

repththeorem*

Jesse Straat

2024-03-27

Abstract

When writing a large manuscript, it is sometimes beneficial to repeat a theorem (or lemma or ...) at an earlier or later point for didactical purposes. However, `thmtools`'s built-in `restatable` only allows replicating theorems *after* they have been stated, and only in the same document. `repththeorem` solves the issue by making use of the `.aux` file, and also introduces its own file extension, `.thm`, to replicate theorems in other files.

Contents

1 Repeating theorems	1
2 Replicating theorems between files	3
2.1 Replicating theorems to subfiles	3
3 Source code	3

1 Repeating theorems

Let's say we define a theorem as follows:

```
\begin{theorem}[Yoneda Lemma]
  For  $(F\colon \mathcal{C}\to \mathbf{Set})$  a functor,
   $([\mathcal{C}^{\mathrm{op}}, \mathbf{Set}](YA, F) \cong F(A))\%$ 
  for all objects  $A$  in  $\mathcal{C}$ .
\end{theorem}
```

Its output is of course

Theorem 1 (Yoneda Lemma). *For $F\colon \mathcal{C} \rightarrow \mathbf{Set}$ a functor, $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}](YA, F) \cong F(A)$ for all objects A in \mathcal{C} .*

Now let's say we want to replicate the theorem within the same document. `makethm` (*env.*) That is what the new environment `makethm` is used for.

```
\begin{makethm}{theorem}{thm:Yoneda}[Yoneda Lemma]{
  For  $(F\colon \mathcal{C}\to \mathbf{Set})$  a functor,
```

*Version v1.0, last revised 2024-03-27.

```

\([\mathcal{C}^{\mathrm{op}}, \mathbf{Set}](YA, F) \cong F(A)\)%
for all objects \((A)\) in \((\mathcal{C})\).
\end{makethm}

```

Its output is the same (in fact, we’ve secretly used `makethm` in the previous example), but the important difference is that we have saved the theorem for later use.

The `makethm` environment takes two mandatory arguments and one optional one. The first mandatory argument is the type of theorem environment, like `theorem`, `lemma`, `definition`, etc. The second is the theorem’s label. The label is mandatory because to replicate the theorem, we need to have a “name” attached to it. `makethm` automatically attaches a `\label`, as well, so `\ref{thm:Yoneda}` becomes `1`. The optional argument is passed right to the optional argument of the theorem environment, giving the name of a theorem

Now let’s say we want to replicate the theorem later or earlier in the text. This may be done if, for example, the theorem is proven at a later point, or we want to “tease” the reader with a powerful theorem that will be proven later in the chapter. To do this, we use the command, as follows.

```

\repthm{theorem}{thm:Yoneda}

```

This outputs the theorem again.

Theorem 1 (Yoneda Lemma). *For $F: \mathcal{C} \rightarrow \mathbf{Set}$ a functor, $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}](YA, F) \cong F(A)$ for all objects A in \mathcal{C} .*

The label of this theorem is a `\ref`, and automatically links to the original theorem statement.

If the original theorem statement exists in a different file, or has not been created yet, we can add a placeholder alt text to the `\repthm` as an optional argument, which only displays if the theorem is undefined. For example, `\repthm{theorem}{thm:foo}[bar]` returns

Theorem ??. *bar*

If we do the same without providing an alt text, we get

Theorem ??.

together with a warning.

Since we’re using the `.aux` file, it is possible to replicate a theorem before it is stated. For example,

```

\repthm{theorem}{thm:later}
\begin{makethm}{theorem}{thm:later}[Later]
  Alligator!
\end{makethm}

```

returns

Theorem 2 (Later). *Alligator*

Theorem 2 (Later). *Alligator*

Note that it is necessary to run a `.tex` file twice to replicate theorems ahead of time, similarly to how one has to run a file twice to make sure the references are correct.

2 Replicating theorems between files

Let's say we have the following files for our project:

```
foo.tex
bar.tex
```

Let's say that we have defined a theorem `thm:baz` in `bar.tex`, and we want to replicate it in `foo.tex`. To achieve this, we first use the `\theoremfile` command in the preamble of `bar.tex`. This compiles all theorems defined in `bar.tex` and outputs them into a file `bar.thm`. To then import these into `foo.tex`, we use `\loadtheorems` `\loadtheorems{bar.thm}` in the preamble, which loads all theorems saved in `bar.thm`. One can then use `\repthm` as usual.

Since the `.aux` file is loaded at `\begin{document}`, putting `\loadtheorems` in the preamble of a file will guarantee that the loaded theorem file will be overwritten by the theorems in the `.aux` file, i.e., theorems defined in the same document. In our example, if we also defined a `thm:baz` in `foo.tex`, loading `bar.thm` into `foo.tex` will not overwrite the local `thm:baz`.

2.1 Replicating theorems to subfiles

Replicating theorems to different files is particularly useful when working in big documents with multiple subfiles. For example, let's say we have the files

```
main.tex
foo.tex
bar.tex
```

Here, `main.tex` is generated by including `foo.tex` and `bar.tex` as chapters, creating a single large document. It is now possible to replicate theorems within the subfiles by running `\theoremfile` in `main.tex`, and then using `\loadtheorems{main.thm}` in `foo.tex` and `bar.tex`. This will allow us to use all theorems in the final `main.tex` in each of the subfiles.

3 Source code

```
1 (*package)
2 \ProvidesPackage{repththeorem}[2024-03-27 v1.0 Repththeorem package]
\theoremfile Using \theoremfile will output all saved theorems into an output file. By default,
if your LATEX file is foo.tex, the output file is foo.thm.
3 \def\repththeorem@theoremfile{\relax}
4 \NewDocumentCommand{\theoremfile}{0{\jobname.thm}}{
5 % 0: the path of the file to which we should save theorems
6 %
7 \def\repththeorem@theoremfile{#1}
8 \newwrite\@thmlist
9 \immediate\openout\@thmlist=#1
10 }
```

`\loadtheorems` If you have exported saved theorems to a file, you can load them into another file using the macro `\loadtheorems`.

```

11 \NewDocumentCommand{\loadtheorems}{ m }{
12 \IfFileExists{#1}{
13 \input{#1}
14 }{
15 \PackageWarning{repththeorem}{%
16 File #1 not found. I will not import any theorems.%
17 }
18 }
19 }

```

`makethm (env.)` On to defining the actual theorems to be saved.

```

20 \NewDocumentEnvironment{makethm}{ m m o +b }
21 % m: the type of theorem environment
22 % m: the name of the theorem
23 % o: optional parameter for environment
24 % b: the content of the theorem
25 %
26 {%
27 \IfValueTF{#3}{% Check if theorem has optional arguments
28 \begin{#1}[#3]\label{#2}
29 }{
30 \begin{#1}\label{#2}
31 }
32 #4
33 \end{#1}
34 \expandafter\long\expandafter\gdef\csname thm@#2\endcsname{#4}%
35 \expandafter\gdef\csname thmdesc@#2\endcsname{#3}%
36 % Saving parameters to aux file
37 \long\gdef\@thmoutput{%
38 \string\expandafter\string\long\string\expandafter%
39 \string\gdef\noexpand\csname thm@#2\string\endcsname{#4}%
40 ^^J%
41 \string\expandafter\string\expandafter%
42 \string\gdef\noexpand\csname thmdesc@#2\string\endcsname{#3}%
43 }
44 \write\@auxout{\@thmoutput}
45 \if\repththeorem@theoremfile\relax
46 % No file has been set
47 \else
48 % We have a theorem file
49 % Saving parameters to theorem file
50 \write\@thmlist{\@thmoutput}
51 \fi
52 }{ }

```

`\repthm` To repeat a theorem, use the `\repthm` command. The type of theorem is not saved, so it is necessary to respecify it each time a theorem is repeated.

```

53 \newcounter{old@counter}
54 \NewDocumentCommand{\repthm}{ m m +o }{
55 % m: the type of theorem environment
56 % m: the name of the theorem
57 % o: alt text

```

```

58 \begingroup
59 \setcounter{old@counter}{\value{#1}}
60 % Save theorem counter so we don't increase it
61 \def\thetheorem{\ref{#2}}
62 \let\@theoremnotdefined\relax
63 %
64 \ifcsname thm@#2\endcsname% Check if theorem is even defined
65 % Theorem is defined
66 \expandafter\edef\expandafter\@thmdesc{\csname thmdesc@#2\endcsname}%
67 \expandafter\let\expandafter\@thm\csname thm@#2\endcsname
68 % Output theorem
69 \IfValueTF{\@thmdesc}{% Check if theorem has name
70 \begin{#1}[\@thmdesc]
71 \@thm
72 \end{#1}
73 }{% No optionals
74 \begin{#1}
75 \@thm
76 \end{#1}
77 }
78 \else
79 % Theorem undefined
80 \IfValueTF{#3}{
81 \begin{#1}
82 #3
83 \end{#1}
84 }{% No theorem or alt text provided: throw warning
85 \begin{#1}
86 \end{#1}
87 \PackageWarning{repththeorem}{%
88 Theorem #2 not defined; rebuild your project.
89 If the issue persists, create the theorem using
90 \begin{makethm} or consider adding alt text to \repthm
91 using the optional parameter%
92 }
93 }
94 \fi
95 \setcounter{#1}{\value{old@counter}}
96 % Reset theorem counter back to original
97 \endgroup
98 }
99 </package>

```

Change History

v1.0
 General: First public release 1

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		L	P		
<code>\@theoremnotdefined</code>	<i>62</i>	<code>\loadtheorems</code>	<i>3</i> , <u><i>11</i></u>	<code>\PackageWarning</code>	<i>15</i> , <i>87</i>
<code>\@thm</code>	<i>67</i> , <i>71</i> , <i>75</i>				
<code>\@thmdesc</code>	<i>66</i> , <i>69</i> , <i>70</i>	M		R	
<code>\@auxout</code>	<i>44</i>	<code>makethm (env.)</code>	<i>1</i> , <u><i>20</i></u>	<code>\reptheorem@theoremfile</code>	
<code>\@thmlist</code>	<i>8</i> , <i>9</i> , <i>50</i>				<i>3</i> , <i>7</i> , <i>45</i>
<code>\@thmoutput</code>	<i>37</i> , <i>44</i> , <i>50</i>	N		<code>\repthm</code>	<i>2</i> , <u><i>53</i></u>
		<code>\newwrite</code>	<i>8</i>		
E				T	
environments:		O		<code>\theoremfile</code>	<i>3</i> , <u><i>3</i></u>
<code>makethm</code>	<i>1</i> , <u><i>20</i></u>	<code>\openout</code>	<i>9</i>	<code>\thetheorem</code>	<i>61</i>