

# Mathematics for L<sup>A</sup>T<sub>E</sub>X—personal notes

Óscar Pereira<sup>1</sup>

12th June 2020

<sup>1</sup>[https://, oscar@randomwalk.eu](https://oscar.randomwalk.eu).

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Theorems and other environments</b>               | <b>2</b> |
| 1.1      | An environment for the right math occasion . . . . . | 2        |
| 1.2      | More environments: theorem and friends . . . . .     | 2        |
| 1.3      | Alter egos . . . . .                                 | 4        |
| 1.4      | Paired delimiters . . . . .                          | 5        |
| <b>2</b> | <b>New and renew'd commands</b>                      | <b>7</b> |
| <b>3</b> | <b>Miscellaneous mathematics</b>                     | <b>8</b> |
| 3.1      | Sets . . . . .                                       | 8        |
| 3.2      | Matrices . . . . .                                   | 8        |
| 3.3      | Relation vs operator . . . . .                       | 8        |
| 3.4      | Number theory . . . . .                              | 9        |
|          | <b>References</b>                                    | <b>9</b> |

# Chapter 1

## Theorems and other environments

### 1.1 An environment for the right math occasion

So the default environment for equations is (duh) `equation`. If it is a long (multi line) equation, use `multline` to display it without alignment; and `split` (inside equation) to have alignment. When there are multiple equations, use `align` to have alignment, `gather` to have multiple equations (on distinct lines) \*without\* alignment, and `flalign` to have horizontal spreading (like table cells). `gathered` and `aligned` (used inside equation) act a la minipage, meaning they occupy the width of their arguments. This can be used to put things side by side.

When writing (say) demonstrations with side notes, use `align*`. The reason is that with `split`, you would have to use `\text` instead of `\tag` for the side notes, but the former is not right aligned (so the notes get placed right next to the math).

As an endnote, the `cases` environment is also used inside equation.

### 1.2 More environments: theorem and friends

For the environments typically required in mathematics texts. Note that:

- `amsmath` must be loaded before `eulervm`, or things bork!
- `\circledS` is also defined by `amssymb`, so `undef` it before loading said package.
- `amssymb` is required (at least) for proof environ black square.
- `amsthm` is required for proof environ (et al.).

```
\usepackage{amsmath}
\let\circledS\undefined
\usepackage{amssymb}
\usepackage{amsthm}
```

...

```
\newtheorem{theorem}{Theorem}[chapter]
\newtheorem{corollary}[theorem]{Corollary}
```

```

\newtheorem{lemma}[theorem]{Lemma}
% 'def' cannot be used as environ name.
\newtheorem{definition}[theorem]{Definition}
\theoremstyle{remark}
\newtheorem{remark}[theorem]{Remark}
\newtheorem{example}[theorem]{Example}

```

It is easy to add new environments of this sort for, e.g. assertions, propositions, etc. The proof environment (i.e. `\begin{proof}` is defined `amsthm`, as indicated above). Speaking of proofs, the traditional QED symbol is a non-filled square; to get a filled one, use:

```

\usepackage{amssymb}
\renewcommand\qedsymbol{${\blacksquare}$}

```

Here is how to make sure all enumerations in theorems and what not are similar:

```

\AtBeginEnvironment{theorem}{%
  \setlist[enumerate]{label={\upshape(\roman*)}}
}

```

To make emphasis in definitions et al. yield upright bold text, use the following code. `remark` environment is exempted because its default text is not italic—so ok to use italics for emphasis.

```

\usepackage{etoolbox}

```

...

```

\AtBeginEnvironment{theorem}{\renewcommand\em{\upshape\bfseries}}
\AtBeginEnvironment{definition}{\renewcommand\em{\upshape\bfseries}}
\AtBeginEnvironment{lemma}{\renewcommand\em{\upshape\bfseries}}
\AtBeginEnvironment{corollary}{\renewcommand\em{\upshape\bfseries}}
% \AtBeginEnvironment{remark}{\renewcommand\em{\upshape\bfseries}}

```

Make optional names for theorems et al bold (doesn't work for `remark` environ; see below), leaving the rest unchanged:<sup>1</sup>

```

\usepackage{amsthm}

```

...

```

\makeatletter
\def\th@plain{%
  \thm@notefont{}% optional name font same as heading font
  \itshape% body font
}
\makeatother

```

---

<sup>1</sup>I don't know *why* this works, but see <https://tex.stackexchange.com/questions/43966/how-to-make-the-optional-title-of-a-theorem-bold-with-amsthm>

To accomplish the same for remark:<sup>2</sup>

```
\makeatletter
\def\th@remark{%
  \thm@headfont{\bfseries}%
  \thm@notefont{}% optional name font same as heading font
  \normalfont % body font
  \thm@preskip\topsep \divide\thm@preskip\tw@
  \thm@postskip\thm@preskip
}
\makeatother
```

If there is no newline after the end of a theorem or similar environment, then the next paragraph should not be indented. This is accomplished thus:

```
\usepackage{etoolbox}
```

...

```
\makeatletter
\patchcmd{\@endtheorem}{\@endpfalse}{}{}{}
\patchcmd{\endproof}{\@endpfalse}{}{}{}
\makeatother
```

By default only the proof environment has an ending symbol, in this case, a black square. To make the same for remark and example environments (any other you might need), do:<sup>3</sup>

```
\newcommand{\addQEDstyle}[2]{%
  \AtBeginEnvironment{#1}{\pushQED{\qed}\renewcommand{\qedsymbol}{#2}}%
  \AtEndEnvironment{#1}{\popQED}}
\addQEDstyle{remark}{$\triangle$}
\addQEDstyle{example}{$\lozenge$}
```

### 1.3 Alter egos

In some circumstances, one might want to repeat a theorem, or a lemma, or a definition in more than one place (e.g. to state a theorem in the main text, while relegating the proof to appendix, but wanting to restate the theorem prior to the proof). This is one way to do it. First, add this to the preamble:<sup>4</sup>

```
\makeatletter
\newtheorem*{rep@theorem}{\rep@title}
```

<sup>2</sup>See <https://tex.stackexchange.com/questions/250035/transform-output-theoremstyleremark-from-italics-to-bold>. Cannot just copy first line, it works but spacing is wrong. So just copy full definition.

<sup>3</sup>See [https://tex.stackexchange.com/questions/16453/denoting-the-end-of-example-remark#comment1002519\\_32394](https://tex.stackexchange.com/questions/16453/denoting-the-end-of-example-remark#comment1002519_32394).

<sup>4</sup>See <https://tex.stackexchange.com/questions/422/how-do-i-repeat-a-theorem-number>.

```

\newcommand{\newrepththeorem}[2]{%
\newenvironment{rep#1}[1]{%
  \def\rep@title{#2 \ref*{##1}}%
  \begin{rep@theorem}}%
  {\end{rep@theorem}}}
\makeatother

```

Then, after the `\newtheorem` declarations, add:

```

\newrepththeorem{theorem}{Theorem}
\newrepththeorem{lemma}{Lemma}
\newrepththeorem{definition}{Definition}

```

Now, to restate say, a theorem, where the original label is `thm:awesome`, do:

```

\begin{repththeorem}{thm:awesome}
  Here is the theorem restated: ...
\end{repththeorem}

```

For lemmas, or definitions, replace with `replemma` or `repdefinition` respectively. Note that any patches to the theorem et al. environments—e.g. making emphasis yield bold upright text—may need to be redone for these new environments.

## 1.4 Paired delimiters

I use commands for basically all delimiters in math mode, including parenthesis and whatnot, because of scaling (size) considerations. Exemplifying makes it simpler:

```

\usepackage{mathtools}

...

\DeclarePairedDelimiter\ceil{\lceil}{\rceil}
\DeclarePairedDelimiter\floor{\lfloor}{\rfloor}
\DeclarePairedDelimiter\parens{\lparen}{\rparen}
\DeclarePairedDelimiter\sparens{\lbracket}{\rbracket}
\DeclarePairedDelimiter\cparens{\lcurly}{\rcurly}
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
\DeclarePairedDelimiter\norm{\lVert}{\rVert}

```

With these settings, using for example `\abs*{x}` yields side bars (absolute value) that grow bigger or smaller as required. Furthermore, with the fixed size (non-starred) version, one can actually set the desired size, e.g. `\abs[\Big]{x}`. However, I usually want the automagically size-adjusting version, so I reverse the starred and non-starred roles like this:<sup>5</sup>

```

\usepackage{etoolbox}

```

---

<sup>5</sup>See <https://tex.stackexchange.com/questions/94410/easily-change-behavior-of-declarepaireddelimiter>.

...

```
\newcommand\swapifbranches[3]{#1{#3}{#2}}
\makeatletter
\MHInternalSyntaxOn
\patchcmd{\DeclarePairedDelimiter}{\@ifstar}{\swapifbranches\@ifstar}{}{}
\MHInternalSyntaxOff
\makeatother
```

So now,  $\abs{x}$  is the automatically resizing version, but to specify a size we must now do  $\abs*[\Big]{x}$ .

## Chapter 2

### New and renew'd commands

To get an equals sign, with “def” written above in tiny script, do:

```
\newcommand\eqdef{\mathrel{\overset{\makebox[0pt]{%  
  \mbox{\normalfont\tiny\sffamily def}}}{=}}}
```

To get the set vertical bar, when you need to use it in places other than sets, do:

```
\newcommand\st{\;\vert\;}
```

For function declaration, the arguments are the letter that names the function, object set, range set:

```
\newcommand\funcdecl[3]{#1\colon #2 \rightarrow #3}
```

For function definition, the arguments are independent variable, function expression (or domain and co-domain):

```
\newcommand\funcdef[2]{#1 \mapsto #2}
```



# Chapter 3

## Miscellaneous mathematics

### 3.1 Sets

For set conditional definition, use the braket package. `\set` is for regular sized-brackets and middle bar, and `\Set` for expanding-as-required brackets and bar.

```
\usepackage{braket}
```

...

```
⋄\set{x \in A | x \notin B}⋄
```

### 3.2 Matrices

Apparently to use `\bm` in a matrix, you need `amsmath`...

And speaking matrices, here is how to use the `array` environment with a caption!

```
\begin{center}
⋄\begin{array}{c|c|c}
  \varphi & \psi & \varphi \rightarrow \psi \\
\hline
  0 & 0 & 1 \\
  0 & 1 & 1 \\
  1 & 0 & 0 \\
  1 & 1 & 1
\end{array}⋄
\captionof{table}{\label{arr:implication_truth_table} Truth table
for the implication connective.}
\end{center}
```

For the record, the `\captionof` command, which also comes from the `caption` package, stands for “caption out of floats” (recall that captions are usually used with floats only).

### 3.3 Relation vs operator

Short and simple:

```
% \mid -> binary relation (same as \mvert)
% \vert -> binary operator (synonym for |, not to use as delim)
```

### 3.4 Number theory

Divides operator: use `\divides` from package `mathabx`. This package redefines a lot of symbols, so it must be loaded after all other math packages!!