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

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

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 \TeX, Donald Arseneau's \texttt{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 \texttt{braket.sty} to \texttt{CONTEXT}.

2 Usage

To use this module add

\begin{verbatim}
\usemodule[mathsets]
\end{verbatim}

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

\texttt{unknown setup 'definemathset'}

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

\begin{verbatim}
\definemathset[EXP] [text=\mfunction{E}]
\end{verbatim}

we can use \texttt{EXP{X}} to get \( E\{X\} \) and \texttt{EXP{X|Y}} to get \( E\{X \mid Y\} \). Scaling of the delimiters and conditional sign are take care automatically. For example

\begin{verbatim}
\startformula
\EXP{\sum_y f(X,y) \, \mid \, Z}
\stopformula
\end{verbatim}

gives

\[ \texttt{EXP{\sum_y f(X,y) \, \mid \, Z}} \]
E \left\{ \sum_y f(X,y) \bigg| 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) \mid Z = z } = \sum_{x,y} \PR{x,y \mid Z=z}
\stopformula
```

gives

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

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) \mid Y = y}
\stopformula
```

gives

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

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

```
\startformula
\EXP{ \sum_{Y} \EXP { \frac{1}{f(X)} \mid Y } }
\stopformula
```

gives

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

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

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

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, \texttt{\textbackslash mathset}, is predefined. It is relatively simple to define sets equivalent to those defined in \texttt{braket.sty}.

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

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

\langle \phi \mid \frac{\partial^2}{\partial t^2} \mid \psi \rangle

3 Implementation

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

\verbatiminput{mathsets}

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

\verbatiminput{mathsets}

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

\verbatiminput{mathsets}

Now we define internal macros to take care of the formatting.

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

\def\dodododedefinemathset[#1]{% 
  \begingroup
  \def\currentmathset{#1}
  \def\currentmathsetgrouplevel{\the\numexpr\currentgrouplevel+2\relax}
  % Not here, else messes subscripts
  % \mathcode\|32768
  % \let\|\mathsetmiddle
  \doifelsenothing{\mathsetparameter{c!text}}
  {\dododododedefinemathset!notext}
  {\doifelse{\mathsetparameter{c!text}}{\v!no}{\dododododedefinemathset!notext}}
  {\docapturemathoplimits\dododododedefinemathset!text}
}

\def\setmathmiddle{% 
  \mathcode\|32768
  \let\|\mathsetmiddle
  \docapturemathoplimits
}

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

\startformula
\PR^{f,g} \{f(X) \mid g(Y)\}
\stopformula

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

We need to be a bit careful not to activate \mid too soon, as it can also occur in sub- and superscripts. For example

\startformula
\EXP_{{X|Y}}\{f(X) \mid Y = y\}
\stopformula

gives

$$E_{X|Y}\{f(X) \mid Y = y\}$$

\def\dododododedefinemathset!notext#1{% 
  \setmathmiddle
  \mathopen{\left\mathsetparameter{c!left}#1\right\mathsetparameter{c!right}\mathclose{}}
}
TODO. Keep the \nolimits to be configurable.

\def\dodododefinemathset!text#1#2\%
{\mathop\{\kern\zeropoint\mathsetparameter\c!text\nolimits\#1%
\setmathmiddle
\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

\begin{verbatim}
\startformula
2\left(\frac{3}{4}\right) \quad \hbox{ vs } \quad 2\biggl(\frac{3}{4}\biggr)
\stopformula
and
\begin{verbatim}
\startformula
\Pr\left(\frac{3}{4}\right) \quad \hbox{ vs } \quad \Pr\biggl(\frac{3}{4}\biggr)
\stopformula
\end{verbatim}
\end{verbatim}

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

\begin{verbatim}
\definemathset[SET][left=,right=]
\startformula
2\{\frac{3}{4}\} \quad \hbox{ and } \quad \Pr\{\frac{3}{4}\}
\stopformula
\end{verbatim}
Also, if its argument is a single character, \texttt{mathop} centers it to with respect to the math-axis. Compare the outputs of

\begin{verbatim}
\ruledhbox{$\mathop{y}_{x}\left\{ A,|,B\right\}$}
\ruledhbox{$\mathop{\kern\zeropoint y}_{x}\left\{ A,|,B\right\}$}
\end{verbatim}

I have added a \texttt{kern/zeropoint} to prevent that.

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

- \{sub\}–{sup}
- \{sup\}–{sub}
- \{sub\}
- \{sup\}
- \{empty\}

\begin{verbatim}
\def\docapturemathoplimits#1\%
\{\doifnextcharelse_%
\{\docaptemathoplimits{#1}\}
\{\doifnextcharelse"_%
\{\docaptemathoplimits{#1}\}
\{\#1{}{}{}\}\}\}
\end{verbatim}

\begin{verbatim}
\def\docapturemathoplimits#1#2#3\%
\{\doifnextcharelse_%
\{\docaptemathoplimits{#1}{#2}{#3}\}
\{\doifnextcharelse"_%
\{\docaptemathoplimits{#1}{#2}{#3}\}
\{#1(#2(#3)){}\}\}\}
\end{verbatim}

\begin{verbatim}
\def\docapturemathoplimits#1#2#3#4#5\%
{#1(#2(#3)#4(#5))}\}
\end{verbatim}

\begin{verbatim}
\setupmathset
\{ \c!left={\{}{},
\c!right={\}{}{},
\c!middle=\vert{},
\c!text=no,\}
\end{verbatim}

\begin{verbatim}
\definemathset[mathset]
\end{verbatim}

4 Change log

5. December 6, 2008

Defined a new macro \texttt{setmathmiddle}. Now \texttt{|} is made active after the subscripts, so that things still work when \texttt{|} is used in the subscripts.

Added \textit{text=no} option, included an interface file, and cleaned up the documentation for \TeX\live\! 2008.

3. June 17, 2007

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

2. April 11, 2007

This version provides some fine tuning of how the sets are displayed by working around two misfeatures of \TeX\ math: \texttt{\left ... \right} always create a math inner atom and \texttt{\mathop} centers its argument if the argument is a single letter.

1. February 25, 2007

First version of the module.

20 \protect
\stopmodule
\definemathset 3
\docapturemathoplimits 6
\setupmathset 3