The `jkmath` package

Jonas Kaerts

March 25, 2018

1 The package options

The package contains a few options with regards to subsets.

subsetorder You are a person that likes your symbols for subsets to resemble the symbols used for ordering numbers. The command `\subset` now displays the symbol $\subseteq$ while a new command `\stsubset` (for strict subsets) can be used for dispaying the symbol $\subset$. Similar behavior occurs with `\supset` and `\stsupset`.

subsetnonorder You are a person that likes variety. Your symbols for subsets do not resemble the usual ordering symbols. the command `\subset` displays the symbol $\subset$ while the symbol `\stsubset` displays as $\subsetneq$. Same for `supset` and `stsupset`.

subsetnonamb You like your notation as unambiguous as possible. The command `\subset` displays the symbol $\subseteq$ while `\stsubset` displays $\subsetneq$. Again similar for `\supset` and `\stsupset`.

The advantage of this approach is that you can convert a document from one style of notation to another by simply changing the package option.

There are also two options, `bbsets` and `bfsets` concerning the display of number systems. They provide the following shorthands:

Command	Option `bbsets`	Option `bfsets`	Usage
`\N`	$\mathbb{N}$	$\mathbf{N}$	Natural numbers
`\Z`	$\mathbb{Z}$	$\mathbf{Z}$	Integers
`\Q`	$\mathbb{Q}$	$\mathbf{Q}$	Rational numbers
`\R`	$\mathbb{R}$	$\mathbf{R}$	Real numbers
`\C`	$\mathbb{C}$	$\mathbf{C}$	Complex numbers
`\F`	$\mathbb{F}$	$\mathbf{F}$	Fields
`\Aff`	$\mathbb{A}$	$\mathbf{A}$	Affine Space
`\PP`	$\mathbb{P}$	$\mathbf{P}$	Projective Space

2 Commands with arrays

2.1 Systems of equations

This package uses the `array`-package to define some useful math alignment. The first is the `system` environment. There are two new column types (`e` and `o`) to get the spacing around operators right. You can then call the code

```
\begin{system}{rorer}
4x & + & 3y & 7\\
2x & - & 5y & 10
\end{system}
```

to get the result

$$\begin{cases} 4x + 3y = \ \ 7 \\ 2x - 5y = 10 \end{cases}.$$

This allows fine control over the alignment of a system of equations while still having the correct spacing. Note that the column type `e` automatically inserts an equality sign.

2.2 Augmented matrices

A second class of commands are the augmented matrices. The environment `augmentedmatrix` takes two arguments n and m and makes a matrix of $n+m$ columns with a vertical rule after the n-th column, allowing the typesetting of systems with (multiple) right hand sides in matrix form. The code

```
\begin{augmentedmatrix}{2}{2}
1 & 2 & 3 & 4\\
5 & 6 & 7 & 8
\end{augmentedmatrix}
```

has the following output:

$$\left(\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{array}\right)$$

At the moment there are two shorthand commands `apmqty` and `ipmqty` which take $m = 1$ and $m = n$ respectively and insert parentheses. These are used for solving systems with one right hand side and for calculating inverse matrices. The shorthand name is inspired by the shorthands in the `physics`-package. The code

```
\amqty{2}{1 & 2 & 3 \\ 4 & 5 & 6}
\neq
\ipmqty{3}{0 & 1 & 0 & 1 & 0 & 0\\
-1 & 0 & 2 & 0 & 1 & 0\\
0 & 0 & 3 & 0 & 0 & 1}
```

produces the following output:

$$\amqty{2}{1 & 2 & 3 \\ 4 & 5 & 6} \neq \ipmqty{3}{0 & 1 & 0 & 1 & 0 & 0\\ -1 & 0 & 2 & 0 & 1 & 0\\ 0 & 0 & 3 & 0 & 0 & 1}$$

3 Delimiters and intervals

I often use a script to check if my code is consistent in its use of delimters since LaTeX allows you to have unmatched parentheses etc. in the text. The commands `\lbrace`, `\rbrace`, `\lbrack` and `\rbrack` are a godsend when I both want my script to give meaningful output and I only need one delimiter (such as in the `system` environment). This package defines similar commands `\lparens` and `\rparens` for parentheses.

Using these delimiter commands the package also defines four types of intervals: `\oointerval`, `\ccinterval`, `\ocinterval` and `\cointerval`. The o and c say whether the left or right endpoint is open or closed. The code

```
\cointerval{1,3} \cup \ccinterval{3,7} = \ccinterval{1,7}
```

typesets the following output

$$\cointerval{1,3} \cup \ccinterval{3,7} = \ccinterval{1,7}.$$

You can define your own shorthands for these commands or redefine the style of the intervals.

4 Sets

A general macro for denoting sets is `\set` which automatically places scalable braces around the argument. A scalable version of `\mid`, called `\where`, is also included. This makes sure the (readable) code

```
\set{x\in\mathbb{R} \where \frac{3}{4}x + 5 = 0}
```

will give the following result:

$$\left\{ x \in \mathbb{R} \;\middle|\; \frac{3}{4}x + 5 = 0 \right\}.$$

A second macro is `\restr` for denoting restritions of functions to subsets of their domain. Simple usage is `\restr{f}_U` which displays $f|_U$.

5 Combinatorics

Using `\genfrac` from `amsmath` the package defines two commands for Stirling numbers of the first and second kind. Example usage:

```
\stirlingfirstkind{n}{k}=
\stirlingsecondkind{-k}{-n}
```

gives the output

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{Bmatrix} -k \\ -n \end{Bmatrix}.$$

Shorthands for these two commands have yet to be defined.

6 Number Theory

Two commands (with identical results) `\legendre` and `\jacobi` are defined to typeset Legendre symbols and Jacobi symbols. The output is identical but their name differs to make the code more readable. Example usage:

```
\jacobi{a}{n} =
\legendre{a}{p_1}^{e_1}
\legendre{a}{p_2}^{e_2}\cdots
\legendre{a}{p_k}^{e_k}
```

gives the output

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

7 Names of mathematicians

This section describes three simple commands `\mobius`, `\cech` and `\erdos` so you can mention Möbius, Čech and Erdős without any pain.