jmsdelim_typeset:Nn #1 #2 {
74   \bool_set:Nn \l_tmpa_bool \g__jmsdelim_bump
75   \int_set:Nn \g__jmsdelim_size {0}
76   \bool_set:Nn \g__jmsdelim_bump \c_false_bool
77
78   \__jmsdelim_hbox_set:Nn #1 {#2}
79
80   \bool_if:NT \g__jmsdelim_bump {
81     \bool_if:NF \l_tmpa_bool {
82       \__jmsdelim_incr_delim_size:
83     }
84   }
85
86   \cs_set_nopar:Npn \jmsdelim_size_cmd: {
87     \__jmsdelim_clist_item:Nn \l__jmsdelim_size_cmds {
88       \g__jmsdelim_size + 1
89     }
90   }
91 }
```

(*End definition for* `__jmsdelim_typeset:Nn`.)

jmsdelim_surround:nnn

```
92 \cs_new:Npn \jmsdelim_surround:nnn #1 #2 #3 {
93   \__jmsdelim_scope:n {
94     \__jmsdelim_typeset:Nn \l_tmpa_box {#3}
95
96     \mathopen\jmsdelim_size_cmd: {#1}
97     \box_use:N \l_tmpa_box
98     \mathopen\jmsdelim_size_cmd: {#2}
99   }
100 }
```

(*End definition for* `jmsdelim_surround:nnn`. *This function is documented on page* *4.*)

jmsdelim_between:nnn

```
101 \cs_new:Npn \jmsdelim_between:nnn #1 #2 #3 {
102   \__jmsdelim_scope:n {
103     \__jmsdelim_typeset:Nn \l_tmpa_box {#2}
104     \__jmsdelim_typeset:Nn \l_tmpb_box {#3}
105     \box_use:N \l_tmpa_box
106     \mathrel{\jmsdelim_size_cmd: {#1}}
107     \box_use:N \l_tmpb_box
108   }
109 }
```

(End definition for `jmsdelim_between:nnn`*. This function is documented on page 4.)*

jmsdelim_between:nnnnn

```
110 \cs_new:Npn \jmsdelim_between:nnnnn #1 #2 #3 #4 #5 {
111   \__jmsdelim_scope:n {
112     \__jmsdelim_typeset:Nn \l_tmpa_box {#4}
113     \__jmsdelim_typeset:Nn \l_tmpb_box {#5}
114
115     \mathopen\jmsdelim_size_cmd: {#1}
116     \box_use:N \l_tmpa_box
117     \mathrel{\jmsdelim_size_cmd: {#2}}
118     \box_use:N \l_tmpb_box
119     \mathclose\jmsdelim_size_cmd: {#3}
120   }
121 }
```

(End definition for `jmsdelim_between:nnnnn`*. This function is documented on page 4.)*

3.3   Document interace

DelimMin

```
122 \NewDocumentCommand\DelimMin{m}{
123   \__jmsdelim_set_delim_size:n {#1}
124 }
```

(End definition for `DelimMin`*. This function is documented on page 1.)*

DelimBump

```
125 \NewDocumentCommand\DelimBump{}{
126   \bool_gset:Nn \g__jmsdelim_bump \c_true_bool
127 }
```

(End definition for `DelimBump`*. This function is documented on page 2.)*

DelimSurround

```
128 \NewDocumentCommand\DelimSurround{mmm}{
129   \jmsdelim_surround:nnn {#1} {#2} {#3}
130 }
```

(End definition for `DelimSurround`*. This function is documented on page 2.)*

DelimBetween

```
131 \NewDocumentCommand\DelimBetween{mmm}{
132   \jmsdelim_between:nnn {#1} {#2} {#3}
133 }
134 \NewDocumentCommand\DelimBetweenSurround{mmmmm}{
135   \jmsdelim_between:nnnnn {#1} {#2} {#3} {#4} {#5}
136 }
```

9

(*End definition for* `DelimBetween`*. This function is documented on page 3.*)

DelimPrn

```
137 \NewDocumentCommand\DelimPrn{m}{
138   \jmsdelim_surround:nnn {(} {)} {#1}
139 }
```

(*End definition for* `DelimPrn`*. This function is documented on page 3.*)

DelimBrk

```
140 \NewDocumentCommand\DelimBrk{m}{
141   \jmsdelim_surround:nnn {[} {]} {#1}
142 }
```

(*End definition for* `DelimBrk`*. This function is documented on page 3.*)

DelimBrc

```
143 \NewDocumentCommand\DelimBrc{m}{
144   \jmsdelim_surround:nnn {\lbrace} {\rbrace} {#1}
145 }
```

(*End definition for* `DelimBrc`*. This function is documented on page 3.*)

DelimBbrk

```
146 \NewDocumentCommand\DelimBbrk{m}{
147   \jmsdelim_surround:nnn {\llbracket} {\rrbracket} {#1}
148 }
```

(*End definition for* `DelimBbrk`*. This function is documented on page 3.*)

DelimGl

```
149 \NewDocumentCommand\DelimGl{m}{
150   \jmsdelim_surround:nnn {\langle} {\rangle} {#1}
151 }
```

(*End definition for* `DelimGl`*. This function is documented on page 3.*)

```
152 \ProcessKeysPackageOptions {jmsdelim}
```

References

[Mun13] Guillaume Munch-Maccagnoni. "Syntax and Models of a non-Associative Composition of Programs and Proofs". PhD thesis. Univ. Paris Diderot, 2013 (cit. on p. 2).

[Mun17] Guillaume Munch-Maccagnoni. *perfectcut – Nested delimiters that consistently grow regardless of the contents*. Sept. 3, 2017. URL: https://ctan.org/pkg/perfectcut (cit. on p. 2).

[Obe16]   Heiko Oberdick. *The `mleftright` package.* May 16, 2016. URL: https://ctan. org/pkg/mleftright (cit. on p. 2).