

perfectcut.sty documentation

Guillaume Munch-Maccagnoni*

January 31st 2014

1 Use

This package supplies the following commands:

Command	Produces
<code>\perfectcut{#1}{#2}</code>	$\langle \#1 \parallel \#2 \rangle$
<code>\perfectbra{#1}</code>	$\langle \#1 $
<code>\perfectket{#1}</code>	$ \#1 \rangle$

The effect of the commands is to determine the size of the brackets depending on the number of nested `\perfectcut` (regardless of the contents). It is intended for use:

- In proof theory, for term notations of sequent calculus,
- In computer science, for the modeling of abstract machines.

It could also be adapted for any visually similar effects as an alternative to `\left`, `\right` and `\middle`. (You can contact the author.)

If the package causes errors see the option `nomathstyle` below.

2 Example

The following states the commutativity of a strong monad:

```
\def\mt{\tilde{\mu}}
\[
\cut t{\mt x.\cut u{\mt y.\cut ve}}
=\cut u{\mt y.\cut t{\mt x.\cut ve}}
\]
```

The following states the idempotency of an adjunction:

*<http://www.pps.univ-paris-diderot.fr/~munch/>

```
\[
\cut t{\mt x.\cut{\mu\alpha.\cut ue}{e'}}
=\cut{\mu\alpha.\cut t{\mt x.\cut ue}}{e'}
\]
```

Using perfectcut.sty

```
\usepackage{perfectcut}
\let\cut\perfectcut
```

The following states the commutativity of a strong monad:

$$\langle t \parallel \tilde{\mu}x. \langle u \parallel \tilde{\mu}y. \langle v \parallel e \rangle \rangle \rangle = \langle u \parallel \tilde{\mu}y. \langle t \parallel \tilde{\mu}x. \langle v \parallel e \rangle \rangle \rangle$$

The following states the idempotency of an adjunction:

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

Using \left, \middle and \right instead

```
\renewcommand{\cut}[2]{\left\langle #1\middle|\mkern-2mu\middle|#2\right\rangle}
```

The following states the commutativity of a strong monad:

$$\langle t \parallel \tilde{\mu}x. \langle u \parallel \tilde{\mu}y. \langle v \parallel e \rangle \rangle \rangle = \langle u \parallel \tilde{\mu}y. \langle t \parallel \tilde{\mu}x. \langle v \parallel e \rangle \rangle \rangle$$

The following states the idempotency of an adjunction:

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

3 Options

Option nomathstyle

The use of `\currentmathstyle` from the package `mathstyle` prevents the exponential blowup in recursions that would occur if we used `\mathpalette` instead. To record the `\currentmathstyle`, `mathstyle` redefines many primitives and is therefore a source of incompatibilities. If you run into such issues, please refer to the `mathstyle` manual.

If you want to disable the loading of `mathstyle`, use the `nomathstyle` option. With the `nomathstyle` option, the style is set to `\cutstyle`, which you must define by hand to be `\displaystyle`, `\textstyle` (default), `\scriptstyle`, `\scriptscriptstyle`. This means that cuts in subscripts and superscripts do not have the proper size unless `\cutstyle` is redefined.

Option realVert

With the option `realVert`, the double bars are obtained with the `\Vert` command. Without it, two `\vert` symbols are used and their spacing is controlled with `\cutinterbarskip`. In addition, without `realVert`, a penalty is added for better line breaks.

Customisation

The following mu-skips can be redefined in your preamble:

Command	Defines
<code>\cutbarskip=5.0mu plus 8mu minus 2.0mu</code>	spacing around bars
<code>\cutangleskip=0.0mu plus 8mu minus 1.0mu</code>	spacing around angles (inside)
<code>\cutangleouterskip=0.0mu plus 8mu minus 0mu</code>	spacing around angles (outside)
<code>\cutinterbarskip=0.8mu plus 0mu minus 0mu</code>	spacing between bars

(1 mu equals $\frac{1}{18}$ -th of an em in the current math font.)

4 Reimplementation of fixed-size delimiters

In addition, I provide the following corrections and generalisations of `\big`, `\bigg`, etc. Why not using the latter? Because both the plain `TEX` and the `amsmath` versions are incorrect when changing math font, font size, math style or `\delimitershortfall`. Moreover, OpenType math fonts in particular offer more sizes. We want a robust solution.

Command	Example
<i>#1-th size of delimiter #2</i>	
<code>\nthleft{#1}{#2}</code>	<code>\nthleft{2}{(</code> (
<code>\nthmiddle{#1}{#2}</code>	<code>\nthmiddle{2}{\Vert}</code>
<code>\nthright{#1}{#2}</code>	<code>\nthright{2}{)}</code>)
<i>delimiter #2 of length at least #1</i>	
<code>\lenleft{#1}{#2}</code>	<code>\lenleft{3.2mm}{[</code> [
<code>\lenmiddle{#1}{#2}</code>	<code>\lenmiddle{3.2mm}{ </code>
<code>\lenright{#1}{#2}</code>	<code>\lenright{3.2mm}{]}</code>]
<i>delimiter #2 of length exactly #1 obtained by scaling the above one</i>	
<code>\reallenleft{#1}{#2}</code>	<code>\reallenleft{3.2mm}{[</code> [
<code>\reallenmiddle{#1}{#2}</code>	<code>\reallenmiddle{3.2mm}{ </code>
<code>\reallenright{#1}{#2}</code>	<code>\reallenright{3.2mm}{]}</code>]

Exemple with `\nthleft`

`\nthleft0(\nthleft1(\nthleft2(\nthleft3(\nthleft4(\nthleft5(`

((((((

Example with `\big`, `\Big`, `\bigg`, `\Bigg`

`(\big(\Big(\bigg(\Bigg(`

(((

Note: `\big` starts at at size 2 in some fonts.

5 License

This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3 of this license or (at your option) any later version. Refer to the README file.