quick-theorems — A quick way to typeset the most common theorem-like environments*

Ilario Bonacina†

Released 2020/09/28

Contents

1 Introduction

This package defines the most common theorem-like environments following the specifications from `amsthm.sty`. That is in *plain* style we define environments for

> Theorem, Lemma, Corollary, Proposition, Fact, Conjecture, Criterion, Assertion.

In *definition* style we define environments for

*This file describes version v1.0, last revised 2020/09/28.

†E-mail: ilariobonacina@gmail.com

Definition, Acknowledgements, Condition, Problem, Open Problem, Example, Exercise, Algorithm, Question, Axiom, Property, Assumption, Hypothesis.

Finally, in *remark* style we define environments for

Remark, Note, Notation, Claim, Summary, Case, Conclusion.

1.1 Features to add

- add styling for theorems with names, e.g. Gauss's Lemma
- add support for Restated theorems
- (maybe) define openproblem with the rest of the other commands, using `\zap@space`

1.2 Known Bugs

- This package make use of `\MakeUppercase` for the names of the environments, it breaks `\autoref` in `\sections` etc.
- The `case` environment is usually used inside proofs. Its numbering should always restart from 1 when used. At the moment this is achieved manually with `\setcounter{thecase}{0}` before the first usage of `case`.
- the `\qed` command conflicts with the definition from quick-math.

2 Usage

This package is loaded calling `\usepackage[<options>]{quick-theorems}` in the preamble of your document. It defines the environments:

```
   theorem          lemma         corollary    proposition      fact
 conjecture       criterion       assertion     definition    condition
   problem         example         exercise     algorithm     question
    axiom          property       assumption    hypothesis     remark
     note           notation          claim     summary case   conclusion
openproblem   acknowledgements
```

The environments are used as usual, e.g.

```
\begin{theorem}
...an amazing theorem goes here...
\end{theorem}
```

All the environments, except `acknowledgements`, are numbered (as usual). This package builds mainly upon `amsthm.sty` and `etoolbox.sty`.

2.1 Styling

Through `<options>` it is strightforward to apply some common styles on the environments. The possibilities are the following.

2.1.1 Fonts

`small-caps`
`slanted`
`italics`

By default the plain and definition style have the environment names in **boldface**. With the option `small-caps` they are rendered SMALLCAPS. With the option `slanted` (default option) the `proof` environment becomes "*Proof.*" and not "*Proof.*" and the body of the plain-like environments is *slanted* and not *italics*. The option `italics` reverts this behaviour.

2.1.2 Numbering

`swap-numbers`

With this option the numbering precedes the name of the environment. The default behaviour is the opposite.

`counter=shared`
`counter=not-shared`

This option tells if all the environments share the same counter (default) or each of them has its own counter.

`follow=global`
`follow=chapter`
`follow=section`
`follow=subsection`

This option tells if the counter(s) should never reset (global), if they should reset at each chapter, section (default), subsection.

2.1.3 QED styles

This package provides some basic styling for the QED symbol redefining the command `\qedsymbol`.

`qed=word` Q.E.D.
`qed=white` □
`qed=grey` ■ (default)
`qed=black` ■

2.2 Macros

`\NewTheoremStyled`

If the user needs to define some new theorem-like environment it can be done with `\NewTheoremStyled{newtheorem}{style}`. It takes 2 arguments, `#1` is the theorem name (the rendered name of the newtheorem is the same name but with the first letter uppercase), `#2` is the style (`definition`, `plain` or `remark`). The command `\shortproof{Very short proof}` can be used to isolate very short proofs, e.g. the previous example expands as ⟦Very short proof⟧.

`\shortproof`

3 Implementation

1 ⟨∗package⟩

We load the required packages.

```
2 \RequirePackage{xcolor} % for the option ``qed=grey''
```

```
\RequirePackage{amsthm}
\RequirePackage{amssymb}
\RequirePackage{stmaryrd} % for the boldface brackets used in \shortproof
\RequirePackage{etoolbox}
```

We create the options for the package

```
\DeclareOption{swap-numbers}{\swapnumbers}
\newif\ifSmallCaps
\DeclareOption{small-caps}{\SmallCapstrue}
\newif\ifSlanted
\DeclareOption{slanted}{\Slantedtrue\Italicsfalse}
\newif\ifItalics
\DeclareOption{italics}{\Italicstrue\Slantedfalse}
\newif\ifSharedCounter
\DeclareOption{counter=shared}{\SharedCountertrue}
\DeclareOption{counter=not-shared}{\SharedCounterfalse}
\newif\ifGlobalCounter
\DeclareOption{follow=global}{\GlobalCountertrue}
\DeclareOption{follow=chapter}{\def\@FollowCounter{chapter}}
\DeclareOption{follow=section}{\def\@FollowCounter{section}}
\DeclareOption{follow=subsection}{\def\@FollowCounter{subsection}}
\newif\ifQEDWhite
\DeclareOption{qed=white}{\QEDWhitetrue\QEDGreyfalse\QEDBlackfalse\QEDWordfalse}
\newif\ifQEDGrey
\DeclareOption{qed=grey}{\QEDGreytrue\QEDWhitefalse\QEDBlackfalse \QEDWordfalse}
\newif\ifQEDBlack
\DeclareOption{qed=black}{\QEDBlacktrue\QEDWhitefalse\QEDGreyfalse\QEDWordfalse}
\newif\ifQEDWord
\DeclareOption{qed=word}{\QEDWordtrue\QEDWhitefalse\QEDGreyfalse\QEDBlackfalse}
```

We process the default options and relax.

```
\ExecuteOptions{slanted, counter=shared, follow=section, qed=grey}
\ProcessOptions\relax
```

3.1 Environment Definitions

We use the `\newtheoremstyle` commands from `amstm.sty` to define the styles `definition`, `plain` and `remark`. We let the corresponding headfonts to be customizable via the commands `\@DefinitionHeadFont`, `\@PlainHeadFont` and `\@RemarkHeadFont`.

`\@NewTheorem` This is the main command used to define the environments.

```
\newcommand\@NewTheorem[1]{%
 \ifGlobalCounter
  \ifSharedCounter
    \expandafter\newtheorem{#1}[definition]{\MakeUppercase#1}
  \else
    \expandafter\newtheorem{#1}{\MakeUppercase#1}
  \fi
 \else
  \ifSharedCounter
```

```
    \expandafter\newtheorem{#1}[definition]{\MakeUppercase#1}
  \else
    \expandafter\newtheorem{#1}{\MakeUppercase#1}[\@FollowCounter]
  \fi
 \fi
}%
```

If needed the user can define extra theorem-like environments with

`\NewTheoremStyled` It takes 2 arguments, the first is the name of the theorem and the second is the style it should use (i.e. `definition` or `plain` or `remark`). The printed name of the theorem is *the same* as the command name but the first letter is uppercase.

```
\newcommand{\NewTheoremStyled}[2]{%
\theoremstyle{#2}
\@NewTheorem{#1}
}%
```

3.1.1 Definition-like Environments

`\@DefinitionHeadFont`

```
\newcommand{\@DefinitionHeadFont}{\ifSmallCaps\scshape\else\bfseries\fi}
```

`definition`

```
\newtheoremstyle{definition}
  {\topsep}   % ABOVESPACE
  {\topsep}   % BELOWSPACE
  {\normalfont}  % BODYFONT
  {0pt}       % INDENT (empty value is the same as 0pt)
  {\@DefinitionHeadFont} % HEADFONT
  {.}         % HEADPUNCT
  {5pt plus 1pt minus 1pt} % HEADSPACE
  {}          % CUSTOM-HEAD-SPEC
%
\theoremstyle{definition}
```

The environments `definition`, `acknowledgements` and `openproblem` are defined by hand below. We start with `definition`.

```
\ifGlobalCounter
\newtheorem{definition}{Definition}
\else
\newtheorem{definition}{Definition}[\@FollowCounter]
\fi
```

Then we define `acknowledgements`. (It is defined separately as it is never numbered.)

```
\newtheorem*{acknowledgements}{Acknowledgements}
```

Then we define `openproblem`. (It is defined separately as "Open Problem" are two words.)

```
\ifGlobalCounter
```

```
 \ifSharedCounter
   \newtheorem{openproblem}[definition]{Open Problem}
 \else
   \newtheorem{openproblem}{Open Problem}
 \fi
\else
 \ifSharedCounter
   \newtheorem{openproblem}[definition]{Open Problem}
 \else
   \newtheorem{openproblem}{Open Problem}[\@FollowCounter]
 \fi
\fi
```

We now define all the other environments using the definition style.

```
\forcsvlist{\listadd\@DefinitionList}{condition, problem, example, %
exercise, algorithm, question, axiom, property, assumption, hypothesis}
\forlistloop\@NewTheorem{\@DefinitionList}
```

3.1.2 Plain-like Environments

\@PlainHeadFont

```
\newcommand{\@PlainHeadFont}{\ifSmallCaps\scshape\else\bfseries\fi}
```

\@PlainBodyFont

```
\newcommand{\@PlainBodyFont}{\ifSlanted\slshape\fi\ifItalics\itshape\fi}
```

plain

```
\newtheoremstyle{plain}
  {\topsep}   % ABOVESPACE
  {\topsep}   % BELOWSPACE
  {\@PlainBodyFont}  % BODYFONT
  {0pt}       % INDENT (empty value is the same as 0pt)
  {\@PlainHeadFont} % HEADFONT
  {.}         % HEADPUNCT
  {5pt plus 1pt minus 1pt} % HEADSPACE
  {}          % CUSTOM-HEAD-SPEC

\theoremstyle{plain}
%
\forcsvlist{\listadd\@PlainList}{theorem, lemma, corollary, proposition, %
fact, conjecture, criterion, assertion}
\forlistloop\@NewTheorem{\@PlainList}
```

3.1.3 Remark-like Environments

\@RemarkHeadFont

```
\newcommand{\@RemarkHeadFont}{\ifSlanted\slshape\fi\ifItalics\itshape\fi}
```

remark

```
\newtheoremstyle{remark}
  {0.5\topsep}   % ABOVESPACE
  {0.5\topsep}   % BELOWSPACE
  {\normalfont}  % BODYFONT
  {0pt}       % INDENT (empty value is the same as 0pt)
  {\@RemarkHeadFont} % HEADFONT
  {.}         % HEADPUNCT
  {5pt plus 1pt minus 1pt} % HEADSPACE
  {}          % CUSTOM-HEAD-SPEC

\theoremstyle{remark}
%
\forcsvlist{\listadd\@RemarkList}{remark, note, notation, claim, %
summary, case, conclusion}
\forlistloop\@NewTheorem{\@RemarkList}
```

3.2 Proof macros

If the `slanted` option is active we redefine `\proofname`

```
\ifSlanted
 \let\oldproofname=\proofname
 \renewcommand{\proofname}{{\slshape\oldproofname}}
\fi
```

`\shortproof` We provide a command to isolate very short proofs.

```
\newcommand{\shortproof}[1]{$\llbracket$#1$\rrbracket$}
```

Finally, we process the options relative to QED and redefine `\qedsymbol` accordingly.

`\qedsymbol`

```
\renewcommand{\qedsymbol}{%
 \ifQEDWhite {\scriptsize$\square$} \fi
\ifQEDGrey {\scriptsize\textcolor{black!50!white}{$\blacksquare$}} \fi
 \ifQEDBlack {\scriptsize$\blacksquare$} \fi
 \ifQEDWord {\scshape q.e.d.} \fi
}%
%
\endinput
```

⟨/package⟩