The thmtools bundle*

Ulrich M. Schwarz, ulmi@absatzen.de

June 22, 2008

Abstract

thmtools is a collection of tools and enhancements for theorem environments.

Remark 1. *Each of the packages has its own documentation in a file named after the package itself. The only difference is that those documentations include the reasonably-well documented source code.*

thm-kv

1 Usage

\declaretheorem The macro \declaretheorem[⟨*key=val-opts*⟩]{⟨*name*⟩} can be used to define a new theorem instead of \newtheorem. It is hoped that \declaretheorem is easier to use than \newtheorem's tangle of mutually-exclusive optional arguments. The following is the list of keywords understood:

parent, numberwithin, within These keys govern when the theorem counter is reset. For example, giving parent=chapter gives you theorems numbered per chapter, so it's equivalent to the second optional argument to \newtheorem. There are three names so you'll remember at least one of them.

sibling, numberlike, sharenumber These keys make the theorem share a common numbering with the given theorem. This is just like giving the first optional argument.

unnumbered, starred If your theorem package supports it, this will call \newtheorem*, i.e. you'll get a theorem that is never numbered. Currently, only amsmath offers this possibility.

name, title, heading \newtheorem takes *two* options, the name of the environment (like lem) and its title (Lemma). \declaretheorem only requires the environment name, and the title defaults to the environment name with the first letter uppercased. If you name your environments lemma, theorem, etc., you don't need to do anything else. Otherwise, you can always manually specify the title.

*This documents thmtools version 2008/06/22 v0.1beta6. Newer releases might be available at http://absatzen.de/.

(pre/post)(head/foot)hook You can specify extra code that will be executed whenever you use the environment. **Warning:** this needs the thm-patch package, and you're responsible for loading it yourself if you want these keys to work. This functionality might be shifted over to thm-patch in future releases.

style This will issue a `\theoremstyle{foo}` for you if you give `style=foo`. Note that currently, no care is taken to prevent this from becoming the default style for subsequent theorems.

thm-restate

2 Usage

`restatable` Only one environment is provided: `restatable`, which takes one optional and two mandatory arguments. The first mandatory argument is the type of the theorem, i.e. if you want `\begin{lemma}` to be called on the inside, give `lemma`. The second argument is the name of the macro that the text should be stored in, for example `mylemma`. Be careful not to specify existing command names! The optional argument will become the optional argument to your theorem command. Consider the following example:

```
\documentclass{article}
\usepackage{amsmath, amsthm, thm-restate}
\newtheorem{lemma}{Lemma}
\begin{document}
  \begin{restatable}[Zorn]{lemma}{zornlemma}\label{thm:zorn}
    If every chain in $X$ is upper-bounded,
    $X$ has a maximal element.

    It's true, you know!
  \end{restatable}
  \begin{lemma}
    This is some other lemma of no import.
  \end{lemma}
  And now, here's Mr. Zorn again: \zornlemma*
\end{document}
```

which yields

Lemma 1 (Zorn)**.** *If every chain in X is upper-bounded, X has a maximal element. It's true, you know!*

Lemma 2. *This is some other lemma of no import.*

Actually, we have set a label in the environment, so we know that it's Lemma 1 on page 1. And now, here's Mr. Zorn again:

Lemma 1 (Zorn)**.** *If every chain in X is upper-bounded, X has a maximal element. It's true, you know!*

Since we prevent the label from being set again, we find that it's still Lemma 1 on page 1, even though it occurs later also.

restatable*

As you can see, we use the starred form `\mylemma*`. As in many cases in LaTeX, the star means "don't give a number", since we want to retain the original number. There is also a starred variant of the `restatable` environment, where the first call doesn't determine the number, but a later call to `\mylemma` without star would. Since the number is carried around using LaTeX' `\label` machanism, you'll need a rerun for things to settle.

2.1 Restrictions

The only counter that is saved is the one for the theorem number. So, putting floats inside a restatable is not advised. You cannot nest restatables either. You *can* use the `\restatable...\endrestatable` version, but everything up to the next matching `\end{...}` is scooped up. I've also probably missed many border cases.

thm-autoref

3 Usage

hyperref's `\autoref` command does not work well with theorems that share a counter: it'll always think it's a Lemma even if it's a Remark that shares the Lemma counter. Load this package to fix it. No further intervention needed.

thm-listof

4 Usage

\listoftheorems

This package provides two main commands: `\listoftheorems` will generate, well, a list of all theorems, lemmas, etc. in your document. This list is hyperlinked if

\listtheoremname

you use hyperref, and it will list the optional argument to the theorem. The heading name is stored in the macro `\listtheoremname` and is "List of Theorems" by default. All other formatting aspects are taken from `\listoffigures`. (As a matter of fact, `\listoffigures` is called internally.)

\ignoretheorems

`\ignoretheorems{`⟨*remark,example,...*⟩`}` can be used to suppress some types of theorem from the LoTh. Be careful not to have spaces in the list, those are currently *not* filtered out.

There's currently no interface to change the look of the list. If you're daring, the code for the theorem type "lemma" is in `\l@lemma` and so on.

thm-patch

5 Usage

This package is maybe not very suitable for the end user. It redefines \newtheorem in a way that lets other packages (or the user) add code to the newly-defined theorems, in a reasonably cross-compatible (with the kernel, theorem and amsthm) way.

Warning: the new \newtheorem is a superset of the allowed syntax. For example, you can give a star and both optional arguments, even though you cannot have an unnumbered theorem that shares a counter and yet has a different reset-regimen. At some point, your command is re-assembled and passed on to the original \newtheorem. This might complain, or give you the usual "Missing \begin{document}" that marks too many arguments in the preamble.

\addtotheorem(pre/post)(head/foot)hook A call to \addtotheorempreheadhook[⟨*kind*⟩]{⟨*code*⟩} will insert the code to be executed whenever a kind theorem is opened, before the actual call takes place. (I.e., before the header "Kind 1.3 (Foo)" is typeset.) There are also posthooks that are executed after this header, and the same for the end of the environment, even though nothing interesting ever happens there. These are useful to put \begin{shaded}...\end{shaded} around your theorems. Note that foothooks are executed LIFO (last addition first) and headhooks are executed FIFO (first addition first). There is a special kind called generic that is called for all theorems. This is the default if no kind is given.

The added code may examine \thmt@thmname to get the title, \thmt@envname to get the environment's name, and \thmt@optarg to get the extra optional title, if any.

aliasctr

6 Usage

\@counteralias{#1}{#2} makes #1 a counter that uses #2's count register. This is useful for things like hyperref's \autoref, which otherwise can't distinguish theorems and definitions if they share a counter.

For detailed information, see Die TeXnische Komödie 3/2006.

parseargs

7 Usage

The main command provided by the package is \parse{⟨*spec*⟩}. *spec* consists of groups of commands. Each group should set up the command \@parsecmd which is then run. The important point is that \@parsecmd will pick up its arguments from the running text, not from the rest of *spec*. When it's done storing the arguments, \@parsecmd must call \@parse to continue with the next element of *spec*. The process terminates when we run out of spec.

Helper macros are provided for the three usual argument types: mandatory, optional, and flag.

unique

8 Usage

Two macros are provided: `\setuniqmark` takes a single parameter, the name, which should be a string of letters. `\ifuniqmark` takes three parameters: a name, a true-part and a false-part. The true part is executed if and only if there was exactly one call to `\setuniqmark` with the given name during the previous LaTeX run.

Example application: legal documents are often very strongly numbered. However, if a section has only a single paragraph, this paragraph is not numbered separately, this only occurs from two paragraphs onwards.

It's also possible to not-number the single theorem in your paper, but fall back to numbering when you add another one.