The uefcsthesis class*

Pauli Miettinen

pauli.miettinen@uef.fi

5th May 2021

Abstract

This package contains the LaTeX class to typeset Bachelor's and Master's theses at University of Eastern Finland's School of Computing.

Contents

---

*This document corresponds to uefcsthesis v0.5.2, dated 2021/05/05.

1 Introduction

This package contains the LaTeX class `uefcsthesis` that is meant for typesetting Bachelor's and Master's theses at University of Eastern Finland's (UEF's) School of Computing. This documentation provides the details about the package itself; separate documentation provides examples on how to use the class to typeset the thesis, and how to use LaTeX in general. Thus, most students should be able to start working with their theses without reading this document first.

It is, however, recommended that everybody using this class *(students and supervisors alike) read Section 3* at some point as it provides the details of the class' features. Section 4 is only needed if one wants to see how the class is implemented, perhaps to improve it or to do debug it.

The design goals of this class were to provide a class that:

1. Makes it easy for students to start writing their theses even with no prior experience on LaTeX.

2. Provides standard LaTeX interface for typesetting the document so that students learn how to use most LaTeX classes.

3. Supports both the more traditional pdfLaTeX engine as well as the more modern LuaLaTeX and XeLaTeX engines.

4. Supports both BibTeX and BibLaTeX.

5. Supports theses in both Finnish and English (in both American and British spelling) and facilitates the use of other languages, especially by supporting UTF-8 encoding.

6. Is easy to adapt if needed.

7. Has clean and well-documented implementation.

2 Installation

The class is normally distributed as a standard LaTeX `.cls` file, and it can be used by copying the `uefcsthesis.cls` file in a place where LaTeX will find it (e.g. in the same folder with the main `.tex` file). To compile the document, the `uef_logo.pdf` file[1] has to be in the same folder as the main `.tex` file (or in a folder defined with the `\graphicspath` command, see [3]).

The source code of the class and this documentation are in `uefcsthesis.dtx` file that has to be used with `uefcsthesis.ins` file. To prepare the `uefcsthesis.cls` file, run

```
latex uefcsthesis.ins
```

---

[1]The file `uef_logo.eps` is needed if standard LaTeX is used instead of pdfLaTeX, LuaLaTeX, or XeLaTeX.

in the command line. This will also produce the `uefcsthesis.xmpd` and `uefcsthesis.bib` files that will be needed to prepare this documentation. To prepare the documentation, run the following commands in the command line:

```
pdflatex uefcsthesis.dtx
bibtex uefcsthesis
makeindex -s gind.ist -o $(PACKAGE).ind $(PACKAGE).idx
makeindex -s gglo.ist -o $(PACKAGE).gls $(PACKAGE).glo
pdflatex uefcsthesis.dtx
while ( grep -q '^LaTeX Warning: Label(s) may have changed' uefcsthesis.log )
do pdflatex uefcsthesis ; done
```

Notice that the last two lines are part of the same command, but they have to be separated with a newline.

If you have standard Unix `make` utility, you can use the package's `Makefile` and issue

```
make uefcsthesis.cls
make uefcsthesis.pdf
```

in the command line to make the class file and documentation.

The documentation depends on the following packages. They all are part of the standard LaTeX distributions such as MiKTeX and TeX Live.

- `inputenc`
- `fontenc`
- `babel`
- `amsmath`
- `microtype`
- `hologo`
- `url`
- `seealso`

- `hypdoc`
- `urlbst`
- `listings`
- `tabularx`
- `booktabs`
- `caption`
- `multicol`

3 Using the Class

To use the class, give its name as a parameter for the LaTeX's `\documentclass` command near the begin of the main `.tex` file:

```
\documentclass{uefcsthesis}
```

The class expects that the text is encoded in UTF-8 encoding. The class also requires that some metadata must be set. An example minimal file is provided in Listing 1.

Listing 1: Minimal `.tex` file for `uefcsthesis.cls`.

```
\documentclass[mscthesis,english,oneside,biblatex]{uefcsthesis}

%% Correct the below with the name of your bibliography file
\addbibresource{minimal.bib}

%% Replace all capital text with your own information.
\title{TITLE} % Title of the thesis
\author{GIVEN NAME}{FAMILY NAME} % Your name
\date{MONTH YEAR} % The month and year of handing in your thesis
\city{CITY} % Either Kuopio or Joensuu
\firstsupervisor{GIVEN FAMILY} % Name of the first supervisor
\secondsupervisor{GIVEN FAMILY} % Name of the second supervisor, if any
\keywords{KEYWORD1\sep KEYWORD2\sep ETC} % Keywords must be separated with \sep

%% To get the ACM CCS classification, you can visit
%% https://dl.acm.org/ccs/ccs.cfm
%% There you can find a tool to generate LaTeX code for the classification
%% Copy it here. You don't need to copy the XML at the begin, though.
%% For example,
%% \ccsdesc[500]{Some Class}

\begin{document}
\maketitle
\begin{abstract}
  WRITE YOUR ENGLISH ABSTRACT HERE
\end{abstract}

\frontmatter
\tableofcontents
\mainmatter

\chapter{Introduction}
\label{cha:intro}

WRITE YOUR INTRODUCTION HERE

WRITE THE REST OF THE THESIS HERE

THIS IS AN EXAMPLE OF USING CITATIONS:
Graph generators are important \citep{metzler18random}.
\citet{kalofolias18from} discuss sets of redescriptions.

%% Next comes the references
\printbibliography[heading=bibintoc]

\backmatter % Do not remove!
%% Possible appendices come here
\end{document}
```

3.1 Dependencies to Other Packages

The `uefcsthesis.cls` class depends on a number of other packages. These are listed below. The packages marked with a star might or might not be loaded, depending on the class options (see Section 3.2).

- amsmath
- amsthm
- appendix
- array
- babel*
- biblatex*
- bm
- booktabs
- caption
- changecntr*
- comment
- csquotes
- etoolbox
- fancyhdr
- fontenc*
- fontspec*
- geometry

- graphicx
- hyperref*
- ifluatex
- ifthen
- ifxetex
- inputenc*
- mathtools
- microtype
- natbib*
- newtxmath
- newtxtext*
- pdfx*
- polyglossia*
- setspace
- textcomp
- tocbibind
- xparse

3.2 Class Options

The class takes a number of options that control its behaviour. The different options are listed in Table 1.

Thesis Type The main options control the thesis type. These are `mscthesis` and `bscthesis`. They also have Finnish aliases, `gradu` and `kandi`. None of these options have any effect on the language of the thesis (i.e. `kandi` doesn't set the language to Finnish nor does `mscthesis` set it to English). The `bscthesis` (and `kandi`) option do set some other features, though: it turns the layout for one-sided printing and changes the math numbering so that math counters do not reset at the start of chapters.

Table 1: Options for loading the class. Options in the same group are mutually exclusive. The default for each group is listed first.

Group	Option	Description
Thesis type	`mscthesis`	Sets the thesis type to Master's thesis
	`gradu`	Same as above
	`bscthesis`	Sets the thesis type to Bachelor's thesis
	`kandi`	Same as above
Language	`finnish`	Sets the language to Finnish
	`english`	Sets the language to (US) English
	`british`	Sets the language to British English
Engine	`polyglossia`	Activates the `polyglossia` package instead of the `babel` package (using `babel` is the default)
Printing	`oneside`	Adapts the layout for one-sided printing
	`twoside`	Adapts the layout for two-sided printing
Finalizing	`final`	Produce finalized output
	`draft`	Produce draft output
Numbers	`lining`	Activates the lining figures
	`osf`	Activates the old-style figures
Bibliography engine	`bibtex`	Activates the support for the BibTeX bibliography engine
	`biblatex`	Activates the support for the `biblatex` bibliography package with the `biber` engine
Citation style	`authoryear`	Activates the author–year citation style
	`numeric`	Activates the numeric citation style
Math numbering	`chaptermathnum`	Reset math environments' numbering at the begin of each chapter
	`runningmathnum`	Keep the same numbering throughout the thesis
Theorem numbering	`sharedtheoremnumbers`	All theorem-like environments share the same numbering
	`separatetheoremnumbers`	Each theorem-like environment has its own numbering
Hyperref-package	`hyperref`	Load the `hyperref` package
	`nohyperref`	Do not load `hyperref`
PDF/A	`nopdfa`	Do not generate the output in PDF/A format
	`pdfa`	Load `pdfx` package and produce PDF/A output; will also load the `hyperref` package

Language and Engine The language options are straightforward. The `uefcsthesis` class supports Finnish, US English, and British English as pre-defined languages using options `finnish`, `english`, and `british`, respectively. Setting the language changes all pre-defined strings and sets the hyphenation correctly.

The actual language support comes from the `babel` or `polyglossia` packages, depending on whether the `polyglossia` option is given. The support is (mostly) transparent to the end user (though `polyglossia` does support more complicated cases), but it does set restrictions to the LaTeX engine used: when using `babel`, one can use any LaTeX engine; when using `polyglossia`, one can use either LuaLaTeX or XƎLaTeX. Notice however that as of 2019, development of `polyglossia` seems to have stalled; currently it is recommended to use `babel` unless some special features of `polyglossia` are needed.

Printing, Finalizing, and Numbers The options `oneside` and `twoside` change the layout of the thesis suitable for one-sided and two-sided output. For digital use, one-sided output is preferable.

Another option controlling the layout is the `osf` option that changes the numbers to old-style figures. Using old-style figures requires one to be careful when writing numbers in the running text: the 'mathematical numbers' should always be surrounded by dollar signs as their outlook is markedly different from the 'text numbers'. The opposite of old-style figures (and the default) is `lining` figures.

The finalizing options `final` and `draft` control the layout of various things. Many standard LaTeX-packages check the `draft` option, and change their behaviour accordingly. For example, the `graphicx` package does not print the figures when the draft option is given, and LaTeX prints black boxes next to overflown lines to highlight them. On the other hand, the `uefcsthesis` class catches the `draft` options to make sure that the `microtype` package keeps working even in the draft mode.

Bibliography Engine and Citation Style LaTeX has (and `uefcsthesis` supports) two bibliographical engines: BibTeX and BibLaTeX with biber. In general, the more modern BibLaTeX package is recommended, but the more standard BibTeX can also be used (and is better if one wants to use the work later with commercial publishers, who do not usually support BibLaTeX). You can mix-and-match the engines and use pdfLaTeX with BibLaTeX, for example.

Whichever bibliograhy engine is selected, `uefcsthesis` allows you to use `natbib`-like citation commands like `\citet` and `\citep`. The commands work in both author-year citation style (the default, with `auhtoryear`) and numerical citation style, which can be activated with the `numeric` option.

With BibLaTeX, the citation style is the `biblatex-apa` package [5]. With BibTeX, we are using the `apacite` package [6]. See their respective documentations for more information regarding the usage of the styles and various bibliography record types and fields supported.

Numbering in Mathematics   There are two features of numbering of 'mathematical objects' that can be controlled in `uefcsthesis`. The first is whether the numbering of equations, as well as theorems, lemmas, and other theorem-like environments resets at the begin of every chapter. This is the default in Master's theses, and these numbers are two-part, like (1.1). In Bachelor's theses, the numbering is running throughout the thesis, and the numbers are one-part, like (1). Option `chaptermathnum` turns on the per-chapter numbering also for Bachelor's theses and `runningmathnum` turn on the running numbering also for Master's theses.

The other numbering feature that can be controlled considers the numbering of theorem-like environments (like `theorem`, `lemma`, `corollary`, etc.) By default (or when given the `sharedtheoremnumbers` option), these share the same counter, that is, Lemma 1 is followed by Theorem 2 which is followed by Corollary 3 and so on. Option `separatetheoremnumbers` separates the counters, and Lemma 1 is followed by Theorem 1 which is followed by Corollary 1 and so on. More information regarding the theorem-like environments is in Section 3.6

Hyperref and PDF/A   The `uefcsthesis` package can automatically generate PDF/A output using the `pdfx` package. This is controlled by the `pdfa` and `nopdfa` options. By default, generation of PDF/A is turned off, as it requires a sidecar file (see Section 3.5) and as PDF/A does not allow comments, it might harm the supervisor's ability to comment the drafts. For the final deliverable, it is recommended to use the `pdfa` option (and to read Section 3.5 before doing that).

Even when no PDF/A is being generated, `uefcsthesis` uses the `hyperref` package to turn the references into hyperlinks and to add the PDF metadata. Sometimes `hyperref` package must be loaded after some user-loaded package; in these situations, the loading of `hyperref` by `uefcsthesis` can be turned off with the `nohyperref` option.

3.3   Document Metadata, Title Page, and Abstract

The `uefcsthesis` class provides standard LaTeX commands for entering the document metadata (title, author, etc.) and for printing the title page and abstract. To support abstracts in different languages (e.g. Finnish and English), many commands are enhances to take an optional argument that defines the language for which the metadata is set.

Entering Document Metadata

The document metadata must be entered in the document preamble, as it is needed at the begin of the document to fill in the PDF metadata.

`\title` The title of the thesis is set by `\title` macro, which can be called as `\title[⟨lang⟩]{⟨title⟩}`. The optional ⟨lang⟩ parameter can be used to set title in languages other than the document's main language. This options use with `\title` and other macros that support it is explained in Section 3.4. If the thesis
`\subtitle` has a subtitle, it's given with `\subtitle[⟨lang⟩]{⟨subtitle⟩}` command.
`\author` The author name is given as `\author{⟨given name(s)⟩}{⟨family name⟩}`.

\date	There's no multilingual support for the author name, as of now. The date of the thesis is set as \date[⟨*lang*⟩]{⟨*date*⟩}. The mandatory argument ⟨*date*⟩ can be given as \today, but this prints the full date, while the standard is to have just the month and the year.
\city	The thesis must indicate whether it's done in Joensuu or Kuopio. This is done using the \city macro as \city{⟨*city*⟩}, where ⟨*city*⟩ is either Joensuu or Kuopio. The \city macro does also support the ⟨*lang*⟩ optional argument, but unlike the other macros, if none is given, the city name is set for all supported languages (Finnish, US English, and British English) as Joensuu and Kuopio are the same in all of them.
\firstsupervisor	The name of the first (primary) supervisor is given as \firstsupervisor{⟨*name*⟩}, where ⟨*name*⟩ is given in 'given name(s) family name' format, for instance,
\secondsupervisor	\firstsupervisor{Pauli Miettinen}. If the thesis has a second supervisor, his or her name is given with \secondsupervisor command in exactly the same way as the first supervisor. These two macros do not take the language as an optional argument.
	If the thesis has more than two supervisors, all but the last supervisors' names should be given in \firstsupervisor, separated with commas. The last supervisor's name is given with \secondsupervisor.
\keywords	The thesis's keywords are given with \keywords[⟨*lang*⟩]{⟨*keywords*⟩} command. The ⟨*keywords*⟩ argument must be a list of keywords separated with \sep command.
\sep	The \sep command creates the correct separator for different use cases (e.g. semicolon in the abstract page, comma in some metadata fields).
\ccsdesc	The ACM 2012 Computing Classification System's (CCS) classification is given with the \ccsdesc macro that has the format \ccsdesc[⟨*level*⟩]{⟨*class*⟩}. The ⟨*level*⟩ argument describes the level of importance of the ⟨*class*⟩. The \ccsdesc commands are expected to be generated with the ACM tool at https://dl.acm.org/ccs/ccs.cfm.
\numberofappendices	If the thesis contains appendices, their number must be inserted with
\appendixpagecount	\numberofappendices{⟨*num*⟩}. The \appendixpagecount{⟨*pages*⟩} macro is used to tell the class how many pages long the appendices in total are, and it is mandatory if \numberofappendices macro is given a value larger than 0.

Special Pages and Document Structure

\maketitle	The thesis title page is printed with the (standard) \maketitle command. The command doesn't take any arguments.
abstract	The abstract of the thesis is written in the abstract environment, like

```
\begin{abstract}
  The text of the abstract
\end{abstract}
```

The abstract environment will also print the abstract page and populate it with the document metadata and page numbers.

To print an abstract in other language, the `abstract` environment can be called with an optional argument: `\begin{abstract}[⟨lang⟩]`. This will generate another abstract page, typeset in language ⟨*lang*⟩.

`\frontmatter` The start of the 'front matter' of the thesis is indicated by the `\frontmatter` macro, which is mandatory in every document using `uefcsthesis`. The front matter has roman pages numbers and it contains at least the table of contents `preface` (generated with the `\tableofcontents` command). In addition of the table of contents, the front matter can contain the preface, that is written inside the `preface` `acknowledgements` environment, and the acknowledgements, written inside the `acknowledgements` environment. These environments do not have the language parameter, but the standard `\selectlanguage` command can be used to typeset them in other languages (see also Section 3.4).

`\mainmatter` The body of the thesis is written after the `\mainmatter` macro, which is again `\backmatter` mandatory. After the list of references, starts the 'back matter', indicated by the `\backmatter` macro. The back matter contains the appendices, if any, but it is nonetheless mandatory, even in there are no appendices, as it is used to count the total number of pages in the thesis.

`appendices` The (possible) appendices come after the `\backmatter` macro and they are enclosed in `apppendices` environment. For example,

```
\backmatter
\begin{appendices}
  \chapter{All The Stuff That Didn't Fit to the Main Text}
\end{appendices}
```

3.4 Multilingual Documents

The `uefcsthesis` package is designed to directly support theses written in Finnish or English, to support including text in other languages, and to be easy to extend to support especially abstract pages in other languages. It also allows to change the pre-defined strings with ease.

The main part of the multilingual support is that `uefcsthesis` assumes UTF-8 encoded input. Both LuaLaTeX and X⅁LaTeX assume that by default, and for pdfLaTeX, we use the `inputenc` package with `utf8` option.

Abstract in the Secondary Language

The previous section detailed how to provide the metadata and produce the abstract page in the thesis' primary language (i.e. in the language selected in the class options). Master's and Bachelor's theses should also include abstract page in the secondary language (English if the thesis is written in Finnish and vice versa). To make that, the necessary metadata has to be provided in the secondary language using the optional ⟨*lang*⟩ argument. At least `\title`, possible `\subtitle`, and `\keywords` have to be given in both languages.

To produce the actual abstract, give the optional ⟨*lang*⟩ argument to the abstract environment. Listing 2 shows an example of a Master's thesis in Finnish

Listing 2: Example of a Finnish MSc thesis with British abstract.

```
\begin{documentclass}[mscthesis,finnish]{uefcsthesis}
\title{Otsikko}
\title[british]{Title}
\author{Oula}{Opiskelija}
\firstsupervisor{Rauni~P.~Rohvessori}
\secondsupervisor{Toini Tutkija}
\keywords{pro gradu –tutkielma\sep tieteellinen kirjoittaminen\sep esimerkki}
\keywords[british]{MSc thesis\sep scientific writing\sep example}
\begin{document}
  \maketitle
  \begin{abstract}
    Suomenkielinen teksti
  \end{abstract}
  \begin{abstract}[british]
    English text
  \end{abstract}
  \frontmatter
  % Rest of the thesis
\end{document}
```

with British abstract.

Abstract in Other Languages and Changing Pre-Defined Strings

Students whose mother tongue is not Finnish or English might wish to include an abstract page in their mother tongue. It is recommended that LuaLATEX or XƎLATEX is used in these cases. If the language needs non-latin alphabet, the `uefcsthesis` class should be loaded with the `polyglossia` option.

Class `uefcsthesis` already loads Swedish in addition to the supported languages. To add another new language, `polyglossia`'s `\setotherlanguage` command must be called first [4, Section 2.1].

There are two ways to add an abstract in other languages. The first is to use the English abstract page and only write the text of the abstract (and potentially title and the keywords) in the other language. This can be done by calling the `abstract` environment with `english` as the optional argument and calling `polyglossia`'s ⟨lang⟩ environment [4, Section 3]. For example,

```
\setotherlanguage{german}
\begin{abstract}[english]
  \begin{german}
    Kurzfassung auf Deutch
  \end{german}
\end{abstract}
```

`\setstring`   The alternative is to print the full abstract page in the target language. This

Table 2: Pre-defined strings used in the abstract page

⟨*string name*⟩	⟨*string text*⟩ (English)	⟨*string text*⟩ (Finnish)
thesistype	Master's thesis	Pro gradu -tutkielma
field	Computer Science	Tietojenkäsittelytiede
departmentname	School of Computing	Tietojenkäsittelytieteen laitos
facultyname	Faculty of Science and Forestry	Luonnontieteiden ja metsätieteiden tiedekunta
universityname	University of Eastern Finland	Itä-Suomen yliopisto
pages	p.	s.
and	and	ja
keywordsname	Keywords	Avainsanat
supervisorname	Supervisor	Ohjaaja
supervisorsname	Supervisors	Ohjaajat
app@o	appendix	liite
app@m	appendices	liitettä

requires that the `uefcsthesis` class is told the pre-defined strings in the new language. This can be done with the `\setstring` command, that is called as follows: `\setstring[`⟨*lang*⟩`]{`⟨*string name*⟩`}{`⟨*string text*⟩`}`. If no ⟨*lang*⟩ is given, the `\setstring` command changes the string for the document main language. The pre-defined strings that are used in the abstract page are listed in Table 2 together with their English and Finnish values.

The language-aware metadata commands (`\title` etc.) work with any language that has been added with `\setotherlanguage` (and Swedish, that is automatically included). A thesis that contains a Swedish abstract might have the following lines (among others):

```
\setstring[swedish]{universityname}{Östra Finlands universitet}
\setstring[swedish]{facultyname}{Naturvetenskapliga och forstvetenskapliga fakulteten}
\title[swedish]{Rubrik}
```

The `\setstring` command can be used to change the pre-defined strings in Finnish, English, and British, as well. If, for example, one wants to change the field of the thesis to Data Science (datatiede) in a Finnish MSc thesis with British abstract, that would require the following commands in the preamble:

```
\setstring{field}{Datatiede}
\setstring[british]{field}{Data Science}
```

Thanks to the UTF-8 support, `uefcsthesis` can support also non-latin alphabets, although writing systems that are not left-right are not tested or supported. The standard font does not support all fonts, though, so you might have to tell `fontspec` (assuming you're using LuaLATEX or XᴇLATEX, as you should) to use

Listing 3: Default minimal `.xmpdata` metatdata file

```
\Title{\uef@pdf@title}
\Author{\@author}
\Keywords{\uef@keywords}
\Subject{\uef@pdf@subject}
\Publisher{\uef@s@universityname}
\endinput
```

different font for Cyrillic text. Standard LaTeX comes with Tempora font, that is a Cyrillic font that looks similar to Times New Roman. To use it, add the following to the document preamble:

```
\newfontfamily\russianfont[Script=Cyrillic]{Tempora}
```

3.5 PDF/A

University of Eastern Finland requires that Master's theses are returned in PDF/A format. `uefcsthesis` allows direct generation of PDF/A output using the `pdfx` package [9]. The generation of the PDF/A output can be activated with the `pdfa` class option. To successfully generate the PDF/A file, `pdfx` requires a sidecar file containing the document metadata. To obtain the default sidecar file, *you rename the file `minimal.xmpdata` to ⟨thesisfile⟩.xmpdata*. If your theses' main `.tex` file is called `opiskelija_oula_gradu.tex`, copy `minimal.xmpdata` to `opiskelija_oula_gradu.xmpdata`. The contents of the `minimal.xmpdata` file are provided in Listing 3.

In most cases, the default file will work fine. One should, however, *always check the metadata*. This can be done, e.g. in Adobe Reader, selecting `File→Document Properties`. If the thesis title, author name, or keywords contain (complex) LaTeX commands, the metadata will not look good. In these cases, one should edit the ⟨thesisfile⟩.xmpdata file. It should be necessary to edit only the contents of the macros `\Title`, `\Author`, and `\Keywords`. The current macros (e.g. `\uef@pdf@title`) can just be replaced with the actual title, author name, and keywords, all in UTF-8 encoding, and separating the keywords with `\sep`. See the documentation of `pdfx` [9] for more information.

If the document uses other fonts than the standard ones (e.g. for typesetting Russian), pdfLaTeX might not produce compliant PDF/A documents. In these cases, it is recommended to use LuaLaTeX. On the other hand, as `uefcsthesis` produces files for the PDF/A-1b standard, if any of the figures contain layers or CMYK colours, the resulting file is not compliant. The validity of the generated file can be checked with the free VeraPDF application.[2]

`\hypersetup`   As `pdfx` loads `hyperref` before you can load it yourself, you have to use the `\hypersetup` to configure the parts of `hyperref` that you might want to change

---

[2]http://verapdf.org/software/, accessed 17 Dec 2018.

(e.g. the colouring of the links). The `uefcsthesis` guarantees that `\hypersetup` exists, but if `hyperref` is not loaded, it doesn't do anything. For more information on how to use `\hypersetup`, refer to the `hyperref` documentation [10].

3.6 Pre-Defined Commands and Environments for Mathematics

The `uefcsthesis` class facilitates the typesetting of mathematics by defining most of the standard mathematical theorem-like environments and some other helper macros.

Theorem-Like Environments

theorem

theorem*

lemma
corollary
proposition
conjecture
proof

definition
problem
example
remark
note

The class uses the `amsthm` package to build the theorem-like environments. It defines a number of standard environments. The standard for these environments is `theorem` that is used to write a theorem. It, like the other theorem-like environments, takes one optional argument, contents of which will be typeset in parentheses after the word **Theorem**. The theorem will be numbered using a two-part number $x.y$, where x is the chapter number and y is the theorem number, except if option `runningmathnum` is given (or the thesis is Bachelor's thesis), in which case the numbering scheme is one-part. The starred version will not print any number.

The other theorem-like environments are `lemma`, `corollary`, `proposition`, and `conjecture`. Their numbering scheme is the same as with `theorem`, and by default, they have the same counter (i.e. Lemma 1 is followed by Theorem 2). This can be changed with option `separatetheoremnumbers`.

The environment to typeset (short) proofs, `proof`, is provided by the `amsthm` class; see its documentation [8] for further information.

The environments `definition`, `problem`, and `example` are typset differently, and they each have their own counters. These counters have the same format as `theorem`, though.

The last two environments, `remark` and `note`, have no numbers.

Convenient Macros for Typesetting Math

\abs

\norm

The `uefcsthesis` class provides few special macros that help typesetting math. The first macro is `\abs`, used to typeset 'absolute value' (i.e., two vertical bars). It's basic form is `\abs[⟨size⟩]{⟨stuff⟩}` that will put ⟨stuff⟩ inside properly-spaced vertical bars. If the optional argument ⟨size⟩ is given, it should be one of LaTeX's delimiter sizing commands, like `\big`, and it will control the size of the bars. The command also has a starred version, `\abs*`, which is equivalent of putting `\left` and `\right` around the bars.

Similarly to `\abs`, `\norm` produces double-vertical bars around its mandatory argument. It has similar sizing commands, so `\norm[\Bigg]{\frac{a}{b}}` will produce tall double-vertical bars.

Finnish mathematical text usually uses 'inverted square brackets' to indicate open end of an interval, like $]a, b[$. Notice, however, that LaTeX typesets the space around the inverted brackets wrongly. `uefcsthesis` provides three commands to facilitate the typesetting of such intervals: `\lointerval`, `\rointerval`, and `\ointerval` for left-open intervals $]a, b]$, right-open intervals $[a, b[$, and open intervals $]a, b[$, respectively. They take the same optional argument or starred form as `\abs` and `\norm`.

Finally, `uefcsthesis` also loads the `bm` package. It allows one to use italic bold-face fonts in math by using the command `\bm` instead of `\mathbf`. Unfortunately, `bm` can sometimes cause issues by using many LaTeX font spots; see the manual [2] for more information, including ways to solve the issue.

3.7 Support for Other Packages

The `uefcsthesis` loads a few more packages, and provides support (mostly, Finnish translation) for further packages if they are loaded.

Tables and Figures

To insert the figures, `uefcsthesis` loads the `graphicx` package. You can use it to include your own figures and to customize the folder where you store the figures; see the package documentation [3] for more information.

To help typesetting tables, `uefcsthesis` loads `tabularx` and `booktabs` packages. Most importantly, the class allows users to use four new table column types: `C`, `L`, `R`, and `X`. Three first, `C`, `L`, and `R` create centered, left-aligned, and right-aligned columns – similar to standard `c`, `l`, and `r` – but set their contents in math mode. The math mode can be cancelled in a particular cell (e.g. in the caption) by surrounding the cell contents in dollar signs. These column types are convenient for typesetting simple tables with numbers, but see the `siunitx` package [11] for more complete approach for typesetting tables with numbers.

The last new column type, `X`, can only be used inside a new environment, `tabularx` and it creates a `p`-type column with automatically adjusting width. See the documentation of the `tabularx` package [1] for more information.

The `tabularx` package loads the `array` package, which provides some useful tools for defining material that will be repeated for every table cell in a column; see [7] for more information.

The `booktabs` package defines four macros for drawing rules in LaTeX tables: `\toprule`, `\midrule`, `\cmidrule`, and `\bottomrule`. These commands should be used instead of old `\hline`, and *vertical rules should never be used in tables*. The documentation for the `booktabs` package [?] explains how to design a good-looking table, and it is a recommended reading for anybody using LaTeX.

Typesetting Algorithms

The `uefcsthesis` package does not provide any special commands for typesetting algorithms, but it supports the standard packages used to do it. Namely,

`uefcsthesis` provides Finnish translations for the floating environment names and 'list-of-algorithms' lists for the packages `algorithm2e` and `algorithmicx`, often used to typeset pseudo-code, and for packages `listings`, `listingsutf8`, and `fancyvrb`, often used to typeset actual program listings.

4 Implementation

Most of the code in this package is for the `uefcsthesis.cls` class file (compiled with argument `cs`), but we also include the basic `xmpdata` file for `pdfx` package (compiled with the argument `xmp`).

```
1 ⟨*cs⟩
```

4.1 Package Options and Ifs

We will record all package options in `\if`s, so we start by defining them.

`uef@mscthesis`
`uef@twoside`
`uef@osf`

These ifs will control the outlook of the thesis: whether it's an MSc or BSc thesis (default: MSc); whether it's printed one or two-sided (default: one); and whether to use old-style figures (default: not).

```
2 \newif\ifuef@mscthesis\uef@mscthesistrue
3 \newif\ifuef@twoside\uef@twosidefalse
4 \newif\ifuef@osf\uef@osffalse
```

`uef@polyglossia`
`uef@finnish`
`\uef@language`

These ifs will control the language of the document; `uef@polyglossia` controls whether to load the `polyglossia` package or the `babel` package (default: `babel`), and `uef@finnish` stores whether the document is in Finnish or in some other language. In any case, the name of the language is stored in `\uef@language`.

```
5 \newif\ifuef@polyglossia\uef@polyglossiafalse
6 \newif\ifuef@finnish\uef@finnishtrue
7 \newcommand{\uef@language}{finnish}
```

`uef@numbertheoremsbychapter`
`uef@singlemathnumber`

These ifs control the behaviour of math environment numbering. If `uef@singlemathnumber` is true it means that Theorems, Lemmas, Propositions, and Conjectures use one number; otherwise, they each have their own number.

```
8 \newif\ifuef@numbertheoremsbychapter\uef@numbertheoremsbychaptertrue
9 \newif\ifuef@singlemathnumber\uef@singlemathnumbertrue
```

`uef@biblatex`
`\uef@biblatexstyle`

Whether to use BIBLATEX or BIBTEX (default: BIBTEX), and which style to use (default: authoryear-comp)

```
10 \newif\ifuef@biblatex\uef@biblatexfalse
11 \newcommand{\uef@biblatexstyle}{authoryear-comp}
```

`uef@hyperref`
`uef@pdfa`

The final if's control whether we load the `hyperref` package and the `pdfx` package. As `pdfx` loads `hyperref`, we don't load the latter if the former is loaded. Sometimes it's also useful to not load `hyperref` here, but to postpone it for later.

```
12 \newif\ifuef@hyperref\uef@hyperreftrue
13 \newif\ifuef@pdfa\uef@pdfafalse
```

Package Options

With the ifs defined, we can define the class options to set the ifs in the correct way.

mscthesis · The most important option is the type of the thesis. MSc theses are by default
bscthesis · two-sided and have their theorem counter(s) reset in every chapter, while BSc
gradu · theses are by default one-sided and don't reset their theorem counter(s) for every
kandi · chapter.

For convenience, there's also Finnish equivalents, but they don't imply the language of the thesis.

```
14 \DeclareOption{mscthesis}{\uef@mscthesistrue}
15 \DeclareOption{bscthesis}{%
16   \uef@mscthesisfalse%
17   \uef@twosidefalse%
18   \uef@numbertheoremsbychapterfalse%
19 }
20 \DeclareOption{gradu}{\uef@mscthesistrue}
21 \DeclareOption{kandi}{%
22   \uef@mscthesisfalse%
23   \uef@twosidefalse%
24   \uef@numbertheoremsbychapterfalse%
25 }
```

oneside · These options allow one to specify one or two sided outlook.
twoside

```
26 \DeclareOption{oneside}{\uef@twosidefalse}
27 \DeclareOption{twoside}{\uef@twosidetrue}
```

polyglossia · The `polyglossia` option activates the `polyglossia` package. Using it also means that the document must be compiled with either LuaLaTeX or XꟼLaTeX.

```
28 \DeclareOption{polyglossia}{\uef@polyglossiatrue}
```

finnish · These options control the language of the thesis. Currently only Finnish and
english · English (with either US or UK spelling) are supported.
british

```
29 \DeclareOption{finnish}{\uef@finnishtrue\renewcommand{\uef@language}{finnish}}
30 \DeclareOption{english}{\uef@finnishfalse\renewcommand{\uef@language}{english}}
31 \DeclareOption{british}{\uef@finnishfalse\renewcommand{\uef@language}{british}}
```

osf · These options control the way the figures are printed. Option `osf` turns on old-style
lining · figures for text, but not for math or tables. Option `lining` uses the lining figures.

```
32 \DeclareOption{osf}{\uef@osftrue}
33 \DeclareOption{lining}{\uef@osffalse}
```

bibtex · These options control the way the bibliography is done. Options `bibtex` and
biblatex · `biblatex` choose between BibTeX and BibLaTeX, and options `numeric` and
numeric · `authoryear` choose between numeric and author-year citation format.
authoryear

```
34 \DeclareOption{bibtex}{\uef@biblatexfalse}
35 \DeclareOption{biblatex}{\uef@biblatextrue}
36 \DeclareOption{numeric}{\renewcommand{\uef@biblatexstyle}{numeric-comp}}
37 \DeclareOption{authoryear}{\renewcommand{\uef@biblatexstyle}{authoryear-comp}}
```

	These options control the numbering of the mathematical theorem-style en-
chaptermathnum	vironments, namely theorems, lemmas, corollaries, and propositions. Options
runningmathnum	`chaptermathnum` and `runningmathnum` control whether the counters will re reset
sharedtheoremnumbers	at the begin of every chapter or not, respectively, while `sharedtheoremnumbers`
separatetheoremnumbers	and `separatetheoremnumbers` control whether all of the environments share a

single counter or if there's a different counter for each environment.

```
38 \DeclareOption{chaptermathnum}{\uef@numbertheoremsbychaptertrue}
39 \DeclareOption{runningmathnum}{\uef@numbertheoremsbychapterfalse}
40 \DeclareOption{sharedtheoremnumbers}{\uef@singlemathnumbertrue}
41 \DeclareOption{separatetheoremnumbers}{\uef@singlemathnumberfalse}
```

	Here we control whether we load the `hyperref` package and whether we should
hyperref	generate PDF/A format output (by loading the `pdfx` package). As `pdfx` loads
nohyperref	`hyperref`, we will have to do all setup of `hyperref` using `\hypersetup`. We will
pdfa	also make sure that we won't load `hyperref` if `pdfa` option is set to true.
nopdfa	

```
42 \DeclareOption{hyperref}{\uef@hyperreftrue}
43 \DeclareOption{nohyperref}{\uef@hyperreffalse}
44 \DeclareOption{pdfa}{\uef@pdfatrue}
45 \DeclareOption{nopdfa}{\uef@pdfafalse}
```

	We catch `draft` and `final` options so that we could act on them if we would want
draft	to.
final	

```
46 \DeclareOption{draft}{\PassOptionsToClass{\CurrentOption}{book}}
47 \DeclareOption{final}{\PassOptionsToClass{\CurrentOption}{book}}
```

At this point, we can process the options.

```
48 \ProcessOptions\relax
```

This is a convenient point to print some info to log for potential debugging etc.

```
49 \ifuef@mscthesis
50   \def\@tempa{Master's thesis}
51 \else
52   \def\@tempa{Bachelor's thesis}
53 \fi
54 \ifuef@twoside
55   \def\@tempb{two-sided}
56 \else
57   \def\@tempb{one-sided}
58 \fi
59 \ifuef@osf
60   \def\@tempc{old-style}
61 \else
62   \def\@tempc{lining}
63 \fi
64 \ifuef@biblatex
65   \ClassInfo{uefcsthesis}{Creating a \@tempa\space in \uef@language . \MessageBreak
66     Using BibLaTeX with style \uef@biblatexstyle .\MessageBreak
67     Layout is \@tempb \MessageBreak
68     and numbers \@tempc
```

```
69   }
70 \else % BibTeX
71   \ClassInfo{uefcsthesis}{Creating a \@tempa\space in \uef@language . \MessageBreak
72     Using BibTeX with style \uef@biblatexstyle .\MessageBreak
73     Layout is \@tempb \MessageBreak
74     and numbers \@tempc
75   }
76 \fi
```

Load the standard `book` class this class is based on

```
77 \ifuef@twoside
78   \PassOptionsToClass{twoside}{book}
79 \else
80   \PassOptionsToClass{oneside}{book}
81 \fi
82 \LoadClass[a4paper,12pt]{book}
```

4.2 Setting Up the Outlook

We set the page geometry using the `geometry` package.

```
83 \RequirePackage[a4paper,%
84 top=3cm,%
85 left=35mm,%
86 right=30mm,%
87 bottom=20mm,%
88 head=0pt,%
89 headsep=0pt,%
90 foot=12pt,%
91 footskip=13mm,
92 ]%
93 {geometry}
```

`uef@modern` Load some helper packages to identify if we're using a modern version of LaTeX (i.e. LuaLaTeX or XeLaTeX). Save the knowledge at `uef@modern`, so we know it later, and load `inputenc` with UTF-8 encoding and `fontenc` if we're not modern.

```
94 \RequirePackage{ifxetex}
95 \RequirePackage{ifluatex}
96 \RequirePackage{ifthen}
97 \newif\ifuef@modern
98 \ifthenelse{\boolean{xetex}\OR\boolean{luatex}}{\uef@moderntrue}{\uef@modernfalse}
99 \ifuef@modern
100 \else
101   \RequirePackage[utf8]{inputenc}
102   \RequirePackage[T1]{fontenc}
103 \fi
```

Next, load the font. We use `newtxtext` and `newtxmath`. However, the AMS packages (`amsmath` and `amsthm`) must be loaded before `newtxmath`. We load `amsthm` here, as `newtxmath` loads `amsmath`.

```
104 \RequirePackage{amsthm}
```

If we're using modern engine, we don't load newtxtext.

```
105 \ifuef@modern\else
106   \ifuef@osf
107     \RequirePackage[theoremfont,osf]{newtxtext}
108   \else
109     \RequirePackage[theoremfont]{newtxtext}
110   \fi
111 \fi
```

We will also load `textcomp` and `bm` packages to improve math fonts. And we use `varg` argument for `newtxmath` to get more distinguishable g.

```
112 \RequirePackage{textcomp}
113 \RequirePackage[varg]{newtxmath}
114 \RequirePackage{bm}
```

If we're using modern engine, we load TeX Gyre Termes as the font (with or without old style figures, depending on the options).

```
115 \ifuef@modern
116   \RequirePackage[no-math]{fontspec}
117   \ifuef@osf
118     \setmainfont[Numbers=OldStyle]{TeX Gyre Termes}
119   \else
120     \setmainfont{TeX Gyre Termes}
121   \fi
122   \DeclareSymbolFont{operators}{\encodingdefault}{\familydefault}{m}{n}
123 \fi
```

FIXME: doesn't support `newtxmath`'s theoremfont option

There is no indent between paragraphs, but there's extra space of 1 em that can be increased to 2 em. The spacing is set to one-and-half using the `setspace` package.

```
124 \setlength{\parindent}{0pt}
125 \setlength{\parskip}{1em \@plus 1em}
126 \RequirePackage[onehalfspacing]{setspace}[2011/12/19]
```

Next we have to load `babel` or `polyglossia` so that we get to set other language-dependant features. But before that, we have to load `csquotes`, that must come before `babel`. We always load Finnish, English, and Swedish as the languages, and set the main language according to the options. Babel must be at least 3.9c to support main= definition.

```
127 \RequirePackage{csquotes}
128 \ifuef@polyglossia
129   \RequirePackage{polyglossia}
130   \ifuef@finnish
131     \setdefaultlanguage{finnish}
132     \setotherlanguages{english,swedish}
133   \else
134     \ifthenelse{\equal{\uef@language}{english}}{%
135       \setdefaultlanguage{english}%
136     }{%
```

```
137        \setdefaultlanguage[british]{english}%
138      }
139      \setotherlanguages{finnish,swedish}
140   \fi % \ifuef@finnish
141 \else % using babel
142   \RequirePackage[main=\uef@language,swedish,finnish,english,british]{babel}[2013/04/07]
143 \fi % \ifuef@polyglossia
```

Load the `microtype` package for better typography. Use option 'final' to apply microtype even when the class is given 'draft' option and use option 'babel' to turn on babel integration.

```
144 \RequirePackage[final,babel]{microtype}
```

4.3   Headers, Matters, and Appendices

First we need to redefine the chapter header style to write everything in one row. The standard `book` class calls `\@makechapterhead` to actually print the chapterhead, so we modify that command

```
145 \def\@makechapterhead#1{%
146   \vspace*{50\p@}% some empty space
147   {\parindent \z@ \raggedright \normalfont
148     \interlinepenalty \@M
149     \Huge \bfseries \thechapter. \hspace{1ex} #1\par\nobreak
150     \vskip 40\p@
151   }%
152 }
```

Then we set the page headers using the `fancyhdr` package. In one-sided documents, we put the page number at the middle of the page, but in two-sided documents, we put it to the outer footer.

```
153 \RequirePackage{fancyhdr}
154 \ifuef@twoside
155   \fancypagestyle{plain}{%
156     \fancyhf{}% Clear all
157     \fancyfoot[RO,LE]{\thepage}%
158   }
159 \else
160   \fancypagestyle{plain}{%
161     \fancyhf{}% Clear all
162     \fancyfoot[C]{\thepage}%
163   }
164 \fi
```

And we also make sure that the empty pages are truly empty, and remove the rule between the header and body.

```
165 \fancypagestyle{empty}{\fancyhf{}}
166 \renewcommand{\headrulewidth}{0pt}
```

Front, Main, and Back Matter

\frontmatter First redefine `\frontmatter` to have roman numbering and plain pagestyle.

```
167 \renewcommand{\frontmatter}{%
168    \cleardoublepage
169    \pagenumbering{roman}
170    \pagestyle{plain}
```

acknowledgements preface
Frontmatter has the special environments `acknowledgements` and `preface`. These are just unnumbered chapters with pre-defined names.

```
171    \newenvironment{preface}{\chapter*{\uef@s@prefacename}}{}
172    \newenvironment{acknowledgements}{\chapter*{\uef@s@acknowledgementsname}}{}
173 }
```

The names of acknowledgements and preface sections are defined for Finnish, English, and British.

```
174 \addto\captionsfinnish{\def\uef@s@prefacename{Esipuhe}}
175 \addto\captionsenglish{\def\uef@s@prefacename{Preface}}
176 \addto\captionsbritish{\def\uef@s@prefacename{Preface}}
177 \addto\captionsfinnish{\def\uef@s@acknowledgementsname{Kiitokset}}
178 \addto\captionsenglish{\def\uef@s@acknowledgementsname{Acknowledgments}}
179 \addto\captionsbritish{\def\uef@s@acknowledgementsname{Acknowledgements}}
```

\mainmatter Next, the main matter. We use arabic numbers and plain page styles. We also print error if somebody tries to use `\preface` or `\acknowledgements` after `\mainmatter`.

```
180 \renewcommand{\mainmatter}{%
181    \cleardoublepage
182    \pagenumbering{arabic}
183    \pagestyle{plain}
184    \def\preface{\ClassError{uefcsthesis}%
185      {Environment 'preface' can only be used in preface}%
186      {You must put the environment 'preface' before the \MessageBreak
187        \noexpand\mainmatter command.}}
188    \def\acknowledgements{\ClassError{uefcsthesis}%
189      {Environment 'acknowledgements' can only be used in preface}%
190      {You must put the environment 'acknowledgements' before the \MessageBreak
191        \noexpand\mainmatter command.}}
192 }
```

\backmatter LastNormalPage
The `\backmatter` macro is essentially a noop for page style, but it has the important task of defining the label `LastNormalPage`, which will be used in the abstract page to count the number of pages excluding the appendices.

```
193 \renewcommand{\backmatter}{%
194    \label{LastNormalPage}
195    \cleardoublepage
196    \pagestyle{plain}
197 }
```

appendices The appendices are written inside the `appendices` environment that is provided by the `appendix` package. We use `titletoc` option to put the appendices to the table of contents. We also provide the Finnish translation for the strings and re-define the English string for the name of the list of appendices.

```
198 \RequirePackage[titletoc,]{appendix}
199 \ifuef@finnish
200   \renewcommand{\appendixname}{Liite}
201   \renewcommand{\appendixtocname}{Liitteet}
202   \renewcommand{\appendixpagename}{Liitteet}
203 \else
204   \renewcommand{\appendixtocname}{List of appendices}
205 \fi
```

4.4   The Bibliography

We load either `apacite` or `biblatex`, depending on what the user asked with the options. For `biblatex`, we use option `natbib=true` to activate the `natbib`-compatible citation command (`\citet`, `\citep`, etc.), `sorting=nyt` to sort using name–year–title ordering, `sortcites=true` to sort the citations inside one `\cite`-type command, and as a style we use the APA style from `biblatex-apa`. Notice that we need to have the translation of the strings available in file `finnish-apa.lbx`. This file is currently shipped separately with `uefcsthesis.cls`.

FIXME: Does not honor the `numeric` option. FIXME: The `authoryear-comp` might not be compatible with the `natbib` style

```
206 \ifuef@biblatex
207   \RequirePackage[natbib=true,
208   sorting=nyt,
209   sortcites=true,
210   style=apa,
211   ]
212   {biblatex}
```

We then update the bibliography name strings

```
213   % Update language-specific strings
214   \DefineBibliographyStrings{english}{%
215     bibliography = {References},
216     references = {References},
217   }
218   \DefineBibliographyStrings{finnish}{%
219     bibliography = {Viitteet},
220     references = {Viitteet},
221   }
```

For `apacite`, we give the `natbibapa` option in order to load the `natbib` package, too.

```
222 \else
223   \RequirePackage[natbibapa]{apacite}
224   \bibliographystyle{apacite}
225   \addto\captionsfinnish{
```

23

```
226    \renewcommand{\bibname}{Viitteet}
227  }
228  \addto\captionsenglish{
229    \renewcommand{\bibname}{References}
230  }
```

To translate the punctuation correct in Finnish, we must change the `\BCBT` and `\BCBL` that define the comma between two authors and comma between the penultimate and last author, if there's more than two authors, to nothing.

```
231  \addto\captionsfinnish{
232    \renewcommand{\BCBT}{}
233    \renewcommand{\BCBL}{}
234  }
235 \fi % Using BibTeX
```

The `tocbibind` package adds references and other lists of to table of contents Only works with BibTeX; with BibLATeX one has to give option `heading=bibintoc` to `\printbibliography`

```
236 \RequirePackage[nottoc]{tocbibind}
```

4.5   Title and Abstract Pages and Multilingual Support

The macros in this section need to support multi-lingual interface. For consistency, this is done using an optional argument, so that `\title{X}` sets the title in the default language, and `\title[finnish]{X}` sets the title in Finnish (be that the default or not). To implement the optional arguments in a clean way, we use the `xparse` package.

```
237 \RequirePackage{xparse}
```

In principle, `xparse` could cause compatibility issues with older LATeX engines. If that is the case, we could use the following `\ifemptyarg` helper macro `\ifemptyarg`.[3]

```
 \def\ifemptyarg#1{%
   \if\relax\detokenize{#1}\relax % H. Oberdiek
     \expandafter\@firstoftwo
   \else
     \expandafter\@secondoftwo
   \fi%
 }
```

The usage of this macro is explained later with the `\setstring` macro.

Macros to Set and Store Thesis Metadata

`\author`
`\@author@first`
`\@author@family`
`\@author`

Author's name is not multilingual, though, so we don't support any language option here. But we do need to separate author's given and family names; re-define

---

[3]https://tex.stackexchange.com/questions/308/different-command-definitions-with-and-without-optional-argument

`\author` to take two arguments and store them in different macros. We also store the full name in GivenName FamilyName order to the standard `\@author`.

```
238 \gdef\@author@first{}
239 \gdef\@author@family{}
240 \renewcommand{\author}[2]{\gdef\@author@first{#1}%
241   \gdef\@author@family{#2}%
242   \gdef\@author{#1\space #2}%
243 }
```

`\uef@supervisor@first`
`\uef@supervisor@second`

The first supervisor's name is stored in `\uef@supervisor@first`, and we issue a warning if it's not defined.

```
244 \def\uef@supervisor@first{\ClassWarning{uefcsthesis}{First supervisor is not defined}}
```

The second supervisor's name can be left empty, and `\uef@supervisor@second` is only defined if the document calls `\secondsupervisor`.

`\firstsupervisor`
`\secondsupervisor`

Notice that it's possible to call `\secondsupervisor` without calling `\firstsupervisor`, but this is obviously not supported (what would it even mean?)

```
245 \newcommand{\firstsupervisor}[1]{\def\uef@supervisor@first{#1}}
246 \newcommand{\secondsupervisor}[1]{\def\uef@supervisor@second{#1}}
```

`\title`
`\@title`

The title of the thesis is set with the standard `\title` macro, and we store it's value to the standard `\@title` macro. But we store it in a language-dependant way, so that activating different languages activates the different titles. The optional language parameter has to be saved as a macro so that it's expandable even if it's just a string, like `finnish`. By default, we use the document's main language from `\uef@language`, set by class options.

```
247 \DeclareDocumentCommand{\title}{ O{\uef@language} m }{
248   \def\lang{#1}
249   \expandafter\addto\csname captions\lang\endcsname{\def\@title{#2}}
250 }
```

`\subtitle`
`\@subtitle`

The (optional) subtitles are handled the same way as the titles. We collect it separately so that we can alter the layout for it and store it in metadata with proper punctuation.

```
251 \DeclareDocumentCommand{\subtitle}{ O{\uef@language} m }{
252   \def\lang{#1}
253   \expandafter\addto\csname captions\lang\endcsname{\def\@subtitle{#2}}
254 }
```

`\date`
`\@date`

The date of the thesis is stored with the standard `\date` macro, but this is also enhanced to take the optional language parameter.

```
255 \DeclareDocumentCommand{\date}{ O{\uef@language} m }{
256   \def\lang{#1}
257   \expandafter\addto\csname captions\lang\endcsname{\def\@date{#2}}
258 }
```

25

\today   The default format for the date has only the month and the year, so we re-define \today to print just these. These are again language-dependant. FIXME: The definition is something like below, but with month names as macros. Check if babel defines macros for them.

```
\renewcommand{\today}{%
    \ifcase \month \or January\or February\or March\or %
    April\or May\or June\or July\or August\or September\or %
    October\or November\or December\fi \number \year}
```

\keywords   The keywords are added the same way as titles, but we store them twice.
\uef@keywords   \uef@keywords stores the keywords in the current language, while \uef@all@keywords
\uef@all@keywords stores all keywords given to the \keywords macro. This way we can list both Finnish and English keywords, for example. As a consequence, the only way to empty \uef@all@keywords is to redefine it to empty.

```
259 \def\uef@keywords{}
260 \def\uef@all@keywords{}
```

\sep   The keywords must be separated with the \sep command. This way they stay compliant with the pdfx package, as it assumes keywords to be separated with \sep. We will re-define \sep to comma for the abstract page, but for other use, we'll use semicolon. pdfx will redefine \sep for its own needs.

```
261 \providecommand{\sep}{\leavevmode\unskip ;\space}
```

The actual implementation of \keywords is similar to other metadata macros.

```
262 \DeclareDocumentCommand{\keywords}{ O{\uef@language} m }{
263   \def\lang{#1}
264   \expandafter\addto\csname captions\lang\endcsname{\def\uef@keywords{#2}}
265   \let\@tempa\uef@all@keywords
266   \def\uef@all@keywords{\@tempa \sep \uef@keywords}
267 }
```

\city   The \city macro is used to set the city of the thesis. This should be either Kuopio
\uef@city   or Joensuu (assuming UEF doesn't open new campuses). City is mandatory but easy to forget (unlike title or author name, say), so we complain if it's left empty.

```
268 \def\uef@city{\ClassError{uefcsthesis}%
269   {Thesis city is not defined}%
270   {You must define the city of the thesis (Joensuu or Kuopio)\MessageBreak
271     using \noexpand\city command}}
```

Also, both Kuopio and Joensuu are the same in Finnish and English (and in other Western European languages), so by default, we set the city name to all default languages (Finnish, English, and British). The optional argument allows for adding new languages or changing the city name for the supported languages.

```
272 \DeclareDocumentCommand{\city}{ o m }{
273   \IfNoValueTF {#1} {
274     \addto\captionsfinnish{\def\uef@city{#2}}
275     \addto\captionsenglish{\def\uef@city{#2}}
```

26

```
276      \addto\captionsbritish{\def\uef@city{#2}}
277   }
278   {
279      \def\lang{#1}
280      \expandafter\addto\csname captions\lang\endcsname{\def\uef@city{#2}}
281   }
282 }
```

FIXME: Make metadata macros print errors if used in document body
(\AtBeginDocument).

Pre-Defined Strings and a Macro to Change Them

\setstring   To support the different languages, all pre-defined strings printed are macros that
can be changed. The macro to change them is called \setstring. If the xparse
package causes issues, the \setstring macro (and the metadata macros above)
can be re-defined using the \ifemptyarg macro explained above. The non-xparse
version of \setstring could be as follows:

```
\newcommand{\setstring}[3][]{%
  \ifemptyarg{#1}%
  {\set@string{\uef@lang}{#2}{#3}}%
  {\set@string{#1}{#2}{#3}}%
}
```

The \setstring macro takes two arguments, the string and the value, and an
optional argument, the language (by default, we use the document's default lan-
guage). The indirection used below is actually not needed, but is left as is to make
the above example of non-xparse version work.

```
283 \DeclareDocumentCommand \setstring { o m m } {%
284   \IfNoValueTF {#1} {%
285      \set@string {\uef@language} {#2} {#3}%
286   }{%
287      \set@string {#1} {#2} {#3}%
288   }%
289 }
```

\set@string   The actual string-setting work is done by \set@string macro that must have the
language parameter. Again, we put the language in a macro to make strings work.

```
290 \newcommand{\set@string}[3]{%
291   \def\lang{#1}
```

All user-editable strings have prefix **uef@s@** to prevent the users to be able to use
\setstring to set other macros to potentially breaking ways. The string name
given to \setstring is the macro name without the **uef@s@** prefix.

```
292 \expandafter\addto\csname captions\lang \endcsname{%
293   \expandafter\renewcommand\csname uef@s@#2\endcsname{#3}%
294 }
295 }
```

27

We can now define the strings needed for the title and abstract pages. We will index them without the `uef@s@` prefix, that is, in the format given to `\setstring`.

thesistype | The thesis type is set based on the class options.

```
296 \ifuef@mscthesis
297   \addto\captionsfinnish{\def\uef@s@thesistype{Pro gradu -tutkielma}}
298   \addto\captionsenglish{\def\uef@s@thesistype{Master's thesis}}
299   \addto\captionsbritish{\def\uef@s@thesistype{Master's thesis}}
300 \else
301   \addto\captionsfinnish{\def\uef@s@thesistype{Kandidaatintutkielma}}
302   \addto\captionsenglish{\def\uef@s@thesistype{Bachelor's thesis}}
303   \addto\captionsbritish{\def\uef@s@thesistype{Bachelor's thesis}}
304 \fi
```

field | The field of the thesis is by default CS.

```
305 \addto\captionsfinnish{\def\uef@s@field{Tietojenkäsittelytiede}}
306 \addto\captionsenglish{\def\uef@s@field{Computer Science}}
307 \addto\captionsbritish{\def\uef@s@field{Computer Science}}
```

Currently, to change the thesis field, one has to use the `\setstring` macro. If different fields are needed, there could be a macro to change this (similar to the metadata macros, like `\title`). For slightly more changes, class option could be an option. If there's more changes than just a few strings, it could make more sense to create different class files using DocStrip's conditional extraction capabilities.

departmentname | The macros `departmentname`, `facultyname`, and `universityname` store the de-
facultyname | partment's, faculty's, and university's names. Probably mostly useful for theses
universityname | that need abstract pages in other languages than Finnish and English.

```
308 \addto\captionsfinnish{%
309   \def\uef@s@departmentname{Tietojenkäsittelytieteen laitos}}
310 \addto\captionsenglish{%
311   \def\uef@s@departmentname{School of Computing}}
312 \addto\captionsbritish{%
313   \def\uef@s@departmentname{School of Computing}}
314 \addto\captionsfinnish{%
315   \def\uef@s@facultyname{Luonnontieteiden ja metsätieteiden tiedekunta}}
316 \addto\captionsenglish{%
317   \def\uef@s@facultyname{Faculty of Science and Forestry}}
318 \addto\captionsbritish{%
319   \def\uef@s@facultyname{Faculty of Science and Forestry}}
320 \addto\captionsfinnish{%
321   \def\uef@s@universityname{Itä-Suomen yliopisto}}
322 \addto\captionsenglish{%
323   \def\uef@s@universityname{University of Eastern Finland}}
324 \addto\captionsbritish{%
325   \def\uef@s@universityname{University of Eastern Finland}}
```

pages | The ever so useful `pages` and `and` strings are needed in many places.
and |
```
326 \addto\captionsfinnish{\def\uef@s@pages{s.}}
```

327 `\addto\captionsenglish{\def\uef@s@pages{p.}}`
328 `\addto\captionsbritish{\def\uef@s@pages{p.}}`
329 `\addto\captionsfinnish{\def\uef@s@and{ja}}`
330 `\addto\captionsenglish{\def\uef@s@and{and}}`
331 `\addto\captionsbritish{\def\uef@s@and{and}}`

keywordsname   And the `keywordsname` string stores the name of the keywords.

332 `\addto\captionsfinnish{\def\uef@s@keywordsname{Avainsanat}}`
333 `\addto\captionsenglish{\def\uef@s@keywordsname{Keywords}}`
334 `\addto\captionsbritish{\def\uef@s@keywordsname{Keywords}}`

\supervisorname
\supervisorsname   There are different strings for theses with one or many supervisors. (There's no special support for languages with different plural form for two and three or more cases, but in such cases, one can just re-define `supervisorsname` to the correct case given the number of supervisors.)

335 `\addto\captionsfinnish{\def\uef@s@supervisorname{Ohjaaja}}`
336 `\addto\captionsenglish{\def\uef@s@supervisorname{Supervisor}}`
337 `\addto\captionsbritish{\def\uef@s@supervisorname{Supervisor}}`
338 `\addto\captionsfinnish{\def\uef@s@supervisorsname{Ohjaajat}}`
339 `\addto\captionsenglish{\def\uef@s@supervisorsname{Supervisors}}`
340 `\addto\captionsbritish{\def\uef@s@supervisorsname{Supervisors}}`

app@o
app@m   The singular and plural versions of the word 'appendix' are stored with the `@o` and `@m` suffices. Hence, to change them inside the main document, one has to use `\makeatletter` and `\makeatother`. This could perhaps be changed in the future.

341 `\addto\captionsfinnish{\def\uef@s@app@o{liite}\def\uef@s@app@m{liitettä}}`
342 `\addto\captionsenglish{\def\uef@s@app@o{appendix}\def\uef@s@app@m{appendices}}`
343 `\addto\captionsbritish{\def\uef@s@app@o{appendix}\def\uef@s@app@m{appendices}}`

\ccsname   The name of the ACM Computing Classification is given in a fixed string `\ccsname` with no way to change it. As the classification itself is always in English, this should suffice.

344 `\def\ccsname{ACM CCS (2012)}`

\uef@logo   The name of the file containing the university logo is not a string, and hence it's not prefixed with `uef@s@`. To change it, one has to redefine `\uef@logo`. To show the logo, we load the `graphicx` package.

345 `\def\uef@logo{uef_logo}`
346 `\RequirePackage{graphicx}`

Title and Abstract Pages

Final thing before we can define the title and abstract pages is to define a helper macro[4] `\uef@selectlanguage`.

---

[4] https://tex.stackexchange.com/questions/136659/how-to-pass-a-macro-as-the-argument-to-selectlanguage

`\uef@selectlanguage` This is used to solve the problem that `babel` assumes that the language parameter given to `\selectlanguage` is a language name (e.g. `\selectlanguage{\english}` is equivalent to `\selectlanguage{english}`). As we need to pass the language name in a macro, we need to make sure it gets expanded before calling `\selectlanguage`.

```
347 \ifuef@polyglossia
348 \let\uef@selectlanguage=\selectlanguage
```

`polyglossia` doesn't allow macros as language names, and hence will expand the parameter correctly.

```
349 \else
350 \newcommand{\uef@selectlanguage}[1]{%
351   \begingroup\edef\x{\endgroup
352     \noexpand\selectlanguage{#1}}\x
353 }
354 \fi
```

`\maketitle` The `\maketitle` macro will just set up the title page using the standard `titlepage` environment from the `book` class. We will turn off the page anchors for the title page. The `\hypersetup` command is guaranteed to exist.

```
355 \renewcommand{\maketitle}{%
356   \hypersetup{pageanchor=false}
357   \begin{titlepage}%
```

We make sure the language is the document's main language, otherwise the title page will look wrong.

```
358     \uef@selectlanguage{\uef@language}
```

For the title, we leave some space at the top, center everything, and set the title in `\LARGE` font and the subtitle, if it's given, in `\Large` font.

```
359     \null\vspace{0.5cm}
360     \begin{center}
361       {\LARGE \@title \par}
362       \ifdefined\@subtitle
363         {\Large \@subtitle \par}
364       \fi
```

There's 1.5 cm between the author and the title.

```
365       \vspace{1.5cm}
366       {\large \@author \par}
```

And thesis type, university logo, affiliation info, and date are pushed to the bottom of the page

```
367       \vfill
368       {\large \uef@s@thesistype\\
369         \vspace{1em}
370         \includegraphics[height=7cm]{\uef@logo}\\
371         \uef@s@departmentname\\
372         \uef@s@field\\
373         \@date \par}
```

```
374     \end{center}
375   \end{titlepage}%
376   \hypersetup{pageanchor=true}
377 }
```

To print the information on the abstract page, we define some helper macros.

\numberofappendices
\appendixpagecount
uef@appendixcounter
uef@appendixpages

Currently, the number of appendices (if any) and the number of pages they have must be specified manually. This option is good to have even if some future implementation allows automatically figuring this information from the LaTeX sources, as students might have to add appendices directly to the final PDF.

```
378 \newcounter{uef@appendixcounter}
379 \newcounter{uef@appendixpages}
380 \newcommand{\numberofappendices}[1]{\setcounter{uef@appendixcounter}{#1}}
381 \newcommand{\appendixpagecount}[1]{\setcounter{uef@appendixpages}{#1}}
```

\uef@printappendixpagenumbers

The number of appendices, and their total page count, is printed with the \uef@printappendixpagenumbers helper macro.

```
382 \def\uef@printappendixpagenumbers{%
383 \ifnum\value{uef@appendixcounter}>0
384   ,\space
385   \ifcase\value{uef@appendixcounter}\relax
```

The counter uef@appendixcounter cannot be zero, as we just checked against it. If it's one, use string \uef@s@app@o; otherwise, use string \uef@s@app@m.

```
386   \or%
387     \arabic{uef@appendixcounter}~\uef@s@app@o%
388   \else
389     \arabic{uef@appendixcounter}~\uef@s@app@m%
390   \fi
```

The total number of appendix pages comes in parentheses. We assume that the string \uef@s@pages doesn't have to differentiate between one and many pages.

```
391   \space (\arabic{uef@appendixpages}~\uef@s@pages)
392 \fi
393 }
```

\uef@printsupervisors

The name(s) of the supervisor are printed by \uef@printsupervisors. The macro checks whether there are one or two supervisors by testing if \uef@supervisor@second is undefined or not. To have more than two supervisors, all but the last supervisor should be listed in \uef@supervisor@first and the list should end with comma if thesis is in English and uses Oxford comma.

```
394 \newcommand{\uef@printsupervisors}%
395 {\@ifundefined{uef@supervisor@second}%
396   {\uef@s@supervisorname :\space \uef@supervisor@first}%
397   {\uef@s@supervisorsname :\space \uef@supervisor@first\space%
398     \uef@s@and\space\uef@supervisor@second}%
399 }
```

\ccsdesc
\@concepts

The ACM CCS 2012 web tool[5] generates LaTeX code with command \ccsdesc. The

---
[5] https://dl.acm.org/ccs/ccs.cfm

code to parse it, below, is copied directly from the `acmart.cls` file (2017/01/07, v1.28), by Boris Veytsman. It uses the `comment` package to make LaTeX to ignore the XML that is printed first. In the future, this script could add the classification terms to the PDF metadata. The parsed concepts with the correct typesetting is stored in the `\@concepts` macro, which we will use later in the `abstract` environment.

```
400 \RequirePackage{comment}
401 \excludecomment{CCSXML}
402 \let\@concepts\@empty
403 \newcommand\ccsdesc[2][100]{%
404   \ccsdesc@parse#1~#2~}
405 %%\RequirePackage{textcomp} % already included
406 \def\ccsdesc@parse#1~#2~#3~{%
407   \expandafter\ifx\csname CCS@#2\endcsname\relax
408     \expandafter\gdef\csname CCS@#2\endcsname{\textbullet\textbf{#2} \textrightarrow }%
409   \g@addto@macro{\@concepts}{\csname CCS@#2\endcsname}\fi
410   \expandafter\g@addto@macro\expandafter{\csname CCS@#2\endcsname}{%
411     \ifnum#1>499\textbf{#3; }\else
412     \ifnum#1>299\textit{#3; }\else
413   #3; \fi\fi}}
```

abstract   The abstract(s) are enclosed in the `abstract` environment that will print the whole abstract page. The environment takes the language of the page as an optional argument; if no argument is given, it uses the document default language.

```
414 \newenvironment{abstract}[1][\uef@language]{%
```

Like `\maketitle`, the `abstract` environment also uses the `titlepage` environment from the `book` class. We use the raw `\titlepage` and `\endtitlepage` commands so that the error messages report the `abstract` environment and not the (unbeknownst to the user) `titlepage` environment. Also similarly to `\maketitle`, we turn off the page anchors.

```
415   \hypersetup{pageanchor=false}
416   \titlepage
```

We set the language to the provided one (or the default), and set the spacing to single.

```
417   \uef@selectlanguage{#1}
418   \singlespacing
```

The university name is printed in all-caps. To make them look a bit better, we use large small caps. Alternatively, some kerning could be used to set the name properly in all-caps. Faculty and department info and thesis field follow.

```
419   \textsc{\large\MakeLowercase{\uef@s@universityname}},
420   \uef@s@facultyname , \uef@city\\
421   \uef@s@departmentname\\
422   \uef@s@field\\[2em]
```

In the abstract page, the author name is printed in FamilyName, GivenName order and the author name is separated from the title with a colon. If subtitle is given, it's separated from the main title with an en-dash and spaces.

```
423    \@author@family , \@author@first : \@title
424    \ifdefined\@subtitle\space -- \@subtitle\fi\\
```

We count the number of pages automatically from `\mainmatter` to `\backmatter` using `LastNormalPage` label. The appendix information is printed if needed.

```
425    \uef@s@thesistype ,\space \pageref{LastNormalPage}~\uef@s@pages %
426    \uef@printappendixpagenumbers\\
```

Finally, we print the supervisor(s) name(s) and date and begin the abstract with boldface text saying 'abstract' in the correct language.

```
427    \uef@printsupervisors\\
428    \@date \par
429    \paragraph{\abstractname :}
430 }%
```

At the end of the abstract page, we'll print the keywords with a header that is always printed, as the keywords are mandatory. If the CCS concepts are defined, we'll also print them. Finally, we'll close the `titlepage` environment.

```
431 {\vspace{2em}
432    \paragraph{\uef@s@keywordsname :} \uef@keywords\par
433    \ifx\@concepts\empty\else\bgroup
434      {\paragraph{\ccsname}\mbox{}\\ \@concepts\par }\egroup
435    \fi
436    \endtitlepage
437    \hypersetup{pageanchor=true}}
```

4.6   Commands for Mathematics

The main thing we need to do to support math is to provide the theorem-like environments in different languages and with different numbering options. Also, the class defines few conveniency macros for typesetting math properly.

Theorem-Like Environments

theoremname
lemmaname
corollaryname
propositionname
conjecturename
definitionname
problemname
examplename
remarkname
notename

We start by defining the theorem-like names in a way that supports `babel` and `polyglossia`. These have the same `uef@s@` prefix as with other strings, and they can be changed, and new languages can be added, with the `\setstring` macro.

```
438 \addto\captionsfinnish{\def\uef@s@theoremname{Lause}}
439 \addto\captionsenglish{\def\uef@s@theoremname{Theorem}}
440 \addto\captionsbritish{\def\uef@s@theoremname{Theorem}}
```

Here we use the term 'Lause' for Theorem in Finnish. Some authors prefer 'Teoreema', but this can be changed with `\setstring{theoremname}{Teoreema}` in Finnish theses.

```
441 \addto\captionsfinnish{\def\uef@s@lemmaname{Lemma}}
442 \addto\captionsenglish{\def\uef@s@lemmaname{Lemma}}
443 \addto\captionsbritish{\def\uef@s@lemmaname{Lemma}}
444 \addto\captionsfinnish{\def\uef@s@corollaryname{Korollaari}}
445 \addto\captionsenglish{\def\uef@s@corollaryname{Corollary}}
446 \addto\captionsbritish{\def\uef@s@corollaryname{Corollary}}
```

```
447 \addto\captionsfinnish{\def\uef@s@propositionname{Propositio}}
448 \addto\captionsenglish{\def\uef@s@propositionname{Proposition}}
449 \addto\captionsbritish{\def\uef@s@propositionname{Proposition}}
450 \addto\captionsfinnish{\def\uef@s@conjecturename{Konjektuuri}}
451 \addto\captionsenglish{\def\uef@s@conjecturename{Conjecture}}
452 \addto\captionsbritish{\def\uef@s@conjecturename{Conjecture}}
453 \addto\captionsfinnish{\def\uef@s@definitionname{Määritelmä}}
454 \addto\captionsenglish{\def\uef@s@definitionname{Definition}}
455 \addto\captionsbritish{\def\uef@s@definitionname{Definition}}
456 \addto\captionsfinnish{\def\uef@s@problemname{Ongelma}}
457 \addto\captionsenglish{\def\uef@s@problemname{Problem}}
458 \addto\captionsbritish{\def\uef@s@problemname{Problem}}
459 \addto\captionsfinnish{\def\uef@s@examplename{Esimerkki}}
460 \addto\captionsenglish{\def\uef@s@examplename{Example}}
461 \addto\captionsbritish{\def\uef@s@examplename{Example}}
462 \addto\captionsfinnish{\def\uef@s@remarkname{Huomautus}}
463 \addto\captionsenglish{\def\uef@s@remarkname{Remark}}
464 \addto\captionsbritish{\def\uef@s@remarkname{Remark}}
465 \addto\captionsfinnish{\def\uef@s@notename{Merkintä}}
466 \addto\captionsenglish{\def\uef@s@notename{Note}}
467 \addto\captionsbritish{\def\uef@s@notename{Note}}
```

The remaining definitions were straight forward.

Now we can define the theorem-like environments. This uses the `amsthm` package, included before the font was loaded. Theorems, Lemmas, Corollaries, Propositions, and Conjectures are in the 'plain' theorem style.

```
468 \theoremstyle{plain}
```

theorem   We will first define `theorem`, and set it to reset with the chapter counter or not, depending on the class options.

```
469 \ifuef@numbertheoremsbychapter
470   \newtheorem{theorem}{\uef@s@theoremname}[chapter]
471 \else
472   \newtheorem{theorem}{\uef@s@theoremname}
473 \fi
```

lemma   The remaining plain-style environments either follow `theorem`'s numbering or not,
corollary   depending on class options.
proposition
conjecture
```
474 \ifuef@singlemathnumber
475   \newtheorem{lemma}[theorem]{\uef@s@lemmaname}
476   \newtheorem{corollary}[theorem]{\uef@s@corollaryname}
477   \newtheorem{proposition}[theorem]{\uef@s@propositionname}
478   \newtheorem{conjecture}[theorem]{\uef@s@conjecturename}
479 \else
480   \newtheorem{lemma}{\uef@s@lemmaname}
481   \newtheorem{corollary}{\uef@s@corollaryname}
482   \newtheorem{proposition}{\uef@s@propositionname}
483   \newtheorem{conjecture}{\uef@s@conjecturename}
484 \fi
```

cor
prop
Some lazy authors don't want to write `corollary` or `proposition`, so we let the short versions `cor` and `prop` to be aliases by re-defining the low-level commands.

485 `\let\cor\corollary \let\endcor\endcorollary`
486 `\let\prop\proposition \let\endprop\endproposition`

Environments for definitions, problems, and examples use the 'definition' theorem style.

487 `\theoremstyle{definition}`

definition
problem
example
Again, the numbers either follow `chapter` or not, depending on the class options.

488 `\ifuef@numbertheoremsbychapter`
489 `  \newtheorem{definition}{\uef@s@definitionname}[chapter]`
490 `  \newtheorem{problem}{\uef@s@problemname}[chapter]`
491 `  \newtheorem{example}{\uef@s@examplename}[chapter]`
492 `\else`
493 `  \newtheorem{definition}{\uef@s@definitionname}`
494 `  \newtheorem{problem}{\uef@s@problemname}`
495 `  \newtheorem{example}{\uef@s@examplename}`
496 `\fi%`

remark
note
Finally, `remark` and `note` use the 'remark' style and have no numbers.

497 `\theoremstyle{remark}`
498 `\newtheorem*{remark}{\uef@s@remarkname}`
499 `\newtheorem*{note}{\uef@s@notename}`

By default, equation numbers are of form (ch.eq), where ch is the chapter number and eq is the equation number within the chapter. This is the correct use when numbering theorems by chapter and we don't have to do anything.

500 `\ifuef@numbertheoremsbychapter%`
501 `\else`

If the user has requested the numbering to not reset with chapters (explicitly or implicitly because this is a Bachelor's thesis), we have to turn that feature off from equations, as well. To do that, we use the `chngcntr` packages.

502 `  \RequirePackage{chngcntr}`
503 `  \counterwithout{equation}{chapter}`
504 `\fi`

Conveniency Macros

The `amsmath` package defines the vertical-bar-as-parenthesis operators `\lvert` and `\rvert` and their double-bar versions `\lVert` and `\rVert`.

\abs
\norm
We use the `mathtools` package to provide the conveniency macros `\abs` and `\norm`

505 `\RequirePackage{mathtools}`
506 `\DeclarePairedDelimiter\abs{\lvert}{\rvert}`
507 `\DeclarePairedDelimiter\norm{\lVert}{\rVert}`

The ISO 31-11:1992 standard defines that open and half-open intervals can be typeset either as (a, b), $(a, b]$, and $[a, b)$ or as $]a, b[$, $]a, b]$, and $[a, b[$. To support the latter notation with correct spacing (note that the spacing here is *not* correct), we have to declare ']' and '[' as left and right parenthesis symbols, respectively.

`\lOpen`
`\rOpen`
We store the symbols as math delimiters `\lOpen` and `\rOpen`.

```
508 \DeclareMathDelimiter{\lOpen}{\mathopen}{operators}{93}{largesymbols}{3}
509 \DeclareMathDelimiter{\rOpen}{\mathclose}{operators}{91}{largesymbols}{2}
```

`\lointerval`
`\rointerval`
`\ointerval`
To facilitate the use of the 'inverted brackets' notation, we provide macros for left-open, right-open, and open intervals as `\lointerval`, `\rointerval`, and `\ointerval`, respectively.

```
510 \DeclarePairedDelimiter\lointerval{\lOpen}{\rbrack}
511 \DeclarePairedDelimiter\rointerval{\lbrack}{\rOpen}
512 \DeclarePairedDelimiter\ointerval{\lOpen}{\rOpen}
```

4.7   Support for Other Packages

The class also supports some other often-used packages. Some of them are pre-loaded (usually to encourage their use), and for others, the support usually means translated strings.

Captions and Tables

We load the `caption` package to customize the outlook of the captions. Namely, the figure captions should be at the bottom, table captions should at the top, and the label (e.g. 'Figure 1' should be set in bold.

```
513 \RequirePackage[%
514   figureposition=bottom,%
515   tableposition=top,%
516   labelfont=bf,%
517 ]{caption}
```

For tables, we load the `booktabs` package , because nobody should make any tables without it,

```
518 \RequirePackage{booktabs}
```

and the `tabularx` package that allows for stretchable p-type columns denoted X.

```
519 \RequirePackage{array}
520 \RequirePackage{tabularx}
```

The `tabularx` package loads the `array` package, but we load it explicitly to indicate that we use its `\newcolumntype` to define centered, left-aligned, and right-aligned math columns C, L, and R.

```
521 \newcolumntype{C}{>{$}c<{$}}
522 \newcolumntype{L}{>{$}l<{$}}
523 \newcolumntype{R}{>{$}r<{$}}
```

Typesetting Algorithms

The class supports the two most popular pseudocode packages, `algorithm2e` and `algorithmicx`, as well as the `listings` package. We don't load them, but we define the necessary strings in `babel` and `polyglossia`.

In `algorithm2e`, the two strings we define are `\algorithmcfname` (the word 'Algorithm' in the label) and `\litalgorithmcfname` (the title for the list of algorithms, if used).

```
524 \addto\captionsfinnish{\def\algorithmcfname{Algoritmi}}
525 \addto\captionsenglish{\def\algorithmcfname{Algorithm}}
526 \addto\captionsbritish{\def\algorithmcfname{Algorithm}}
527 \addto\captionsfinnish{\def\listalgorithmcfname{Algoritmiluettelo}}
528 \addto\captionsenglish{\def\listalgorithmcfname{List of Algorithms}}
529 \addto\captionsbritish{\def\listalgorithmcfname{List of Algorithms}}
```

The `algorithmicx` package uses the `algorithms` bundle to generate the floating `algorithm` environment. The `algorithms` bundle again uses the `float` package, which allows us to change the name of the float with `\floatname{`⟨*floatenv*⟩`}{`⟨*name*⟩`}` command. To avoid having to check if `float` is loaded, we provide a dummy command:

```
530 \providecommand{\floatname}[2]{}
```

The name of the floating environment is `algorithm`, which we re-define using the `\floatname`. The 'list of algorithms' name is `\listalgorithmname`.

```
531 \addto\captionsfinnish{\floatname{algorithm}{Algoritmi}}
532 \addto\captionsengligh{\floatname{algorithm}{Algorithm}}
533 \addto\captionsbritish{\floatname{algorithm}{Algorithm}}
534 \addto\captionsfinnish{\def\listalgorithmname{Algoritmiluettelo}}
535 \addto\captionsenglish{\def\listalgorithmname{List of Algorithms}}
536 \addto\captionsbritish{\def\listalgorithmname{List of Algorithms}}
```

In `listings` (and `listingsutf8`), the two strings we define are `\lstlistingname` and `\lstlistlistingname`.

```
537 \addto\captionsfinnish{\def\lstlistingname{Listaus}}
538 \addto\captionsenglish{\def\lstlistingname{Listing}}
539 \addto\captionsbritish{\def\lstlistingname{Listing}}
540 \addto\captionsfinnish{\def\lstlistlistingname{Listausluettelo}}
541 \addto\captionsenglish{\def\lstlistingname{List of Listings}}
542 \addto\captionsbritish{\def\lstlistingname{List of Listings}}
```

4.8 PDF/A and Hyperlink Support

The PDF metadata is collected at the end of the preamble, so that the user has had time to fill it in. We use the `etoolbox` package for the `\AtEndPreamble`. The metadata commands must be defined before loading `pdfx` or `hyperref`. The correctly-formatted metadata is stored in special macros that start `uef@pdf@`.

```
543 \RequirePackage{etoolbox}
544 \AtEndPreamble{
```

`\uef@pdf@title` We separate the subtitle, if given, with standard hyphen in metadata.

```
545 \ifdefined\@subtitle
546   \def\uef@pdf@title{\@title{} - \@subtitle}
547 \else
548   \def\uef@pdf@title{\@title}
549 \fi
```

`\uef@pdf@author` Author's name should always be defined, but just in case, we check for it, and store the name in Family, First order.

```
550 \ifdefined\@author@first
551   \def\uef@pdf@author{\@author@family , \@author@first}
552 \else
553   \def\uef@pdf@author{}
554 \fi
```

`\uef@pdf@subject` The PDF subject metadata field is somewhat vague. We store the thesis type, it's field, and the name of the university and the faculty. We could also store the CCS classification, but currently we don't do that.

```
555 \def\uef@pdf@subject{\uef@s@thesistype, %
556   \uef@s@field. \uef@s@universityname, %
557   \uef@s@facultyname}
558 }
```

Sample `.xpmdata` File

The package reads XMP metadata from a file called `\jobname.xmpdata`. For most use cases of `uefcsthesis`, this file can simply direct to the above-defined macros.

```
559 ⟨/cs⟩
560 ⟨*xmp⟩
561 \Title{\uef@pdf@title}
562 \Author{\@author}
```

The `pdfx` package puts the contents of the `\Keywords` macro in `XMP-dc:Subject` field, and the contents of the `\Subject` macro in `XMP-dc:Description` field.[6]

```
563 \Keywords{\uef@keywords}
564 \Subject{\uef@pdf@subject}
```

As the publisher, we use just the university, as is common with theses.

```
565 \Publisher{\uef@s@universityname}
566 ⟨/xmp⟩
567 ⟨*cs⟩
```

If the author has used complicated macros (or math) in `\author`, `\title`, or `\keywords`, this system might fail. In such situations, the author should edit the `\jobname.xmpdata` file manually.

---

[6]For more information, see http://www.dublincore.org/documents/dces/.

Loading the Packages

If the user has requested the generation of the PDF/A file, we load the `pdfx` package at the end of the preamble. That way we know that we have the metadata defined before. `pdfx` requires file called `\jobname.xmpdata`, and our default file uses the above `uef@pdf@` strings to fill in the metadata. We set the type of the PDF as PDF/A-1b. Using the 2b version would be better, as it supports layered figures, but alas, the fonts are not entirely compatible (they do not define all CID's). Also, we will make sure the language is the correct one so that we get the right extensions of the macros.

The `colorprofiles` package is loaded by `pdfx`, but MiKTeX doesn't download the package unless we specifically ask for it and `pdfx` only loads it if it exists. We also set the minimum version of `pdfx` to rather recent, and set the `pdfstartview` option of `hyperref` to empty. For these, see also https://coursepages.uta.fi/mtta1-latex/wp-content/uploads/sites/55/2019/03/pdfa-guide.pdf.

```
568 \ifuef@pdfa
569   \AtEndPreamble{%
570     \uef@selectlanguage{\uef@language}%
571     \RequirePackage{colorprofiles}
572     \RequirePackage[a-1b,mathxmp]{pdfx}[2018/12/22]%
573     \hypersetup{pdfstartview=}
574   }
575 \fi
```

If user did not request PDF/A, we load `hyperref` (`pdfx` loads it otherwise). `hyperref` should always be loaded at the very end, and in case there are any problems, it's loading here can be prevented.

```
576 \ifuef@hyperref
577 \ifuef@pdfa\else
578   \RequirePackage{hyperref}
```

If loaded, `pdfx` will set the PDF metadata. Here, we set it using `hyperref` as `pdfx` hasn't been loaded.

```
579   \AtBeginDocument{
580   \let\oldsep\sep
581   \def\sep{,\space}
582   \hypersetup{pdfauthor={\uef@pdf@author},
583     pdftitle={\uef@pdf@title},
584     pdfkeywords={\uef@all@keywords},
585     pdfsubject={\uef@pdf@subject}
586   }
587   \let\sep\oldsep
588   }%
589 \fi\fi % \ifhyperref\ifpdfa
```

`\hypersetup`   We use the `\hypersetup` command in the title and abstract pages without checking that it exists, so to be safe, we provide a dummy command that does nothing if `\hyperref` is not loaded.

```
590 \providecommand{\hypersetup}[1]{}
```

4.9 Last Bit

The very last thing to do is to turn the pagestyle to empty and end with DocStrip.

```
591 \pagestyle{empty}
592 ⟨/cs⟩
```

4.10 Minimal Example Files

This section contains the sources for the minimal example files. We generate four files, for all combinations of

$$\{\texttt{classic}, \texttt{modern}\} \times \{\texttt{English}, \texttt{Finnish}\} .$$

These examples are all protected with the `ex` quard

```
593 ⟨*ex⟩
```

We primarily divide by language, starting with English.

```
594 ⟨*en⟩
595 %% This is a minimal example of using the uefcsthesis class.
596 %% This generates an English MSc thesis with one-sided layout.
```

The command to use to compile and the `\documentclass` command depend on the engine.

```
597 ⟨*modern⟩
598 %% To compile, use either lualatex or xelatex, for example,
599 %% $ lualatex minimal_modern.en.tex
600 %% $ biber minimal_modern.en
601 %% $ lualatex minimal_modern.en.tex
602 %% or use latexmk:
603 %% $ latexmk -lualatex minimal_modern.en.tex
604 %%
605 \documentclass[mscthesis,english,oneside,biblatex]{uefcsthesis}
606
607 %% Correct the below with the name of your bibliography file
608 \addbibresource{minimal.bib}
609 ⟨/modern⟩
610 ⟨*classic⟩
611 %% To compile, use pdflatex, for example,
612 %% $ pdflatex minimal_classic.en.tex
613 %% $ bibtex minimal_classic.en
614 %% $ pdflatex minimal_classic.en.tex
615 %% or use latexmk:
616 %% $ latexmk -pdf minimal_classic.en.tex
617 %%
618 \documentclass[mscthesis,english,oneside]{uefcsthesis}
619 ⟨/classic⟩
```

The metadata and main text are the same for classic and modern.

```
620
621 %% Replace all capital text with your own information.
```

```
622 \title{TITLE} % Title of the thesis
623 \author{GIVEN NAME}{FAMILY NAME} % Your name
624 \date{MONTH YEAR} % The month and year of handing in your thesis
625 \city{CITY} % Either Kuopio or Joensuu
626 \firstsupervisor{GIVEN FAMILY} % Name of the first supervisor
627 \secondsupervisor{GIVEN FAMILY} % Name of the second supervisor, if any
628 \keywords{KEYWORD1\sep KEYWORD2\sep ETC} % Keywords must be separated with \sep
629
630 %% To get the ACM CCS classification, you can visit
631 %% https://dl.acm.org/ccs/ccs.cfm
632 %% There you can find a tool to generate LaTeX code for the classification
633 %% Copy it here. You don't need to copy the XML at the begin, though.
634 %% For example,
635 %% \ccsdesc[500]{Some Class}
636
637 \begin{document}
638 \maketitle
639 \begin{abstract}
640   WRITE YOUR ENGLISH ABSTRACT HERE
641 \end{abstract}
642
643 \frontmatter
644 \tableofcontents
645 \mainmatter
646
647 \chapter{Introduction}
648 \label{cha:intro}
649
650 WRITE YOUR INTRODUCTION HERE
651
652 WRITE THE REST OF THE THESIS HERE
653
654 THIS IS AN EXAMPLE OF USING CITATIONS:
655 Graph generators are important \citep{metzler18random}.
656 \citet{kalofolias18from} discuss sets of redescriptions.
657
658 %% Next comes the references
```

The way the bibliography is included again depends on the engine.

```
659 ⟨modern⟩\printbibliography[heading=bibintoc]
660 ⟨*classic⟩
661 %% Update the following with the name of your bibliography file
662 \bibliography{minimal}
663 ⟨/classic⟩
```

The footer is again shared.

```
664
665 \backmatter % Do not remove!
666 %% Possible appendices come here
667 \end{document}
668 ⟨/en⟩
```

Next, we'll generate the Finnish example files. The structure is the same as with the english ones.

```
669 ⟨*fi⟩
670 %% Tämä on yksinkertainen esimerkki uefcsthesis-luokan käytöstä.
671 %% Tämä tiedosto tuottaa pro gradu -tutkielman yksipuoleisella asettelulla.
672 ⟨*modern⟩
673 %% Tuottaaksesi PDF-tiedoston, käytä joko lualatex- tai xelatex-ohjelmaa.
674 %% Esimerkiksi:
675 %% $ lualatex minimal_modern.fi.tex
676 %% $ biber minimal_modern.fi
677 %% $ lualatex minimal_modern.fi.tex
678 %% Vaihtoehtoisesti voit käyttää latexmk-ohjelmaa:
679 %% $ latexmk -lualatex minimal_modern.fi.tex
680 %%
681 \documentclass[mscthesis,finnish,oneside,biblatex]{uefcsthesis}
682
683 %% Korvaa seuraavasta minimal.bib lähdeviitetietokantatiedostosi nimellä.
684 \addbibresource{minimal.bib}
685 ⟨/modern⟩
686 ⟨*classic⟩
687 %% Tuottaaksesi PDF-tiedoston, käytä pdflatex-ohjelmaa.
688 %% Esimerkiksi:
689 %% $ pdflatex minimal_classic.fi.tex
690 %% $ bibtex minimal_classic.fi
691 %% $ pdflatex minimal_classic.fi.tex
692 %% Vaihtoehtoisesti voit käyttää latexmk-ohjelmaa:
693 %% $ latexmk -pdf minimal_classic.fi.tex
694 %%
695 \documentclass[mscthesis,finnish,oneside]{uefcsthesis}
696 ⟨/classic⟩
697
698 %% Korvaa isolla kirjoitetut tekstit omilla tiedoillasi.
699 \title{OTSIKKO} % Työsi otsikko
700 \author{ETUNIMI}{SUKUNIMI} % Nimesi
701 \date{KUUKAUSI VUOSI} % Työsi valmistumiskuukausi ja -vuosi
702 \city{KAUPUNKI} % Joko Kuopio tai Joensuu
703 \firstsupervisor{ETUNIMI SUKUNIMI} % Ensimmäisen ohjaajan nimi
704 \secondsupervisor{ETUNIMI SUKUNIMI} % Toisen ohjaajan, jos on, nimi
705 \keywords{AVAINSANA1\sep AVAINSANA2\sep JNE} % Avainsanat erotetaan \sep-komennolla
706
707 %% ACM:n CCS-luokittelun LaTeX-komennot saa luotua ACM:n työkalulla osoitteessa
708 %% https://dl.acm.org/ccs/ccs.cfm
709 %% Kopioi työkalun tuottama LaTeX-koodi tähän (alun XML-koodia ei tarvitse
710 %% kopioida). Esimerkiksi:
711 %% \ccsdesc[500]{Some Class}
712
713 \begin{document}
714 \maketitle
715 \begin{abstract}
```

```
716    KIRJOITA SUOMENKIELINEN TIIVISTELMÄSI TÄHÄN
717 \end{abstract}
718
719 \frontmatter
720 \tableofcontents
721 \mainmatter
722
723 \chapter{Johdanto}
724 \label{cha:johdanto}
725
726 KIRJOITA JOHDANTOKAPPALEESI TÄHÄN
727
728 KIRJOITA LOPUT TYÖSTÄSI TÄHÄN
729
730 TÄMÄ ON ESIMERKKI VIITTAUSTEN KÄYTÖSTÄ:
731 Verkkogeneraattorit ovat hyödyllisiä \citep{metzler18random}.
732 \citet{kalofolias18from} etsivät jälleenkuvausten joukkoja.
733
734 %% Seuraavaksi tulee viiteluettelo
735 ⟨modern⟩\printbibliography[heading=bibintoc]
736 ⟨*classic⟩
737 %% Korvaa alta minimal oman lähdetietokantatiedostosi nimellä
738 \bibliography{minimal}
739 ⟨/classic⟩
740
741 \backmatter % Älä poista!
742 %% Mahdolliset liitteet tulevat tähän
743 \end{document}
744 ⟨/fi⟩
```

The minimal bib-file has only two records.

```
745 ⟨*bib & ex⟩
746 @incollection{metzler18random,
747   author = {Metzler, Saskia and Miettinen, Pauli},
748   title = {Random Graph Generators for Hyperbolic Community Structures},
749   booktitle = {{Proc. 7th International Conference on Complex Networks and Their Applications (
750   editor = {Aiello, L. and Cherifi, C. and Cherifi, H. and Lambiotte, R. and Li{\'o}, P. and Ro
751   publisher = {Springer},
752   address = {Cham},
753   year = {2018},
754   pages = {680-693},
755   doi = {10.1007/978-3-030-05411-3_54},
756 }
757 @article{kalofolias18from,
758   author = {Kalofolias, Janis and Galbrun, Esther and Miettinen, Pauli},
759   title = {From sets of good redescriptions to good sets of redescriptions},
760   journal = {Knowl. Inf. Syst.},
761   volume = 57,
762   number = 1,
763   pages = {21--54},
```

```
764    year = 2018,
765    doi = {10.1007/s10115-017-1149-7},
766 }
767 ⟨/bib & ex⟩
```

And that finishes the minimal example files.

```
768 ⟨/ex⟩
```

4.11   Bibliography Records

Here are the bibliography records for the manual. We don't want to have these in
the minimal examples, hence we require not to have the ex tag.

```
769 ⟨*bib&!ex⟩
770 @misc{graphicx,
771    title = {Packages in the 'graphics' bundle},
772    author = {Carlisle, D. P.},
773    year = 2017,
774    url = {https://ctan.org/pkg/latex-graphics},
775    lastchecked = {6 December 2018},
776 }
777 @misc{urlbst,
778    title = {The {\textsf{urlbst}} package},
779    author = {Gray, Norman},
780    year = 2011,
781    url = {https://ctan.org/pkg/urlbst},
782    lastchecked = {6 December 2018},
783 }
784 @misc{polyglossia,
785    title= {Polyglossia: An Alternative to {Babel} for \hologo{XeLaTeX} and \hologo{LuaLaTeX}},
786    author = {Charette, Fran{\c{c}}ois and Reutenauer, Arthur},
787    year = 2018,
788    url = {https://ctan.org/pkg/polyglossia},
789    lastchecked = {9 December 2018},
790 }
791 @misc{pdfx,
792    title = {Generation of {PDF/X}- and {PDF/A}-compliant {PDFs} with \hologo{pdfTeX} -- \textsf-
793    author = {Radhakrishnan, C. V. and Thành, Hàn Th{\'{\^e}} and Moore, Ross and Selinger, Peter
794    year = 2017,
795    url = {https://ctan.org/pkg/pdfx},
796    lastchecked = {9 December 2018},
797 }
798 @misc{amsthm,
799    title = {Using the \textsf{amsthm} Package},
800    author = {{Publications Technical Group, American Mathematical Society}},
801    year = 2017,
802    url = {https://ctan.org/pkg/amsthm},
803    lastchecked = {9 December 2018},
804 }
805 @misc{array,
806    title = {A new implementation of \hologo{LaTeX}'s \textsf{tabular} and \textsf{array} environ-
```

807   author = {Mittelbach, Frank and Carlisle, David},
808   year = 2018,
809   url = {https://ctan.org/pkg/array},
810   lastchecked = {9 December 2018},
811 }
812 @misc{tabularx,
813   title = {The \textsf{tabularx} package},
814   author = {Carlisle, David},
815   year = 2016,
816   url = {https://ctan.org/pkg/tabularx},
817   lastchecked = {9 December 2018},
818 }
819 @misc{siunitx,
820   title = {\textsf{siunitx} --- A comprehensive (SI) units package},
821   author = {Wright, Joseph},
822   year= 2018,
823   url = {https://ctan.org/pkg/siunitx},
824   lastchecked = {9 December 2018},
825 }
826 @misc{bm,
827   title = {The \textsf{bm} package},
828   author = {Carlisle, David and Mittelbach, Frank},
829   year = 2017,
830   url = {https://ctan.org/pkg/bm},
831   lastchecked = {10 December 2018},
832 }
833 @misc{hyperref,
834   title = {Hypertext marks in \hologo{LaTeX}: a manual for \textsf{hyperref}},
835   author = {Rahtz, Sebastian and Oberdiek, Heiko},
836   year = 2017,
837   url = {https://ctan.org/pkg/hyperref},
838   lastchecked = {10 December 2018},
839 }
840 @misc{biblatex-apa,
841   title = {{APA} {B}ib\hologo{LaTeX} style: Citation and References macros for {B}ib\hologo{La
842   author = {Kime, Philip},
843   year = 2018,
844   url = {https://ctan.org/pkg/biblatex-apa},
845   lastchecked = {23 December 2018},
846 }
847 @misc{apacite,
848   title = {The apacite package: Citation and reference list with \hologo{LaTeX} and \hologo{Bib
849   author = {Meijer, Erik},
850   year = 2013,
851   url = {https://ctan.org/pkg/apacite},
852   lastchecked = {23 December 2018},
853 }
854 ⟨/bib&!ex⟩

45

References

[1] D. Carlisle. The tabularx package, 2016. URL: https://ctan.org/pkg/tabularx [cited 9 December 2018].

[2] D. Carlisle and F. Mittelbach. The bm package, 2017. URL: https://ctan.org/pkg/bm [cited 10 December 2018].

[3] D. P. Carlisle. Packages in the 'graphics' bundle, 2017. URL: https://ctan.org/pkg/latex-graphics [cited 6 December 2018].

[4] F. Charette and A. Reutenauer. Polyglossia: An alternative to Babel for X_ELATEX and LuaLATEX, 2018. URL: https://ctan.org/pkg/polyglossia [cited 9 December 2018].

[5] P. Kime. APA BibLATEX style: Citation and references macros for BibLATEX, 2018. URL: https://ctan.org/pkg/biblatex-apa [cited 23 December 2018].

[6] E. Meijer. The apacite package: Citation and reference list with LATEX and BIBTEX according to the rules of the American Psychological Association, 2013. URL: https://ctan.org/pkg/apacite [cited 23 December 2018].

[7] F. Mittelbach and D. Carlisle. A new implementation of LATEX's tabular and array environment, 2018. URL: https://ctan.org/pkg/array [cited 9 December 2018].

[8] Publications Technical Group, American Mathematical Society. Using the amsthm package, 2017. URL: https://ctan.org/pkg/amsthm [cited 9 December 2018].

[9] C. V. Radhakrishnan, H. T. Thành, R. Moore, and P. Selinger. Generation of PDF/X- and PDF/A-compliant PDFs with pdfTEX – pdfx.sty, 2017. URL: https://ctan.org/pkg/pdfx [cited 9 December 2018].

[10] S. Rahtz and H. Oberdiek. Hypertext marks in LATEX: a manual for hyperref, 2017. URL: https://ctan.org/pkg/hyperref [cited 10 December 2018].

[11] J. Wright. siunitx — a comprehensive (si) units package, 2018. URL: https://ctan.org/pkg/siunitx [cited 9 December 2018].

Change History

0.5.1

General: Do not use polyglossia in minimal modern examples . . . 40

0.5.2

General: Replace *et al.* with *ym.*

and print dates in *01.01.2021* format in Finnish bibLaTeX APA. 23

Index

For index terms written in typewriter font, the following convention apply: 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. For all other index terms, the numbers refer to the pages of this document.