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.

2 Document interface

`\DelimMin` `\DelimMin{`⟨intexpr$_{min}$⟩`}`

This sets the minimum delimiter size to ⟨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 cocompletion, 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`:

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

```
\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 appallingly 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.

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

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

Behavior under subscripts By default, delimiter sizes are capped under subscripts and superscripts because the alternative is unaesthetic. For instance, consider the following somewhat contrived examples:

$$\int_{\left(\sum_i a_i\right)} \int_{\left(\sum_i a_i\right)}$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{#1}{#2}%
}
\[
  \int_{\DelimPrn{\Sum{i}{a_i}}}
  \int_{\DelimPrn{\DelimMin{4}\Sum{i}{a_i}}}
\]
```

Because the emitted delimiter size under a subscript does *not* determine the actual amount of space used, it is in most cases not correct for this delimiter size to have an effect on its non-subscript context. For this reason, judicious use of the `\DelimProtect` command is recommended in the case of subscripts.

2.1 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.

$$\left|\sum_i b_i\right|$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{#1}{#2}%
}
\[\DelimSurround{\vert}{\vert}{\Sum{i}{b_i}}\]
```

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

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

$$a \,\Big\|\, \sum_i b_i$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{#1}{#2}%
}
\[\DelimBetween{\Vert}{a}{\Sum{i}{b_i}}\]
```

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

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

$$\left\{\sum_i a\cdot b_i \;\middle|\; a\in A\right\}$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{#1}{#2}%
}
\[
  \DelimBetweenSurround{\lbrace}{\vert}{\rbrace}{
    \Sum{i}{a\cdot b_i}
  }{a\in A}
\]
```

`\DelimProtect` `\DelimProtect{⟨body⟩}`

Executes ⟨body⟩ in a sandbox, preventing its state updates from bubbling outward; this is useful in case of subscripts and superscripts. The following command demonstrates incorrect sizing in the presence of a high delimiter size within a subscript:

$$\left(\sum_{\left(\sum_i a_i\right)} F\right)$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{#1}{#2}%
}
\[
  \DelimPrn{\Sum{\DelimPrn{\DelimMin{4}\Sum{i}{a_i}}}{F}}
\]
```

Using a combination of `\DelimProtect` and `\DelimMin`, the formatting can be corrected locally.

$$\left(\sum_{\left(\sum_i a_i\right)} F\right)$$

```
\NewDocumentCommand\Sum{mm}{%
  \DelimMin{1}{\textstyle\sum}_{\DelimProtect{#1}}{#2}%
}
\[
  \DelimPrn{
    \DelimMin{2}
    \Sum{\DelimPrn{\DelimMin{4}\Sum{i}{a_i}}}{F}
  }
\]
```

2.2 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).

2.3 Configuration and options

\DelimSetup `\DelimSetup{⟨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. By default, the standard `\big`, `\Big`, `\bigg`, `\Bigg` sequence is replaced by custom versions that behave differently in script size. This behavior can be overridden as follows:

```
\DelimSetup{
  size commands = {\relax,\big,\Big,\bigg,\Bigg}
}
```

3 Interface for macro authors

The internals of jmsdelim are implemented in expl3.

jmsdelim_scope:nn `jmsdelim_scope:nn {⟨pre⟩} {⟨post⟩}`

This is the fundamental control structure for authors of custom delimiting commands; ⟨pre⟩ is a block of code that renders things to temporary boxes, and {⟨post⟩} is code that *uses* these boxes, placing them relative to some delimiters. The function of `\jmsdelim_scope:nn` is to watch for the delimiter size updates induced by ⟨pre⟩, and set the delimiter size commands correctly before executing ⟨post⟩. Both ⟨pre⟩ and ⟨post⟩ are to be executed in the same block level.

jmsdelim_hbox_set:Nn `jmsdelim_hbox_set:Nn {⟨box⟩} {⟨contents⟩}`

This command is meant to be used inside the ⟨pre⟩ block of `\jmsdelim_scope:nn`; it typesets ⟨contents⟩ in the box named by ⟨box⟩, correctly propagating the math style.

jmsdelim_size_cmd: `jmsdelim_size_cmd:`

This command is meant to be used inside the ⟨post⟩ block of `\jmsdelim_scope:nn` to set the size of a given delimiter; it behaves like `\big`, etc.

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.

`jmsdelim_protect:n` `jmsdelim_protect:n {⟨body⟩}`

Executes ⟨body⟩ in a sandbox, preventing its state updates from bubbling upward.

4 jmsdelim implementation

```
⟨*package⟩
\RequirePackage{expl3}
\RequirePackage{l3keys2e}
\RequirePackage{xparse}
\RequirePackage{ifluatex}
\RequirePackage{scalerel}
\ProvidesExplPackage {jmsdelim} {2020/11/02} {0.2.0}
  {Compositional delimiter sizing}
⟨@@=jmsdelim⟩
```

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

```
\keys_define:nn { jmsdelim } {
  size~commands .clist_set:N = \l__jmsdelim_size_cmds,
}
\keys_set:nn { jmsdelim } {
 size~commands = {relax,jmsdelim_big:n,jmsdelim_Big:n,jmsdelim_bigg:n,jmsdelim_Bigg:n},
}

\cs_new:Npn \jmsdelim_big:n #1 {
  {\mathchoice{\big #1} {\big #1}{\big #1}{#1}}
}

\cs_new:Npn \jmsdelim_Big:n #1 {
  {\mathchoice{\Big #1} {\Big #1}{\big #1}{#1}}
}

\cs_new:Npn \jmsdelim_bigg:n #1 {
  {\mathchoice{\bigg #1} {\bigg #1}{\big #1}{#1}}
}

\cs_new:Npn \jmsdelim_Bigg:n #1 {
  {\mathchoice{\Bigg #1} {\Bigg #1}{\big #1}{#1}}
}
```

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

```
\int_new:N \g__jmsdelim_size
\int_new:N \g__jmsdelim_size_up
\int_gset:Nn \g__jmsdelim_size {0}
\int_gset:Nn \g__jmsdelim_size_up {0}
```

4.1 Internals

__jmsdelim_clist_item:Nn

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

```
\cs_new:Npn \__jmsdelim_clist_item:Nn #1 #2 {
  \clist_item:Nn #1 {
    \int_min:nn { #2 } {\clist_count:N #1}
  }
}
```

(*End definition for* __jmsdelim_clist_item:Nn.)

__jmsdelim_setup_sizes:

```
\cs_new:Npn \__jmsdelim_setup_sizes: {
  \int_gset:Nn \g__jmsdelim_size {
    \int_max:nn \g__jmsdelim_size \g__jmsdelim_size_up
  }

  \cs_set_eq:Nc \jmsdelim_size_cmd: {
    \__jmsdelim_clist_item:Nn \l__jmsdelim_size_cmds {
      \g__jmsdelim_size + 1
    }
  }
}
```

(*End definition for* __jmsdelim_setup_sizes:.)

4.1.1 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 [Seg16], 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

```
\cs_new:Npn \__jmsdelim_luatex_save_mathstyle:N #1 {
  \ifcase \mathstyle
    \cs_set_eq:NN #1 \displaystyle
  \or
    \cs_set_eq:NN #1 \crampeddisplaystyle
  \or
    \cs_set_eq:NN #1 \textstyle
  \or
    \cs_set_eq:NN #1 \crampedtextstyle
  \or
    \cs_set_eq:NN #1 \scriptstyle
```

```
  \or
    \cs_set_eq:NN #1 \crampedscriptstyle
  \or
    \cs_set_eq:NN #1 \scriptscriptstyle
  \or
    \cs_set_eq:NN #1 \crampedscriptscriptstyle
  \fi
}
```

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

`__jmsdelim_restore_mathstyle:n`

```
\cs_new:Npn \__jmsdelim_restore_mathstyle: {
  \SavedStyle
}
```

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

`__jmsdelim_save_mathstyle:n`

```
\cs_new:Npn \__jmsdelim_save_mathstyle:n #1 {
  \ifluatex
    \__jmsdelim_luatex_save_mathstyle:N \__jmsdelim_restore_mathstyle:
    #1
  \else
    \ThisStyle{#1}
  \fi
}
```

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

4.2 Public interface for macro authors

`jmsdelim_scope:nn`

```
\cs_new:Npn \jmsdelim_scope:nn #1 #2 {
  \group_begin:
    \int_set:Nn \l_tmpa_int \g__jmsdelim_size_up
    \int_gset:Nn \g__jmsdelim_size_up 0
    \int_gset:Nn \g__jmsdelim_size 0
    \group_begin:
      \__jmsdelim_save_mathstyle:n {
        #1
        \__jmsdelim_setup_sizes:
        #2
      }
    \group_end:
   \int_gset:Nn \g__jmsdelim_size_up {\int_max:nn \g__jmsdelim_size_up \l_tmpa_int}
  \group_end:
}
```

(*End definition for* `jmsdelim_scope:nn`. *This function is documented on page* 5.)

jmsdelim_hbox_set:Nn

```
\cs_new:Npn \jmsdelim_hbox_set:Nn #1 #2 {
   \mode_if_math:TF
     { \hbox_set:Nn #1 {$\m@th\__jmsdelim_restore_mathstyle: #2$} }
     { \hbox_set:Nn #1 { #2 } }
}
```

(*End definition for* jmsdelim_hbox_set:Nn. *This function is documented on page* 5.)

jmsdelim_surround:nnn

```
\cs_new:Npn \jmsdelim_surround:nnn #1 #2 #3 {
  \jmsdelim_scope:nn {
    \jmsdelim_hbox_set:Nn \l_tmpa_box {#3}
  }{
    \mathopen\jmsdelim_size_cmd: {#1}
    \box_use:N \l_tmpa_box
    \mathclose\jmsdelim_size_cmd: {#2}
  }
}
```

(*End definition for* jmsdelim_surround:nnn. *This function is documented on page* 5.)

jmsdelim_protect:n

```
\cs_new:Npn \jmsdelim_protect:n #1 {
  \group_begin:
    \int_set:Nn \l_tmpa_int \g__jmsdelim_size_up
    \int_set:Nn \l_tmpb_int \g__jmsdelim_size
    \group_begin: #1 \group_end:
    \int_gset:Nn \g__jmsdelim_size_up \l_tmpa_int
    \int_gset:Nn \g__jmsdelim_size \l_tmpb_int
  \group_end:
}
```

(*End definition for* jmsdelim_protect:n. *This function is documented on page* 6.)

jmsdelim_between:nnn

```
\cs_new:Npn \jmsdelim_between:nnn #1 #2 #3 {
  \jmsdelim_scope:nn {
    \jmsdelim_hbox_set:Nn \l_tmpa_box {#2}
    \jmsdelim_hbox_set:Nn \l_tmpb_box {#3}
  }{
    \box_use:N \l_tmpa_box
    \mathrel{\jmsdelim_size_cmd: {#1}}
    \box_use:N \l_tmpb_box
  }
}
```

(*End definition for* jmsdelim_between:nnn. *This function is documented on page* 5.)

jmsdelim_between:nnnnn

```
\cs_new:Npn \jmsdelim_between:nnnnn #1 #2 #3 #4 #5 {
  \jmsdelim_scope:nn {
    \jmsdelim_hbox_set:Nn \l_tmpa_box {#4}
    \jmsdelim_hbox_set:Nn \l_tmpb_box {#5}
  }{
    \mathopen\jmsdelim_size_cmd: {#1}
    \box_use:N \l_tmpa_box
    \mathrel{\jmsdelim_size_cmd: {#2}}
    \box_use:N \l_tmpb_box
    \mathclose\jmsdelim_size_cmd: {#3}
  }
}
```

(*End definition for* jmsdelim_between:nnnnn. *This function is documented on page 6.*)

4.3 Document interace

DelimMin

```
\NewDocumentCommand\DelimMin{m}{
  \int_gset:Nn \g__jmsdelim_size_up {\int_max:nn \g__jmsdelim_size_up {#1}}
}
```

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

DelimSurround

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

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

DelimBetween

```
\NewDocumentCommand\DelimBetween{mmm}{
  \jmsdelim_between:nnn {#1} {#2} {#3}
}
```

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

DelimBetweenSurround

```
\NewDocumentCommand\DelimBetweenSurround{mmmmm}{
  \jmsdelim_between:nnnnn {#1} {#2} {#3} {#4} {#5}
}
```

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

DelimProtect

```
\NewDocumentCommand\DelimProtect{m}{
  \jmsdelim_protect:n {#1}
}
```

(*End definition for* `DelimProtect`. *This function is documented on page 4.*)

`DelimPrn`

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

(*End definition for* `DelimPrn`. *This function is documented on page 4.*)

`DelimBrk`

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

(*End definition for* `DelimBrk`. *This function is documented on page 4.*)

`DelimBrc`

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

(*End definition for* `DelimBrc`. *This function is documented on page 4.*)

`DelimBbrk`

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

(*End definition for* `DelimBbrk`. *This function is documented on page 4.*)

`DelimGl`

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

(*End definition for* `DelimGl`. *This function is documented on page 4.*)

```
\ProcessKeysPackageOptions {jmsdelim}
```

References

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

[Seg16] Steven B. Segletes. `scalerel` – *Constrained scaling and stretching of objects*. Dec. 29, 2016. URL: https://ctan.org/pkg/scalerel (cit. on p. 7).