

Example of the Use of the Style for Writing Theses at CS, SFU

by

Brown Burger

M.Sc., Even Better University, 1994

B.Sc. (Hons.), Even Better University, 1990

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

© Brown Burger 2013
SIMON FRASER UNIVERSITY
Fall 2013

All rights reserved.

However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name: Brown Burger
Degree: Doctor of Philosophy
Title of Thesis: Example of the Use of the Style for Writing Theses at CS, SFU

Examining Committee: Dr. Brian Funt
Chair

Dr. Torsten Möller,
Professor, Senior Supervisor

Dr. Ghassan Hamarneh,
Associate Professor, Senior Supervisor

Dr. Mirza Faisal Beg,
Associate Professor, SFU Examiner

Dr. Christopher R. Johnson,
External Examiner, Distinguished Professor of
Computer Science, University of Utah

Date Approved: September 9th, 2013

Partial Copyright Licence



The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU's own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU's rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author's written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author's knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013

Abstract

Here you put the abstract of the thesis. adkgakfhlsufp osudif sDHOs dfus sdfgsuduf sf

To whomever whoever reads this!

“Don’t worry, Gromit. Everything’s under control!”
— *The Wrong Trousers*, AARDMAN ANIMATIONS, 1993

Acknowledgments

Here go all the people you want to thank.

Contents

Approval	ii
Partial Copyright License	iii
Abstract	iv
Dedication	v
Quotation	vi
Acknowledgments	vii
Contents	viii
List of Tables	x
List of Figures	xi
List of Programs	xii
Preface	xiii
1 Purpose	1
1.1 Main features of csthesis.sty	1
1.2 Distribution	2
1.3 How to use csthesis.sty	3
2 Changes from sfuthesis.sty	4
2.1 Page layout	4
2.2 Line spacing	5
2.3 Front matter	5
2.3.1 Title page	5

2.3.2	Approval page	5
2.3.3	Dedication and Quotation pages	6
2.3.4	Contents	6
2.3.5	Preface	6
2.4	Back matter	6
2.4.1	Appendices	6
2.4.2	Bibliography	6
2.4.3	Index	7
3	introduction	8
3.1	literature review	8
3.2	what are the algorithms in keyword extraction	8
3.3	our approach our algorithm	11
3.4	visualization of trends	11
3.5	conclusion	11
	Bibliography	12
	Appendix A Appendices:Sectioning	14
A.1	Section	14
A.1.1	Subsection	14
	Appendix B Illustrating Lists	15
B.1	List of Programs	15
B.2	Formatting of Lists	16

List of Tables

2.1	Example of a table	5
B.1	First meaningless table in Appendix B	16
B.2	Second meaningless table in Appendix B	16

List of Figures

2.1	Example of a figure	5
-----	-------------------------------	---

List of Programs

B.1	Example of the new program environment	15
B.2	Second program	15

Preface

Here go all the interesting reasons why you decided to write this thesis.

Chapter 1

Purpose

This document outlines the functionality of the style file (package) `csthesis.sty`, suitable for writing theses in $\text{\LaTeX}2_{\epsilon}$ at SFU. The style file modifies and generalizes the `sfthesis.sty`, designed by Margaret Sharon from ACS. The changes were done for two reasons: 1) to reflect the most recent regulations on SFU theses;¹ and 2) to improve the overall look of the theses. The changes will be illustrated and commented on in Chapter 2; the following section summarizes the main features of the present package.

1.1 Main features of `csthesis.sty`

- fully conforms to the most recent regulations on SFU theses;
- offers a visually pleasing design;
- supports all optional thesis sections (Dedication, Quotation, Preface, etc.) and simplifies their inclusion;
- supports the `twoside` option of \LaTeX and coordinates it with two-sided printing—this is not allowed for the library copies, but you should definitely consider it for other copies;
- comes with a full documentation, describing all available commands and identifying which parts of the code can be modified and how to achieve specific effects (like adding/removing white space between signatures on Approval page);
- comes with two files, illustrating the use of the package in writing theses (you are reading one of them). The following section describes all the files in the distribution in more detail.

¹ Please refer to <http://www.lib.sfu.ca/theses> to check for any changes to requirements.

1.2 Distribution

The style file `csthesis.sty` is accompanied by several supporting files. These are:

- `READ.ME`: the file containing the copying and distribution limitations, the installation instructions, the address where to sent bug reports, and the listing of all the files included in the `csthesis` bundle;
- `csthesis.ins`: the installation file for `csthesis.sty`. The bundle comes unpacked and ready-to-use, so you don't need to generate `csthesis.sty` this way, but the file is included to allow for a more compact, packed version of the bundle.
- `csthesis.dtx`: the documented code for `csthesis.sty`. It lists all the user commands defined by the style file and explains the code. In particular, it points out the places where the code should be modified in case you run into troubles like not being able to fit the Approval section on one page.
- `thes-short.tex`: an example of a thesis written using `csthesis.sty`. The thesis is “short” in that it contains only the required parts of the front matter (Abstract, Acknowledgment, Contents) but no optional parts (Dedication, Quotation, List of Figures, List of Tables, List of “other things,” Preface). In addition, the thesis does not contain the optional parts of the back matter—appendices and Index.
- `thes-full.tex`: an example of the “full” version of the same thesis, containing all the optional parts from the front matter, several appendices, and Index.
- the `files/` subdirectory: the auxiliary files for `thes-short.tex` and `thes-full.tex`.
- the `doc/` subdirectory: the “printouts” of the documentation for the package and for the two example theses.

You can use either of the two versions of the thesis as a guide in designing your own thesis file—but it's in your interest to take a look at both. The difference between the two versions is accomplished by 1) the switches located right under the beginning of the `document` environment, 2) inclusion/exclusion of the files containing optional thesis parts (Preface, appendices, Index, etc.), and 3) the inclusion of commands for generating the List of “other things” in the preamble of `thes-full.tex`.

The two thesis files demonstrate the use of the commands provided by the `csthesis.sty` package. If a more thorough explanation of some command(s) is required, consult the `csthesis.dvi` guide.

1.3 How to use `csthesis.sty`

If you're starting writing from scratch, all you need to do is to put the following two lines in the beginning of your thesis document:

```
\documentclass[11pt]{report}%% or 10pt, or 12pt
\usepackage{csthesis}
```

This makes all the commands defined in `csthesis.sty` available to you; you can consult the two examples theses and the `csthesis.dvi` to see what commands you have at your disposal and how they should be used.

If you're upgrading from `sfthesis.sty` to `csthesis.sty`, you also need to include the two lines mentioned above. In addition, though, you need to change some things in your thesis document because the two thesis formats are not fully compatible. In particular, you should:

- replace the `\afterpreface` command of `sfthesis.sty` with `\lists` and `\beforetext`, inserted in appropriate places (see the example theses);
- in case you're having one or more appendices, issue `\appendix` command after the last chapter of your thesis before the first appendix. In addition, treat each appendix as a chapter, i.e., the main heading of the appendix should use the `\chapter` command.

These should be all changes you need to do—if there is more, let me know (I haven't used `sfthesis.sty`, so I am to some extent guessing here based on the coding differences between the two packages).

Chapter 2

Changes from sfuthesis.sty

2.1 Page layout

All the changes in this section are not required by the regulations; they were made for better visual impact.

The four margins were set by sfuthesis.sty to the values recommended in the regulations; these were kept unchanged. The space between the header and the text was slightly reduced and made dependent on the chosen point size of the font (10, 11, or 12)¹—it now corresponds to one empty line of text.

The space between the text and the footer is set to the result of adding half an empty line to the value of the space between the header and the text. This centers the text vertically on the page with respect to the other material. If, for any reasons, you decide to adjust the vertical placement of the text, only the value of `\headsep` needs to be modified.

The first footnote is pushed slightly lower under the horizontal bar which separates footnotes from the text (see below). In addition, small vertical space separates individual footnotes on the page.²

The vertical space before the `paragraph` and `subparagraph` sections was slightly reduced from the original values set by `LATEX`. The new values work better for the line spacing used in the text (see Appendix A).

¹This text is set with 10pt size, which seems to be the best choice overall from the three possibilities.

²This is an example of a second footnote on the page. I will make it longer than the first one. As can be seen, the two footnotes are separated by small vertical space.

This is a new paragraph in the second footnote. There is no additional vertical space between paragraphs in a footnote.

2.2 Line spacing

The style file defines three line spacing lengths: `\textstretch`, `\tighttextstretch`, and `\doublestretch`. The first is used for the regular text. The second, which basically corresponds to the single spacing on a typewriter, is used for footnotes (see page 4) and inside tables (see Table 2.1) and figures (see Figure 2.1)—including captions. The last corresponds to the double spacing on a typewriter and is used for the thesis title, as required by the regulations.

This shows that single line spacing
is used inside figures.

Figure 2.1: Example of a figure

This shows the same thing for tables.

Table 2.1: Example of a table

In addition, I recommend using single spacing also for bibliography and index (if included), as it is done in this document.³ The corresponding files `files/bibl.tex` and `files/ind.tex` show how to specify that single spacing be used.

2.3 Front matter

2.3.1 Title page

The title itself is set in bold face, capitalized, and double spaced. The vertical white space was changed from fixed to stretchable (and partially shrinkable) to allow for a better fit of the material—titles have varying amounts of lines and people can have several previous degrees.

2.3.2 Approval page

Essentially the same, only made easier to modify if the text doesn't fit on a single page or if you want to change the amount of white space between signatures.

³This is now the accepted format for references.

2.3.3 Dedication and Quotation pages

The `csthesis.sty` package has a built-in support for optional Dedication and Quotation pages. Both dedication and quotation are typeset flushed right, dedication in *italics* and quotation *slanted* (both are illustrated in the full version of this document). The pages are not titled, but they are numbered, and a corresponding entry is added to the Contents table.

There are no constraints on the look of these pages imposed by the regulations, you can change the design as you see fit.

2.3.4 Contents

The table of Contents was reverted back to the original design (for `report.sty`) from \LaTeX . There are no specific requirements in the regulations as to how the table should look like, but all the changes made to it in `sfuthesis.sty`—reducing white space between chapters, putting dotted lines on the chapter level, or redefining the behavior of the `\appendix` command—looked decidedly awful and so I got rid of them.

2.3.5 Preface

The regulations allow for a Preface or Foreword section, which could contain a short blurb about “why the author came to study the subject of the thesis.” It’s probably useless because nobody will want to write anything like that, but it’s incorporated in the package for completeness. The optional inclusion of Preface necessitated the modification of the `\afterpreface` command of `sfuthesis.sty`, which was replaced by two commands: `\lists`, and `\beforetext`.

2.4 Back matter

2.4.1 Appendices

As mentioned in Section 2.3.4, `sfuthesis.sty` redefines the `\appendix` command of \LaTeX . Again, there does not seem to be any reason to do that (the regulations are silent on the formatting of appendices); and the original design looks better. Thus, the present style reverts to \LaTeX ’s original design, where appendices are treated as chapters and numbered alphabetically. (The change in numbering is accomplished by issuing the `appendix` command before the text of the first appendix.)

2.4.2 Bibliography

The `csthesis.sty` package does not influence the typesetting of Bibliography directly, but—as I said in Section 2.2—I recommend changing the line spacing from normal to single, as was done in

`files/bibl.tex`. See the (fake) Bibliography of this document as an example of the result.

2.4.3 Index

The same comment applies to the optional Index section (see `files/ind.tex` for how to adjust the line spacing and the Index section of the full version of this document for the result).

Chapter 3

introduction

motivation , primarily problem to solve

we need jobs to match skills we need to detect skills

real motivation secondary problem that we are actually solving

how to detect skills. why it is challenging and different than usual keyword extraction problems

3.1 literature review

3.2 what are the algorithms in keyword extraction

NLP introduction The automatic analysis of text involves a deep understanding of natural language by machines, a reality from which we are still very far off [3]. Hitherto, online information retrieval, aggregation, and processing have mainly been based on algorithms relying on the textual representation of web pages. Such algorithms are very good at retrieving texts, splitting them into parts, checking the spelling and counting the number of words. When it comes to interpreting sentences and extracting meaningful information, however, their capabilities are known to be very limited. NLP in fact, requires high level symbolic capabilities(Dyer 1994) including:

- 1-creation and propagation of dynamic

bindings; 2-manipulation of recursive, constituent structures, acquisition and access of lexical, semantic, and episodic memories; control of multiple learning/processing modules and routing of information among such modules; grounding of basic-level language constructs (e.g., objects and actions) in perceptual/motor experiences; representation of abstract concepts. All such capabilities are required to shift from mere NLP to what is usually referred to as natural language understanding (Allen, 1987). Today, most of the existing approaches are still based on the syntactic representation of text, a method that relies mainly on word co-occurrence frequencies. Such algorithms are limited by the fact that they can process only the information that they can see. As human text processors, we do not have such limitations as every word we see activates a cascade of semantically related concepts, relevant episodes, and sensory experiences, all of which enable the completion of complex NLP tasks such as word-sense disambiguation, textual entailment, and semantic role labeling in a quick and effortless way.[3]

what is tagging? Tagging is the process of labeling web resources based on their content. Each label, or tag, corresponds to a topic in a given document. Unlike metadata assigned by authors, or by professional indexers in libraries, tags are assigned by end-users for organizing and sharing information that is of interest to them. The organic system of tags assigned by all users of a given web platform is called a folksonomy. [17]

what is keyphrase? what is keyphrase extraction? what is the goal of extracting keyphrases? Many journals ask their authors to provide a list of keywords for their articles. We call these keyphrases, rather than keywords, because they are often phrases of two or more words, rather than single words. We define a keyphrase list as a short list of phrases (typically five to fifteen noun phrases) that capture the main topics discussed in a given document. This paper is concerned with the automatic extraction of keyphrases from text. Keyphrases are meant to serve multiple goals. For example, (1) when they are printed on the first page of a journal article, the goal is summarization. They enable the reader to quickly determine whether the given article is in the reader's fields of interest. (2) When they are printed in the cumulative index for a journal, the goal is indexing. They enable the reader to quickly find a relevant article when the reader has a specific need. (3) When a search engine form has a field labelled keywords, the goal is to enable the reader to make the search more precise. A search for documents that match a given query term in the keyword field will yield a smaller, higher quality list of hits than a search for the same term in the full text of the documents. Keyphrases can serve these diverse goals and others, because the goals share the requirement for a short list of phrases that captures the main topics of the documents. We define automatic keyphrase extraction as the automatic selection of important, topical phrases from within the body of a document. Automatic keyphrase extraction is a special case of the more general task of automatic keyphrase generation, in which the generated phrases do not necessarily appear in the body of the given document. **what is the differences between indexing and keyphrase list**

We discuss related work by other researchers in Section 3. The most closely related work

involves the problem of automatic index generation (Fagan 1987, Salton 1988, Ginsberg 1993, Nakagawa 1997, Leung and Kan 1997). One difference between keyphrase extraction and index generation is that, although keyphrases may be used in an index, keyphrases have other applications, beyond indexing. Another difference between a keyphrase list and an index is length. Because a keyphrase list is relatively short, it must contain only the most important, topical phrases for a given document. Because an index is relatively long, it can contain many less important, less topical phrases. Also, a keyphrase list can be read and judged in seconds, but an index might never be read in its entirety. Automatic keyphrase extraction is thus a more demanding task than automatic index generation. [18] **difference between keyphrase extraction and information extraction** Keyphrase extraction is also distinct from information extraction, the task that has been studied in depth in the Message Understanding Conferences (MUC-3 1991, MUC-4 1992, MUC-5 1993, MUC-6 1995). Information extraction involves extracting specific types of task-dependent information. For example, given a collection of news reports on terrorist attacks, information extraction involves finding specific kinds of information, such as the name of the terrorist organization, the names of the victims, and the type of incident (e.g., kidnapping, murder, bombing). In contrast, keyphrase extraction is not specific. The goal in keyphrase extraction is to produce topical phrases, for any type of factual, prosaic document. [18]

in [18] automatic keyphrase extraction is approached as a supervised learning task. We treat a document as a set of phrases, which must be classified as either positive or negative examples of keyphrases. This is the classical machine learning problem of learning from examples. In Section 5, we describe how we apply the C4.5 decision tree induction algorithm to this task (Quinlan 1993). There are several unusual aspects to this classification problem. For example, the positive examples constitute only 0.2% of the examples. The experiments in this paper use five collections of documents, with a combined total of 652 documents. The collections are presented in Section 4. In our first set of experiments (Section 6), we evaluate nine different ways to apply C4.5. In preliminary experiments with the training documents, we found that bagging seemed to improve the performance of C4.5 (Breiman 1996a, b, Quinlan 1996). Bagging works by generating many different decision trees and allowing them to vote on the classification of each example. We experimented with different numbers of trees and different techniques for sampling the training data. The experiments support the hypothesis that bagging improves the performance of C4.5 when applied to automatic keyphrase extraction. During our experiments with C4.5, we came to believe that a specialized algorithm, developed specifically for learning to extract keyphrases, might achieve better results than a general-purpose learning algorithm, such as C4.5. Section 7 introduces the GenEx algorithm. GenEx is a hybrid of the Genitor steady-state genetic algorithm (Whitley 1989) and the Extractor parameterized keyphrase extraction algorithm (Turney 1997, 1999).⁴ Extractor works by assigning a numerical score to the phrases in the input document. The final output of Extractor is essentially a list of the highest scoring phrases. The behaviour of the scoring function is determined by a dozen numerical parameters.

Genitor tunes the setting of these parameters, to maximize the performance of Extractor on a given set of training examples. The second set of experiments (Section 8) supports the hypothesis that a specialized algorithm (GenEx) can generate better keyphrases than a general-purpose algorithm (C4.5). Both algorithms incorporate significant amounts of domain knowledge, but we avoided embedding specialized procedural knowledge in our application of C4.5. It appears that some degree of specialized procedural knowledge is necessary for automatic keyphrase extraction. The third experiment (Section 9) looks at subjective human evaluation of the quality of the keyphrases produced by GenEx. On average, about 80

supervised

unsupervised

graph-based algorithms

what works and what fails

Automatic keyphrase extraction

3.3 our approach our algorithm

our algorithm a bit of introduction

data acquisition

data cleaning

the results

3.4 visualization of trends

3.5 conclusion

Bibliography

- [1] Hiyan Alshawi. *Memory and Context for Language Interpretation*. Studies in Natural Language Processing. Cambridge University Press: Cambridge, New York, 1987.
- [2] Nicholas Asher. From discourse macro-structure to micro-structure and back again: Discourse semantics and the focus/background distinction. In Hans Kamp and Barbara H. Partee, editors, *Proceedings of the workshops on Context Dependence in the Analysis of Linguistic Meaning*, volume 1: Papers, pages 21–51. IMS, University of Stuttgart, 1995.
- [3] E. Cambria and B. White. Jumping nlp curves: A review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, 9(2):48–57, May 2014. 8, 9
- [4] Gennaro Chierchia. *Dynamics of Meaning: Anaphora, Presupposition and the Theory of Grammar*. The University of Chicago Press: Chicago, London, 1995.
- [5] Herbert H. Clark and Susan E. Haviland. Comprehension and the given–new contract. In Roy O. Freedle, editor, *Discourse Production and Comprehension*, number 1 in Discourse Processes: Advances in Research and Theory, pages 1–40. Ablex Publishing Corporation: Norwood, NJ, 1977.
- [6] Östen Dahl. Topic-comment structure revisited. In Östen Dahl, editor, *Topic and Comment, Contextual Boundness and Focus*, number 6 in Papers in Textlinguistics, pages 1–24. Helmut Buske Verlag: Hamburg, 1974.
- [7] Paul Dekker and Herman Hendriks. Files in focus. In Elisabet Engdahl, editor, *Integrating Information Structure into Constraint-based and Categorical Approaches*, ESPRIT Basic Research Project 6852, Dynamic Interpretation of Natural Language, DYANA-2 Deliverable R1.3.B, pages 29–37. ILLC, University of Amsterdam, 1994.
- [8] Elisabet Engdahl and Enric Vallduví. Information packaging and grammar architecture: A constraint-based approach. In Elisabet Engdahl, editor, *Integrating Information Structure into Constraint-based and Categorical Approaches*, ESPRIT Basic Research Project 6852, Dynamic Interpretation of Natural Language, DYANA-2 Deliverable R1.3.B, pages 41–79. ILLC, University of Amsterdam, 1994.
- [9] Nomi Erteschik-Shir. *The Dynamics of Focus Structure*. Number 84 in Cambridge Studies in Linguistics. Cambridge University Press: Cambridge, New York, Melbourne, 1997.
- [10] L. T. F. Gamut. *Logic, Language, and Meaning*, volume 1: Introduction to Logic. The University of Chicago Press: Chicago, London, 1991.

- [11] L. T. F. Gamut. *Logic, Language, and Meaning*, volume 2: Intensional Logic and Logical Grammar. The University of Chicago Press: Chicago, London, 1991.
- [12] Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.
- [13] Jeanette K. Gundel. *The Role of Topic and Comment in Linguistic Theory*. PhD thesis, University of Texas, 1974. Published by the Indiana University Linguistics Club, Bloomington, 1977.
- [14] Jeanette K. Gundel. Universals of topic–comment structure. In Michael Hammond, Edith A. Moravcsik, and Jessica R. Wirth, editors, *Studies in Syntactic Typology*, volume 17 of *Typological Studies in Language*, pages 209–239. John Benjamins: Amsterdam, Philadelphia, 1988.
- [15] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art.
- [16] Herman Hendriks and Paul Dekker. Links without location. In Paul Dekker and Martin Stokhof, editors, *Proceedings of the Tenth Amsterdam Colloquium*, pages 339–358. ILLC—Dept. of Philosophy, University of Amsterdam, 1996.
- [17] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. 9
- [18] PeterD. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000. 10
- [19] Kai von Fintel. A minimal theory of adverbial quantification. In Hans Kamp and Barbara H. Partee, editors, *Proceedings of the workshops on Context Dependence in the Analysis of Linguistic Meaning*, volume 1: Papers, pages 153–193. IMS, University of Stuttgart, 1995.

Appendix A

Appendices:Sectioning

Appendices appear in the Contents table on the level of chapters and numbered alphabetically starting from A. You can include normal sectioning commands (notice the vertical spacing between different levels):

A.1 Section

A section is numbered and appears in the Contents.

A.1.1 Subsection

Same holds for a subsection.

Subsubsection

But not for other sectioning commands, unless you adjust the `secnumdepth` and `tocdepth` counters to higher values (by default both are set to 2).

Paragraph A paragraph is inlined.

Subparagraph A subparagraph is inlined and indented.

Appendix B

Illustrating Lists

This appendix is present only in the full version of the thesis. It illustrates in more detail the formatting of the Lists of Tables, Figures, etc. In addition, it also illustrates the use of the “other list” facility of `csthesis.sty`. Let us start with the latter.

B.1 List of Programs

In the preamble of `thes-full.tex`, a new type of a floating environment—`program`—is defined, together with the `\otherlist` command for typesetting the List of Programs. The list is formatted in the same way as the lists of Figures and Tables, and an appropriate entry is added into Contents.

Because programs are defined here as floating environments, they are typeset with the same (tighter) line-spacing as figures and tables:

This shows that single line spacing
is used inside programs.

Program B.1: Example of the new `program` environment

A second example of a program environment.

Program B.2: Second program

B.2 Formatting of Lists

The tables (figures, programs, etc.) are organized and sorted by chapters (appendices), with additional small vertical space separating the chapter blocks. To illustrate this, I include two new tables here (see List of Tables, etc., in the beginning of the thesis for the result):

To be silly or not to be silly
that is the question!

Table B.1: First meaningless table in Appendix B

$$F = nd^2$$

Table B.2: Second meaningless table in Appendix B