Requirements for Scientific Theses in Computer Science

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachgebiet Algorithmik

Preamble

This document was written to help students to get a thorough understanding of what a scientific thesis (in Computer Science) is about. It makes the expectations and grading criteria transparent and can be seen as a guideline. Following this guide closely ensures a very good grade. The goal is a consistent and predictable grading. However, this document cannot be a replacement for regular communcation between student and supervisor.

Disclaimer: This document solely reflects the standards of the Algorithmics group at the Department of Computer Science of TU Darmstadt.

1. Writing a Thesis on Your Own Scientific Work

In scientific research, every statement and conclusion is based on facts or proofs. Every insight presented in a thesis needs to have a clear chain of reasoning.

First of all, you must unambiguously distinguish between insights that were discovered by you in the course of the thesis work and those that originate from other sources (including earlier work of yours). Therefore, it is important to correctly reference other work to make this distinction obvious to the reader.

There is a distinction between primary and secondary sources: a secondary source cites insights published somewhere else (the primary source). Generally, you should work with the primary sources and reference them. Exceptionally, you may decide to follow the presentation in a scientific survey article or textbook. Then you should reference both sources. Any other source such as Wikipedia articles or blog entries should only be used in case there is no scientific source; information from sources of the latter kind are not acceptable as reliable evidence for your conclusions.

By definition, scientific results are always verified beyond doubt, comprehensible for the uninitiated scientist in the field of Computer Science, and reproducible.

2. Research Question

Basically speaking, the purpose of a scientific thesis is to answer one or more research questions in a sound and rigorous way. The answer to a research question must not be obvious or already available in existing literature. A research question should be formulated in a precise and unambiguous way. It is important that there is a definite answer to the research question (which is unknown beforehand).

The answer to the research question needs to be rigorously proven in the thesis. The type of proof depends on the research question: it might be a formal proof (i.e. by mathematical argumentation) or an empirical proof through systematic experiments, in which the key variables for the research question(s) are gathered systematically.

Experiments need to be designed in a way that they are reproducible - otherwise, the answer to a research question is not proven. If not reproducible, the experimental setting was not suitable to answer the research question in the first place.

Next two examples of research questions in Computer Science and how they can be answered through a systematic empirical study:

Thesis title: Detection of Fraudulent PayPal Transactions

Research Questions: How can fraudulent transactions be detected using Support Vector Machines? What are the performance (average computation time, quantiles) and quality characteristics (precision and recall) of the chosen approach?

Methodology: Design a procedure and evaluate it on a well-chosen large dataset of transactions. Measure response times and compare the computed results against ground truth (i.e. a reasonably chosen set of annotated instances). Evaluate the performance and quality characteristics mentioned above.

Thesis title: Teaching Dijkstra's Algorithm Online

Research Questions: What is an intuitive way to teach Dijkstra's algorithm? How can a visualization supplement a textual description? How does the visualization improve the learning process in comparison to a textual description?

Methodology: Conduct a study with N participants split into two groups. The first group learns Dijkstra's algorithm in limited time using the textual description only. The second group learns Dijkstra's algorithm supported by the visualization, within the same time limit. Afterwards, both groups have to take an exam about Dijkstra's algorithm. Does the second group perform significantly better than the first group in this exam?

3. Terminology

It is important to use the established specific terminology of the respective field. To formulate statements in the thesis precisely, it is important to always address an entity by the same term (no synonyms, no periphrases). If no terminology has been established for an entity, the thesis needs to introduce an unambiguous term and use it throughout the whole document.

4. Structure

There is no standardized structure for scientific theses. However, in empirical Computer Science, most published thesis documents are organized in the following way:

- Abstract: summarizes the thesis in a short and precise way.
  - Typically between 100 to 250 words.
  - It motivates the research question:
    Why is it relevant / important?
    In case non-triviality is not obvious: why exactly is it not trivial?
  - It describes the approach to solve the problem / answer the research question.
  - It discusses the results/insights of the thesis and contains a very short conclusion in view of the research question(s).
- Introduction:
  - Introduces and motivates the research question: Why is it relevant / important?
    In case non-triviality is not obvious: why is it not trivial?
    (both more detailed than in the Abstract)
  - It tells the reader what to expect (and possibly what not to expect if this is not obvious) from this thesis, and how the thesis is structured.
- Related Work:
  - Discusses the state of the art. Cite and describe all sources that conducted research on this or very similar topics. The main intention is to make it clear to the reader that your research is novel.
  - Cites all relevant literature:
    * Other approaches to solve the problem at hand
    * Approaches this thesis will be based on

- Contribution[1]: Distinguishes the thesis at hand from those described in related work. How does your work differ from other approaches? Why is it worthwhile?
- Foundations:
  - Every insight that does not originate from this thesis but which is important to understand the approach/insights presented in this thesis.
  - May contain a section about the terminology that is used throughout the thesis.
- Approach (split into different sections): This is the main part of the thesis. It presents the methodological approaches. The exact content depends on the topic of the thesis.
- Evaluation:
  - This is an essential and indispensable part of the thesis.
  - Describes the setup and methodology of the experiment(s).
  - Shows the results of the measurements in tables, statistical plots, and other figures. Describes and discusses the results.
  - Argues rigorously and convincingly why these data answer the research question(s).
- Conclusion and Future Work:
  - Shortly concludes, summarizing the key takeaways
  - Presents future work (e.g. ways to further improve the presented approach). Some rough ideas presented in a concise manner.
    This part is - by definition - off-topic. Thus, it should not be very long.
- Bibliography: Structured list of cited publications
- List of Figures (optional)
- List of Tables (optional)
- Appendix: Optional - can contain, for example:
  - Proofs or other details that were too long and not relevant for the reader's understanding of the main statements made in the thesis
  - Tables from the evaluation that were too long and not relevant for the main conclusions
  - Code fragments that support the reader's understanding
  - Manuals and documentation

[1] can also be part of the introduction

5. Developing an Approach

The following points outline the steps to approach scientific theses in experimental computer science:

- The first step is to research and analyze existing state of the art approaches. A good starting point is the Google literature search engine *scholar.google.com*. Sources you find there and at Universitäts- und Landesbibliothek Darmstadt (ULB) may be only available from the university network. Make use of all its features (e.g. BibTeX export and going through relevant literatur until you are at most recent publications). Organize your literature in a structured manner (e.g. using literature management software tools).
- If the problem at hand can already be solved by existing approaches: identify shortcomings and problems that were not addressed by existing approaches. Otherwise, find problems that are closely related to the problem you are trying to solve. Note that the closest approach might not neccesarily originate from your initial problem domain.
- Develop an approach that either addresses the shortcoming of an existing approach or solves a new/modified problem definition.
- Implement your approach.
- Verify the correctness of your implementation. There is no point in highly optimized code that fails to solve the actual problem.
- Evaluate your approach and compare the results to existing approaches. If your approach has optional optimizations you need to evaluate the base version and the optimized version. Compare the results to show that your modification is an improvement upon the baseline version.

6. Thesis Contents

What is important:

The theoretical approach is the pivotal element of the thesis. Everything that is not related to the pure theory should not be part of sections describing the theoretical approach:

What is not part of the description of the theoretical approach:

- Program code and the used programming language (i.e. the same approach could be implemented in other languages). Exceptions: for example if the research question would be something like “How can the Java garbage collector performance be improved using [...]”.

- Frameworks or libraries used. Here, only the theoretical functioning of the framework/library is important. If the functionality implemented in a library was published in a scientific paper, this should be cited.

Implementation details like the programming language and the hardware the computational study was conducted on should be part of the Evaluation chapter.

7. Colloquium

The colloquium is mandatory as stated in the official requirements. It is basically a short presentation of your findings. Your target audience should be computer scientists with no specific knowledge of your specific thesis topic. So you do not need to explain basic concepts in computer science (e.g. how a hash table works). The structure of your presentation can be derived from the table of contents of your thesis: first, you motivate the problem you want to solve and argue why your contribution is worthwhile. Shortly outline existing approaches/literature and compare them to your activities. Present your approach and the results of your evaluation. Draw a short but meaningful conclusion. Present a short live demonstration if appropriate.

8. Official Minimum Requirements

Source
TU Darmstadt - Computer Science Department
Course Description Handbook

Learning Content Independent study of a scientific research question in Computer Science based on scientific principles in limited time. The research question, approach, as well as the results will be documented in a written thesis and presented in a colloquium.

Outcomes A student should be able:

- to independently conduct scientific research on a complex, state of the art research topic
- to combine and apply the knowledge, methods and expertise acquired during the course of studies
- to research and evaluate the relevant literature
- systematically analyze the topic and create a chain of reasoning
- to weigh the validity of pro- and contra arguments in a comprehensible way

- to evaluate the results and classify them regarding the current research
- to document the results in writing according to scientific principles
- present and defend the results

9. Common Misconceptions

- Communication between student and supervisor is key for a high quality thesis. Bad or no communication is the most common cause for bad grades. This can easily be prevented. It should be in the interest of the student to have regular meetings with the supervisor in order to synchronize the common understanding of the thesis.
- Manage your time. Do not underestimate the time it takes to actually write your thesis document. Do not postpone writing.
- Stay within the scope of your thesis topic. Ideally, your thesis topic is clearly defined. If there is no clear line, you need to discuss this with your supervisor.
- Work independently. Your supervisor will not give you detailed step-by-step instructions on how to debug your programs, use Git, setup your IDE, etc.
- The most important part is your thesis document. Neither your presentation, nor the code you write is as important as your thesis document. Implementing additional features cannot compensate a poorly written thesis. There is no excuse for a missing evaluation or analysis of related work.
- Prepare for meetings with your supervisor (e.g. prepare a demonstration of the current state of your program, prepare questions you want to ask). Only ask for an appointment when there is something to discuss. Take notes and ask questions if you did not understand everything your supervisor explained to you.
- In “bleeding edge” research, you need to be prepared to adjust the focus of your thesis.