The PemberleyMath Package

Adrian Rettich

February 19, 2020

1 Synopsis

The PemberleyMath package loads some useful packages and provides a slew of commands I need often.
If you want to load this package for some of its new macros without changing anything about code you have already written, simply load it with the *newmacrosonly* option.

2 Loaded Packages

PemberleyMath loads **pemberley**, **amsmath**, **amssymb**, **amsfonts**, **mathtools**, **mathdots**, **faktor**, and **braket**.

3 Formatting

By default, PemberleyMath shows numbers on only those equations that need them, i.e. those that are referenced somewhere in your document. You can suppress this behavior (such that all equations are numbered) by passing the *equationnumbers* option.
Finally, PemberleyMath swaps the commands \phi and \varphi, such that the former now produces φ and the latter produces ϕ. You can keep the original commands (though why you would want to escapes me) by passing the *keepphi* option. The same goes for the \epsilon (now produces ε) and \varepsilon (now ϵ) commands together with the *keepepsilon* option.

4 Macros

PemberleyMath provides the following additional macros.

4.1 Number Spaces

\NN, \ZZ, \QQ, \RR, \CC Aliases for \mathbb{letter}, i.e. $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$. These can be used without manually entering math mode.

4.2 Math Operators

Note that these can only be used in math mode.

\Hom, \Obj, \id Give $\mathrm{Hom}, \mathrm{Obj}$, and id, respectively.

\ker, \im, \coker, \coim Kernel, image, cokernel and coimage. These override the default behavior by shrinking the space before the next character to make parens-less notation look a bit nicer, as in $\ker f$, $\mathrm{im} g^2$. You may want to add a \thinspace when using parens or chaining operators, as in $\mathrm{coker}\,\mathrm{coim}(h)$.

4.3 Sets

The **braket** package provides a wonderful option for typesetting sets in math mode.

\set{x | x\in \RR} Gives $\{\, x \mid x \in \mathbb{R} \,\}$.

This is the same as the \Set command in **braket**. Note that in Pemberley, the command is spelled with a lowercase s because \Set is reserved for the category Set. To suppress this change and retain the original \set and \Set commands from **braket**, load PemberleyMath with the option *keepset*.
Furthermore, PemberleyMath provides macros for standard intervals (you can put whatever you want in place of "0,1"):

\intervaloo{0,1} The open interval $(0, 1)$.

\intervalcc{0,1} The closed interval $[0, 1]$.

\intervaloc{0,1} The half open interval $(0, 1]$.

\intervalco{0,1} The half open interval $[0, 1)$.

Finally:

\emptyset Is changed to mean $\varnothing$ rather than $\emptyset$. The latter is still accessible as \uglyemptyset, and you can disable the new behavior by loading PemberleyMath with the *keepemptyset* option.

\uglyemptyset Produces $\emptyset$.

4.4 Categories

I like typesetting category names in smallcaps. A few are provided by default: \Grp, \Ab, \Ring, \Rng, \Top, and \Pair produce Grp, Ab, Ring, Rng, Top, Pair, respectively. There is no need to change to math mode. \Set produces Set, unless you loaded PemberleyMath with the *keepset* option.
You can use \cat{#1} to typeset categories not in the above list: \cat{Foo} produces Foo.
Finally, some categories take arguments:

\Mod{R} produces ${}_R\textsc{Mod}$ for the category of R-modules.

\pair{X}{Y} produces $\mathfrak{P}_{X,Y}$.

\Ch{\Ab} produces $\textsc{Ab}_\bullet$ for the category of chain complexes over $\textsc{Ab}$.

\CoCh{\Ab} produces $\textsc{Ab}^\bullet$ for cochain complexes.

4.5 Formal Logic

\iff Produces $\Leftrightarrow$, which is shorter than the standard $\Longleftrightarrow$.

\implies Produces $\Rightarrow$, which is shorter than the standard $\Longrightarrow$.

\follows Produces the corresponding $\Leftarrow$.

\land Produces ${\scriptstyle\wedge}$, which to me looks a lot nicer than the default $\wedge$. You can still access the latter as \wedge.

\lor Produces ${\scriptstyle\vee}$, which to me looks a lot nicer than the default $\vee$. You can still access the latter as \vee.

\eu Stands for "exists uniquely" and produces $\exists!$.

4.6 Maps and Relations

\definedas Produces $\coloneqq$.

\iso I like to use $\cong$ to mean "isomorphic to".

\homot Likewise, $\sim$ would be "homotopic to".

\subset Pemberley changes this command to mean $\subseteq$ rather than $\subset$. The old symbol still exists as \strictsubset.

\strictsubset Gives $\subset$.

\supset As with \subset, Pemberley changes this command to mean $\supseteq$ rather than $\supset$. The old symbol still exists as \strictsupset.

\strictsupset Gives $\supset$.

\mto, \eto, \ito Like \to ($\to$), these produce arrows for maps, namely for injective (**m**onomorphisms) $\hookrightarrow$, surjective (**e**pimorphisms) $\twoheadrightarrow$, and bijective (**i**somorphisms) $\xrightarrow{\sim}$ maps.

\blank Gives a blank for function or functor definitions, as in x\mapsto(x,\blank), which renders as $x \mapsto (x, \_)$.

4.7 Commutative Diagrams

At some point I might move these to a dedicated package for typesetting commutative diagrams with tikz.

\isolabel gives you a $\sim$ to put over arrows.

\commutes gives you a nice little $\circlearrowleft$ to put in the center of your diagram.

4.8 Miscellaneous Symbols

\where Can be used to typeset constraints: \sum a_i\where a_i \in A renders as $\sum a_i, \quad a_i \in A$.

\powerset{set} Nicely typesets the power set of a set, as in $\mathcal{P}\mathbb{R}$. This is slightly different from simply putting a \mathcal{P} in front of your set, as the latter results in bad spacing: $\mathcal{P}\,\mathbb{R}$.

\indexify{#1}, \exponentify{#1} You can use these to make your indices and exponents smaller under specific circumstances. For example, \RRˆ 1 ($\mathbb{R}^1$) looks fine, but \RR_{\geq 1} ($\mathbb{R}_{\geq 1}$) really should be $\mathbb{R}_{\scriptscriptstyle\geq\!1}$. The latter is achieved via \RR_{\indexify{\geq\negthinspace 1}}. \exponentify works the same way, but for exponents.

\abs{foo} Nicely sets the absolute value, as in $x = |foo|$.

\inv The inverse of something: f\inv produces f^{-1}.

\transp The transpose: a\transp produces $a^\top$.

5 Options

PemberleyMath accepts the same options as the base Pemberley package (with the same effects), as well as the following additional options:

newmacrosonly Do not redefine any macros, and do not change any formatting.

keepset Do not redefine \set and \Set. Instead, keep the versions from **braket**.

keepphi Do not swap \phi and \varphi.

keepepsilon Do not swap \epsilon and \varepsilon.

keepemptyset Do not renew the \emptyset macro.

equationnumbers Number *all* equations, not just those you reference somewhere.

6 License