



How to Write a Thesis

Harald Zankl
Harald.Zankl@uibk.ac.at

3 December 2010

Abstract

This note gives a short description on how to write a scientific document. It is primarily aimed at computational logic students to ensure a uniform presentation of their work. It also provides some hints on structuring and organizing documents.

Contents

1	Introduction	1
2	Formatting	1
2.1	General Rules	1
2.2	Code Listings	2
3	Graph Drawing	2
4	Referencing Work by Others	2
5	Coming to an End	2
6	Conclusion	3
	Bibliography	3

Throughout the entire document emphasized text is used as explanation.

1 Introduction

Try to be very precise when writing the introduction. The reader should get an idea of what is going on in the course of the document. It is also the right place to motivate why the performed work is interesting, e.g.,

This document gives some hints on how to structure and organize a thesis. It does not contain explicit help on L^AT_EX. For that issue please refer to a short introduction in German [?] or a not so short introduction in English [?]. To ensure a uniform layout this note further fixes some conventions when typesetting in L^AT_EX and lists some useful packages.

At the end of an introduction a rough outline of the document should be presented.

After discussing some basic formatting guidelines in Section 2, Section 3 presents a package for graph drawing before the focus is put on how to reference work by others in Section 4. Section 5 briefly states how the last period of writing the thesis is organized before some concluding remarks are given in Section 6.

2 Formatting

Start a section/chapter with a short overview.

This section is concerned with simple formatting guidelines. After recalling general rules in Section 2.1, hints for listing source code are discussed in Section 2.2. program listings

2.1 General Rules

Using L^AT_EX most of the formatting task is easy or even automatic. Nevertheless there are some rules that have to be adopted.

- Always specify a caption for both figures and tables. Refer to the tables in the text and explain them. If you do not center figures and tables there should be a really good reason for not doing so!
- Headings that consist of more than one word are written capitalized. Consider the heading of Section 4 for example.

Please use the dedicated environments for corollaries, definitions, examples, lemmas, and theorems. A short hint what follows is appropriate.

The next example demonstrates arithmetic over natural numbers.

Example 2.1. We have $3 + 2 = 5$ and $3 \times 2 = 5$.

If you reference figures and tables the first letter is always capitalized. The same holds for sections and chapters.

Figure 1 shows that figures (same as tables) should always be captioned by a full sentence (which is therefore concluded by a full stop).



Figure 1: The short logo of the Computational Logic group.

```
let main () = Format.printf "Hello_World!\n";;  
  
main ();;
```

Listing 1: Hello World program in OCaml.

2.2 Code Listings

If you include (parts of) your source code please do it similar to Listing 1 which shows a hello world program in OCaml. Aligning listings at the top or bottom of a page usually eases reading.

A whole variety of syntax highlighting for various programming languages can be chosen by the `\lstset` command. For further information please consult [?].

3 Graph Drawing

For most graphs the XY-pic package [?] is quite sufficient. More advanced graphics are managed by TiKZ. The official project site of this package is sourceforge.net/projects/pgf.

4 Referencing Work by Others

A thorough scientific work is to a high extent founded on references to others people's work. BibTeX is a tool for such tasks. If you are not familiar with that tool, click www.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html for a short explanation. In BibTeX journal articles like [?] are cited differently from conference papers like [?]. The difference can be seen in the file `biblio.bib`.

5 Coming to an End

Read this section at least twice: Once before you start with typesetting and once after finishing it!

After the pure writing part (the introduction (one to two pages) and the abstract (at most a quarter of a page) should definitely be written last!), make sure to use a spell checker. We mention `ispell`, which will suffice for your project. It is available from <http://www.gnu.org/software/ispell/ispell.html>.¹ Note

¹Always use a spell checker before sending a draft version of the thesis to your supervisor.

that spell checkers neither reveal misuse of homonyms, e.g., to, too, and two nor spot all misprints, i.e., “add” instead of “and”, “is” instead of “it” and so on. Careful proof reading might reduce some of these typos but since one is usually immune against own errors, proof reading by somebody else is strongly advised. Search for singular/plural (especially third person singular) mistakes with attention.

Although L^AT_EX usually does a good job in line and page breaking, make sure that your tables and figures fit nicely in the text. The same holds for program listings, mathematical formulas, etc.

Concerning the references, check that the way you cite them is consistent, i.e., you should not refer to one item by “Proc. of the 7th International conference ...” and to another one by “8th Conference on ...” and a third one by “Proceedings of the sixth ...”.

At the very end please check the date displayed on the front page. The format should be <day> <month> <year>.

6 Conclusion

This part briefly recaptures the problem and stresses your contributions. This is also the right place for mentioning future work or related research.

This note gives a comprehensive guide for computational logic students on how to organize their scientific documents. In order to get started with L^AT_EX some useful packages are mentioned.