

# Unofficial UW Computer Science Thesis Template

by

Your Name Here

A thesis  
presented to the University of Waterloo  
in fulfilment of the  
thesis requirement for the degree of  
Doctor of Philosophy  
in  
Computer Science

Waterloo, Ontario, Canada, 2008

©Your Name Here 2008



**AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Your Name Here



# Abstract

The abstract should be maybe 200 words or so describing what your thesis is about. Remember that it'll end up going into library databases, and so it's probably a good idea to avoid using math notation that won't survive that process. If possible, stick to stuff that will look good in flat ASCII.



# Acknowledgements

Acknowledgements go here. It's usual to thank your supervisor, and your spouse or partner if you have one. If you're on a scholarship like NSERC you're probably obligated to credit that here also.





*Dedication goes here,  
if you're using one.*



# Table of Contents

<b>Author's Declaration</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Dedication</b>	<b>ix</b>
<b>Table of Contents</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 The template file</b>	<b>3</b>
2.1 Front matter . . . . .	3
2.2 Back matter . . . . .	4
2.2.1 Bibliography . . . . .	4
2.3 Checklist for using the template file . . . . .	5
<b>3 The mskthesis style</b>	<b>7</b>
3.1 Other packages . . . . .	7
3.2 Fonts . . . . .	7
3.3 Theorems and theorem-like environments . . . . .	8
3.4 Maintenance hooks . . . . .	10
3.5 Clever references . . . . .	10
3.6 Support for EPS figures . . . . .	11
3.7 Definitions and index entries . . . . .	12
3.8 Short section names . . . . .	14

3.9 Math notation shortcuts . . . . .	14
3.10 PDF bookmarks . . . . .	15
3.11 Bibliography and index intro paragraphs . . . . .	15
3.12 Result summary tables . . . . .	15
3.13 Other minor formatting tweaks . . . . .	15
<b>4 Other stuff in this package</b>	<b>17</b>
<b>5 Conclusion</b>	<b>19</b>
<b>Bibliography</b>	<b>21</b>
<b>Index</b>	<b>23</b>

## List of Tables



## List of Figures





# Chapter 1

## Introduction

This template is based on the style file and other infrastructure I developed for writing my own PhD thesis [5]. I think that when properly used it'll result in output that both looks good and satisfies the University of Waterloo's requirements for electronic thesis submission, though obviously those requirements are constantly in flux and I can make no promises. They did accept my electronically-submitted thesis on the first try.

For my actual thesis I used a couple of layers of additional code (Perl scripts and such) to automatically manage file dependencies and keep count of TODO and FIXME items. Much of that is too ugly to share, though I may clean it up in the future if this stripped-down and cleaned-up template proves popular. For the moment, I'm assuming that people who want to use such systems will prefer to build their own.

This package was designed for a computer science thesis. It's assumed that the user is an intermediate to advanced-level  $\text{\LaTeX}$  hacker.

This template is released to the public domain, notwithstanding the copyright notice on the front page (which, of course, is only included so you'll know what one should look like in your own thesis).

—Matthew Skala



## Chapter 2

# The template file

The main template is in the file *template.tex*. You'll probably want to rename it to something like "thesis.tex" for actual use. Note that some of my utility programs assume that name. You'll probably also want to put your individual chapters in separate  $\text{\TeX}$  files and include them with `\input`.

### 2.1 Front matter

The regulations specify this front matter:

- Title page
- Abstract
- Acknowledgements
- Dedication (optional)
- Table of Contents
- List of Tables
- List of Illustrations

There are specified page numbers for each of these things but those can be ignored if appropriate; the actual point is that in two-sided printing they should each start on a right-hand (odd-numbered) page. The thing called "List of Illustrations" would normally be called a List of Figures in computer science, and I confirmed with GSO that making that substitution is acceptable. I preferred not to put a heading of "dedication" on my dedication page; that also seems to be in line with common practice.

I think it makes sense to list all the front-matter items in the Table of Contents.  $\LaTeX$  will not do that by default; the template file includes `\addcontentsline` commands to insert ToC entries for the front matter items. There is also an issue of how to list them in the PDF bookmarks. By default these are all chapter-level entities. I prefer to have a fake chapter called “Front matter” with all the front-matter items appearing inside it as if they were at the section level instead. To accomplish that, I run a utility (included) called *fixbookmarks* after each run of *pdflatex* to bump the front-matter bookmarks down a level. See the *Makefile* for its use. This step is optional; without it, you just get the front-matter items appearing at the chapter level.

fixbookmarks

Front matter should be numbered with lower-case Roman numerals (i, ii, iii, iv, and so on) and then the numbers start over at 1 with the first page of the main matter and continue on into the back matter. The template does that.

## 2.2 Back matter

My template assumes that there’s going to be a Bib $\TeX$  bibliography and an index built with `makeindex`, so there are chapter-level items added for those. Modify to taste. There are also some commented-out lines at the end for generating a list of TODO items. In my actual thesis I used an elaborate set of scripts for tracking the number of outstanding TODO items, and I don’t want to document and share all of that at this point; but if you want to get a simple list, you can uncomment the relevant lines at the end of the template.

Beware of leaving change-tracking features uncommented in the final version. I screwed up with that myself and ended up getting my introduction named “Introduction (0,1)” in the final electronic version posted through UWSpace.

### 2.2.1 Bibliography

thesis.bib  
conferences.bib

The bibliography databases are assumed to live in *thesis.bib* and *conferences.bib*. See those files for example data. I’ve included my actual *conferences.bib* file because it’ll probably be useful to others; I used the cross-reference feature of Bib $\TeX$  extensively, so when I cite a conference paper I have an entry for the paper itself [4] which just gives authors, title, page numbers, and a cross-reference to the conference’s own entry [2].

The  $\LaTeX$  default behaviour is to roll the information for the cross-referenced item (that is, the name of the conference, who edited the proceedings, and so on) into the bibliographic entry for the individual paper if you cite only one paper from a given conference, but if you cite more than one, then the entries for individual papers just say “in such-and-such other volume” and point at a unified

entry for the whole conference. I made a few changes to this behaviour for my own purposes.

First, I hacked the bibliographic style to refer to conferences by their names instead of who edited the proceedings; if you refer to let's say SPIRE'05, you want to look it up as SPIRE'05, not "Consens and Navarro" [1]. That functionality is included with this template. One reason for it is that it's often difficult or impossible to figure out who edited a volume of proceedings (not all have editors listed, especially not in convenient online sources), and with no disrespect to proceedings editors, I don't think that the names of the editors are really the key by which people look for conference proceedings.

Also, in my actual thesis I used a separate and not included Perl script to generate `\nocite` entries for all the collections (mostly conference proceedings) I mentioned, so that even if I only cited one paper from SISAP'08 [3], I'd get separate entries for the paper and the conference. My reason for doing that was that people don't really want the details of where the conference was and who edited the proceedings when they look up an individual paper—the interesting thing for someone reading the bibliography was "It was in SISAP'08 and I can look up just what that means with this handy number, should I want to." But that's a bit more of a bold decision. It's not standard comp sci practice, and although my committee accepted it, I'm not sure I can recommend it to you. Anyway, if you want to do exactly as I did, you'll need to generate your own `\nocite` entries or ask nicely for a copy of my script.

Of course you can use a different Bib<sub>T</sub><sub>E</sub>X style instead of my custom one, if you wish.

I like to organize my entries with keys consisting of the first author's surname, a colon, and the first word I think of as major in the title. So, for instance, my thesis is indexed under `Skala:Aspects` [5]. In the case of authors with accents and stuff in their names I just bash it into plain unaccented ASCII. Within the *thesis.bib* file I keep the entries in alphabetical order, and I have lines like `%%A%%`, `%%B%%`, and so on to mark the sections starting with A, B, and so on. Those strings would not naturally occur in the file in any other context. That makes it easy to navigate with my text editor's search feature—I could just search for the tag at the start of whichever letter-section I wanted and then page down to the right spot. My final bibliography file contained about 500 entries, so some kind of organization like this was a must, but of course you'll probably have your own preferred way of doing it.

## 2.3 Checklist for using the template file

Be sure you do all of these things:

- Put your title and name into the `\title` command and its enclosed `\author` command.
- Put your title and name into the *hyperref* options in the *mskthesis.sty* file. Look for the “FIXME FIXME FIXME” line in that file. This should be the only thing in the style file that’s absolutely necessary to edit, but see the next chapter for more about this file.
- Put your title onto the title page. If it’s a long title, use `\\` to insert line breaks as shown.
- Put your name onto the title page (two places).
- Put the date onto the title page (two places).
- Put your degree and field into the title page if it’s not going to be a PhD in comp sci.
- Put your name into the Author’s Declaration.
- Replace the sample Abstract, Acknowledgements, and Dedication, with your own material; or delete the Dedication if you don’t want one.
- If you have no tables, or no figures, delete the appropriate lists.
- Replace the “main matter” with the actual text of your thesis. You will probably want to put it chapter-by-chapter in `\input` files instead of really including it in the template.
- Adjust the bibliography lines if you want to use a different Bib<sub>T</sub><sub>E</sub>X style from mine, different database filenames, or similar.
- Uncomment the TODO-list lines if you want to include a TODO list.

## Chapter 3

# The mskthesis style

Most of the formatting stuff is in the file *mskthesis.sty*. You'll want to edit it at least a little to put in your name and title for the PDF metadata, and also to take a look at the commands available. Some of them are fairly specific to my own thesis and you may want to get rid of them. This chapter of the template file documents what's in that file. I suggest reading the file yourself to get some idea of how it all fits together. mskthesis.sty

### 3.1 Other packages

My style uses many other style files, some of which should have come with your  $\text{\LaTeX}$  installation and others of which you can get from CTAN. Try typesetting this document, when it complains about not having packages get them, and so on.

Note that this is all designed to work with recent *pdflatex*. Since UW requires PDF as the submission format I think that's reasonable. I don't know how well, or if, it'll work with other versions of  $\text{\LaTeX}$ .

### 3.2 Fonts

I'm using Charter as the text font, and the MathDesign math fonts to match with it. However, I really didn't like MathDesign's lowercase delta and ended up hacking the font file to make it look better ("δ"). I've included the hacked font files in this package. I don't know how well they'll work on your system; I think including them in the current directory when you do your typesetting will cause  $\text{\LaTeX}$  to find them and use them properly, but depending on your installation it may be a little harder. Fonts in  $\text{\LaTeX}$  are, unfortunately, still a bit of a black art. Also included are some *pdflatex* config files and so on that should probably

cause it to find the fonts correctly. I can't promise that that will actually work on anyone's system but mine, though.

The MathDesign Charter math fonts include “formal script” math script capitals, such as are typically used by pure-math people and physicists. Those look great as long as you use them one at a time; however, in my thesis I used script for complexity classes (like  $\mathcal{NPC}$ ), with two or more script letters side by side, and the formal script letters look really bad when you do that. They only work when used one at a time. I specified the `cal` option to the `mathdesign` package to make it revert to the default  $\text{\LaTeX}$  calligraphic font for math script, which looks better when the script letters are used next to each other. If you're sure you won't be using more than one script letter at a time, you might like to remove this option and go to MathDesign's default of formal script.

I also specify Helvetica for sans-serif, and later in the style file specify it for most headings. Some people think Helvetica is overused, but I like the way it looks in my thesis. Don't override it unless you're sure you know what you're doing. On my system so-called Helvetica is actually something like “Bitstream Nimbus Sans,” but that's close enough. Arial is not close enough. Don't use Arial. Fortunately,  $\text{\LaTeX}$  probably won't let you anyway.

### 3.3 Theorems and theorem-like environments

I define a bunch of theorem-like environments as follows, using the `ntheorem` package. A copy of `ntheorem` is included with this template. Be aware that that package does get updated from time to time and new versions may or may not work well with my code; that's why I included a copy of the one I used.

#### **Theorem 3.1**

This is a theorem created with `\begin{theorem}... \end{theorem}`.

#### **Lemma 3.2**

This is a lemma created with `\begin{lemma}... \end{lemma}`.



**Corollary 3.3**

This is a corollary created with `\begin{corollary}... \end{corollary}`.

**Example 3.1**

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

**Conjecture 3.2**

This is a conjecture created with `\begin{conjecture}... \end{conjecture}`.

**Definition 3.3**

This is a definition created with `\begin{definition}... \end{definition}`.

**Note 3.4**

This is a note created with `\begin{note}... \end{note}`.

**Proof** This is proof created with `\begin{proof}... \end{proof}`. ■

Some people like to put theorem statements in italics. I don't; I think it makes them hard to read. Note also that the UW thesis regulations say lengthy blocks of italics should be avoided (because of being hard to read) although they don't specifically forbid, and you could probably get away with, the common usage of italics for theorem statements. My style puts theorem statements in boxes instead, to make them stand out on flipping through the document while keeping the contents of the box in normal text style. Using boxes in this way, when combined with the other packages I'm using, requires some low-level  $\TeX$  hacking to make it all work right. See the style file for the code if you're interested.

Boxed theorem-like things (theorems, lemmata, and corollaries) share one set of numbers, and unboxed theorem-like things (examples, conjectures, definitions, and notes) share another set of numbers. That's not specified by the rules; my reason for doing it is that having fewer sets of numbers makes it easier for people to navigate the document. If I'm looking for "Corollary 1.2" in a chapter that might have only two Corollaries and a bunch of Theorems and Lemmata all numbered separately, then it could be almost anywhere—only finding Corollary 1.1 could help me—whereas if Corollaries share their numbers with Theorems and Lemmata, then every time I see a Theorem or Lemma I also get a clue for where to find Corollary 1.2.

Proofs, as you can see, don't get numbers but do get square "QED" symbols in understated grey-green. This colour is defined in the style file as `pmcolour`. The reason for it is so that the QED symbols won't draw too much attention to

themselves on the page (as they might if they were solid black, for instance). In black-and-white printout they come out as grey. The proof environment should follow, but not be included inside, the corresponding theorem or similar environment.

### 3.4 Maintenance hooks

While writing your thesis you may want to include notes to yourself about things to add or change later. You can add a TODO item with `\TODO{do this later}`; the result looks like **do this later**. You can add a FIXME item with `\FIXME{this should be done too}`; the result looks like **this should be done too**. Note that the TODO gives you a circle in the margin, which you can check off on your printout when you do whatever it is you were supposed to do, and the FIXME gives you a script  $\mathcal{F}$ . Both also get entered in the “list of todo items” which will appear at the end of your thesis if you uncommented those lines in the template.

In my actual thesis I also had a script that would count how many of these existed per chapter and add them to the chapter headings so that I’d be able to see the counts in the table of contents. In retrospect, that was probably a bad idea. It made the *Makefile* too complicated, among other things.

### 3.5 Clever references

I used the `cleveref` package to do smart references, and then added some features of my own, and then after I talked to them about it the `cleveref` people added features too, incorporating some of my ideas, so I ended up with a document that may or may not be compatible with the latest version of `cleveref`. You can explore upgrading if you want, or use my hacks and the old version of `cleveref` that I used, which is included with this template. There’s also some nasty code in there to make it play with `hyperref`, which does not in general play nicely with other packages. The `cleveref` people claim that this is `hyperref`’s fault, and I think they’re probably right, but to you and me it doesn’t matter who’s right, as users we just want our documents to work. Anyway, the main user-level commands for the resulting kludged-together system are `\cref` and `\Cref`.

To typeset a reference, whether it’s to a theorem, a lemma, a note, a figure, or whatever, use a command like `\cref{label}`. That should automatically pick up what kind of thing you’re referring to and produce text like “Figure 1.2” or “Lemma 3.4.”

The `cleveref` package normally uses the distinction between `\cref` and

`\Cref` to tell whether the name should be capitalized or not, like “figure 1.2” or “Figure 1.2.” However, I wanted it to be capitalized all the time anyway, so instead, I made the `\Cref` form generate an abbreviated form, like “Fig. 1.2” and “Lem. 3.4.” I used that occasionally in tables where I was short on horizontal space.

As a further exception, references to what  $\TeX$  called “equations” don’t get a name at all with `\cref`, they just appear as things like “(5.6),” in order to avoid addressing the issue of whether something like

$$1 < 2 \cup 3 \tag{3.1}$$

which doesn’t actually say that two things are equal, should really be called an “equation.” If you use `\Cref` for it it’ll be called an Equation anyway, damn the torpedoes, which is what you need at the start of a sentence.

You can use multiple labels separated by commas in the `\cref` or `\Cref` commands and it should do the right thing, even when there are multiple different kinds of labels included, so as to (in the extreme) generate references like “Theorem 1.2; Lemmata 3.4 and 5.6; and Corollaries 7.8, 9.10–9.15, and 11.12.” I think you have to sort them yourself, though; it will put the numbers out of order if you cite them out of order.

My modifications to the `cleveref` package, embodied in the enclosed style file, cause it to use what is known as “the serial comma” or “the Oxford comma.” This is a controversial point. Some people like the serial comma. I do. Others do not like it. It is not the default in `cleveref` and (although it’s subsequently been added as an option) there was no support for it in the version of `clever` I ended up using. One can make the argument that the UW thesis regulations may require the serial comma, because they endorse the *Chicago Manual of Style*, and the *Chicago Manual of Style* says you should use the serial comma. Really, though, nobody’s going to notice whether you use it or not—probably not even if you’re in the English department! Anyway, if you care about this point then you’re free to read my source code and investigate it further.

I recommend, though the code does not require this, that it’s a good idea name your labels with prefixes specifying what they are, such as “fig:figure-label,” “thm:theorem-label,” and so on.

### 3.6 Support for EPS figures

If you have a simple EPS file you want to turn into a figure (which you’ll actually need to convert to a PDF file to make *pdflatex* pick it up properly) you can do that quickly with the `\myepsfig` macro, which takes three arguments: the scaling

factor, the filename, and the caption. A label will be created with a name like “fig:filename,” obeying the naming convention mentioned above. I also created a couple of macros called `\tallgpscale` and `\widegpscale` for scaling factors I used repeatedly with GNUplot output; those are likely not to be useful to you and you can probably delete them.

### 3.7 Definitions and index entries

defined term

I like to put each *defined term* in italics when I define it, and add a marginal note for it, so that the definition will be easy to find. The `\defnterm` macro does this. With one argument, that gets typeset in italics, stuck in the margin, and an index entry is created.

Then there are more complicated things you can do. With something like `\defnterm[goose]{geese}`, the word “geese” gets written in the text, but the marginal note and index entry will both be “goose.” That’s useful if you want to mention something in the plural but index it in the singular, or vice versa. In general, you should make a firm decision and stick to it about which form will be indexed. I settled on singulars in the index.

I wanted page numbers in the index to appear in a different (sans-serif) font for definitions from the one used for other references. That’s accomplished with the `\indexdef` macro, which is a hook used by the bar syntax of `\index`, as in `\index{foobar|indexdef}`. See the [L<sup>A</sup>T<sub>E</sub>X](#) indexing documentation for more about that.

Sometimes you may want the behaviour of `\defnterm` without creating an index entry. In that case use `\defntermni`, which is otherwise identical to `\defnterm`.

I wanted to make it easy to find things that might or might not be indexed in inverted form. For instance, a “pink unicorn” should be found under both “pink unicorn” and “unicorn, pink.” I decided that the main entry should be “unicorn, pink” and then “pink unicorn” should say “see unicorn, pink.” To help accomplish this kind of thing I created several helper macros for creating index entries:

- `\indextwo{pink}{unicorn}` creates a reference under “unicorn, pink” with the page number and one under “pink unicorn” saying “see unicorn, pink.”
- `\indextwo[one-horned horse]{pink}{unicorn}` creates a reference under “one-horned horse, pink” with the page number and one under “pink unicorn” saying “one-horned horse, pink.” The idea here is that the optional argument replaces the main heading, but only in the main entry. In practice

I used that for grouping together named things, like the “Central Limit Theorem” and “Minkowski Inequality” both under a main entry of “theorem (named).”

- `\indextwobar{pink}{unicorn}{indexdef}` is just like `\indextwo` except it takes another argument that will be used with the bar syntax of `\index`, as for definitions or see-alsos. Use it with “(” or “)” to do page ranges. It also accepts an optional argument in square brackets as above.
- `\indexthree{invisible}{pink}{unicorn}` creates a main entry under “unicorn, pink, invisible” and cross-references under “invisible pink unicorn” and “pink unicorn, invisible.”
- The `\indexthree` macro can also take an optional argument to replace the last term in the main entry, just like `\indextwo`.
- There is a `\indexthreebar` macros which has the obvious properties.

In retrospect I’m not sure whether this index organization worked well or not; there are a lot of cross-references that might have been better just reiterating the page numbers instead of sending readers to other parts of the index. Also, the definition thing didn’t work out as well as I’d hoped, because I had intended to define things each in one place and at my supervisor’s urging I ended up instead repeating a bunch of definitions, so that entries like “site” in my index end up having as many as six “here is the definition” page numbers. But maybe you can do better with yours.

As usual, you can just delete or not use these macros if you don’t want an index or want to do your index in a different way.

There is an issue with “see” entries in that if there are multiple instances of the same “see” entry on different pages, as may be created by using my macros, then `makeindex` may see them as different and cause there to be more than one “see” entry mentioned in the index. In my *Makefile* I run a Perl script just before running `makeindex`, which munges the `.idx` file to assign all “see” entries to page 999 rather than their actual page numbers. The page numbers for “see” entries do not actually appear in the index, so it’s all right, and that causes the duplicates to disappear in the sorting.

#### **Note 3.5**

This template document only has a few semi-random index entries so that you can see what an index looks like; it’s not intended to be thoroughly indexed.

### 3.8 Short section names

This is a very specific hack, but: I had one section whose name was too long to appear comfortably in the running page header. The `\shortsecname{short name}` macro was created for that; call it right after the `\section` macro, with a shortened version of the section name to put in the running page header.

### 3.9 Math notation shortcuts

You'll probably want to delete or radically change these, but here are some commands I defined for notation I used a lot.

Command	Result
<code>\disteq</code>	$\stackrel{d}{=}$
<code>\distbitends</code>	$\stackrel{d}{\longleftrightarrow}$
<code>\disttends</code>	$\stackrel{d}{\rightarrow}$
<code>\maxover{a}{b}</code>	$\max^{(a)}\{b\}$
<code>\maxover[x]{a}{b}</code>	$\max_x^{(a)}\{b\}$
<code>\minover{a}{b}</code>	$\min^{(a)}\{b\}$
<code>\minover[x]{a}{b}</code>	$\min_x^{(a)}\{b\}$
<code>\bin</code>	$\text{bin}$
<code>\enc</code>	$\text{enc}$
<code>\lcs</code>	$\text{lcs}$
<code>\E[foo]</code>	$E[\text{foo}]$
<code>\V[foo]</code>	$V[\text{foo}]$
<code>\BPP</code>	$\mathcal{BPP}$
<code>\NP</code>	$\mathcal{NP}$
<code>\NPC</code>	$\mathcal{NPC}$
<code>\P</code>	$\mathcal{P}$
<code>\UP</code>	$\mathcal{UP}$
<code>\bigcdot</code>	$\cdot$
<code>\Bigcdot</code>	$\bullet$
<code>\sT</code>	$\mathsf{T}$
<code>\sF</code>	$\mathsf{F}$
<code>\sS</code>	$\div$
<code>\sW</code>	$\star$
<code>\s0</code>	$1$
<code>\sZ</code>	$0$
<code>\AbbrevZi</code>	$\left  \frac{1}{2} - z_i \right $
<code>\AbbrevZin</code>	$\left  \frac{1}{2} - z_{i+n} \right $
<code>\AbbrevZj</code>	$\left  \frac{1}{2} - z_j \right $

Note that defining `\P` as above causes us to lose that command’s original meaning of “paragraph symbol”; not a problem in my case because I didn’t need that symbol. The `\bigcdot` and `\Bigcdot` commands are for multiplying together big things like integrals where a regular centre-dot looks too small; or when I have a chain of things strung together with regular centre-dots and then a centred ellipsis in between, where I’d like some contrast between the multiplication dots and the ellipsis dots.

I also redefined a couple of existing math commands to use notational conventions I preferred: `\tfrac{a}{b}` now looks like  $\frac{a}{b}$  (it formerly put the numerator directly above the denominator like  $\frac{a}{b}$ ) and `\vec{x}` now looks like  $\mathbf{x}$  (formerly a yucky arrow-accent).

### 3.10 PDF bookmarks

To support having a nice PDF outline (usable in some PDF readers) corresponding to the structure of the document, there’s a `\mybookmark` macro defined. It’s used by the template to add links for unnumbered chapter-like things such as the table of contents. It shouldn’t be necessary for you to call it yourself.

### 3.11 Bibliography and index intro paragraphs

There are macros defined to give a couple of explanatory sentences for page numbers in the bibliography and index. These, too, you probably don’t need to touch unless you’re using different conventions from mine.

### 3.12 Result summary tables

I created a special environment for some tables I was making in my introduction; the `resultsummary` environment which creates a box with special paragraph formatting inside, and the `\resultsummaryhead` which makes a heading to be used inside the box. You can probably ignore or delete these; they are included for completeness.

### 3.13 Other minor formatting tweaks

See the bottom of the style file. I redefine paragraph indentation and baseline stretch; make the page numbers in the table of contents just a little wide because with three-digit page numbers and the fonts I’m using, that turns out to be necessary; and rename the Table of Contents to be a Table of Contents (instead

of just a “Contents”) because that’s specified by the thesis regulations. Really they probably wouldn’t care, but.

And I tell the interpreter to list the included files in the log (which is useful for automatically generating Makefile dependencies), and that breaks in displayed math should be allowed, because I have a lot of long displays and would have a lot of short pages without that.



## Chapter 4

### Other stuff in this package

This template comes with a few other little goodies that may be of interest. Here's a summary of the files included.

*MDCH.sfd*: FontForge source file for my modified font (with the nicer lower-case delta).

*Makefile*: a very basic makefile showing the sequence of utilities to run to generate a PDF file, with reference resolution and so on.

*cleveref.\**: The `cleverer` package in the version I used.

*conferences.bib*: A large, but not complete, collection of BibTeX records for computer science conferences. Be warned that I have normalized the titles of most conference proceedings volumes to all follow substantially the same format (in terms of things like whether to list numbers as numerals or words and whether to call something “Proceedings of X” or “X, Proceedings”) even if the result is not exactly what is listed on the title pages of the physical volumes. Depending on your point of view that may either be shockingly bad scholarship, or a great moral victory for the forces of sanity. My committee passed it, anyway.

*count*: A Perl script that generates a bunch of progress-indicator counts based on the assumption that the main thesis file is called *thesis.tex*. I ran this periodically and dumped the results onto my home page so that my visitors could share my joy.

*fixbookmarks*: Already mentioned—a Perl script that modifies the PDF bookmarks file to move front matter chapter-level items into section-level items filed under “Front matter.”

*framed.sty*: The version I used of the `framed` package, which might be worth keeping around because that package interacts with others and may be “fixed” in an incompatible way in the future.

*gennocites*: Generates `\nocite` commands to ensure splitting conference proceeding volumes into separate bibliographic entries even if they are only used

once. Nonstandard practice, use at own risk.

*gpepsmod* and *gpepsmod.dat*: Changes the EPS files output by GNUpLOT to be prettier (solid, coloured lines instead of dash patterns).

*hacked-epstopdf*: Slightly modified version of *epstopdf* with extra commands to force it to always embed fonts and never subset them. I think that's preferable to ensure that your figures will be reproducible even on systems that (for instance) think Arial is an acceptable substitute for Helvetica.

*md-chri7m.\**: Modified MathDesign Charter font (with nicer delta).

*mdbch.cfg*: MathDesign Charter config file causing it to use small caps.

*mskslides.sty*: Bonus undocumented style file! Creates slides that look similar to the style of the thesis template. I used this for my defence slides.

*mskthesis.bst* and *mskthesis.sty*: Bibliography, and  $\text{\LaTeX}$  document, style files.

*ntheorem.\**: The *ntheorem* package in the version I used.

*pdftex.\**: Config files for *pdflatex*. I don't know if these are sufficient to make the fonts work right, but they may help. Or not. If you have trouble, try deleting them.

*template.pdf*: This document.

*template.tex*: Source code for this document.

*thesis.bib*: Bibliographic database for this document, containing just a few entries to show the format.

## Chapter 5

## Conclusion

That's it. Have fun. Visit my web page at <http://ansuz.sooke.bc.ca/>.



# Bibliography

The numbers at the end of each entry list pages where the reference was cited. In the electronic version, they are clickable links to the pages.

- [1] Mariano P. Consens and Gonzalo Navarro, editors. *String Processing and Information Retrieval: 12th International Conference (SPIRE'05)*, volume 3772 of *Lecture Notes in Computer Science*, Buenos Aires, Argentina, November 2–4, 2005. Springer. [5](#), [21](#)
- [2] *Proceedings of the 1st International Workshop on Similarity Search and Applications (SISAP'08)*, Cancún, Mexico, April 11–12, 2008. IEEE Computer Society. [4](#), [21](#)
- [3] Matthew Skala. Measuring the difficulty of distance-based indexing. In SPIRE'05 [\[1\]](#), pages 103–114. [5](#)
- [4] Matthew Skala. Counting distance permutations. In SISAP'08 [\[2\]](#), pages 69–76. [4](#)
- [5] Matthew Adam Skala. *Aspects of Metric Spaces in Computation*. PhD thesis, University of Waterloo, 2008. [1](#), [5](#)



# Index

Page numbers printed like **123** indicate definitions. Other relevant page numbers are printed like 123.

clever references, *see* references, clever

conferences.bib, **4**

copyright

author's declaration, **iii**

defined term, **12**

fixbookmarks, **4**

index entries, **12**

mskthesis.sty, **7**

recursion, *see* recursion

references

clever, **10**

template.tex, **3**

theorems, **8**

thesis.bib, **4**