Notes for the Summer 2020 University of Utah Styles for Mathematics Department Dissertations and Theses in LaTeX

Nelson H. F. Beebe
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
USA
Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org
WWW URL: http://www.math.utah.edu/~beebe
Telephone: +1 801 581 5254
FAX: +1 801 581 4148

15 June 2020

Contents

Foreword

This document accompanies a set of sample theses and style files for use in the University of Utah *Mathematics Department only*, during a test period that is expected to last a year or two, wherein the Department, rather than the University of Utah Thesis Office in the Graduate School,

has the authority to determine acceptable document formatting. The files are available from links at `http://www.math.utah.edu/pub/uumath-thesis`.

For now, students from other campus departments must use the previous version available at `http://www.math.utah.edu/pub/uuthesis`.

1   Introduction

A new design of the University of Utah support for typesetting dissertations and theses in LaTeX builds upon the standard `book` class, and makes available modern capabilities, including extended bibliography styles, microtypography, optional hypertext links in PDF files, running page headers, and optional single spacing of front matter and bibliographies.

The new design emphasizes minimal changes to internals of standard LaTeX commands, so that it should be possible to use almost any of the nearly 5000 styles available in recent TeX Live distributions.

Along with the new design, many of the previous formatting restrictions on dissertations and theses have been relaxed, so authors should be able to concentrate on good exposition and logical markup, without spending much time at all on tweaking line breaking, and figure and table placement.

Several of the default penalties and page fractions in LaTeX that control whether floats (usually, figures and tables) can go on the current page or not have been loosened to make it more likely that they appear close to their point of definition in your input document.

Provided that BiBTeX is used to generate formatted reference lists, they too should require little user attention, apart from care to ensure consistency in journal and publisher names, and to supply protective bracing of proper nouns, and German text, in titles and thesis types to prevent unwanted downcasing.

Extended bibliography styles are available that can be used to produce richer reference lists, including international document and publication identifiers, Library of Congress call numbers, and optionally, hyperlinked Web addresses in DOI and URL values.

2   Online resources

Writing a large document is not a simple task, and there are many fine points of typography and writing that may not be obvious to authors with little prior experience with LaTeX. There are four actively maintained Web documents that authors are urged to consult *before* beginning the writing task, and reference collection:

- *A Brief Introduction to TeX and LaTeX*
  `http://www.math.utah.edu/pub/tex/doc/latex-overview/index.html`
  `http://www.math.utah.edu/pub/tex/doc/latex-overview/index.pdf`

- *Typesetting and Writing Hints for Theses and Dissertations*
  http://www.math.utah.edu/pub/uuthesis/writing-tips.html
  http://www.math.utah.edu/pub/uuthesis/writing-tips.pdf

- *BIBTEX Information and Tutorial*
  http://www.math.utah.edu/pub/bibnet/bibtex-info.html
  http://www.math.utah.edu/pub/bibnet/bibtex-info.pdf

- *A Font Family Sampler*
  http://www.math.utah.edu/pub/uuthesis/font-sampler/font-sampler.pdf

Those documents, and several others, can be found at this site:

http://www.math.utah.edu/pub/tex/doc/

In particular, two newer documents there, *Typesetting multilingual verbatim text with pdflatex* and *Typesetting multilingual text with lualatex and xelatex*, contain more detail about handling additional writing systems, and mixing text with left-to-right and right-to-left writing directions.

3   A top-level LATEX file

Each of the sample thesis top-level LATEX files begins with an automatically produced section that shows just the LATEX commands, without the detailed comments, to give an overview of the structure of a LATEX document for dissertations and theses. Here is an even shorter example:

```
\documentclass[11pt]{book}
\usepackage {uuthesis-base}     % MANDATORY package for a minimal thesis
\usepackage {uuthesis-updates}  % MANDATORY and LAST of all uuthesis-* packages
\usepackage {mythesis}          % for personally defined macros
\author              {Jane Doe}
\title               {Integral Transforms in Quantum Physics}
\thesistype          {dissertation}
\dedication          {For my parents, Kim and Terry.}
\degree              {Doctor of Philosophy}
\approvaldepartment  {Mathematics}
\department          {Department of Mathematics}
\graduatedean        {Alice B. Toklas}
\departmentchair     {Petrus Marcus Aurelius Featherstone-Hough}
\committeechair      {Cornelius L{\'a}nczos}
\firstreader         {Hans Bethe}
\secondreader        {Niels Bohr}
\thirdreader         {Max Born}
```

```
\fourthreader          {Paul A. M. Dirac}
\chairtitle            {Professor}
\submitdate            {May 2020}
\copyrightyear         {2020}
\firstdateapproved     {21 February 2020}
\seconddateapproved    {22 February 2020}
\thirddateapproved     {23 February 2020}
\fourthdateapproved    {24 February 2020}
\begin{document}
    \titlepage                    % MANDATORY
    \copyrightpage                % optional, but strongly recommended
    \dissertationapproval         % MANDATORY
    \preface {abstract} {Abstract}  % optional, but strongly recommended
    \dedicationpage               % optional
    \tableofcontents              % MANDATORY
    \listoffigures                % MANDATORY (if there are figures)
    \listoftables                 % MANDATORY (if there are tables)
    \uumaintext    % Start normal page numbering: parts and chapters follow.
    \pagestyle {headings}
    \include {chap1}
    \include {chap2}
    \include {chap3}
    \include {chap4}
    \appendix      % Chapters, sections are now appendix style A, A.1, A.2, B, C, D, ...
    \include {appa}
    \include {appb}
    \bibliographystyle {plain}
    \bibliography {\jobname}
\end {document}
```

Notice that about half the contents of the top-level LATEX file are just routine declarations of values for various fields in the front matter that is unique to a University of Utah dissertation or thesis.

It is almost always a good idea to supply the bulk of the document in separate files, one for each chapter or appendix. That makes it easier to see the overall structure, and to rearrange the order of their inclusion if that later seems advisable. It also narrows the scope of an editing disaster.

At the University of Utah, Masters degree committees have only a chair, and two other members, and the degree document is called a *thesis*. Doctoral degree committees generally have a chair and four other members. Rarely, there might be two co-chairs, or a fifth member. A TEX command in the uuthesis-approval.sty file therefore tests whether a third reader has been defined. If not, the heading on the approval page contains the phrase **STATEMENT OF THESIS APPROVAL**. Otherwise, it says **STATEMENT OF DISSERTATION APPROVAL**.

For the case of a co-chair, code the input like this:

```
\committeechair{\mbox{\small Cornelius L{\'a}nczos and Arnold Sommerfeld}}
```

A smaller font is likely needed to get both names on the underlined field, and boxing the names prevents unwanted line breaks.

To handle larger committees, the style supports `\fifthreader` through `\ninthreader`, and corresponding `\fifthdateapproved` through `\ninthdateapproved`, declarations.

From Summer 2019, the `uumath-thesis` distribution has a new directory, `honors-thesis-0`, that shows how an undergraduate honors thesis might be typeset. Its markup is almost identical to that for masters and doctoral degrees, but its title page is quite different, because it also contains the committee approval. For such a thesis in the University of Utah Honors College, there is only one faculty supervisor, plus an honors faculty advisor for the department, along with the department chair, and the Dean of the Honors College (*not* that of the department's college).

4   Software tools

Three common errors are hard for humans to find, but easy for computers:

- spelling errors;

- doubled-word errors;

- delimiter-balance errors.

The Mathematics Department provides several spell checkers, including `aspell`, `hunspell`, `ispell`, `myspell`, and `spell`. They are best used in a mode where words that are not in their default dictionaries, but known to be correctly spelled, are accumulated in a private exception list, and supplied to the spell checker as an extra dictionary. That way, you only get a report once for each suspect word, and subsequent checks silently accept them.

The `emacs` text editor has a convenient command bound to the keystroke `M-$` (meta dollar) that checks the spelling of the word before the cursor. It also has `M-x flyspell-mode` to turn on dynamic checking, and `M-x flyspell-buffer` and `M-x flyspell-region` to run the checks on demand. Other editors may offer similar conveniences.

Automatic spelling *correction* by software, however, is generally inadvisable: technical documents have far too many unusual words that are likely to be badly mangled by such actions.

The `dw` utility can be used to find doubled words in a file; its source code is freely available at `http://www.math.utah.edu/pub/dw`. It is usually best used with a tool to strip LaTeX markup, like one of these:

```
% detex           < myfile.tex | dw
% detex -m -n -s < myfile.tex | dw
```

The second form shows additional command line options for the Department of Mathematics version that remedy some of the deficiencies of detex in TeX Live distributions.

The emacs text editor has the command M-x next-doubled-word to do a similar task in an editing session.

Tools for detecting mismatched delimiters, such as braces, brackets, and parentheses are not common, yet such errors are exceptionally difficult to spot during proofreading. The chkdelim utility is one such tool.[1] The emacs text editor has a command M-x show-paren-mode to toggle matching of delimiters; however, that is primarily helpful during text entry, and cannot be easily used to check an existing file for such errors.

The common mathematical notation for closed (*including* endpoint) and open (*excluding* endpoint) intervals

$$[a, b] \qquad \text{means values } x \text{ such that } a \leq x \leq b,$$
$$[a, b) \qquad \text{means values } x \text{ such that } a \leq x < b,$$
$$(a, b] \qquad \text{means values } x \text{ such that } a < x \leq b,$$
$$(a, b) \qquad \text{means values } x \text{ such that } a < x < b,$$

introduces delimiter mismatches in two of the four cases, cluttering the diagnostics from chkdelim with unwanted warnings. The easy solution is to avoid the mismatches entirely by defining suitable macros in your private style file:

```
\newcommand {\CC} [2] {{$ [#1, #2] $}}
\newcommand {\CO} [2] {{$ [#1, #2) $}}
\newcommand {\OC} [2] {{$ (#1, #2] $}}
\newcommand {\OO} [2] {{$ (#1, #2) $}}
```

We intentionally gave those macros two arguments, rather than just one, because that allows you to display the interval bounds with different typography, perhaps using color, or changes in font sizes, or font shapes, just by redefinition of the internals of those four macros. We also put their math mode inside a braced group, so that they can be used in both text mode, and in math mode.

Finding grammatical errors, and outright nonsensical statements, is not easily done by any software that we have access to. Instead, careful reading by human proofreaders is called for. Students should be willing to ask a fellow student in a similar field for proofreading help, and be prepared to do so for others.

---

[1] Source code is available at http://www.math.utah.edu/pub/chkdelim

5   Document font size

The historical practice at the University of Utah has been to use 11pt fonts (where 72.27 points is exactly 1 inch or 25.4 millimeters). That size is comfortable for most humans with normal or corrected eyesight. Thus, a dissertation or thesis top-level LaTeX file would normally begin

```
\documentclass[11pt]{book}
```

The LaTeX book style defaults to a 10pt size unless overridden by an 11pt or 12pt bracketed option. No other sizes are standardly available. The question then arises: what if I, or some of my committee members, have reduced visual acuity, and require larger fonts?

The solution is to switch to the extended book class, extbook. It accepts additional size options: 8pt, 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, and 20pt. Your LaTeX file could then start like this:

```
\documentclass[20pt]{extbook}
```

You might then prepare two copies of your document: one in the normal font size, and another in a suitable larger size. The former would serve for the draft and final copies submitted to the Graduate School, while the latter would be the working copies for you or your graduate advisory committee.

Please note that font sizes are assigned by the font designer, and need not precisely specify character extents: the glyph **X** in a 10pt font might actually be 9.6pt or 10.3pt from top to bottom. In particular, if you mix fonts from different families, be prepared to scale them to improve their visual matching. For example, the preamble of this document has the command

```
\RequirePackage [scaled = 0.85] {DejaVuSansMono}
```

to load an alternate fixed-width font at a somewhat reduced size. The font-sampler resource noted on page 4 has much more about choosing alternate fonts.

We discuss later on page 35 how to control line spacing, which depends on the font size.

Remember that changing fonts, or font sizes, affects line breaking, page breaking, and page numbers, so you need to be careful to run the various production tools a sufficient number of times to achieve consistency.

6   Minimal packages for LaTeX

Two packages are required in your top-level LaTeX document:

```
\usepackage {uuthesis-base}    % MANDATORY FIRST package for a minimal thesis
\usepackage {uuthesis-updates} % MANDATORY LAST of all uuthesis-* packages
```

The first provides the minimal set of definitions needed for a dissertation or thesis, and that package is expected to be kept frozen for years at a time. However, we anticipate that further experience may call for slight tweaks in definitions, and thus, we use a second, and smaller, package that may change more frequently. Each logs a banner with version number and modification date.

Between those two package inclusions, you can have any number of `\usepackage` commands to incorporate other standard LaTeX packages that you might require. However, please avoid any that change the overall page dimensions, such as the `geometry` package: it is *your* responsibility to ensure that your final document conforms to regulation page dimensions.

Many authors like to introduce customized commands, either for convenience in input preparation, or for achieving a desired typographical effect, or for guaranteeing consistent formatting. Feel free to do so, and keep your customizations together in one place in a private style file that you include as the last package in your top-level LaTeX file. A typical example looks like this:

```
\usepackage {mythesis}          % for personally defined macros
```

Your package name is arbitrary, but it should not conflict with the names of standard packages: the suggested name is safe.

7   Front matter

The sample file that starts in Section 3 on page 4 has about 20 lines that supply values for various fields that occur on the title page, and in the front matter. They are likely to be obvious, and all that you need to do is change the sample values to those needed for your document.

At the University of Utah, a master's thesis needs only a chair and two readers, so if the `\thirdreader` and `\fourthreader` commands are omitted, or hidden inside comments, then the approval-page format is automatically somewhat different from that for a doctoral degree.

The `\degree` macro supplies the actual text naming the degree; it could have considerable variation across departments and colleges, so its value cannot be used to control how the approval page is formatted.

When you supply personal names for various people recorded in the front matter, please ensure that you use their preferred names for professional publication: the committee member whom you know personally as *Sam Jones* might publish as *Samantha Q. Jones*. Chairs and Deans are time-limited appointments, so make sure that you supply current values. If you are uncertain of their names, check the Web sites of your Department, and the Graduate School, for correct values shortly before you produce the draft copies given to your committee members.

The `\submitdate` value changes each term, as mandated by the Graduate School, and the committee member approval dates are those on which they signed your paperwork after a successful oral defense. Although the approval dates are often identical, they need not be. You

might have one or more members who attended your defense via a remote video link, and their paperwork might therefore be delayed by several days. If a member is sufficiently distant that signing produces too large a delay, consult with your Graduate Coordinator or Department Chair: they may be able to authorize a substitute physical signature.

The signed approval page that is submitted to the Graduate School looks similar to that in your document, but the latter intentionally lacks the signatures. It would be a terrible mistake to have images of personal signatures embedded in electronic documents where they could then be used for forgery.

8   Typesetting the front matter

In LaTeX, typesetting is not permitted until the `\begin{document}` command has been processed. Thus, the front matter has to be typeset by the ten lines that follow that command in the sample in Section 3. Those lines should not require modification, although you are free to comment out those marked optional. That flexibility is why we chose not to hide those ten lines inside a one-line macro.

9   Typesetting the remainder

Typesetting of your own prose begins with the first chapter, and as recommended earlier, you should have separate input files for each. If your computer is slow, you can speed up typesetting with a preamble command like

```
\includeonly{chap2,chap3}
```

to skip reading all but the two named files. LaTeX then uses the auxiliary files from the skipped chapters to supply cross references and page numbers. Those references are soon likely to be somewhat out of date, but you can then comment out the above command, run LaTeX a couple of times, and uncomment the command to continue working with mostly correct references.

10   Microtypography

One of the design goals of PDF-LaTeX was to extend TeX with microtypographical refinements that can improve line breaking and page breaking, and reduce the need for hyphenation. That support was worthy of part of a doctoral dissertation, and is described in a 250-page book available in recent TeX Live installation trees in the location `texmf-dist/doc/latex/microtype/microtype.pdf`. You may be able to display it on your system with the command `texdoc microtype`.

If you wish to make a quick test for your own document, just add the line

```
\usepackage {microtype}
```

in your L̸ATEX file preamble. If you first save a copy of your output PDF file without those enhancements, then you can later run the `diffpdf` command on the old and new PDF files to see what it does. For an earlier version of this document, page breaks remained largely the same, but most hyphenations were eliminated: 23 out of 26 disappeared, leaving an average of just one hyphenation every seven pages. With subsequent minor tweaks to the text, *all remaining hyphenations* have been eliminated from this document.

11   L̸ATEX sectional commands

Standard L̸ATEX classes recognize several sectional commands, from `\part` and `\chapter`, through `\section`, down to `\subparagraph`. Each normally takes a single argument, the title of the sectional unit. However, if you have long titles, then you should supply an optional argument with a shorter title that is used in the front matter lists, and in running page headers.

The double use of the optional argument is a problem for University of Utah dissertations and theses, because regulations require full sectional names in the table of contents. Fortunately, there is a simple way to meet those requirements.

Here is a recommended way to write such headings or captions:

```
\chapter [The First,
          having a short alias
          for the running page headers,
          and a rather long definition
          that takes multiple lines with
          user-defined line breaks]
          {The First,                   \\
          having a short alias          \\
          for the running page headers,  \\
          and a rather long definition   \\
          that takes multiple lines with \\
          user-defined line breaks}
\chaptermark {The First}

\section {Ugandan termites (\protect\bioname{Coptotermes formosanus})
          found near Lake Victoria in 2017.}
\sectionmark {Ugandan termites in 2017}

\caption [Ugandan termites in 2017]
          {Ugandan termites (\protect\bioname{Coptotermes formosanus})
```

```
found near Lake Victoria in 2017.  The soldiers have
red-colored heads, while the workers have pale-colored heads.
Termites build mounds, and some African varieties produce
enormous ones, up to 9 meters high, and 13 meters long.}
```

Vertical alignment of the bracketed short, and braced long, forms makes them easier to read and verify. In captions, the short form should *not* end in punctuation, whereas the long form should contain complete sentences, and rarely, might even have several paragraphs.

The argument of `\chaptermark` and `\sectionmark` is used in the running page headers. In the `\chapter` command, we had to supply both arguments, because the line-breaking requirements differ in the two locations. Here, we omitted explicit line break commands, `\\`, in the bracketed argument, and supplied them in the braced argument. To adjust the entry appearance in the table of contents, you could insert different line breaks in the bracketed argument.

Please note, however, that long sectional headings are **rarely needed**: most well-designed technical books limit headings to at most five words, short enough to fit on a single line in both the text, and the front matter.

12   Running page headers

Previous versions of LaTeX support for University of Utah dissertations and theses did not permit running page headers, even though they are helpful to readers, particularly in large documents.

The new styles documented here automatically produce running heads in an italic font, and separated from the body text by a thin horizontal rule.

13   Including publications as chapters

Some students publish journal articles or book chapters before they graduate, and it is desirable to recognize that important work by incorporating verbatim page images of such publications as separate chapters. That leverages prior work for the student, makes the Graduate School editors happy, because those pages are essentially fixed images that they do not need to read or correct, and emphasizes to human readers that part of the work has already undergone external professional review and validation.

The final two of the four doctoral sample theses show how such inclusions are done. The essential steps are:

- Add a `\usepackage {uuthesis-chapterbib}` command in the LaTeX file preamble, because the Graduate School requires separate bibliographies in each chapter when one or more of them is a previously published document.

- Add a `\usepackage {pdfpages}` command in the LaTeX file preamble.

- Write a short introduction of a half-page or so to begin the chapter. There you state that you got permission from the publisher to include the PDF file, and other information that might be pertinent, such as your share of the work when there are multiple authors.

- Use the `\uudummyXXX` macros to create table-of-contents data for your document to reflect the sectional structure of the included PDF file. Their page number values are those in the included file; the macros automatically adjust them to values suitable for your document.

- Issue the `\setupuuchapterbib` command to get things started on a new page.

- Include a suitably scaled version of the PDF file, with a command like this:

```
\includepdf [
              pages = -,            % want all document pages
              scale = 0.85,         % adjust to fit thesis page box
              pagecommand = {\pagestyle{plain}} % bare page numbers
            ]
            {mypaper.pdf}
```

The major point where care is needed is selection of the scale factor: both the height and width of the included document need to fit in the normal text area of your document.

14   Bibliographies

Any of the four standard BibTeX styles, **abbrev**, **alpha**, **plain**, and **unsrt**, can be used for formatting your bibliography from BibTeX entries. They were designed in the mid-1980s to correspond to common practices in journals in computer science and mathematics. Since then, hundreds more styles have been developed to meet the needs of various journals and publishers. However, for a dissertation or thesis, you are strongly advised to stick to the simplest possible, narrowing our recommendations to just **alpha** or **plain**, producing alphanumeric tags, or bracketed numbers.

The **unsrt** style should be avoided: it is unfriendly to readers of your document, because entries in your reference list appear in citation order, making the list effectively a random one when viewed alone.

The advent of the Web, and electronic publishing, however, has made it possible to enrich reference lists with extra data that make it easier for your readers to precisely identify journals, to readily find books in library catalogs, on library shelves, and at bookseller Web sites, and to reach article texts directly with a single selection in a PDF file viewer or Web browser. The author of this document has therefore developed several additional BibTeX styles, including four

that extend the original basic styles. They are available in the sample theses as the base names **x-abbrev**, **x-alpha**, **x-plain**, and **x-unsrt**. Because of their enhanced capabilities, they require additional macro definitions supplied by a

```
\usepackage {x-bst}
```

command in your LATEX file preamble.

The LATEX input that BIBTEX produces in the .bbl file is significantly enhanced, and each field is wrapped in a macro that controls how, and whether, it is typeset. For example, to suppress typesetting of, say, Library of Congress call numbers, your input file could end with

```
\showLCCNfalse
\bibliography{\jobname}
```

Consult the beginning of your .bbl file to see the dozens of test macros, \showXXX, that you can turn on with \showXXXtrue, and off with \showXXXfalse.

The markup is sufficiently detailed that a usable BIBTEX file can be automatically recovered by software from the formatted reference list in the .bbl file. That could be valuable for a publisher, who could accept author reference lists in any of the extended styles, but could convert them back to BIBTEX form for regeneration in a journal-specific style.

If you have

```
\usepackage {hyperref}
```

near the end of your LATEX file preamble, then numerous commands are redefined to add hypertext links in your output PDF file, which must then be produced directly with pdflatex, rather than via the older route of latex to DVI to PostScript to PDF. In particular, URL and DOI values then automatically have such links, which is a great convenience for your readers, and also helps publishers in the important job of recording cross links between documents, such as those long provided by the *Science Citation Index* service. Links to references make it easy to start with a current publication, and then step backward in time to find the original works on which it is based.

The second of the four sample doctoral files, sample-thesis-1.ltx, has been changed from

```
\bibliographystyle {plain}
```

to

```
\usepackage {x-bst}          % in preamble
...
\bibliographystyle {x-plain} % in preamble, or close to \bibliography command
```

in order to allow you to compare the extended typeset reference list with the briefer lists in the other three sample theses.

15   Indexes

When documents were produced on typewriters, indexing was a tedious and error prone task that would have to be repeated at each subsequent modification of the material. Thus, dissertations and theses have traditionally lacked indexes, because it was viewed as too onerous a job for a document that would often have only a small number of human readers. However, with computer support, index preparation is not difficult, and with a day or two of work, you can likely prepare a thorough index of your document.

Three of the four sample doctoral theses show how to create index entries in the document files, then process them with grep and makeindex to make one or more indexes. The key idea is that the basic index entries that you create can have additional markup that allows groups of them to be easily extracted by grep into separate index files that makeindex can read, sort, and format into other files that are read by LATEX.

Thus, instead of an index command

```
\index{tiger (Panthera tigris)}
```

we instead use

```
\index{tiger (\bioname{Panthera tigris})}
```

The \bioname wrapper macro serves two purposes: it allows the Latin name to be set in a different font (traditionally, *italics*), and it allows grep to prepare a separate index file that shows only the entries that contain the string bioname.

Similar wrappers can be provided for other classes of index phrases, such as corporations, software package names, tradenames, and so on.

You can easily later exclude entries *without changing your document* by eliminating them from your indexes with a simple command stage of the form

```
grep -v housefly
```

The -v option inverts the pattern matching, eliminating all index lines that contain *housefly*, which might be too numerous to usefully index.

To eliminate multiple index phrases, use the extended pattern matcher like this:

```
egrep -v 'flea|fruitfly|gnat|housefly|louse|midge|no-see-um'
```

16   Dissertation and thesis styles

Here is a summary of the currently available packages that extend the standard LATEX book or extbook classes for University of Utah dissertations and theses:

uuthesis-approval : Macros to typeset the committee approval page.

uuthesis-base : The large package that extends the book class. It is *mandatory* and is always the *first* package loaded. It is expected to remain frozen for years at a time.

uuthesis-chapterbib : Macros for producing separate bibliographies in each chapter. This package is *mandatory* if you include PDF files of other documents in one or more chapters.

uuthesis-color : Macros for producing headings in color. See page 25 for details.

uuthesis-index : Macros for supporting document indexes; it includes the standard LaTeX `makeidx` package, and then modifies that package's definitions.

uuthesis-scaled-headings : Internal package to supply five relative font sizes in headings. The Department of Mathematics has tentatively chosen a smaller size for 2018–2020, so student authors should not issue any of the commands listed in that file to select heading font sizes. The `sample-thesis-3` directory contains files `sample-thesis-3-uu*.pdf` that illustrate the supported sizes.

uuthesis-singlespace-bib : Experimental package for producing single-spaced reference lists. It is illustrated in `sample-thesis-3.pdf`.

uuthesis-updates : Mandatory package that should always be loaded *last*. It supplies any small tweaks that are needed to override settings in other `uuthesis-*` packages, which are expected to be frozen for long periods. It may change according to recommendations or requirements of the Graduate School, so before submitting your draft for review by their editors, make sure that you have the latest version of this package, which you can find from links at the Web site `http://www.math.utah.edu/pub/uumath-thesis`.

Three additional experimental styles are described later on page 26.

17   Overlapping numbers in contents and lists

Each sectional heading, and each caption of a figure or a table, generates lines like these in auxiliary files with extensions `.lof`, `.lot`, and `.toc`:

```
\contentsline {figure}  {\numberline {1.11} {\ignorespaces Graph of cosine}} {14}
\contentsline {table}   {\numberline {1.12} {\ignorespaces Cosine properties}} {16}
\contentsline {section}    {\numberline {5.14} Summary} {37}
\contentsline {subsection} {\numberline {6.3.1} Anatomy of gnats} {43}
```

In the master file `texmf-dist/tex/latex/base/latex.ltx` that implements the core of LaTeX, the `\contentsline` command is defined as a macro with only one argument. Its invocation expands that to an internal command, such as `\l@figure`, whose definition differs among document classes.

For the LaTeX book class, its definition in `texmf-dist/tex/latex/base/book.cls` looks something like this:

```
\newcommand* {\l@figure} {\@dottedtocline {1} {1.5em} {2.3em}}
\let \l@table = \l@figure
```

Thus, the commands `\l@figure` and `\l@table` both expand to a reference to the internal command `\@dottedtocline`. That command is defined to require five arguments. The first three appear in the `\@dottedtocline` reference, and the remaining two are supplied by the last two braced groups in the `\contentsline` command. The final expansion then becomes something like this:

```
\@dottedtocline {1}                                          % level
                {1.5em}                                      % indent
                {2.3em}                                      % numwidth
                {\numberline {1.11}{\ignorespaces Graph of cosine}} % text
                {14}                                         % page
```

The second and third arguments are *hard-coded dimensions* of boxes. Thus, if the figure number is long, such as 13.207, or 6.1.4.17, it will likely overflow the box, with the result that the figure number overlaps the caption text.

This excursion into LaTeX internals tells us that it may sometimes be necessary for a document author to redefine those commands to enlarge the number box. Because of the use of the special character `@` in the command names, such definitions must be supplied in a private style file, such as the one named `mythesis.sty`. They might look like this:

```
%%% My figure/table numbers can be long, so increase the
%%% number box width from its default of 2.3em to 3.4em:
\newcommand* {\l@figure} {\@dottedtocline {1} {1.5em} {3.4em}}
\let \l@table = \l@figure
```

The `\let` command ensures that the list of figures and the list of tables are handled in the same way. Without that command, `\l@table` retains its default definition.

It is clearly better to replace the fixed box dimensions with user-definable length variables, so version 2.01 of `uuthesis-updates.sty` has been augmented with this code:

```
\newlength {\fignumwidth}
\newlength {\pagenumwidth}
\newlength {\parnumwidth}
```

```
\newlength {\secnumwidth}
\newlength {\subparnumwidth}
\newlength {\subsecnumwidth}
\newlength {\subsubsecnumwidth}
\newlength {\tabnumwidth}

\setlength {\fignumwidth}        {2.3em}
\setlength {\pagenumwidth}       {1.55em}
\setlength {\parnumwidth}        {5em}
\setlength {\secnumwidth}        {2.3em}
\setlength {\subparnumwidth}     {6em}
\setlength {\subsecnumwidth}     {3.2em}
\setlength {\subsubsecnumwidth}  {4.1em}
\setlength {\tabnumwidth}        {2.3em}

\renewcommand* {\l@figure}       {\@dottedtocline {1} {1.5em} {\fignumwidth}}
\renewcommand* {\l@table}        {\@dottedtocline {1} {1.5em} {\tabnumwidth}}
\renewcommand* {\l@section}      {\@dottedtocline {1} {1.5em} {\secnumwidth}}
\renewcommand* {\l@subsection}   {\@dottedtocline {2} {3.8em} {\subsecnumwidth}}
\renewcommand* {\l@subsubsection}{\@dottedtocline {3} {7.0em} {\subsubsecnumwidth}}
\renewcommand* {\l@paragraph}    {\@dottedtocline {4} {10em}  {\parnumwidth}}
\renewcommand* {\l@subparagraph} {\@dottedtocline {5} {12em}  {\subparnumwidth}}

\renewcommand  {\@pnumwidth}     {\pagenumwidth}
```

Now, an author who needs to widen the default figure-number box in the list of figures just needs this simple command in the document preamble, *after* the `\usepackage{uuthesis-updates}` command:

```
\setlength {\fignumwidth} {3.4em}
```

Even better, instead of introducing yet another hard-coded magic dimension, ask TₑX to compute it for you according to its expected contents:

```
\settowidth {\fignumwidth} {99.99}  % enough for 2-digit parts
```

Clearly, the indentations could be parametrized as well, but they are unlikely to require user customization.

18   Multilingual text

Dissertations and theses at the University of Utah are written in English, but in some cases, such as for a student taking a degree in linguistics, or a student wishing to cite short passages in a language that requires a nonroman script, it is desirable to be able to typeset text in other languages. With the help of additional packages, LaTeX can do so.

The first issue is the encoding of the non-English characters in the input files. Historically, there has been a huge mess with incompatible encodings used in various countries, with names like Latin-1, Latin-2, …, or equivalently, ISO 8859-1, ISO 8859-2, … for languages native to Europe, and many others for African and Asian languages. Fortunately, the Unicode character set introduced in the late 1980s has largely solved the problem by offering a combined repertoire targeted at the writing systems of all of the world's human languages. Unicode can ultimately support about two million characters, but in the March 2020 version 13.0 of Unicode, 143,859 characters have been encoded from 154 scripts. Most of them are the ideographs required for Chinese and Japanese. Those large numbers mean that representations as 8-bit and 16-bit integers are inadequate, because they would permit only 256 or 65 536 different characters. Unicode characters require up to 23 bits, but the encoding rules do not permit all $2^{23} = 8\,388\,608$ possible values.

In addition, there is a strong need to support the most widely used existing code — ASCII (American Standard Code for Information Interchange). ASCII was introduced in 1963, and defines only 7-bit encodings, sufficient for lowercase and uppercase letters, digits, punctuation, and 33 control characters. It has been the dominant character set on most computers and operating systems since the mid 1970s.

In their early attempts to support Unicode, Microsoft Windows and the Java programming language adopted a 16-bit encoding called UTF-16; escape sequences are defined to allow two 16-bit integers to encode the full range of Unicode.

The rest of the world, including all operating systems in the Unix family, of which macOS, and the many flavors of BSD, GNU/Linux and Solaris, are members, has instead chosen the UTF-8 encoding, which requires from one to four 8-bit bytes to represent any Unicode character. Importantly, ASCII characters occupy the first 128 positions of UTF-8, so *all existing text files in ASCII are automatically valid Unicode files in UTF-8 encoding*.

In order to typeset arbitrary UTF-8 files, LaTeX needs additional packages to identify the input encoding, and it must also have fonts that provide the character glyphs.

Here is a snippet of preamble commands for a multilingual document:

```
\usepackage [utf8]                                        {inputenc}
\usepackage [T2A, T1]                                     {fontenc}
\usepackage [french, german, greek, polish, russian, english] {babel}
```

The text in this section is written in UTF-8, with *no explicit TEX accents*. The text samples are taken from various European newspaper sites, and from the Kermit site at `http://kermitproject.` `org/utf8.html`.

Accented letters in Latin-based scripts can be displayed in verbatim environments, but Greek and Cyrillic cannot without additional package support.

Typesetting of this document has been tested with annual TEX Live releases from 2013 to 2020; older versions do not handle this file.

Here is the actual input as displayed in verbatim mode:

```
\textbf{Albanian text}:
    Gatishmëria juaj për t'ju bashkuar kësaj kauze tregon për një
    shpirt të madh e human të qytetarëve të Kuçovës.

\textbf{French text}:
    Des milliers de Polonais ont participé, lundi 14 janvier
    dans la soirée, à des rassemblements contre la violence.

\textbf{German text}:
    Kohleausstieg { Gibt es Notfallpläne für großflächige Blackouts?

\textbf{Icelandic text}:
    Heildarkostnaur ríkisins vegna kísilversins PCC á Bakka vi
    Húsavík endai í rúmlega 4,2 milljörum króna.

\textbf{Polish text}:
    Matki by␣ly wniebowziete! Wielka Orkiestra Światecznej Pomocy
    przekaza␣la kilkadziesiat wygodnych foteli dla Górnoślaskiego
    Centrum Zdrowia Dziecka w Katowicach.

\textbf{Romanian text}:
    Poliţiştii i-au făcut semn să oprească, dar şoferul şi-a văzut
    liniştit de drum. De ce a refuzat conducătorul auto să tragă
    pe dreapta.
```

Here is the text display of those unrelated samples:

Albanian text: Gatishmëria juaj për t'ju bashkuar kësaj kauze tregon për një shpirt të madh e human të qytetarëve të Kuçovës.

French text: Des milliers de Polonais ont participé, lundi 14 janvier dans la soirée, à des rassemblements contre la violence.

German text: Kohleausstieg – Gibt es Notfallpläne für großflächige Blackouts?

Icelandic text: Heildarkostnaur ríkisins vegna kísilversins PCC á Bakka vi Húsavík endai í rúmlega 4,2 milljörum króna.

Polish text: Matki były wniebowziete! Wielka Orkiestra Światecznej Pomocy przekazała kilkadziesiat wygodnych foteli dla Górnoślaskiego Centrum Zdrowia Dziecka w Katowicach.

Romanian text: Poliţiştii i-au făcut semn să oprească, dar şoferul şi-a văzut liniştit de drum. De ce a refuzat conducătorul auto să tragă pe dreapta.

To display a fragment of Russian text in the Cyrillic alphabet, supply it as the argument of the command `\textcyrillic{...}`:

Отменная позиция: кому выгодны массовые задержки авиарейсов «Известия» выяснили, как российские перевозчики зарабатывают на непогоде

If your readers do not know the foreign alphabet or writing system, it can sometimes be helpful to supply a transliteration into the Latin alphabet:

Otmennaya pozitsiya: komu vygodny massovyye zaderzhki aviareysov ≪Izvestiya≫ vyyasnili, kak rossiyskiye perevozchiki zarabatyvayut na nepogode.

Several other Asian and European languages also use the Cyrillic alphabet, sometimes with additional letter shapes, as in this sample in Kyrgyz:

Премьер-министр Мухаммедкалый Абылгазиев Алматы шаарында Евразия өкмөттөр аралык кеңешинин тар курамдагы жыйынына катышууда.

If you deal with large blocks of text in other languages, you will soon encounter the need for their correct hyphenation. The **babel** package provides commands for identifying the input language, and selecting its corresponding hyphenation patterns. Run the command `texdoc babel` to view its documentation.

19   Color naming in LaTeX

The standard LaTeX **color** and **colortbl** packages define the interface to the TeX typesetting engine, and provide default definitions of just a few named colors:

```
\definecolor {black}   {rgb}  {0, 0, 0}
\definecolor {white}   {rgb}  {1, 1, 1}

\definecolor {black}   {gray} {0}
\definecolor {white}   {gray} {1}
```

```
\definecolor {red}     {rgb}  {1, 0, 0}
\definecolor {green}   {rgb}  {0, 1, 0}
\definecolor {blue}    {rgb}  {0, 0, 1}

\definecolor {cyan}    {cmyk} {1, 0, 0, 0}
\definecolor {magenta} {cmyk} {0, 1, 0, 0}
\definecolor {yellow}  {cmyk} {0, 0, 1, 0}
```

In most cases, except for *black* and *white*, those colors are too bright for routine use.

Consequently, several years ago the author of this document created the **rgb** package that defines names for the roughly 750 colors that are named in the X11 Window System, and displayable with the `showrgb` command. You can view a subset of them on the Web.[2] For historical reasons, different members of the Unix operating system family may have slightly different color names and definitions. The **rgb** package was prepared by merging `showrgb` output from several different operating systems.

The X11 Window System allows color names to contain spaces, but ignores any spaces, and letter casing, when the names are used in command-line options. You must remove those spaces for LaTeX use:

```
% showrgb | grep forest
 34 139 34 forest green

% grep forest rgb.sty
\definecolor{forestgreen}{rgb}{0.133,0.545,0.133} % #228b22 34 139 34 ForestGreen
```

The X11 named colors have been criticized in a conference talk[3] for their inconsistency and unpredictability, but most of those color names have been in use for more than three decades, so we just have to accept them. You can always redefine them, or add your own names, such as with LaTeX definitions like these:

```
% For official color of universities in Utah, see:
%      https://identity.byu.edu/colors/
%      https://www.suu.edu/mc/brand/font-color.html
%      https://www.usu.edu/prm/identity/pdf/USU_Visual_Identity.pdf
%      https://umc.utah.edu/resources/branding-guidelines/visual-style/
%      https://www.uvu.edu/marketing/branding/
%      https://www.weber.edu/brand/color_palette.html
```

---

[2]See `http://cng.seas.rochester.edu/CNG/docs/x11color.html` and `https://www.w3schools.com/colors/colors_x11.asp`

[3]See `https://www.youtube.com/watch?v=HmStJQzclHc`

```
\definecolor {byublue}   {rgb} {0.0,    0.1333, 0.3333} % official (hex 002255)
\definecolor {byutan}    {rgb} {0.7725, 0.6863, 0.4902} % official (hex c5af7d)

\definecolor {suured}    {rgb} {0.7686, 0.0784, 0.1412} % official (hex c41424)
\definecolor {suublack}  {rgb} {0.1373, 0.1216, 0.1255} % official (hex 231f20)
\definecolor {suuwhite}  {rgb} {1.0,    1.0,    1.0   } % official (hex ffffff)

\definecolor {usublue}   {rgb} {0.0,    0.2,    0.4   } % official (hex 003366)
\definecolor {usucopper} {rgb} {0.6745, 0.5530, 0.4510} % official (hex ac8d73)
\definecolor {ususilver} {rgb} {0.6431, 0.6824, 0.7098} % official (hex a4aeb5)

\definecolor {utahred}   {rgb} {0.8,    0.0,    0.0   } % official (hex cc0000)

\definecolor {uvugreen}  {rgb} {0.1529, 0.3137, 0.2196} % official (hex 275038)

\definecolor {wsupurple} {rgb} {0.2863, 0.1373, 0.3961} % official (hex 492365)
\definecolor {wsugray}   {rgb} {0.3412, 0.3137, 0.2784} % official (hex 575047)
```

If you want to mix your own colors, X11 tools like agave, gpick, kcolorchooser, kcoloredit, xcolors, and xcolorsel make it easy to do so.

In some cases, you may wish to have alternate names for already-defined colors. This author's coloralias package allows a football fan to define them like this:

```
\definecoloralias {aggieblue}     {usublue}
\definecoloralias {aggiecopper}   {usucopper}
\definecoloralias {aggiesilver}   {ususilver}

\definecoloralias {cougarblue}    {byublue}
\definecoloralias {cougartan}     {byutan}

\definecoloralias {utered}        {utahred}

\definecoloralias {thunderbirdred} {suured}

\definecoloralias {wildcatpurple}  {wsupurple}
\definecoloralias {wildcatgray}    {wsugray}
```

Armed with convenient names for your favorite colors, you can now typeset with them like this:

```
This is \colorbox{byutan}{black text on BYU tan}.

This is {\color{usucopper} text in USU copper}.

This is \colorbox{utahred}{\color{yellow} yellow text on Utah
red}.

\begin{center}
    \newcommand {\headingstuff} {\color{black}\large \bf}
    \begin{tabular}{>{\columncolor{utahred}\color{white}\bfseries}l
                    >{\columncolor{utahred}\color{white}\bfseries}r}
        \rowcolor{grey90} \headingstuff Campus &
                          \headingstuff Fall 2017 Enrollment \\
        %
        \rowcolor{byublue}   BYU     & 20,000 \\
        \rowcolor{suured}    SUU     & 10,000 \\
        \rowcolor{usucopper} USU     & 25,000 \\
        \rowcolor{utahred}   Utah    & 33,000 \\
        \rowcolor{uvugreen}  UVU     & 37,000 \\
        \rowcolor{wsupurple} WSU     & 28,000 \\
         %
        \rowcolor{grey90} \headingstuff Total & \headingstuff 153,000
    \end{tabular}
\end{center}
```

This is black text on BYU tan .
This is text in USU copper.
This is yellow text on Utah red .

Campus	Fall 2017 Enrollment
BYU	20,000
SUU	10,000
USU	25,000
Utah	33,000
UVU	37,000
WSU	28,000
Total	153,000

Our sample table reveals a problem that you should always keep in mind when using adjacent,

or distinguishing, colors: the *Utah* and *UVU* rows have red and green backgrounds, and may look similar to individuals with a degree of red/green color blindness.

Depending on your output device, and your own color acuity, you may find it difficult to distinguish between the *BYU* and *WSU* backgrounds, and the *SUU* and *Utah* backgrounds. A simpler design could be obtained by alternating row colors, *without* modifying the table markup at all, like this:

```
\begin{center}
    \definecoloralias {byublue}   {grey70}
    \definecoloralias {suured}    {grey80}
    \definecoloralias {usucopper} {grey70}
    \definecoloralias {utahred}   {grey80}
    \definecoloralias {uvugreen}  {grey70}
    \definecoloralias {wsupurple} {grey80}
    \newcommand {\headingstuff} {\color{black}\large \bf}
    ...
\end{center}
```

Campus	Fall 2017 Enrollment
BYU	20,000
SUU	10,000
USU	25,000
Utah	33,000
UVU	37,000
WSU	28,000
Total	153,000

Once the enclosing `center` environment is left, the previous definitions of the color names are automatically restored.

20   Colored headings

The `uuthesis-color` package provides alternate definitions of commands for document sections, theorem-like environments, and verbatim environments that allow you to define their text colors. It defines default values for three colors, `utahheadingcolor`, `utahtheoremcolor`, and `verbatimcolor`, as well as three primary colors, `utahblue`, `utahgreen`, and `utahred`. You can override the defaults with your own subsequent explicit color definitions:

```
\definecolor {utahheadingcolor} {rgb} {0.9, 0.0, 0.0}
\definecolor {utahtheoremcolor} {rgb} {0.0, 0.9, 0.0}
\definecolor {verbatimcolor}    {rgb} {0.0, 0.0, 0.9}
```

Alternatively, you can use the alias feature to leverage existing definitions, without having to know their details:

```
\definecoloralias {utahheadingcolor} {utahred}
\definecoloralias {utahtheoremcolor} {uvugreen}
\definecoloralias {verbatimcolor}    {utahblue}
```

If you decide to use colored headings in your document, avoid garish colors, like *hot pink* and *blazing orange*, and ensure that the colors are easily readable on the screen, on output from one or more models of color printers, and on black-and-white printers. Some PDF viewers, such as `atril` and `evince` on GNU/Linux systems, give you the option of printing a color document using only black, white, and shades of grey. Otherwise, just use a photocopier to reprint a monochrome or greyscale version of a colored output page from your document.

21   Colored covers

Three experimental styles make small changes to definitions used in the cover page produced by `\titlepage`:

uuthesis-color-cover : Typeset all text on the cover page in the `utahheadingcolor`.

uuthesis-color-cover-title : Typeset just the title on the cover in the `utahheadingcolor`.

uuthesis-color-cover-title-author : Typeset just the title and author on the cover in the given `utahheadingcolor`.

The first can be used anywhere in the preamble. The last two styles modify the internal definitions saved by the `\author` and `\title` commands, so they must be included *after* those commands.

These three styles are subject to departmental review, and might be withdrawn, or modified, in the future. One of them is illustrated in `sample-thesis-3.pdf`.

22   Fixing overfull boxes

In each of the sample theses, the `\documentclass` command is followed by these two lines:

```
\overfullrule = 0pt                 % do not show overfull boxes
\overfullrule = 100pt               % make overfull boxes really visible!
```

The first is the default value. The second produces a 100pt-wide black rule in the right margin of any line that is *overfull*, that is, protrudes into the margin. That action makes problem lines easily visible in a document viewer. It is your job to fix such problems, which can be done in several ways:

- If you have not already used it, add `\usepackage{microtype}` to your document preamble; it often does wonders for hyphenation and overfull-box problems.

- If the protruding text could benefit from a special hyphenation rule this single time, insert a discretionary hyphen, `\-`, at a suitable point, for example, `syz\-ygy`. English hyphenation practices avoid starting or ending fragments shorter than three characters, but that is such a peculiar technical word of Greek origin that you might make an exception: `sy\-zygy`, or `syzy\-gy`, or `sy\-zy\-gy`.

- If the protruding text could benefit from a special hyphenation rule in multiple places in your document, add a command like this to your preamble:

  ```
  \hyphenation
  {
      anti-dis-es-tab-lish-ment-ar-i-an-ism
      syz-ygy
      white-space
  }
  ```

- TeX does not hyphenate words that contain explicitly accented letters, unless discretionary hyphens allow it to do so: `{\'e}l{\'e}\-ments d'analyse` and `Paul Erd\-{\H{o}}s`.

- If hyphenation is not possible, reword the sentence or paragraph. Switch from passive to active voice for crisper, and often shorter, text. Many words in human languages have synonyms, and a thesaurus of such words can help you find some. Our local filesystems contain the Project Gutenberg file `/u/cl/doc/roget.text` that you can search in a text editor. A copy of that file is also available from the `http://www.math.utah.edu/pub/tex/doc/` site.

- If the overfull line is in a verbatim display, you may be able to adjust the text slightly, or else reduce the font size, making sure to surround the outer `small` environment with *one or more empty lines*, like this:

  ```
      ...

  \begin{small}
  ```

```
        \begin{verbatim*}
            ...
        \end{verbatim*}
    \end{small}

    ...
```

- Wrap the current paragraph like this, making sure that there is at least *one blank line* before *and* after the environment:

```
    ...

    \begin{sloppypar}
        ...
    \end{sloppypar}

    ...
```

- Investigate the LaTeX package **hyphenat** to see whether its features could help solve your line-breaking problems by providing more opportunites for automatic hyphenation.

- If none of the above is feasible, you can tweak TeX's parameters that affect paragraph breaking; see Chapter 14 of *The TEXbook*. Take care to localize their effect, by putting them in a group or environment. Something like this is generally safe in prose:

```
    \begin{rm}
        \pretolerance = 200
        \tolerance    = 400
        ...
    \end{rm}
```

23   Managing your document with `make`

The Unix `make` utility ranks as one of the most important pieces of software ever written, because it can be used to dramatically simplify computer processing, hiding most of the details from the user, and minimizing the work done by both the user, and the computer.

`make` can potentially be implemented on any operating system that supports a command shell, and with care, the `Makefile` can be used unchanged across *every* system that has a similar shell. In the Unix family, that means the POSIX Standard Bourne shell, `/bin/sh`.

Each of the sample theses is accompanied by a `Makefile` that contains declarations of file dependencies, and rules to build files from other files. The details are project specific, but the general procedures to use them can be made relatively generic by widespread agreement of a minimal standard set of target names that every `Makefile` should implement. They include

all : The first, and default, target to be made when no target is specified on the command line. Its dependencies and rules should result in `make` creating everything that can be built automatically in the current directory.

check : Run a validation suite to test correct operation of the project software

clean : Remove intermediate files that are unneeded when the project is complete.

distclean : Remove *all* project files that `make` can later rebuild automatically on any system.

install : Install built and validated software in standard conventional locations.

maintainer-clean : Remove *all* project files that `make` can later rebuild automatically, including ones that might require special tools available only on certain systems.

The goal is that on *all* systems, it should be possible to build, validate, and install the project with a simple recipe:

```
% make all check install
```

When the GNU `autoconf` system is used to create a `configure` script for automatic site-specific customization of other files, the recipe is extended to this:

```
% ./configure && make all check install
```

Tens of thousands of open-source and free software packages can now be easily built that way, so it is useful to memorize that command line.

After a successful build, you can cleanup with

```
% make clean
```

To prepare for bundling the project into an archive for distribution to others, you might run something like this, assuming that your current directory is named `project-3.2.1`:

```
% make distclean
% cd ..
% tar cfz project-3.2.1.tar.gz project-3.2.1
% cp project-3.2.1.tar.gz ~/public_html/software/project/
```

Such commands would be conventionally provided in the `Makefile` under a target labeled `dist`.
The syntax of the `Makefile` is not hard to learn, because there are only *four* different kinds of nonempty lines in it:

1. *Comment lines* begin with optional whitespace and a sharp (`#`), and are ignored. Blank lines are also ignored.

2. *Definition lines* look like `name = value`, and define a name, normally spelled in uppercase letters, that can be used to recover its value elsewhere in the `Makefile` with `${name}` or `$(name)`. If the name is only a single letter, then the parentheses or braces may be omitted: `$X` is equivalent to `${X}` and `$(X)`.

3. *Dependency lines* look like `name1 name2 ... : dep1 dep2 ...`. The *target* names to the left of the colon depend on the names to the right. The names may be files, or targets in the `Makefile`.

4. *Command lines* begin with a tab character (*not* spaces), and follow a dependency line to supply the shell commands to be executed to make the targets up to date.

In addition, there are built-in rules for common dependencies in numerous programming languages, although not for TeX and LaTeX, and there are special shorthands that are convenient to use in command lines in a `Makefile`:

`$@` : The current target

`$<` : The *first* dependency.

`$$` : A single dollar sign, needed because that character is often present in shell command lines.

`$*` : In a rule definition [see below], the stem of a target, which is often the basename of a file. If the target is `mydoc.tex`, then `$*` expands to `mydoc` (without the trailing dot).

Here is a simple example relevant for a LaTeX document:

```
CP          = cp -p
FIGFILES    = fig1.pdf fig2.pdf
MYDOC       = mydoc
PDFLATEX    = pdflatex
RM          = rm -f
WEBDIR      = $(HOME)/public_html/doc

all:        pdf
```

```
check:
        @echo
        @echo There is no validation site yet for this project
        @echo

clean:
        -$(RM) *.aux *.log

distclean:  clean
        -$(RM) $(MYDOC).pdf

install:    $(MYDOC).pdf
        $(CP) $< $(WEBDIR)/

$(MYDOC).pdf:       $(MYDOC).ltx $(FIGFILES)
        $(PDFLATEX) $(MYDOC).ltx
        $(PDFLATEX) $(MYDOC).ltx

pdf:        $(MYDOC).pdf
```

Apart from the first, the target line order is arbitrary, because make reads the entire file before doing anything. Alphabetical variable and target order is common and convenient.

The four definition lines reduce repetition in the Makefile, and allow overriding its variables by command-line assignments:

```
% make MYDOC=newdoc all check install
```

The variable HOME is not defined in the Makefile; make automatically imports all environment variables from its parent shell, before executing its internal definitions and rules, which can therefore override default values from the environment. However, if you run make -e, then environment variables are imported *after* the internal ones, overriding them.

The simple command make, without arguments, requests making of the first target, which by convention is called all. That in turn depends on the target pdf, which depends on the file target $(MYDOC).pdf. That depends on the LATEX file, plus two figure files in PDF format; if any of those is newer than the target file, make must run the following command lines, which here request two invocations of pdflatex, needed to get cross references up to date.

If the targets are newer than their dependencies, then make terminates quickly with a report Nothing to do.

A leading hyphen (-) on a command line tells make that it does not matter if the command fails; normally, it terminates on the first failing command.

A leading at-sign (`@`) on a command line prevents the commands from being printed on the console.

You can test what `make` would do, without actually doing it, by giving it the `-n` (no-op) option; that is how you can check correctness of your `Makefile` rules.

The special target `.SUFFIXES` stands for the list of file extensions that appear in built-in and user-defined dependency rules. You can clear it by giving it an empty value, like this:

```
.SUFFIXES :
```

You can add to its current definition by giving it a nonempty value:

```
.SUFFIXES : .ltx .dvi .pdf .tex
```

That is useful in user-provided rules, such as these:

```
LATEX        = latex
PDFLATEX     = pdflatex
PDFTEX       = pdftex
TEX          = tex

.ltx.dvi:
        $(LATEX) $<

.ltx.pdf:
        $(PDFLATEX) $<

.tex.dvi:
        $(TEX) $<

.tex.pdf:
        $(PDFTEX) $<
```

The first of those says that to process a file with extension `.ltx` to a file with extension `.pdf`, run the command `pdflatex` with an argument of the file name of the LaTeX file. Thus, in a directory containing a file `mydoc.ltx`, the command `make mydoc.pdf` executes the command `pdflatex mydoc.ltx`. If there is a plain TeX file called `simple.tex`, then `make simple.dvi` runs the command `tex simple.tex`.

The built-in rules suffice for compiling files in common programming languages, so you may not even need to have a `Makefile`. For example, in a directory with C++, C, and Fortran program files `one.cc`, `two.c`, and `three.f`, then the command `make one two three` would run something like this:

```
CC -o one one.cc
cc -o two two.c
f77 -o three three.f
```

More information can be found in manual pages: run the command `man make`.

There is book-length documentation of the much-extended GNU version: run `info make`, or inside an `emacs` editor session, type `C-h i` to enter the `info` system, then `m make` to select the menu item for `make`. Navigation in `info` files is easy: you can learn the half dozen commands in a minute or two: just type `h` to enter the help system. Type `q` to quit from `info`. Inside an `emacs` session, if you reenter the `info` system, you are exactly where you were when you left it. That way, you can quickly switch between reading documentation, and typing your document or program.

For maximal portability of your `Makefile`, avoid use of the GNU `make` extensions.

24 Managing your document with arara

When several standard tools must be run to produce your document, you can use the `arara` program to process directives that define their order. It is available for all systems supported by recent TeX Live releases, with book-length documentation in `texmf-dist/doc/support/arara/arara-manual.pdf` under the TeX Live installation directory.

The `arara` program is useful on desktop operating systems that emphasize button-clicking over programmability and customizability, a situation that too many students find themselves in.

The sample theses have all been augmented with a special directive hidden in a comment near the start of the top-level LaTeX document file:

```
% arara: make: { }
```

You can then typeset one of them like this:

```
% arara sample-thesis-0

  __ _ _ _ __ __ _ _ __ __ _
 / _' | '__/ _' | '__/ _' |
| (_| | | | (_| | | | (_| |
 \__,_|_|  \__,_|_|  \__,_|

Processing 'sample-thesis-0.tex' (size: 15 KB, last modified:
03/01/2019 15:55:01), please wait.

(Make) The Make program ............................... SUCCESS
```

```
Total: 3.75 seconds
```

If your system lacks the make program, you can approximate it with comment directives like these:

```
% arara: clean:      { extensions: [ aux, bbl, idx, ind, lof, log, lot, toc ] }

% arara: pdflatex:  { }
% arara: bibtex:    { }
% arara: makeindex: { }

% arara: pdflatex:  { }
% arara: bibtex:    { }
% arara: makeindex: { }

% arara: pdflatex:  { }
% arara: bibtex:    { }
% arara: makeindex: { }

% arara: pdflatex:  { }
```

Running arara would first remove files with the extensions given in the clean directive, then run the tool groups three times for consistency reasons, and finally run pdflatex a fourth time to complete the job.

If you use a GUI for LaTeX processing, you may be able to customize it with a button to run arara; otherwise, you need to open up a console or terminal window, and run it there.

Unlike make, arara has no understanding of file dependencies, and no ability to run arbitrary shell commands, as we need to do for the complex processing needed for the multiple indexes and the bibliographies in the sample theses. Thus, running arara a second time repeats the steps unnecessarily, whereas a second make run would avoid all of the steps.

25 Experiments

The double spacing that pervades typical university dissertations and theses is a relic of the days of typewriters, when it was expensive, or impractical, to retype document pages. The extra vertical space gave editors room for corrections and comments, and for that purpose alone, double spacing is still useful today. However, such spacing decreases readability, and makes the document look less like the book that it really is, and increases its length and printing costs.

Here are two experiments that we have explored to return to normal spacing for the front matter, and for reference lists, neither of which requires much proofreading, because both should be automatically generated, by LaTeX from sectional commands and float captions, and by BibTeX from the bibliographic databases. The experiments require minimal changes to the top-level LaTeX file, and if they are acceptable to, and preferred by, the Graduate School thesis editors, we may consider making them the default.

First, change the spacing of the three lists in the front matter:

```
\begin{singlespace}
    \tableofcontents             % MANDATORY
    \listoffigures               % MANDATORY (if there are figures)
    \listoftables                % MANDATORY (if there are tables)
\end{singlespace}
```

Second, include a preamble package to change reference-list spacing:

```
\usepackage {uuthesis-singlespace-bib}
```

That style file is a two-line modification of a 20-line definition of the LaTeX `thebibliography` environment: it inserts a `\singlespace` command immediately after the unnumbered list heading produced in the *begin* block, and a `\doublespace` command at the end of the *end* block.

A more careful design would instead use TeX grouping to limit the scope of the `\singlespace`, but that will take a few more changes to the environment definition. It would also have to handle the case of chapter bibliographies, which it currently ignores.

Vertical spacing is largely controlled by two settings in the **uuthesis-base** style file:

```
\newcommand {\doublespace}                {\setlength {\baselineskip} {19pt}}
\newcommand {\singlespace}                {\setlength {\baselineskip} {13pt}}
```

Notice that they are *not* parametrized to the document point size, so if you typeset your document in an enlarged font size for visually impaired readers, you need to override those settings, which are appropriate for the recommended 11pt choice.

Notice also that the `\doublespace` command does *not* double the 13pt - 11pt = 2pt interline space, nor does it double the vertical distance between successive text baselines. To do that, its value would have to be increased to 26pt.

It might be better to instead define the `\doublespace` command in terms of the LaTeX standard `\baselinestretch` command, like this:

```
\renewcommand {\baselinestretch} {1.461}    % == 19pt / 13pt
\newcommand    {\doublespace}
              {\setlength {\baselineskip} {\baselinestretch \normalbaselineskip}}
```

That way, a draft printing with extra space for corrections could be produced with a single preamble command

```
\renewcommand {\baselinestretch} {1.75}
```

and then changed to

```
\renewcommand {\baselinestretch} {1}
```

for a final single-spaced document.

Of course, whenever typesetting parameters are changed, it is essential to recompile the document from scratch. If the `make` utility is used, which is strongly recommended, then all that you need to do is run the command

```
% make distclean all
```

The `distclean` target removes all automatically generated files, and the `all` target runs the production tools the correct number of times to ensure consistency of cross referencing and indexing.