

Research Proposal

Titel der Arbeit

Max Musterle

Date of Submission: TT. Monat 20YY
Advisors: Prof. Dr. Marie Schmidt
Dr. Be TreuerInZwei
Be TreuerInDrei, M. Sc.



Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen und Komplexität

General instructions

In the *Optimization under Resource Constraints* group we normally register a thesis only after a research proposal has been written down/sketched. This proposal serves as a rough plan for the thesis, it is the basis on which the supervisors assess whether the student's plan is feasible to complete within the given time horizon, and it is a written reminder of what has been agreed upon. Deviations from the research proposal are not unusual, but both supervisor and student should agree on them.

This document provides a template for the research proposal. It starts with a number of sections (Section 1-Section 9) that are relevant for most research proposals in our group. It may be that there are individual sections that do not apply or that are much less relevant than others for your thesis. You may change the order of these sections, as appears most appropriate for you.

This template is a modified version of the Chair 1 my-thesis.tex / my-seminar.tex template using a modified version of the chair1-stylesheet. This should allow you to copy the content of your proposal directly to your thesis draft - so that you can avoid rewriting and reformatting paragraphs that you have already included in your proposal.

Section 10-Section 14 specify some environments, rules for citation, and other formalities, many of which may not yet be relevant for the proposal - but some of them may already be useful.

1 Motivation / Context

Give a (brief) motivation for your work. Often (but not always), the motivation relates to some application, in which the results of your research would be relevant.

State the context / bigger picture research question towards whose answer your research will contribute (if any).

2 State of research

Give a brief overview of what is known already in the literature

If relevant for your thesis, also include the 'state of practice' (how is the problem that you plan to solve currently solved in practice), possibly as an individual subsection / question.

Have a look at Section Section 13 to see how to cite properly.

2.1 'Question 1' that you had when looking at the literature

Use short subsections to structure this section to indicate the different search directions or focus points that you want to take when reviewing the existing literature. It may help to pose these as questions. Be open about what you do not know yet: pose the respective 'question' here, and sketch the current state of your reading, detail how you plan to search for the respective materials - or simply state 'todo' as temporary answer.

3 My research question(s)

State 1-3 research questions.

1. Research question 1
2. Research question 2
3. Research question 3

Note: this section is reserved for the research question that you aim to answer in your thesis. If there is a 'bigger picture' question that your research contributes to, please include it in Section 1.

4 My model / optimization problem

Some theses in our group are based on a predefined optimization problem. If so for your thesis, please provide the definition here. (In this case this section would likely be located before the research questions, possibly even before the 'state of research'.)

For some theses, modeling the situation of interest in a mathematical way *is the goal / one of the goals* of the thesis. In that case, this section can be used to collect (preliminary) modeling ideas.

5 Solution approaches

This section refers to algorithmic approaches (including mathematical programming approaches) to be used / developed in the thesis. Many students in our group apply existing algorithms, modify existing algorithms, or develop new algorithmic approaches to solve the respective optimization problems. Please state what you plan in that respect. Include characteristics of the algorithm (e.g., relating to running time and solution quality (exact? heuristic? approximation algorithm?)) where available and appropriate.

If you plan to use an approach that you still need to learn about: do you have any source(lecture/textbook/journal publication) to learn from?

Do you plan to implement the solution approach? Which language do you plan to use, which libraries / routines /etc are already available and can be used?

6 Provable results

Is there anything that you would like to prove (formally/mathematically) in your thesis? Think of properties of the optimization problem that you consider, and of the solution approaches (running time, solution quality) that you aim to develop.

Important remark: not only positive results are relevant, counterexamples (showing that a certain property does *not* apply) can also give very valuable insights.

7 Experimental results

Are there any statements that you would like to verify experimentally in your thesis?

7.1 Analysis/Question/Statement 1

Sketch the experiment(s) that you plan to undertake, relating (as far as applicable) to

- selection of test instances and parameters,
- performance indicators that you want to track
- adequate aggregation and visualization of results

8 Time schedule

Break your thesis down into several steps. Explicitly include the writing of each thesis chapter as individual step. Make a time schedule (list, table, Gantt chart, whatever you prefer, but it should contain explicit dates) starting from the first thesis meeting and ending at the submission date - and indicate where you stand right now. Plan in time for proofreading in the end. Explicitly account for known absences (holidays, exam periods, etc)

9 Risks and pitfalls

Is there anything that endangers the plan, as sketched so far? What is your 'backup plan'?

10 Umgebungen

Wer bei uns eine Arbeit schreibt, kommt um Definitionen, Sätze, Beweise, Abbildungen und Tabellen nicht herum (für das Proposal wird man aber sicherlich noch nicht alles brauchen).

...

Definition 1. *Dies ist eine Definition, welche sich sehr leicht erstellen lässt.*

Eine kurze Überleitung von einer Definition zu einem Satz hilft dem Leser zu verstehen, wohin die Reise gehen soll.

Manchmal braucht man auch noch ein Lemma, damit der Beweis dann leichter von der Hand geht.

Lemma 2 (Andres Lemma). *Ein Lemma vorneweg erleichtert die Beweisführung.*

Proof. Nach dem Distributivgesetz gilt:

$$(a + d) \cdot c = ac + dc$$

□

Theorem 3 (Finkscher Hauptsatz). *Wichtige und grundlegende Sätze lassen sich leicht hervorheben.*

Proof. Der Satz gilt offensichtlich, denn nach Lemma 2 gilt:

$$\sum_{i=1}^n 1 = n$$

Zudem wird der Beweis automatisch mit einem q.e.d.-Symbol¹ beendet.

□

Auch wenn wir Lemma 2 und Theorem 3 einen Namen verpasst haben, macht man das in der Regel aber eher nicht.

Corollary 4. *Aus einem Satz lassen sich oft einfache Resultate ableiten, die dann in einem Korollar festgehalten werden.*

11 Verweise

Auf Sätze, wie z.B. Theorem 3, lässt sich mithilfe des Befehls `\cref{labelname}` verweisen, wenn man in der Satz-Umgebung einen "Label" mit `\label{labelname}` gesetzt hat. Genauso können wir auf den nächsten Abschnitt, also Section 12, verweisen. Ist

¹Nach Möglichkeit sollte vermieden werden, dass der Beweis mit einer Leerzeile abgeschlossen wird, so wie es beim Beweis von Lemma 2 der Fall ist. Das kann zum Beispiel durch Hinzufügen einer Floskel wie "Damit ist die Aussage des Satzes bewiesen." erreicht werden. Übrigens funktionieren Fußnoten genau so. Wichtig ist hier, dass direkt vor dem Kommando kein Leerzeichen oder eine Leerzeile ist.

ein Verweis besonders weit weg, kann man `\cpageref{labelname}` verwenden, um dem Leser zu sagen, auf welche Seite er springen muss – Section 12 ist auf Page 6. Wir benutzen hier das Paket (`cleveref`), welches nicht nur die Nummer sondern auch automatisch das richtige Label hinzufügt. Benötigt man nur die Nummer, so verwendet man `\ref{labelname}`. Üblicherweise beginnt man einen Labelnamen mit dem Typ der Umgebung, auf die man verweist, also z.B. `\label{fig:trapez}` für eine Abbildung (engl. *figure*). Ach ja, zum Hervorheben (engl. *emphasize*) eines *neuen Begriffs* verwendet man den Befehl `\emph{neuer Begriff}`, wenn der neue Begriff zum ersten Mal verwendet wird.

12 Gleitelemente

Gleitelemente (oder auch *floats*) *gleiten* durch das Dokument, und $\text{\LaTeX}$ ermittelt selbstständig eine geeignete Stelle im PDF, um diese anzuzeigen. Die wichtigsten Gleitelemente sind Abbildungen (`figure`), Algorithmen (`algorithm`) und Tabellen (`table`).

In diesem Abschnitt erläutern wir die einzelnen Gleitelemente. Eine Übersicht über die verschiedenen Gleitregeln findet sich am Ende in Table 1.

12.1 Abbildungen

Abbildungen, wie z.B. Figure 1, sind schnell eingefügt. *Wichtig:* Fügen Sie eine Abbildung immer erst nach der ersten Referenz auf die Abbildung ein!

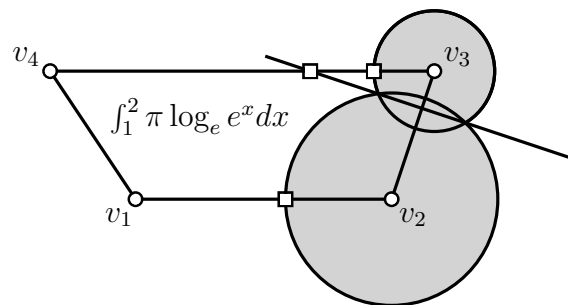


Fig. 1: Das ist eine Abbildung. Sie floatet nach der Regel *here* (h), d.h. $\text{\LaTeX}$ wird versuchen die Abbildung möglichst direkt vor dem untigen Absatz ("Im Allgemeinen...") zu platzieren. Weitere Positionierungsregeln werden in Table 1 erläutert.

Im Allgemeinen braucht man die Endung der Bilddatei beim Einbinden mit dem Paket `\includegraphics` nicht mitanzugeben. Es empfiehlt sich, alle Bilddateien in einen Unterordner (z.B. `figures/`) abzulegen. In mehrteilige Abbildungen kann man mit dem Paket `subcaption` jede mit einer eigenen Bildunterschrift versehen. Man kann sich dann im Text sowohl auf die Teilabbildungen (z.B. Figure 2a) als auch auf die Gesamtabbildung (z.B. Figure 2) beziehen.

Bei Abbildungen ist die Bildunterschrift üblicherweise unter der eigentlichen Grafik. Für alle Gleitelemente gilt, dass das `\label{...}` direkt nach der `\caption{...}` gesetzt

Algorithm 1: BinäreSuche(int[] A , int n , int x)	
Input: sortiertes Feld A , Länge n , gesuchtes Element x	
Output: true genau dann, wenn x in A enthalten ist	
1	$l = 0$
2	$r = n - 1$
3	while $l \leq r$ do
4	$m = \lfloor (l + r)/2 \rfloor$
5	if $A[m] == x$ then
6	return true
7	else if $x < A[m]$ then
8	$r = m - 1$
9	else
10	$l = m + 1$
11	return false

wird, damit bei einem Click auf die Referenz (`\cref{...}`) auch an die richtige Stelle gesprungen wird (und nicht etwa auf den Absatz darüber).

Wichtig: Man sollte sich im Text auf jede Abbildung wenigstens einmal beziehen.

12.2 Algorithmen

Mit der `algorithm`-Umgebung (aus dem Paket `algorithm2e.sty`) ist es nicht schwer, Algorithmen in Pseudocode zu setzen, siehe Algorithm 1.

Das gleiche geht problemlos auch ohne Zeilennummern, dazu benutzt man einfach in der `algorithm`-Umgebung den Befehl `\LinesNotNumbered`.

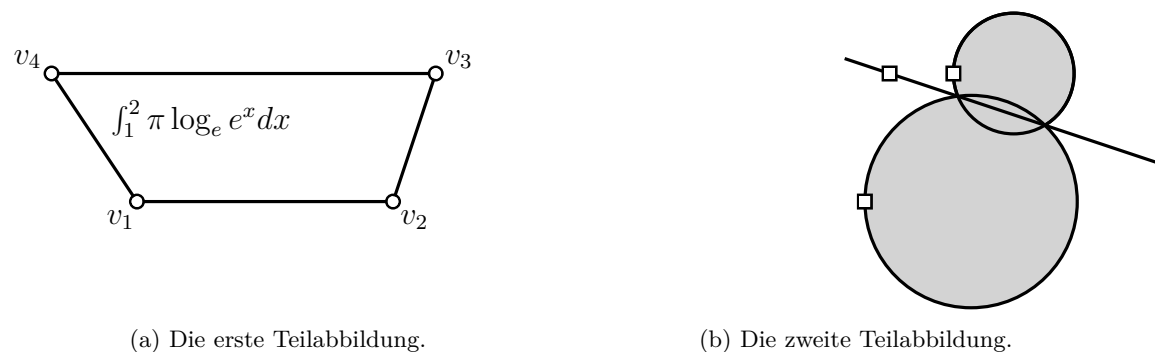


Fig. 2: Das ist eine Abbildung, die aus zwei Teilen besteht. Sie versucht erst *bottom* platziert zu werden, an sonsten erzeugt sie eine *float page*.

Tab. 1: Positionierungsparameter für Gleitelemente. Diese Tabelle selbst hat die Wahl zwischen *top* und *bottom*.

	Wert	Erklärung
[t]	<i>top</i>	Das Element wird oben auf einer Seite angezeigt.
[b]	<i>bottom</i>	Das Element wird unten auf einer Seite angezeigt.
[p]	<i>page</i>	Das Element wird auf einer neuen Seite angezeigt. Diese Seite wird auch <i>float page</i> genannt; sie kann mehrere [p]-Elemente beherbergen.
[h]	<i>here</i>	Das Element wird nah an der Stelle angezeigt, an der es im Code definiert ist. Das kann irgendwo auf der Seite sein.

12.3 Tabellen

Um L^AT_EX etwas in seiner Entscheidung zu beeinflussen, haben wir die Möglichkeit, für jedes Gleitelement einen oder mehrere Positionierungsparameter anzugeben. Eine Übersicht findet sich in Table 1.

Innerhalb der eigentlichen “Tabelle” befindet sich ein `tabularx`-Element, dass den Inhalt darstellt. Elemente dieses Typs haben diverse Vorteile gegenüber normalen `tabular`-Elementen. Zum einen lässt sich die Breite der Tabelle festlegen (hier 90% der Textbreite), zum anderen stehen neben den klassischen zentrierten, links- oder rechtsbündigen (l,r,c) Spalten auch Spalten mit fester Breite zur Verfügung. Diese finden auch in Table 1 Verwendung. Besonders zu beachten ist, dass sich Text in p oder x Spalten wie text in normalen Text-Absätzen verhält, also mit automatischem Zeilenumbruch und Silbentrennung.

13 Literaturverweise

Auch Verweise auf andere Arbeiten, wie die von Mustermann und Musterfrau [MM11], sind ganz einfach. Wenn es von einem Artikel eine Konferenzversion [BDLN02] und eine Zeitschriftenversion [BDLN05] gibt, so sollten Sie stets die Zeitschriftenversion (oder ein Buch [GJ79]) zitieren. Tragen Sie bei einem Konferenzartikel neben Titel, Namen der Autoren und des Tagungsbandes gegebenenfalls auch die Namen der Herausgeber in die bib-Datei ein. ISBN-Nummern sind nicht nötig. Eine große Hilfe beim Identifizieren einer Quelle ist der Digital Object Identifier² – kurz *DOI* genannt. Sofern möglich, sollte stets der DOI einer zitierten Arbeit angegeben werden. Seien Sie vorsichtig mit Umlauten [BKNS10] und Unterstrichen in der bib-Datei.

Wichtig: Zitieren Sie so, dass der Satz, in dem ein Zitat vorkommt, auch ohne das Zitat noch Sinn ergibt; das Zitat soll eine Zusatzinformation sein.

Gut: Binucci et al. [BDLN05] beschäftigen sich mit der Beschriftung von Graphen.

²<https://www.doi.org/>

Schlecht: [BDLN05] beschäftigt sich mit der Beschriftung von Graphen.

Möchte man einen clickbaren Link auf eine Website, wie zum Beispiel OpenStreetMap oder <http://www.graphdrawing.org/> setzen, verwendet man einfach `\href{link}{name}`.

14 Geladene Pakete

Folgende Pakete werden von der Klasse `info1thesis` automatisch mittels `\usepackage` geladen:

- `caption` mit den Optionen `font=small`, `format=hang`, `labelfont=bf`, `figurename=Fig.`, `tablename=Tab`.
- `subcaption` mit `labelfont=normalfont`
- `algorithm2e` mit `linesnumbered`, `algoruled`, `longend`, `vlined`
- `hyperref` mit `bookmarks`, `bookmarksnumbered`, `pdfusetitle`, `pdfencoding=auto`
- `amsmath`, `amsfonts`, `amssymb`, `amsthm`, `graphicx`, `xcolor`, `enumerate`, `booktabs`

Um ein einheitliches Aussehen der Abschlussarbeiten zu gewährleisten, sollte an diesen Parametern nichts geändert werden.

References

- [BDLN02] Carla Binucci, Walter Didimo, Giuseppe Liotta, and Maddalena Nonato: Labeling heuristics for orthogonal drawings. In Petra Mutzel, Michael Jünger, and Sebastian Leipert (editors): *Proc. 9th Int. Symp. Graph Drawing (GD'01)*, volume 2265 of *Lect. Notes Comput. Sci.*, pages 139–153. Springer-Verlag, 2002, 10.1007/3-540-45848-4_12.
- [BDLN05] Carla Binucci, Walter Didimo, Giuseppe Liotta, and Maddalena Nonato: Orthogonal drawings of graphs with vertex and edge labels. *Computational Geometry: Theory & Applications*, 32(2):71–114, 2005, 10.1016/j.comgeo.2005.02.001.
- [BKNS10] Michael A. Bekos, Michael Kaufmann, Martin Nöllenburg, and Antonios Symvonis: Boundary labeling with octilinear leaders. *Algorithmica*, 57(3):436–461, 2010, 10.1007/s00453-009-9283-6.
- [GJ79] Michael R. Garey and David S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
- [MM11] Max Mustermann and Monika Musterfrau: Beispiele in der anwendung. *Beispiele und Muster*, 61(2):306–320, 2011.