R
TU Rheinland-Pfälzische
P Technische Universität
Kaiserslautern
Landau

Seminar Template for Computer Science

Your Name

Technische Universität Kaiserslautern, Department of Computer Science

Note: This report is a compilation of publications related to some topic as a result of a student seminar. It does not claim to introduce original work and all sources should be properly cited.

Give an abstract of your paper. You may also give a very short explanation why it could be useful for the reader, i. e. a short motivation.

1 Introduction

The introduction includes the motivation for the presented work. In general, the author will explain the addressed problem and why it is of interest to solve that problem. The most important thing is to explain the papers contribution, e. g. a paper may introduce a new (more efficient) algorithm or/and the proof of an (existing) algorithm or just a method to solve the addressed problem. A general introduction can not be given due to the diversity and variety of problems and their solutions. Finally, the section can be closed by giving an overview of the rest of the paper. E. g. "The rest of the paper is structured as follows: Section 2 gives an overview of related work and existing solutions of the addressed problem. . . ."

2 Related Work

Related work is a very important part of a paper. Basically, the reader has to be convinced that the author of a paper prepared himself carefully. Existing work has to be cited and the author has to explain advantages/disadvantages and limitations of work that is similar to the presented work. In particular, research is the keyword: the author is advised to use libraries or the internet to search for existing solutions of the addressed problem. You are advised to connect sections by short sentences, e. g. "The next section introduces the solution of the addressed problem".

3 The Solution

Whatever the addressed problem is: now the author can start to explain her/his work. After the explanation, experiences, results or just a discussion of the presented work has to be given in Section 4.

4 Conclusions, Results, Discussion

This section will usually be the last section in a paper. Authors may split results and conclusions into two separate sections.

5 How to install LaTeX

Linux Your distribution will most likely offer a Latex package. For Ubuntu you can install the `texlive-full` package via:

```
sudo apt install texlive-full
```

Mac Download and install the MacTeX distribution (`https://www.tug.org/mactex/`).

Windows You can install TeX Live (`https://www.tug.org/texlive/windows.html`) or MiK-TeX (`https://miktex.org/howto/install-miktex`).

Make sure that the `latexmk` tool is also installed. You can find installation instructions for `latexmk` at `http://mg.readthedocs.io/latexmk.html#installation`.

There are also online editors with dedicated Latex support. The SCI offers a free hosting of ShareLatex, the open-source equivalent to the commercial overleaf version. Please check their webpage for more information.

6 How to build this document

In essence, building a LaTeX document is as simple as running

```
latexmk -pdf seminar.tex
```

Here `seminar.tex` is the name of the main `tex`-file.

The `latexmk` tool integrates several tools including `pdflatex` and `biber`. It makes sure that the individual tools are run in the right order and often enough. If you (or your editor) run the individual tools remember that it might be necessary to compile multiple times. For example, without `latexmk` this document requires the following sequence of commands in order to get all citations and references right:

```
pdflatex seminar
bibtex seminar
pdflatex seminar
pdflatex seminar
```

Note that as long as no citations of bibliography data change between builds (and you do not delete any of the generated files) two runs of `pdflatex` are enough to rebuild the document.

Makefike This template also contains a `makefile`, so that this document can be compiled by running `make`. Moreover `make watch` can be used to recompile the document whenever it changes. The Makefile is also configured with Synctex, which enables some PDF-viewers to jump from the document to the correct position in the LaTeX code.

6.1 Editors

There are a number of IDE-like LaTeX editors (none of which fully satisfies the authors). Most are basically a text editor with some gimmicks, but those can already facilitate writing LaTeX documents significantly.

TeXStudio standalone editor with built-in pdf viewer (with source ↔ pdf sync), automatic build, instant preview . . .

Kile KDE-user's default choice

TeXlipse Eclipse plugin for LATEX which is reasonably powerful.

Emacs with AUCTeX If you like Emacs, this is what you want to use.

TeXShop for MAC users

Your favorite text editor probably has LATEX syntax highlighting at least. Look for plugins that add more features.

7 More on LATEX

A whole collection of tutorials and further information is collected by the German TEX user group [5].

We consider the following material especially useful:

- "The Not So Short Introduction to LATEX 2_ε", a basic tutorial for LATEX.

- The LATEX cheat sheet at `http://www.stdout.org/~winston/latex/`.

- Overviews on typesetting mathematics are available on
    - Wikibooks (en.wikibooks.org/wiki/LaTeX/Mathematics),
    - Wikipedia (en.wikipedia.org/wiki/Help:Displaying_a_formula) and
    - Mathematics Stack Exchange (meta.math.stackexchange.com/q/5020).

- Access package documentation by executing `texdoc packagename` on shell (needs TeX Live).

- Find symbols by drawing them with Detexify.

- Command-line program `lacheck` is often better at detecting syntactical errors than the standard `pdflatex` itself; in fact `pdflatex`'s error messages are hardly ever useful.

- A very comprehensive and up-to-date repository of LATEX packages is available on CTAN.

- If you want to use other fonts, visit the LATEX Font Catalogue. However, fonts can be a ficklish issue in LATEX, see also tex.stackexchange.com/q/25249.

- For any LATEX question, visit tex.stackexchange.com.

Note that since TEX and LATEX are decades old and many features have been added by packages over the years, lots of information on the internet are obsolete by now. A famous example are umlauts and other "non-standard" character (äöüßåø) which used to require special commands; nowadays these work out of the box with Unicode source files. See the source of this document to see how it is done.

In case of doubt, we recommend visiting TeX – LaTeX Stack Exchange; not only does the site cover many problems already but the community is very knowledgeable in all things LATEX.

There are also some particular features we consider noteworthy.

```scala
1  def sort[T](list: List[T])(implicit ord: Ordering[T]): List[T] = {
2    list match {
3      case Nil => Nil
4      case x :: xs =>
5        // partition list based on pivot element x
6        val (lo, hi) = xs.partition(ord.lt(_, x))
7        sort(lo) ++ (x :: sort(hi))
8    }
9  }
```

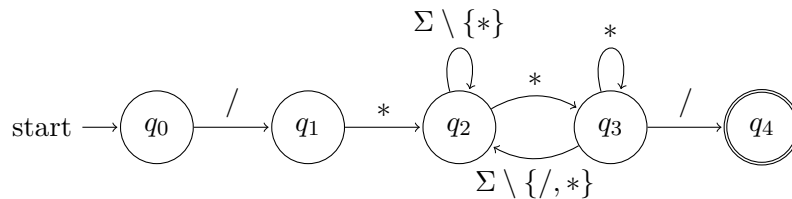Figure 1: An example algorithm. Do you recognize it? Line 6 is crucial!

Figure 2: Example for a TikZ image; here using the `automata` library.

- Referencing (things which have a `label`) and citing[1]. For instance,

  - "`See Section~\ref{sec:editors}`" compiles to "See Section 6.1" and

  - "`a tutorial~\cite{danteintro}`" compiles to "a tutorial [5]".

  There are more commands (and packages) for finer control, such as `\eqref`, `\pageref`, `\nameref` and `\textcite`.

  The most convincing reason to use these is probably that – since links are computed given the current source – numbering, names and links survive rearrangements of the material.

- (Pseudo-)Code can be typeset by a number of packages, e. g. `listings`. See Listing 1 for an example.

- With the `todonotes` package it is easy to leave little reminders in your document.

  Should we explain more?

- TikZ is a great package for creating all sorts of graphics, see e. g. Figure 2. Find an extensive list of examples (with sources) on texample.net/tikz/examples. Your first graphics in TikZ *will* be time-eaters, but with some practice simple sketches are done in minutes. With a little bit of extra tuning, the same sources produce figures of unsurpassed quality.

- You can use document class `beamer` for creating presentation slides. It introduces some special syntax; best check the documentation and/or search the web.

---

[1]Citations are defined in `references.bib` as BibTeX entries. You can find BibTex entries on most web-pages where you can access papers and articles.

Figure 3: Some Google Scholar features we would like to highlight: back-references (find all papers that cite a given article) and BibTeX export (you may need to enable this in Google Scholar settings).

8 General research tools

Google Scholar & backwards search
Google Scholar can be used to find papers and books of all kinds and often has links to PDF versions that are freely available, e. g., on authors' websites. Moreover, Google Scholar can be used for finding other relevant articles in a field by backward-searching for papers that reference a given paper; see screenshot in Figure 3. Google Scholar has some basic support for exporting entries to BibTeX; the results often need some manual polishing, though.

dblp is a searchable index of computer science bibliography with BibTex entries for all articles.

Mendeley is a (free) web service for storing references (including PDFs).
The collected references can be exported to BibTeX.

aspell or some other spell checker.
As a general rule, typos that can be found by running a spell checker are unacceptable.

Git or some other version control system (like Subversion)

Very useful for keeping old versions around and highly recommended. It works best on text files, but can also handle binary files. A comprehensive introduction is available as free ebook [3].

Git repositories can easily be cloned, e. g., onto an external drive for backup. That way, you get incremental backups of your work basically for free.

As Git (and all other versioning systems) typically operate in a *line-based* way, it is very advisable to break your text into short lines (≈ 100 character). For documents that are edited collaboratively, it has been good practice to start a new line for each sentence (or even sub-clause)[2].

---

[2] Recall that a newline is treated by LaTeX like an ordinary space. Only empty lines indicate a new paragraph.

9 Bibliography

The section has been copied from the original template of the EPTCS style ([4]).

We request that you use \bibliographystyle{eptcs} [4]. Compared to the original LaTeX \bibliographystyle{plain}, it ignores the field month, and uses the extra bibtex fields eid, doi, ee and url. The first is for electronic identifiers (typically the number n indicating the nth paper in an issue) of papers in electronic journals that do not use page numbers. The other three are to refer, with life links, to electronic incarnations of the paper.

Almost all publishers use digital object identifiers (DOIs) as a persistent way to locate electronic publications. Prefixing the DOI of any paper with http://dx.doi.org/ yields a URI that resolves to the current location (URL) of the response page[3] of that paper. When the location of the response page changes (for instance through a merge of publishers), the DOI of the paper remains the same and (through an update by the publisher) the corresponding URI will then resolve to the new location. For that reason a reference ought to contain the DOI of a paper, with a life link to corresponding URI, rather than a direct reference or link to the current URL of publisher's response page. This is the role of the bibtex field doi. DOIs of papers can often be found through http://www.crossref.org/guestquery; the second method *Search on article title*, only using the **surname** of the first-listed author, works best.

Often an official publication is only available against payment, but as a courtesy to readers that do not wish to pay, the authors also make the paper available free of charge at a repository such as arXiv.org. In such a case it is recommended to also refer and link to the URL of the response page of the paper in such a repository. This can be done using the bibtex fields ee or url, which are treated as synonyms. These fields should not be used to duplicate information that is already provided through the DOI of the paper. You can find archival-quality URL's for most recently published papers in DBLP — they are in the bibtex-field ee. In fact, it is often useful to check your references against DBLP records anyway, or just find them there in the first place.

---

[3]Nowadays, papers that are published electronically tend to have a *response page* that lists the title, authors and abstract of the paper, and links to the actual manifestations of the paper (e.g. as dvi- or pdf-file). Sometimes publishers charge money to access the paper itself, but the response page is always freely available.

References

[1] Adams et al. (1986): *The Seville Statement on Violence. 6th International Colloquium* on *Brain and Aggression.*

[2] L. Barroca (1995): *A style option for rotated objects in LaTeX/*, 2.10 edition.

[3] Scott Chacon (2009): *Pro Git (Expert's Voice in Software Development).* Apress. Available at `http://www.git-scm.com/book`.

[4] (Retrieved on March 18, 2019): *The EPTCS bibliography style.* Available at `http://biblio.eptcs.org`.

[5] Deutschsprachige Anwendervereinigung TeX e. V. (2014): *Erste Schritte mit TeX, LaTeX und Co.* Available at `http://www.dante.de/tex/TeXAnfaenger.html`.

[6] A. E. Gill (1980): *Some Simple Solutions for Heat-Induced Tropical Circulation. Quart. J. R. Met. Soc.* 106, pp. 447–462.

[7] M. Goossens, F. Mittelbach & A. Samarin (1994): *The LaTeX/ Companion.* Addison-Wesley.

[8] David S Johnson (2001): *A Theoretician's Guide to the Experimental Analysis of Algorithms. American Mathematical Society* 220(5–6), pp. 1–36. Available at `http://www2.research.att.com/~dsj/papers/experguide.pdf`.

[9] D. E. Knuth (1984): *The TeX/book.* Addison-Wesley.

[10] L. Lamport (1985): *LaTeX/ – A Document Preparation System*, first edition. Addison-Wesley.

[11] T. Oetiker (1996): *The Not So Short Introduction to LaTeX/*, 2.2 edition.

[12] Raphael Reitzig (2015): *ltx2any – Yet another LaTeX build wrapper, with one or two nifty features.* Available at `https://github.com/akerbos/ltx2any/`.

[13] W. Shakespeare (1946): *Hamlet.* F. S. Crofts & Co., New York.

[14] The LaTeX/3 Project Team (1995): *LaTeX/ for authors.*