FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Name Surname

Thesis title

Name of the department

Supervisor of the bachelor thesis:	Supername Supersurname
Study programme:	study programme
Study branch:	study branch

Prague YEAR

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author’s signature

Dedication. It is nice to say thanks to supervisors, friends, family, book authors and food providers.

Title: Thesis title

Author: Name Surname

Department: Name of the department

Supervisor: Supername Supersurname, department

Abstract: Abstract.

Keywords: key words

Contents

Introduction

Introduction should answer the following questions, ideally in this order:

1. What is the nature of the problem the thesis is addressing?
2. What is the common approach for solving that problem now?
3. How this thesis approaches the problem?
4. What are the results? Did something improve?
5. What can the reader expect in the individual chapters of the thesis?

Expected length of the introduction is between 1–4 pages. Longer introductions may require sub-sectioning with appropriate headings — use `\section*` to avoid numbering (with section names like ‘Motivation’ and ‘Related work’), but try to avoid lengthy discussion of anything specific. Any “real science” (definitions, theorems, methods, data) should go into other chapters.

You may notice that this paragraph briefly shows different “types” of ‘quotes’ in TeX, and the usage difference between a hyphen (-), en-dash (–) and em-dash (—).

It is very advisable to skim through a book about scientific English writing before starting the thesis. I can recommend ‘*Science research writing for non-native speakers of English*’ by Glasman-Deal [1].

Chapter 1

Important first chapter

First chapter usually builds the theoretical background necessary for readers to understand the rest of the thesis. You should summarize and reference a lot of existing literature and research.

You should use the standard *citations.*

Use \emph command like this, to highlight the first occurrence of an important word or term. Reader will notice it, and hopefully remember the importance.

Obtaining bibTeX citation Go to Google Scholar[1], find the relevant literature, click the tiny double-quote button below the link, and copy the bibTeX entry.

This footnote is an acceptable way to 'cite' a webpages or URLs. Document without proper titles, authors and publishers generally do not form citations. Consequently, avoid citations of wikipedia pages.

Saving the citation Insert the bibTeX entry to the file `refs.bib`. On the first line of the entry you should see the short reference name — from Scholar, it usually looks like `author2015title` — you will use that to refer to the citation.

Using the citation Use the `\cite` command to typeset the citation number correctly in the text; a long citation description will be automaticaly added to the bibliography at the end of the thesis. Always use a non-breakable space before the citing parenthesis to avoid unacceptable line breaks:

```
Trees utilize gravity to invade ye
noble sires~\cite{newton1666apple}.
```

Why should I bother with citations at all? For two main reasons:

- You do not have to explain everything in the thesis; instead you send the reader to refer to details in some other literature. Use citations to simplify the detailed explanations.

[1] `https://scholar.google.com`

- If you describe something that already exists without using a citation, the reviewer may think that you *claim* to have invented it. Expectably, he will demand academic correctness, and, from your perspective, being accused of plagiarism is not a good starting point for a successful defense. Use citations to give the credit to people who invented what you build upon.

How many citations should I use? Cite any non-trivial building block or assumption that you use, if it is published in the literature. You do not have to cite trivia, such as the basic definitions taught in the introductory courses.

The rule of thumb is that you should read, understand and briefly review at least around 4 scientific papers. A thesis that contains less than 3 sound citations will spark doubt in reviewers.

There are several main commands for inserting citations, used as follows:

- Knuth [2] described a great system for typesetting theses.
- We are typesetting this thesis with LaTeX, which is based on TeX and Metafont [2].
- The TeX was actually expanded to LaTeX by Lamport [3].
- Revered are the authors of these systems! [2, 3]

Chapter 2

More complicated chapter

After the reader gained sufficient knowledge in chapter 1, you can jump to your own material and conclusions.

You will need definitions (see definition 1 below in section 2.1), theorems (theorem 1), general mathematics, algorithms (algorithm 1), and tables (table 2.1). Figures 2.1 and 3.1 show how to make a nice figure. See figure 2.2 for an example of TikZ-based diagram.

See documentation of package `booktabs` for hints on typesetting tables. As a main rule, *never* draw a vertical line.

2.1 Some mathematics

Definition 1 (Triplet)**.** *Given stuff X, Y and Z, we will write a* triplet *of the stuff as (X, Y, Z).*

Theorem 1 (Car coloring)**.** *All cars have the same color. More specifically, for any set of cars C, we have*

$$(\forall c_1, c_2 \in C)\ \textsc{Colour}(c_1) = \textsc{Colour}(c_2).$$

Proof. Use induction on sets of cars C. The statement holds trivially for $|C| \leq 1$. For larger C, select 2 overlapping subsets of C smaller than $|C|$ (thus same-colored). Overlapping cars need to have the same color as the cars outside the overlap, thus also the whole C is same-colored. □

This is plain wrong though.

Column A	Column 2	Numbers	More
Asd	QWERTY	123123	–
Asd qsd 1sd	BAD	234234234	This line should be helpful.
Asd	INTERESTING	123123123	–
Asd qsd 1sd	PLAIN WEIRD	234234234	–
Asd	QWERTY	123123	–
Asd qsd 1sd	GOOD	234234299	–
Asd	NUMBER	**123123**	–
Asd qsd 1sd	DIFFERENT	234234234	(no data)

Table 2.1 An example table. Table caption should clearly explain how to interpret the data in the table. Use some visual guide, such as boldface or color coding, to highlight the most important results (e.g., comparison winners).

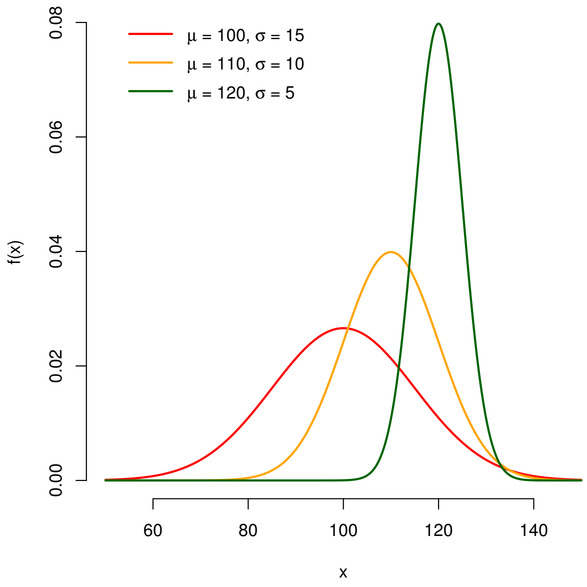

Figure 2.1 A figure with a plot, not entirely related to anything. If you copy the figures from anywhere, always refer to the original author, ideally by citation (if possible). In particular, this picture — and many others, also a lot of surrounding code — was taken from the example bachelor thesis of MFF, originally created by Martin Mareš and others.

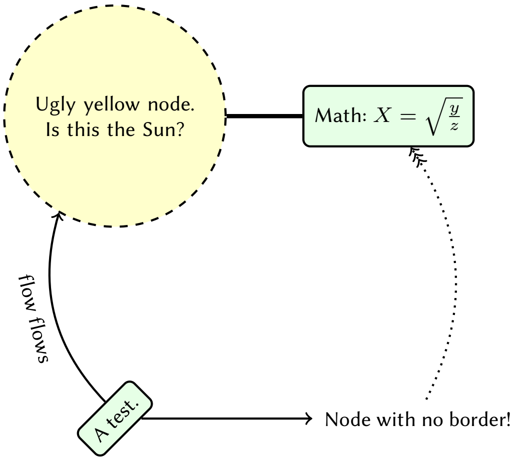

Figure 2.2 An example diagram typeset with TikZ.

Algorithm 1 Algorithm that executes an action with high probability.

```
function ExecuteWithHighProbability($A$)
    $r \leftarrow$ a random number between $0$ and $1$
    $\epsilon \leftarrow 0.00000000001$
    if $r \geq \epsilon$ then
        execute $A$                                ▷ We discard the return value
    else
        print: Not today, sorry.
    end if
end function
```

Chapter 3

Results and discussion

You should have a separate chapter for presenting your results (generated by the stuff described previously, in our case in chapter 2). Remember that your work needs to be validated rigorously, and no one will believe you if you just say that ‘it worked well for you’.

Instead, try some of the following:

- State a hypothesis and prove it statistically
- Show plots with measurements that you did to prove your results (e.g. speedup). Use either `R` and `ggplot`, or Python with `matplotlib` to generate the plots.[1] Save them as PDF to avoid printing pixels (as in figure 3.1).
- Compare with other similar software/theses/authors/results, if possible
- Show example source code (e.g. for demonstrating how easily your results can be used)
- Include a ‘toy problem’ for demonstrating the basic functionality of your approach and detail all important properties and results on that.
- Include clear pictures of ‘inputs’ and ‘outputs’ of all your algorithms, if applicable.

It is sometimes convenient (even recommended by some journals, including Cell) to name the results sub-sections so that they state what exactly has been achieved. Examples follow.

[1] Honestly, the plots from `ggplot` look much better.

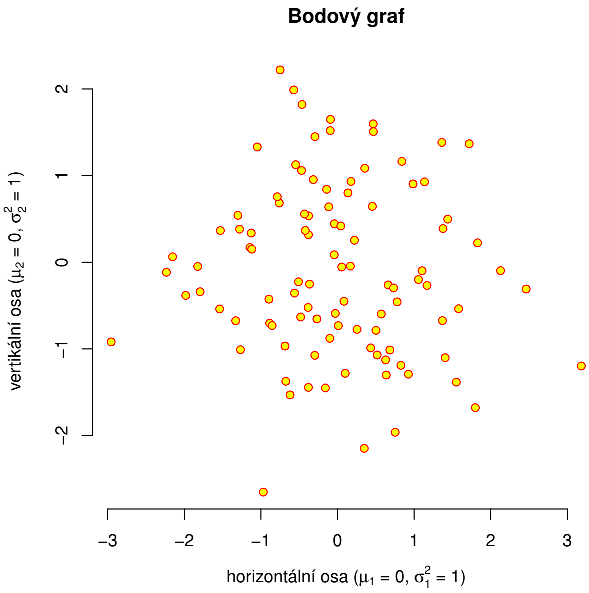

Figure 3.1 This caption is a friendly reminder to never insert figures "in text," without a floating environment, unless explicitly needed for maintaining the text flow (e.g., the figure is small and developing with the text, like some of the centered equations, as in theorem 1). All figures *must* be referenced by number from the text (so that the reader can find them when he reads the text) and properly captioned (so that the reader can interpret the figure even if he looks at it before reading the text — reviewers love to do that).

3.1 SuperProgram is faster than naive OldAlgorithm

3.1.1 Scalability estimation

3.1.2 Precision of the results

3.2 Amount of code reduced by CodeRedTool

3.2.1 Example

3.2.2 Performance on real codebases

3.3 What is a discussion?

After you present the results and showing that your contribution works, it is important to *interpret* them, showing what they mean for the more general public. Separate discussion sections are common in life sciences where ambiguity is common and intuition is sometimes the only thing that the authors have; exact sciences and mathematicians do not use them as often.

Despite of that, it is nice to precisely set your output to the environment, answering:

- What is the potential application of the result?
- Does the result solve a problem that other people encountered?
- Did the results point to any new (surprising) facts?
- Why is the result important for your future work (or work of anyone other)?
- Can the results be used to replace (and improve) anything that is used currently?

If you do not know the answers, you may want to ask the supervisor. Also, do not worry if the discussion section is half-empty or completely pointless; you may remove it completely without much consequence. It is just a bachelor thesis, not a world-saving avenger thesis.

Conclusion

You should summarize what was achieved by the thesis. In a few paragraphs, try to answer the following:

- Was the problem stated in the introduction was solved? (Ideally include a list of successfully achieved goals.)
- What is the quality of the result? Is the problem solved for good and the mankind does not need to ever think about that again, or just partially improved upon? (Is the incompleteness caused by overwhelming problem complexity that would be out of thesis scope, or any theoretical reasons, such as computational hardness?)

  That is quite common.
- Does the result have any practical applications that improve upon something realistic?
- Is there any good future development or research direction that could further improve the results of this thesis? (This is often summarized in a separate section called 'Future work'.)

Bibliography

[1] Hilary Glasman-Deal. *Science research writing for non-native speakers of English.* World Scientific, 2010.

[2] Donald Ervin Knuth. *TEX and METAFONT: New directions in typesetting.* American Mathematical Society, 1979.

[3] Leslie Lamport. *LATEX: a document preparation system: user's guide and reference manual.* Addison-wesley, 1994.

Appendix A

Using CoolThesisSoftware

Use this appendix to tell the readers (specifically the reviewer) how to use your software. A very reduced example follows; expand as necessary. Description of the program usage (e.g., how to process some example data) should be included as well.

To compile and run the software, you need dependencies XXX and YYY and a C compiler. On Debian-based Linux systems (such as Ubuntu), you may install these dependencies with APT:

```
apt-get install \
  libsuperdependency-dev \
  libanotherdependency-dev \
  build-essential
```

To unpack and compile the software, proceed as follows:

```
unzip coolsoft.zip
cd coolsoft
./configure
make
```

The program can be used as a C++ library, the simplest use is demonstrated in listing 1. A demonstration program that processes demonstration data is available in directory `demo/`, you can run the program on a demonstration dataset as follows:

```
cd demo/
./bin/cool_process_data data/demo1
```

After the program starts, control the data avenger with standard `WSAD` controls.

Listing 1 Example program.

```
#include <CoolSoft.h>
#include <iostream>

int main() {
  int i;
  if(i = cool::ProcessAllData());
    std::cout << i << std::endl;
  else
    std::cerr << "error!" << std::endl;
  return 0;
}
```