LaThalesians: The Thalesians' LaTeX library

2016.09.24

Contents

*Current maintainer, contactable on paul@thalesians.com

1 Overview

The LaThalesians library comprises a heterogeneous collection of LaTeX packages, which facilitate the typesetting of the Thalesians' work in mathematics, computer science, and finance. It was originally developed by Paul Bilokon to support his academic and professional work and the library's scope still reflects some of his personal biases, *viz.*:

- mathematical finance,
- econometrics,
- programming,
- scientific computing,
- algorithms,
- statistics,
- stochastic analysis,
- probability theory,
- domain theory,
- computability theory.

It is hoped that, as more people get involved in the development and maintenance of this library, its scope will become more balanced and will more faithfully reflect the diverse activities of the Thalesians.

Rather than being structured as a single package, LaThalesians is a suite of packages, each package name starting with `lathalesians-`. Thus the modules may be used individually, depending on the user's specific needs. *Modularity* is one of the design principles that guided us in this library's development.

Another one is *simplicity*: tasks that occur often in our research and development work should be made easy. The syntax should be straightforward and easy to remember. LaTeX commands that we type in often should be brief.

But not too brief: they should still be unambiguous and easy to remember. *Readability* and *clarity* are also important to us. Finding the right balance between simplicity and readability is an art more than a science.

We are believers in *domain-specific languages*. Therefore we often define LaTeXcommands to represent the concepts from a particular research area rather than typesetting instructions. This is done at the cost of introducing more words into our language. We believe that these are the very words that we need. Let's express what things are, rather than what they should look like.

Finally, we believe that *truth* and *beauty* should go hand in hand. Both should be present in the content. Form should do justice to the content's truth and present it in a way that is beautiful. We don't claim that we have achieved this in LaThalesians. However, this is indeed our striving, our intention. We will be very much obliged for any recommendations on how to make the library's output more aesthetically pleasing.

We, the Thalesians, would be very much obliged to you for your contributions to this library. It is far from perfect now. In many ways it is quite defective. Please help us make it both useful and beautiful.

2 Usage

To use each individual package, simply include the appropriate `\usepackage` in the preamble of your document, before `\begin{document}`, for example:

```
\usepackage{lathalesians-commons}
```

Some of the packages take parameters as a comma-separated key-value pairs list. In this case, the syntax is as follows:

```
\usepackage[<key=value list>]{<package name>}
```

Here is an example of such usage:

```
\usepackage[usecolour=true,defncolour={0,.5,0}]{lathalesians-commons}
```

3 Packages

The bulk of the library consists in LaTeX packages, each named `lathalesians-xxx.sty`. In this section we shall go through each individual package, starting with the most generally useful.

3.1 `lathalesians-commons`: general-purpose utilities for LaTeX

This package defines some general-purpose commands and environments. For example, it defines the `\defn` command, which highlights definitions and add the defined terms to the `general` index. It also contains commands for basic typesetting, such as underlining, overlining, etc. Some of these commands are shorter aliases of existing LaTeX commands.

3.1.1 Usage

```
\usepackage[<key=value list>]{lathalesians-commons}
```

The following options are allowed in the comma-separated key=value list:

Option	Possible values	Default value	Description
usecolour	true, false	true	use colour, e.g. in definitions
defncolour	{r,g,b}	{.5,0,0}	colour for terms being defined
anauthorcolour	{r,g,b}	{0,0,.5}	colour for author names
nbcolour	{r,g,b}	{1,0,0}	colour for emphasised text

Table 1: Options supported by lathalesians-commons

Example:

```
\usepackage[usecolour=true,defncolour={0,.5,0}]{lathalesians-commons}
```

3.1.2 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
\LaThalesians	\LaThalesians	LaThalesians	the name of this library
\ol	\ol{Thales}, $\ol{123}$	$\overline{\text{Thales}}$, $\overline{123}$	overline
\ul	\ul{Thales}, $\ul{123}$	$\underline{\text{Thales}}$, $\underline{123}$	underline
\defn	\defn{G_{δ} set}	G_δ **set**	definition (see details below)
\anauthor	\anauthor{Georg~Cantor}	**Georg Cantor**	reference to an author (see details below)
\generalindexentry	see below	see below	general index entry
\authorsindexentry	see below	see below	authors index entry
\nb	\nb{Important!}	*Important!*	emphasised text
\atitle	\atitle{Functional Analysis}	*Functional Analysis*	title of a work
\commentary	\commentary{Some comments.}	*Some comments.*	comments
\species	\species{Homo sapiens}	*Homo sapiens*	biological species
\french	\french{t{\^e}te-{\`a}-t{\^e}te}	*tête-à-tête*	inclusions in French
\german	\german{Grundbegriffe}	*Grundbegriffe*	inclusions in German
\latin	\latin{inter alia}	*inter alia*	inclusions in Latin
\shortquote	see below	see below	environment for short quotes
\longquote	see below	see below	environment for long quotes
\eg	\eg	e.g.	
\etc	\etc	etc.	
\ie	\ie	i.e.	
\interalia	\interalia	*inter alia*	
\viz	\viz	*viz.*	

Table 2: Summary of commands and environments defined in lathalesians-commons

3.1.3 Definitions: \defn and \generalindexentry

\defn formats a defined term and optionally adds it to the index called general. The syntax

```
\defn{<defined term>}
```

will typeset the <defined term> as **defined term** and add it to the general index. To avoid adding the defined term to that index, use the syntax

```
\defn{<defined term>}[]
```

Instead of adding <defined term> to the general index, one may add <index entry> using the following syntax:

```
\defn{<defined term>}[<index entry>]
```

For example,

```
\defn{Cartesian product}[product!Cartesian]
```

will typeset **Cartesian product** and add "product, Cartesian" to the general index. One may add up to four index entries in a single \defn command, for example:

```
\defn{Cartesian product}[product!Cartesian][Cartesian product|see{product, Cartesian}]
```

The above is equivalent to

```
\defn{Cartesian product}[]\generalindexentry{product!Cartesian}\generalindexentry{Cartesian
    ↪ product|see{product, Cartesian}}
```

The command

```
\generalindexentry{<index entry>}
```

will add <index entry> to the index called general without adding any text to the body of the document where this command occurs.

3.1.4 Authors: \anauthor and \authorsindexentry

\anauthor formats the name of an author and optionally adds it to the index called authors. The syntax

```
\anauthor{<name of author>}
```

will typeset the <name of author> as **name of author** and add it to the authors index. To avoid adding the author's name to that index, use the syntax

```
\anauthor{<name of author>}[]
```

Instead of adding <name of author> to the authors index, one may add <index entry> using the following syntax:

```
\anauthor{<name of author>}[<index entry>]
```

For example,

```
\anauthor{Georg~Cantor}[Cantor, Georg]
```

will typeset **Georg Cantor** and add "Cantor, Georg" to the authors index. One may add up to four index entries in a single \anauthor command, for example:

```
\anauthor{Georg~Cantor}[Cantor, Georg Ferdinand Ludwig Philipp][Cantor, Georg|see{Cantor, Georg
    ↪ Ferdinand Ludwig Philipp}]
```

The above is equivalent to

```
\anauthor{Georg~Cantor}[]\authorsindexentry{Cantor, Georg Ferdinand Ludwig Philipp}\
    ↪ authorsindexentry{Cantor, Georg|see{Cantor, Georg Ferdinand Ludwig Philipp}}
```

The command

```
\authorsindexentry{<index entry>}
```

will add <index entry> to the index called authors without adding any text to the body of the document where this command occurs.

3.1.5 Building indices

Here is an example of how you can include the indices in your document. Notice that, for the indices to work, you need the package `multind`.

```
\usepackage{multind}

\makeindex{general}
\makeindex{authors}

\begin{document}

...

\addcontentsline{toc}{section}{References}
\bibliographystyle{alpha}
\bibliography{lathalesians-bibliography}

\printindex{general}{Subject index}
\printindex{authors}{Index of authors}

\end{document}
```

To build the two indices, first LaTeXify the document, then run

```
makeindex general
makeindex authors
```

Then LaTeXify the document again. You can see an example of the result at the end of this document. For more information on this process, see `https://en.wikibooks.org/wiki/LaTeX/Indexing`

3.1.6 Quotes

The environment `longquote` is useful for typesetting long quotes. The following LaTeX code

```
\begin{longquote}
A vulgar Mechanik can practice what he has been taught or seen done, but if he is in an error he
    ↪ knows not how to find it out and correct it, and if you put him out of his road, he is at
    ↪ a stand; whereas he that is able to reason numbly and judiciously about figure, force and
    ↪ motion, is never at rest til he gets over every rub.
\end{longquote}
```

gives rise to

> A vulgar Mechanik can practice what he has been taught or seen done, but if he is in an error he knows not how to find it out and correct it, and if you put him out of his road, he is at a stand; whereas he that is able to reason numbly and judiciously about figure, force and motion, is never at rest til he gets over every rub.

The environment `shortquote` is currently defined as an alias to `quote`. The following LaTeX code

```
\begin{shortquote}
A vulgar Mechanik can practice what he has been taught or seen done, but if he is in an error he
    ↪ knows not how to find it out and correct it, and if you put him out of his road, he is at
    ↪ a stand; whereas he that is able to reason numbly and judiciously about figure, force and
    ↪ motion, is never at rest til he gets over every rub.
\end{shortquote}
```

gives rise to

> A vulgar Mechanik can practice what he has been taught or seen done, but if he is in an error he knows not how to find it out and correct it, and if you put him out of his road, he is at a stand; whereas he that is able to reason numbly and judiciously about figure, force and motion, is never at rest til he gets over every rub.

3.2 `lathalesians-font`: font setup

This package has no key-value options. It simply configures the fonts. At Thalesians, we prefer to use the Palatino typeface family. According to Wikipedia,

> Palatino is the name of a large typeface family that began as an old style serif typeface designed by Hermann Zapf initially released in 1948 by the Linotype foundry. In 41999 Zapf revised Palatino for Linotype and Microsoft called Palatino Linotype. The revised family incorporated extended Latin, Greek, Cyrillic character sets.
>
> Under the collaboration of Zapf and Akira Kobayashi the Palatino typeface family was expanded. Linotype released the Palatino nova, Palatino Sans, and Palatino Sans Informal families, expanding the Palatino typeface families to include humanist sans-serif typefaces. Palatino nova was released in 2005, while the others were released in 2006.
>
> Named after 16th century Italian master of calligraphy Giambattista Palatino, Palatino is based on the humanist fonts of the Italian Renaissance, which mirror the letters formed by a broad nib pen; this gives a calligraphic grace. But where the Renaissance faces tend to use smaller letters with longer vertical lines (ascenders and descenders) with lighter strokes, Palatino has larger proportions, and is considered much easier to read.
>
> It remains one of the most widely-used (and copied) text typefaces, has been adapted to virtually every type of technology, and is one of the ten most used serif typefaces. It is one of several typefaces by Zapf, each showing influence of the Italian Renaissance letter forms. The group includes Palatine, Sistina, Michaelangelo tiling, and Aldus, which takes inspiration from printing types cut by Francesco Griffo c. 1495 in the print shop of Aldus Manutius.

This document is an example of what the end result will look like.

3.3 `lathalesians-formatting`: formatting

This package includes the basic machinery for formatting margins, paragraphs and indentation.

3.3.1 Usage

```
\usepackage[<key=value list>]{lathalesians-formatting}
```

The following options are allowed in the comma-separated key=value list:

Option	Possible values	Default value	Description
`usegeometry`	`true`, `false`	`true`	use `geometry` package to set up margins
`geometry`	as specified by `geometry` package	see below	`geometry` arguments

Table 3: Options supported by `lathalesians-commons`

If `usegeometry=true` (the default case), then the `geometry` option will be passed on to the `geometry` package. The default value of this option is

```
left=1in,right=1in,top=1in,bottom=1in
```

Here is an example of the script's usage with the arguments supplied explicitly[1]:

```
\usepackage[usegeometry=true,geometry={left=1in,right=1in,top=2in,bottom=2in}]{lathalesians-
    ↪ formatting}
```

In addition to optionally setting the margins, the package will configure the paragraphs and indentation as follows:

```
% Medium space before a \par:
\setlength{\parskip}{\medskipamount}
% Don't indent first lines of paragraphs:
\setlength{\parindent}{0pt}
```

3.4 `lathalesians-theorems`: theorems

This package sets the naming and numbering conventions for propositions, theorems, corollaries, etc., as used in the Thalesians' works. It is fairly self-explanatory, so we shall illustrate its use with a single example:

```
\begin{hypothesis}
This is a hypothesis.
\end{hypothesis}

\begin{proposition}
This is a proposition.
\begin{proof}
And its proof.
\end{proof}
\end{proposition}

\begin{proposition}
This is another proposition.
\end{proposition}

\begin{corollary}
This is a corollary.
\end{corollary}

\begin{theorem}
This is a theorem.
\end{theorem}

\begin{lemma}[some named lemma]
This is a lemma.
\end{lemma}

\begin{definition}
This is a \defn{definition}.
\end{definition}

\begin{remark}
This is a remark.
\end{remark}

\begin{example}
This is an example.
\end{example}
```

This gives rise to the following:

[1] We don't actually need to specify `usegeometry=true`, as this is the default setting anyway.

Hypothesis 3.1. *This is a hypothesis.*

Proposition 3.2. *This is a proposition.*

Proof. And its proof. ∎

Proposition 3.3. *This is another proposition.*

Corollary 3.4. *This is a corollary.*

Theorem 3.5. *This is a theorem.*

Lemma 3.6 (some named lemma)**.** *This is a lemma.*

Definition 3.7. This is a **definition**.

Remark 3.8. This is a remark.

Example 3.9. This is an example.

3.5 `lathalesians-code`: programming

This package enables us to include in our documents portions of computer code and programs' output. Many of the commands defined in this package are wrappers around those defined in the package `listings`, developed by **Carsten Heinz**, **Brooks Moses** and its current maintainer, **Jobst Hoffmann**.

3.5.1 Inlining code: `\code`

Use `\code` to inline code in your LaTeX prose. The syntax is as follows:

```
\code[<key=value list>]<character><source code><same character>
```

While we say here `<same character>`, the command also understands bracket pairs. Thus `\code{3+5}`, `\code!3+5!` and `\code|3+5|` will lead to the same result: `3+5`.

We can use `\code` to include small portions of code in our prose. For example, `\code{c := a + b}` will give rise to `c := a + b`. We can also use it to refer to variables, methods, and classes. To talk about a variable named `foo`, use `\code{foo}` or (for example) `\code!foo!`, which will give the same result. Similarly, you can use `\code{someMethodName}` or `\code!someMethodName!` to typeset `someMethodName`.

`\code` is a wrapper around `\lstinline` of the package `listings`. The comma-separated `[<key=value list>]` is passed on to `\lstinline`. For example, `\code{print("sum", 3+5)}` will result in `print("sum", 3+5)`, which looks different from the result of `\code[language=python]{print("sum", 3+5)}` — **`print`**`("sum", 3+5)`.

3.5.2 Referring to programs, files, and directories

The commands `\program` and `\file` have a similar syntax to `\code`. Use `\program` to refer to programs and their modules. For example, `\program{lathalesians-code}` will appear as `lathalesians-code`. Use `\file` to refer to files and directories, for example `\file{C:\Program Files\MiKTeX 2.9}` will be rendered as `C:\Program Files\MiKTeX 2.9`.

3.5.3 Code snippets

When we need to include entire code snippets in our programs, rather than small expressions, as we did with `\code`, we use the environment `Snippet` or its sibling, `snippet`. Both of these commands are `listings` environments. The only difference between the two is that `Snippet` includes a caption and increments the listings counter. All `Snippets` are numbers, whereas `snippets` are not. The syntax of `snippet` is identical to that of `Snippet`:

```
\begin{Snippet}[<key=value list>]{<style>}
...
\end{Snippet}
```

The <key=value list> is passed on to `lstset`. The <style> is case-sensitive and is one of: `BUGS`, `CPP`, `Java`, `LaTeX`, `Python`, `q`, `plain` and `output`. The styles `plain` and `output` are special: `plain` does not highlight any syntax by default, so in some sense we get a fancy verbatim environment; `output` is used for typesetting program output.

The following listing was created using the LaTeX code

```
\begin{Snippet}[caption=An example from \protect\url{https://wiki.python.org/moin/SimplePrograms
    ↪ }: 8-Queens Problem (define your own exceptions)]{Python}
BOARD_SIZE = 8

class BailOut(Exception):
    pass

...
\end{Snippet}
```

Notice that we used the `Snippet` environment, beginning with the uppercase letter, we ensure that a full numbered listing is created:

Listing 1 An example from `https://wiki.python.org/moin/SimplePrograms`: 8-Queens Problem (define your own exceptions)

```
BOARD_SIZE = 8

class BailOut(Exception):
    pass

def validate(queens):
    left = right = col = queens[-1]
    for r in reversed(queens[:-1]):
        left, right = left-1, right+1
        if r in (left, col, right):
            raise BailOut

def add_queen(queens):
    for i in range(BOARD_SIZE):
        test_queens = queens + [i]
        try:
            validate(test_queens)
            if len(test_queens) == BOARD_SIZE:
                return test_queens
            else:
                return add_queen(test_queens)
        except BailOut:
            pass
    raise BailOut

queens = add_queen([])
print queens
print "\n".join(". "*q + "Q " + ". "*(BOARD_SIZE-q-1) for q in queens)
```

If instead we used the `snippet` environment, beginning with the lowercase letter, there would be a border but no caption, and the listing count would not be incremented. Of course, in this case it would be pointless to provide additional `caption` parameter to the `listings` package, so instead we would use

```
\begin{snippet}{Python}
BOARD_SIZE = 8

class BailOut(Exception):
    pass

...
\end{snippet}
```

to produce

```
BOARD_SIZE = 8

class BailOut(Exception):
    pass
...
```

If instead of including the Python source code in our LaTeX source we prefer to load it from a file, we can use `Source` or its sibling, `source` to obtain the same effect as, respectively, `Snippet` and `snippet` above. Thus

```
\source[caption=Another example from \protect\url{https://wiki.python.org/moin/SimplePrograms}:
    ↪ Prime numbers sieve with fancy generators]{Python}{examples/eg.py}
```

will give us:

```
import itertools

def iter_primes():
     # an iterator of all numbers between 2 and +infinity
     numbers = itertools.count(2)

     # generate primes forever
     while True:
         # get the first number from the iterator (always a prime)
         prime = numbers.next()
         yield prime

         # this code iteratively builds up a chain of
         # filters...slightly tricky, but ponder it a bit
         numbers = itertools.ifilter(prime.__rmod__, numbers)

for p in iter_primes():
    if p > 1000:
        break
    print(p)
```

To include the output of the above program, we can use

```
\Source[caption=Some output]{output}{examples/output.txt}
```

produces

Listing 2 Some output

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,
↪ 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191,
↪ 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283,
↪ 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401,
↪ 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509,
↪ 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631,
↪ 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751,
↪ 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877,
↪ 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997
```

3.6 `lathalesians-analysis`: analysis

This package introduces some commands and environments to simplify the typesetting of works in basic mathematical analysis. Of course, much of mathematics relies on analysis, so this package is useful in many different kinds of mathematical work. We regard set theory as an integral part of mathematical analysis, so much of the package deals with sets.

3.6.1 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
`\defeq`	`$A \defeq \set{1,2,3}$`	$A := \{1,2,3\}$	is equal by def. to (left to right)
`\eqdef`	`$\set{1,2,3} \eqdef A$`	$\{1,2,3\} =: A$	is equal by def. to (right to left)
`\idwith`	`$a_i \idwith b_i$`	$a_i \leftrightarrow b_i$	is identified with
`\tendsto`	`$x \tendsto \infty$`	$x \to \infty$	tends to
`\Implies`	`$A \Implies B$`	$A \Rightarrow B$	implies, only if
`\OnlyIf`	`$A \OnlyIf B$`	$A \Rightarrow B$	ditto
`\ImpliedBy`	`$A \ImpliedBy B$`	$A \Leftarrow B$	implied by, if
`\If`	`$A \If B$`	$A \Leftarrow B$	ditto
`\Iff`	`$A \Iff B$`	$A \Leftrightarrow B$	if and only if
`\N`	`$\N$`	$\mathbb{N}$	natural numbers
`\Nz`	`$\Nz$`	$\mathbb{N}^0$	ditto, explicitly incl. 0
`\Nnz`	`$\Nnz$`	$\mathbb{N}^*$	nonzero natural numbers
`\Z`	`$\Z$`	$\mathbb{Z}$	integers
`\Zp`	`$\Zp$`	$\mathbb{Z}^+$	positive integers
`\Znn`	`$\Znn$`	$\mathbb{Z}_0^+$	nonnegative integers
`\Zn`	`$\Zn$`	$\mathbb{Z}^-$	negative integers
`\Znz`	`$\Znz$`	$\mathbb{Z}^*$	nonzero integers
`\Q`	`$\Q$`	$\mathbb{Q}$	rationals
`\Qp`	`$\Qp$`	$\mathbb{Q}^+$	positive rationals
`\Qnn`	`$\Qnn$`	$\mathbb{Q}_0^+$	nonnegative rationals
`\Qn`	`$\Qn$`	$\mathbb{Q}^-$	negative rationals
`\Qnz`	`$\Qnz$`	$\mathbb{Q}^*$	nonzero rationals
`\R`	`$\R$`	$\mathbb{R}$	reals
`\Rp`	`$\Rp$`	$\mathbb{R}^+$	positive reals
`\Rnn`	`$\Rnn$`	$\mathbb{R}_0^+$	nonnegative reals
`\Rn`	`$\Rn$`	$\mathbb{R}^-$	negative reals
`\Rnz`	`$\Rnz$`	$\mathbb{R}^*$	nonzero reals
`\C`	`$\C$`	$\mathbb{C}$	complex numbers
`\Cnz`	`$\Cnz$`	$\mathbb{C}^*$	nonzero complex numbers
`\Collection`	`$\Collection{F}$`	$\mathcal{F}$	name of a collection
`\Set`	`$\Set{A}$`	A	name of a set
`\collection`	`$\collection{A,B,\ldots}$`	$\{A, B, \ldots\}$	definition of a collection
	`$\collection{O\in\Collection{T}}[O\text{ clopen}]$`	$\{O \in \mathcal{T} \mid O \text{ clopen}\}$	
`\set`	`$\set{1,2,3}$`	$\{1,2,3\}$	definition of a set
	`$\set{x\in\N}[x\text{ even}]$`	$\{x \in \mathbb{N} \mid x \text{ even}\}$	
`\seqel`	`$\seqel{a_k}$`	(a_k)	name of element of a sequence
	`$\seqel{a_k}[k\in\N]$`	$(a_k)_{k\in\mathbb{N}}$	
	`$\seqel{a_k}[k=1][\infty]$`	$(a_k)_{k=1}^{\infty}$	
`\sequence`	`$\sequence{2, 4, 6, \ldots}$`	$(2,4,6,\ldots)$	definition of a sequence
`\tuple`	`$\tuple{2, 4, 6}$`	$(2,4,6)$	definition of a tuple
`\st`	`$x\in\N \st x\text{ even}$`	$x \in \mathbb{N} \mid x$ even	such that
`\sset`	`$\Z \sset \R$`	$\mathbb{Z} \subseteq \mathbb{R}$	subset

\Sset	$\R \Sset \Z$	$\mathbb{R} \supseteq \mathbb{Z}$	superset
\psset	$\Z \psset \R$	$\mathbb{Z} \subsetneq \mathbb{R}$	proper subset
\pSset	$\R \pSset \Z$	$\mathbb{R} \supsetneq \mathbb{Z}$	proper superset
\setdiff	$A \setdiff B$	$A \setminus B$	set difference
\symmdiff	$A \symmdiff B$	$A \triangle B$	symmetric difference
\setcomplement	$\setcomplement{A}$	A^{c}	set complement
\closure	$\closure{A}$	$\overline{A}$	closure
\interior	$\interior{A}$	A°	interior
\embed	$X \embed Y$	$X \hookrightarrow Y$	topological embedding
\embedsinto	$X \embedsinto Y$	$X \hookrightarrow Y$	ditto
\functype	$\functype{\C}{\Rnn}$	$\mathbb{C} \to \mathbb{R}_0^+$	function type
	$\functype[f]{\C}{\Rnn}$	$f : \mathbb{C} \to \mathbb{R}_0^+$	
\funcdefn	$\funcdefn{x}{x^2}$	$x \mapsto x^2$	function definition
	$\funcdefn[f]{x}{x^2}$	$f : x \mapsto x^2$	
\indfunc	$\indfunc{A}(x)$	$\mathbf{1}_A(x)$	indicator function of a set
\zerofunc	$\zerofunc(x)$	$\mathbf{0}(x)$	zero function
\argmin	$\argmin_{x\in A} f(x)$	$\arg\min_{x \in A} f(x)$	arguments of the minima
\argmax	$\argmax_{x\in A} f(x)$	$\arg\max_{x \in A} f(x)$	arguments of the maxima
\norm	$\norm{x}_2$	$\|x\|_2$	norm

Table 4: Summary of commands and environments defined in `lathalesians-analysis`

3.7 `lathalesians-linear`: linear algebra

3.7.1 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
\transpose	$\transpose{v}$	v^{T}	transpose
\tr	$\tr{A}$	$\operatorname{tr} A$	trace
\rank	$\rank{A}$	$\operatorname{rank} A$	rank
\nul	$\nul{A}$	$\operatorname{nul} A$	nullity
\V	$\V{v}$	$\mathbf{v}$	name of a vector
\M	$\M{A}$	$\mathbf{A}$	name of a matrix

Table 5: Summary of commands and environments defined in `lathalesians-linear`

To typeset matrix and vector definitions, we prefer to use the `pmatrix` environment from `amsmath`[2]:

```
\begin{gather*}
\V{u} \defeq \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}, \\
\V{v} \defeq \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \\
\V{v} = \transpose{\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}}, \\
\M{M} \defeq \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}.
\end{gather*}
```

This gives rise to:

$$\begin{aligned}
\mathbf{u} &:= \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}, \\
\mathbf{v} &:= \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \\
\mathbf{v} &= \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}^{\mathsf{T}}, \\
\mathbf{M} &:= \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}.
\end{aligned}$$

[2]For more information on typesetting matrices see `http://tex.stackexchange.com/questions/26434/where-is-the-matrix-command`

3.8 `lathalesians-probability`: probability theory and statistics

This package defines some commands and environments to simplify the typesetting of works in probability and statistics.

By extension, this package is useful for works in stochastic analysis, stochastic filtering and control, econometrics, and mathematical finance.

3.8.1 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
`\SampleSpace`	`$\SampleSpace$`	Ω	sample space
`\Measures`	`$\Measures[X]$`	$\mathcal{M}(X)$	measure space
	`$\Measures$`	$\mathcal{M}$	
`\ProbMeasures`	`$\ProbMeasures[X]$`	$\mathcal{P}(X)$	probability measure space
	`$\ProbMeasures$`	$\mathcal{P}$	
`\Borel`	`$\Borel[\R]$`	$\mathcal{B}(\mathbb{R})$	Borel σ-algebra
	`$\Borel$`	$\mathcal{B}$	
`\Time`	`$\Time$`	$\mathbb{T}$	time of a process
`\State`	`$\State$`	$\mathbb{S}$	state of a process
`\aew`	`\aew`	a.e.	almost everywhere
`\as`	`\as`	a.s.	almost surely
`\convlaw`	`$\convlaw$`	$\xrightarrow{\sim}$	converges in law
`\convprob`	`$\convprob$`	$\xrightarrow{p}$	converges in probability
`\convas`	`$\convas$`	$\xrightarrow{\text{a.s.}}$	converges almost surely
`\convweak`	`$\convweak$`	$\Rightarrow$	converges weakly
`\support`	`$\support \mu$`	$\operatorname{supp}\mu$	support of a measure
`\Prob`	`$\Prob$`	$\mathbb{P}$	probability
	`$\Prob[A]$`	$\mathbb{P}[A]$	
`\ProbMeasure`	`$\ProbMeasure{Q}$`	$\mathbb{Q}$	probability measure
	`$\ProbMeasure{Q}[A]$`	$\mathbb{Q}[A]$	
`\given`	`$\Prob[A \given B]$`	$\mathbb{P}[A \mid B]$	given (conditioned on)
`\E`	`$\E[X]$`	$\mathbb{E}[X]$	expectation operator
	`$\E[X][\ProbMeasure{Q}]$`	$\mathbb{E}_{\mathrm{Q}}[X]$	
	`$\E$`	$\mathbb{E}$	
`\Var`	`$\Var[X]$`	$\mathsf{Var}[X]$	variance
	`$\Var[X][\ProbMeasure{Q}]$`	$\mathsf{Var}_{\mathrm{Q}}[X]$	
	`$\Var$`	Var	
`\Cov`	`$\Cov[X,Y]$`	$\mathsf{Cov}[X,Y]$	covariance
	`$\Cov[X,Y][\ProbMeasure{Q}]$`	$\mathsf{Cov}_{\mathrm{Q}}[X,Y]$	
	`$\Cov$`	Cov	
`\Cor`	`$\Cor[X,Y]$`	$\mathsf{Cor}[X,Y]$	correlation
	`$\Cor[X,Y][\ProbMeasure{Q}]$`	$\mathsf{Cor}_{\mathrm{Q}}[X,Y]$	
	`$\Cor$`	Cor	
`\Normal`	`$\Normal{\mu}{\sigma^2}$`	$\mathcal{N}(\mu,\sigma^2)$	normal distribution
`\WS`	`$\WS{\mu}{\sigma^2}$`	$\mathcal{WS}(\mu,\sigma^2)$	wide sense distribution
`\distributed`	`$X \distributed \Normal{\mu}{\sigma^2}$`	$X \sim \mathcal{N}(\mu,\sigma^2)$	distributed as
`\normalpdf`	`$\normalpdf{\mu}{\sigma^2}$`	$\varphi(\mu,\sigma^2)$	normal p.d.f
	`$\normalpdf[x]{\mu}{\sigma^2}$`	$\varphi(x;\mu,\sigma^2)$	

Table 6: Summary of commands and environments defined in `lathalesians-probability`

3.9 `lathalesians-categories`: category theory

This package defines symbols for some common category names. The naming and typesetting conventions are those defined in [GHK+03]. The descriptions of the categories can be found in the appendix of [GHK+03].

3.9.1 Summary of commands and environments

Code	LaTeXed result
`$\CatAL$`	$\mathbf{AL}$
`$\CatALop$`	$\mathbf{AL}^{\mathrm{op}}$
`$\CatALG$`	$\mathbf{ALG}$
`$\CatALGDOM$`	$\mathbf{ALGDOM}$
`$\CatALGDOMD$`	$\mathbf{ALGDOM}_D$
`$\CatALGDOMG$`	$\mathbf{ALGDOM}_G$
`$\CatArL$`	$\mathbf{ArL}$
`$\CatArLop$`	$\mathbf{ArL}^{\mathrm{op}}$
`$\CatBCSOB$`	$\mathbf{BCSOB}$
`$\CatBF$`	$\mathbf{BF}$
`$\CatCCSOB$`	$\mathbf{CCSOB}$
`$\CatCL$`	$\mathbf{CL}$
`$\CatCLd$`	$\mathbf{CL}_d$
`$\CatCLm$`	$\mathbf{CL}_m$
`$\CatCLop$`	$\mathbf{CL}^{\mathrm{op}}$
`$\CatCONT$`	$\mathbf{CONT}$
`$\CatCPOSP$`	$\mathbf{CPOSP}$
`$\CatCS$`	$\mathbf{CS}$
`$\CatCSEM$`	$\mathbf{CSEM}$
`$\CatDAR$`	$\mathbf{DAR}$
`$\CatDCPO$`	$\mathbf{DCPO}$
`$\CatDCPOp$`	$\mathbf{DCPO}_{\perp}$
`$\CatDCPOps$`	$\mathbf{DCPO}_{\perp !}$
`$\CatDCPOD$`	$\mathbf{DCPO}_D$
`$\CatDCPOG$`	$\mathbf{DCPO}_G$
`$\CatDCPOFILT$`	$\mathbf{DCPOFILT}$
`$\CatDL$`	$\mathbf{DL}$
`$\CatDLat$`	$\mathbf{DLat}$
`$\CatDOM$`	$\mathbf{DOM}$
`$\CatDOMD$`	$\mathbf{DOM}_D$
`$\CatDOMG$`	$\mathbf{DOM}_G$
`$\CatDOMFILT$`	$\mathbf{DOMFILT}$
`$\CatFRM$`	$\mathbf{FRM}$
`$\CatFRMz$`	$\mathbf{FRM}_0$
`$\CatFS$`	$\mathbf{FS}$
`$\CatGRAPH$`	$\mathbf{GRAPH}$
`$\CatH$`	$\mathbf{H}$
`$\CatINF$`	$\mathbf{INF}$
`$\CatINFup$`	$\mathbf{INF}^{\uparrow}$
`$\CatLAT$`	$\mathbf{LAT}$
`$\CatLCSOB$`	$\mathbf{LCSOB}$
`$\CatLDOM$`	$\mathbf{LDOM}$
`$\CatPOID$`	$\mathbf{POID}$
`$\CatPOSET$`	$\mathbf{POSET}$
`$\CatPOSETD$`	$\mathbf{POSET}_D$
`$\CatPOSETG$`	$\mathbf{POSET}_G$
`$\CatSCFRM$`	$\mathbf{SCFRM}$
`$\CatSCFRMi$`	$\mathbf{SCFRM}_1$
`$\CatSCTOP$`	$\mathbf{SCTOP}$
`$\CatSEM$`	$\mathbf{SEM}$
`$\CatSEMI$`	$\mathbf{SEMI}$
`$\CatSET$`	$\mathbf{SET}$
`$\CatSLCTOP$`	$\mathbf{SLCTOP}$
`$\CatSOB$`	$\mathbf{SOB}$
`$\CatSUP$`	$\mathbf{SUP}$
`$\CatSUPu$`	$\mathbf{SUP}^{\wedge}$
`$\CatSUPz$`	$\mathbf{SUP}^{0}$
`$\CatTCPOSP$`	$\mathbf{TCPOSP}$
`$\CatTOP$`	$\mathbf{TOP}$
`$\CatUPS$`	$\mathbf{UPS}$

Table 7: Categories supported by `lathalesians-categories`

3.10 `lathalesians-domains`: domain theory

This package introduces some commands for writing papers on the branch of mathematics called **domain theory**, which studies special kinds of partially ordered sets (posets) [GHK+03].

3.10.1 Usage

```
\usepackage[<key=value list>]{lathalesians-domains}
```

The following options are allowed in the comma-separated key=value list:

Option	Possible values	Default value	Description
`ssfnotation`	`a`, `b`	`b`	whether `\ssf` is equivalent to `\ssfa` or `\ssfb`

Table 8: Options supported by `lathalesians-domains`

Example:

```
\usepackage[ssfnotation=a]{lathalesians-domains}
```

3.10.2 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
`\lteq`, `\lesseq`	`$a \lteq b$`	$a \sqsubseteq b$	"Less than or equal to" for posets
`\lt`, `\less`	`$a \lt b$`	$a \sqsubset b$	"Less than" for posets
`\eq`	`$a \eq b$`	$a = b$	"Equal" for posets, same as $=$
`\gt`, `\greater`	`$a \gt b$`	$a \sqsupset b$	"Greater than" for posets
`\gteq`, `\greatereq`	`$a \gteq b$`	$a \sqsupseteq b$	"Greater than or equal to" for posets
`\wb`, `\waybelow`	`$a \wb b$`	$a \ll b$	"Way-below" for dcpo's
`\wa`, `\wayabove`	`$a \wa b$`	$a \gg b$	"Way-above" for dcpo's
`\join`	`$a \join b$`	$a \sqcup b$	Join of two elements
`\meet`	`$a \meet b$`	$a \sqcap b$	Meet of two elements
`\setjoin`	`$\setjoin A$`	$\bigsqcup A$	Join (supremum) of a set
`\setmeet`	`$\setmeet A$`	$\bigsqcap A$	Meet (infimum) of a set
`\dsetjoin`	`$\dsetjoin A$`	$\bigsqcup^{\uparrow} A$	Directed supremum
`\fsetmeet`	`$\fsetmeet A$`	$\bigsqcap^{\downarrow} A$	Filtered infimum
`\belowset`	`$\belowset{A}$`	$\downarrow A$	Downward closure w.r.t. below
`\aboveset`	`$\aboveset{A}$`	$\uparrow A$	Upward closure w.r.t. below
`\waybelowset`	`$\waybelowset{A}$`	$\Downarrow A$	Downward closure w.r.t. way-below
`\wayaboveset`	`$\wayaboveset{A}$`	$\Uparrow A$	Upward closure w.r.t. way-below
`\ssfa`	`$\ssfa{U}{s}$`	$(U \searrow s)$	Single-step function (notation a)
`\ssfb`	`$\ssfb{U}{s}$`	${}^{s}\chi_{U}$	Single-step function (notation b)
`\ssf`	`$\ssf{U}{s}$`	${}^{s}\chi_{U}$	Notation a or b (default)
`\topfuncspace`	`$\topfuncspace{X}{Y}$`	$C(X, Y)$	Topological function space
`\domfuncspace`	`$\domfuncspace{X}{Y}$`	$[X \to Y]$	Domain-theoretic function space
`\Intervals`	`$\Intervals{[0,1]}$`	$\mathbf{I}[0, 1]$	Interval domain
`\UpperSpace`	`$\UpperSpace{X}$`	$\mathbf{U}X$	Upper space
`\PPD`	`$\PPD{\Intervals{[0,1]}}$`	$\mathbf{P}\,(\mathbf{I}[0, 1])$	Probabilistic power domain (p.p.d.)
`\NPPD`	`$\NPPD{\Intervals{[0,1]}}$`	$\mathbf{P}^{1}\,(\mathbf{I}[0, 1])$	Normalised p.p.d.
`\Maximals`	`$\Maximals{\Intervals{[0,1]}}$`	$\mathrm{Max}\,(\mathbf{I}[0, 1])$	Maximal elements
`\Minimals`	`$\Minimals{\Intervals{[0,1]}}$`	$\mathrm{Min}\,(\mathbf{I}[0, 1])$	Minimal elements
`\Con`	`$\Con_{(A,\lteq)}(a_1, \ldots, a_n)$`	$\mathrm{Con}_{(A,\sqsubseteq)}(a_1, \ldots, a_n)$	Consistency predicate

Table 9: Summary of commands and environments defined in `lathalesians-domains`

Notice that the LaTeX kernel already defines `\leq` and `\geq` as aliases for the plain TeX symbols `\le` and `\ge`, which are rendered as $\leq$ and $\geq$, respectively. These are the equivalents of $\sqsubseteq$ and $\sqsupseteq$, respectively, for the natural ordering found in frequently used posets such as the natural numbers.

In domain theory one may be dealing with multiple posets in the same document. Say you are dealing with posets A and B. You may wish to distinguish between the binary relations for the two posets. One fairly standard way is to use subscripts: `$a \lteq_A b$` and `$a \lteq_B b$` are rendered as $a \sqsubseteq_A b$ and $a \sqsubseteq_B b$, respectively.

By default, `\ssf` is equivalent to `\ssfb`. However, one can change make it equivalent to `\ssfa` using the `ssfnotation` key-value option:

```
\usepackage[ssfnotation=a]{lathalesians-domains}
```

3.11 `lathalesians-computability`: computability theory

This small package, still very much under construction, defines some commands and environments to simplify the typesetting of works in computability theory.

3.11.1 Summary of commands and environments

Command	Example: code	Example: LaTeXed	Description
`\ComputableFunctions`	`$\ComputableFunctions$`	$\mathcal{C}$	computable functions
`\PartialRecursiveFunctions`	`$\PartialRecursiveFunctions$`	$\mathcal{R}$	partial recursive functions
`\partfunc`	`$\delta \partfunc \Sigma^{\omega} \rightarrow M$`	$\delta :\subseteq \Sigma^{\omega} \rightarrow M$	partial function
`\partfunc`	`$\partfuncdefn[\delta]{\Sigma^{\omega}}{M}$`	$\delta :\subseteq \Sigma^{\omega} \mapsto M$	
`\halts`	`$\halts{P(a_1, a_2, \ldots)}$`	$P(a_1, a_2, \ldots)\downarrow$	eventually stops
`\nothalts`	`$\nothalts{P(a_1, a_2, \ldots)}$`	$P(a_1, a_2, \ldots)\uparrow$	never stops

Table 10: Summary of commands and environments defined in `lathalesians-computability`

4 Bibliography: `lathalesians-bibliography`

The file `lathalesians-bibliography.bib` contains the BibTeX bibliography of some of the works that we quote in our research. It is not exhaustive, and we shall be much obliged if you help us expand it.

The convention that we use for the BibTeX keys is as follows:

```
<authorsurname>-<year>
```

or

```
<authorsurname>-<year>-<number>
```

The `<authorsurname>` is the surname of the first author in lower case and without any spaces, dashes, etc., e.g.: `edalat`, `vanmill`. The `year` is in the `yyyy` format, e.g. 2016. The number is included only when there is a clash, thus we have `edalat-1998`, `edalat-1998-1`, and `edalat-1998-2`.

Here is how you can include the bibliography in your document:

```
\bibliographystyle{alpha}
\bibliography{lathalesians-bibliography}
```

Then you can `\cite{edalat-1998}`, for example, and the result will look like [ES98].

References

[ES98] Abbas Edalat and Philipp Sünderhauf. A domain-theoretic approach to computability on the real line. *Theoretical Computer Science*, 210(1):73–98, January 1998.

[GHK+03] Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana Stewart Scott. *Continuous Lattices and Domains*. Number 93 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2003.

Subject index

Index of authors