

The background of the entire page is a dense, overlapping pattern of rectangles. Each rectangle is filled with a series of horizontal stripes in varying shades of gray. The rectangles are of different sizes and are positioned at various angles, creating a complex, layered visual effect.

ConT_EXt

title : Math Sets
subtitle : ConT_EXt port of braket.sty
author : Aditya Mahajan
date : July 3, 2008


```
\definemathset 3
\docapturemathoplimits 6
```

```
\setupmathset 3
```

1 Introduction

I write a lot of probability expressions which look like this.

$$E \left\{ \sum_y f(X, y) \middle| Z \right\}$$

The usual way to input them is as follows

```
\startformula
\mfunction{E} \left\{ \sum_{y} f(X,y) \,,\middle|\,, Z \right\}
\stopformula
```

We need to ensure that the delimiters and the *conditional* sign `|` scale properly, and the spacing around the conditional sign is correct. As a result, the input is markup heavy, and consequently difficult to read.

In \LaTeX , Donald Arseneau's `braket.sty` can be used to input such expressions in a natural manner, and automatically takes care of the scaling of delimiters and the conditional signs. (The actual package only provides this functionality of bra and ket notation, hence the name, but can be easily extended to probability expressions also. This module is a partial port of `braket.sty` to \ConTeXt).

2 Usage

To use this module add

```
\usemodule[mathsets]
```

on the top of your file. This module defines one command `\definemathset` for defining new math-sets. The syntax of this command is:

```
\definemathset [1...] [2...2...]

1 IDENTIFIER
2 text      = no TEXT
  left      = \{ TEXT
  middle    = \vert TEXT
  right     = \} TEXT
```

The first argument is the name of the set to be defined. Thus, after

```
\definemathset[mathset]
```

`\mathset` is available as a command. The second argument to `\definemathset` are optional assignments `text`, `left`, `middle`, `right`. For example, if we can use `text` to specify what comes at the beginning of the math-set. By default, `text=no` which causes no text to appear, but we can change that to any text that we want (Note that `\mfunction` tells \ConTeXt to use the current math text font)

```
\definemathset[EXP] [text=\mfunction{E}]
```

We can use `\EXP{X}` to get $E\{X\}$ and `\EXP{X|Y}` to get $E\{X|Y\}$. Scaling of the delimiters and conditional sign are taken care automatically. For example

```
\startformula
\EXP{\sum_y f(X,y) | Z }
\stopformula
```

gives

$$E\left\{\sum_y f(X,y) \middle| Z\right\}$$

Compare the above input with the one used in the first example.

By default, the contents of the set are surrounded by curly brackets (or braces); we can change them by using `left` and `right` keys. For example.

```
\definemathset[PR] [text={\mfunction{Pr}},left=(,right=)]
\startformula
\EXP{ \sum_y f(X,y) | Z = z } = \sum_{x,y} \PR{x,y | Z=z}
\stopformula
```

gives

$$E\left\{\sum_y f(X,y) \middle| Z = z\right\} = \sum_{x,y} \Pr(x,y | Z = z)$$

We also provide a mechanism for changing the conditional bar using the `middle` key, although I am not sure if this is needed by anyone. For example, consider the following contrived example

```
\definemathset[VAR] [text={\mfunction{Var}}, left=(, right=), middle=\Vert]
\startformula
\VAR{f(X,Y) | Y = y }
\stopformula
```

gives

$$\text{Var}(f(X,Y) \parallel Y = y)$$

This module also takes care of correct nesting of math-sets, so

```
\startformula
\EXP{ \sum_Y \EXP{ \frac{1}{f(X)} | Y } }
\stopformula
```

gives

$$E\left\{\sum_Y E\left\{\frac{1}{f(X)} \middle| Y\right\}\right\}$$

If you do not want some `|` to be considered as conditional signs, nest them inside a group `{}`. For example, to get

$$\left\{x \in \mathbf{R}^2 \middle| 0 < |x| < \frac{3}{16}\right\}$$

we typed

```

\startformula
\mathset{ x\in {\bf R}^2 \mid 0<{|x|}<\frac {3}{16} }
\stopformula

```

We can also use limits after the command, for example:

```

\startformula
\EXP_X{F(X, Y) \mid Y = y }
\stopformula

```

gives

$$E_X \{F(X, Y) \mid Y = y\}$$

Only one set, `\mathset`, is predefined. It is relatively simple to define sets equivalent to those defined in `braket.sty`.

```

\definemathset[BRAKET] [left=\langle, right=\rangle]

\startformula
\BRAKET{ \phi \mid \frac{\partial^2}{\partial t^2} \mid \psi }
\stopformula

```

$$\left\langle \phi \left| \frac{\partial^2}{\partial t^2} \right| \psi \right\rangle$$

3 Implementation

Most of the ideas are simply a CONTeXtified version of the code in `braket.sty`. I mostly used `bracket.sty` to define commands for probability and expectation. So, I have also added the option of declaring such operators using `text=no` option for `\definemathset`.

```

1 \writestatus {loading} {ConTeXt Math Sets Module}
2 \startmodule[mathsets]
3 \unprotect

```

Since two letter codes are reserved for system modules, and CONTeXT seems to be running out of those, I choose a more verbose variable to store options.

```

4 \definesystemvariable {mathset} % Math Set

```

`\setupmath..` To specify the default values of text, left, middle, and right delimiters

```

5 \def\setupmathset
  {\dosingleargument\getparameters[ \??mathset ]}

```

`\definemat..` To define a new math set.

```

6 \def\definemathset
  {\dodoubleargument\dodefinemathset}

```

Now we define internal macros to take care of the formatting

```

7 \let\currentmathset\empty
  \let\currentmathsetgrouplevel\empty

8 \def\mathsetmiddle
  {\ifnum\currentmathsetgrouplevel=\currentgrouplevel
    \expandafter\firstoftwoarguments
  \else
    \expandafter\secondoftwoarguments
  \fi
  {\egroup\,\middle\mathsetparameter\c!middle\,\bgroup}
  {\mathsetparameter\c!middle}}

9 \def\mathsetparameter#1%
  {\executeifdefined{\??mathset\currentmathset#1}{\executeifdefined{\??mathset#1}\empty}}

10 \def\doddefinemathset[#1][#2]%
    {\getparameters[\??mathset#1][#2]
     \setvalue{#1}{\doddefinemathset[#1]}}

```

Since `|` is already active, we do not have to make it active again.

```

11 \def\doddefinemathset[#1]%
    {\begingroup
     \def\currentmathset{#1}
     \edef\currentmathsetgrouplevel{\the\numexpr\currentgrouplevel+2\relax}
     \mathcode'\|32768
     \let|\mathsetmiddle
     \doifelsenothing{\mathsetparameter\c!text}
       {\dododdefinemathset!notext}
       {\doifelse{\mathsetparameter\c!text}{\v!no}
        {\dododdefinemathset!notext}
        {\docapturemathoplimits\dododdefinemathset!text}}}}

```

`\docapturemathoplimits` is to capture limits that may follow the text command. This allows the following to work

```

\startformula
  \PR~{f,g} {f(X) | g(Y)}
\stopformula

```

$$\Pr^{f,g}(f(X) | g(Y))$$

```

12 \def\dododdefinemathset!notext#1%
    {\mathopen{}\left\mathsetparameter\c!left
     {#1}%
     \right\mathsetparameter\c!right\mathclose{}}%
    \endgroup}

```

TODO. Keep the `\nolimits` to be configurable.

```

13 \def\dododdefinemathset!text#1#2%
    {\mathop{\kern\zeropoint\mathsetparameter\c!text}\nolimits#1%
     \left\mathsetparameter\c!left
     {#2}%
     \right\mathsetparameter\c!right%
     \endgroup}

```

The extra group in the definition of `\dodododefinemathset!` is so that such expressions turn out correct

$$E \left\{ \left(\frac{a}{b} \right) \middle| \left(\frac{a}{\sum c} \right) \right\}$$

The `\left` and `\right` generate a math atom of type inner, while for math sets, we want a math math open atom. To see the difference, consider

```
\startformula
2\left(\frac {3}{4} \right) \quad \hbox{ vs } \quad
2\biggl( \frac {3}{4} \biggr)
\stopformula
```

and

```
\startformula
\Pr\left(\frac {3}{4} \right) \quad \hbox{ vs } \quad
\Pr\biggl( \frac {3}{4} \biggr)
\stopformula
```

which gives (notice the spacing before the parenthesis)

$$2 \left(\frac{3}{4} \right) \quad \text{vs} \quad 2 \left(\frac{3}{4} \right)$$

and

$$\Pr \left(\frac{3}{4} \right) \quad \text{vs} \quad \Pr \left(\frac{3}{4} \right)$$

I will assume that if `text` is something, then the default behaviour is desirable, if `text` is empty, then I add `\mathopen` and `\mathclose`. Using `\mathopen` to correct the spacing is due to Frank Mittelbach, see <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3853>

Mathset module ensures that we get the correct spacing in both cases

$$2 \left(\frac{3}{4} \right) \quad \text{and} \quad \Pr \left(\frac{3}{4} \right)$$

which was typed as

```
\definemathset[SET][left=(,right=)]
\startformula
2\SET{\frac{3}{4}} \quad \hbox{ and } \quad
\PR{ \frac{3}{4} }
\stopformula
```

Also, if its argument is a single character, `\mathop` centers it to with respect to the math-axis. Compare the outputs of

```
\ruledhbox{$\mathop{y}\nolimits_x\left\{A\,,\middle|\,,B\right\}$}
\ruledhbox{$\mathop{\kern\zeropoint y}\nolimits_x\left\{A\,,\middle|\,,B\right\}$}
yx.{A|B}
yx.{A|B}
```

I have added a `\kern\zeropoint` to prevent that.

`\docapture..` The next macro captures math limits. This should probably go to some general purpose module. There are three different valid inputs

1. An operator with neither subscript nor superscript.
2. An operator with one subscript or superscript.
3. An operator with both subscript and superscript.

So we scan for four arguments, to capture the following situations

- $_{\text{sub}}^{\text{sup}}$
- $^{\text{sup}}_{\text{sub}}$
- $_{\text{sub}}$
- $^{\text{sup}}$
- $\langle \text{empty} \rangle$

```

14 \def\docapturemathoplimits#1%
    {\doifnextcharelse_%
     {\dodocapturemathoplimits{#1}}
     {\doifnextcharelse^%
      {\dodocapturemathoplimits{#1}}
      {#1{}}}}

15 \def\dodocapturemathoplimits#1#2#3%
    {\doifnextcharelse_%
     {\redocapturemathoplimits{#1}{#2}{#3}}
     {\doifnextcharelse^%
      {\redocapturemathoplimits{#1}{#2}{#3}}
      {#1{#2{#3}}}}}

16 \def\redocapturemathoplimits#1#2#3#4#5%
    {#1{#2{#3}{#4{#5}}}}

17 \setupmathset
    [ \c!left={\{} ,
      \c!right={\}},
      \c!middle=\vert,
      \c!text=no,]

18 \definemathset[mathset]

```

4 Changelog

4. July 3, 2008

Added `text=no` option, included an interface file, and cleaned up the documentation for T_EXlive 2008.

3. June 17, 2007

Added `\docapturemathoplimits` macro. This prevents a serious bug in the previous version, due to which things like `\mathset_{..}` did not work.

2. April 11, 2007

This version provides some fine tuning of how the sets are displayed by working around two mis-features of T_EX math: `\left ... \right` always create a math inner atom and `\mathop` centers its argument if the argument is a single letter.

1. February 25, 2007

First version of the module.

```
19 \protect  
   \stopmodule
```

```
\definemathset 3  
\docapturemathoplimits 6
```

```
\setupmathset 3
```