Johannes Gutenberg-Universität Mainz
Fachbereich 08 / Institut für Informatik
Studiengang Informatik (B.Sc.)

Titel der Arbeit

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt von
Name (Matrikelnummer)
am Abgabedatum

Betreuer:	Name
Erstgutachter:	Name
Zweitgutachter:	Name

Zusammenfassung

Der Abstract einer Abschlussarbeit sollte eine kurze Zusammenfassung enthalten, damit der Leser nach einigen Sätzen einen Eindruck davon bekommt, welches Thema bearbeitet wurde.
Dieses Dokument dient als Vorlage und gleichzeitig als kleine Anleitung, um eine Abschlussarbeit mit LaTeX zu erstellen. Um das Template für die eigene Abschlussarbeit zu verwenden, kann einfach der vorhandene Text gelöscht und eigener Text hinzugefügt werden. Das Dokument enthält keine ausführliche Erklärung für das Arbeiten mit LaTeX, da es hierzu eine Vielzahl von Tutorials im Internet gibt. Stattdessen enthält es einige Tipps und Richtlinien. Der Quellcode ist ausführlich dokumentiert, damit es einfach ist das Template für die eigene Arbeit anzupassen.

Abstract

English abstract if required, otherwise delete this part.

Inhaltsverzeichnis

1 Einleitung

Die Einleitung erklärt den Kontext der eigenen Arbeit und führt zur Fragestellung hin, die bearbeitet wurde. Es sollte klar werden, in welchem Bereich die Arbeit verfasst wurde und warum sie relevant ist. Im Gegensatz zum Abstract wird die Arbeit hier nicht zusammengefasst. Am Ende der Einleitung kann der Aufbau der restlichen Arbeit erläutert werden.

2 Hauptteil

Der Hauptteil besteht üblicherweise aus mehreren Kapiteln, die verschiedene Aspekte der Arbeit beleuchten. Es ist ratsam für jedes Kapitel ein eigene Tex-Datei anzulegen, damit der Quellcode übersichtlich bleibt. Prinzipiell gibt es für wissenschaftliche Arbeiten bzw. wissenschaftliches Schreiben einen grundlegenden „Bauplan“:

1. Einführung (Introduction) - Welches Themengebiet behandelt die Arbeit und warum? Was sind die Ziele?

2. Stand der Technik (Related Work) - Welche Arbeiten und verwandte Forschungsprojekte gibt es in diesem Themengebiet bereits?

3. Grundlagen (Background) - Welche Grundlagen bilden das Fundament der Arbeit und werden zum Verständnis benötigt?

4. Umsetzung (Design and Implementation) - Dies ist das Hauptkapitel. Das Kapitel wird entsprechend des Arbeitsthemas angepasst, je nach dem ob man im Rahmen der Arbeit beispielsweise etwas selbstständig entwickelt (Software, Hardware, sonstiges) oder eine Literaturanalyse durchführt (usw.). Fragestellung: Was mache ich im Rahmen meiner Arbeit und wie wird dies realisiert?

5. Experimentelle Validierung (Evaluation) - Falls praktische Experimente wie etwa Benchmarks oder Systemtests im Rahmen einer Arbeit Sinn machen: Wie sieht mein Testaufbau aus? Welche Experimente werden durchgeführt? Was sind die Ergebnisse und wie interpretiere ich diese?

6. Schlussbetrachtungen (Conclusion) - Zusammenfassung der Arbeit und Ausblick auf mögliche Folgearbeiten.

Je nach dem um welche Art des wissenschaftlichen Schreibens (z.B. Abschlussarbeit, Seminararbeit, Bericht zu Forschungsprojekt, Fachartikel / Paper, etc.) es sich handelt, kann der „Bauplan“ entsprechend angepasst werden.

2.1 Verwendung dieser Vorlage

Dieses Template ist für die Verwendung mit pdflatex gedacht. Am einfachsten ist es die Vorlage in Overleaf [1] zu öffnen. Overleaf ist eine Onlineanwendung zum Arbeiten mit LaTeX, was den Vorteil hat, dass nichts lokal installiert werden muss und mit jedem Betriebssystem gearbeitet werden kann, das über einen Browser verfügt. Außerdem ist es möglich mit mehreren Personen gemeinsam an einem Projekt zu arbeiten. Die Vorlage kann auch mit einer lokalen Installation verwendet werden. Die Website von Overleaf bietet zudem einige gute Tutorials zum Arbeiten mit LaTeX: `https://www.overleaf.com/learn`.

Fehler und Warnungen, die beim Compilieren erzeugt werden, sollten direkt behoben werden, da es später schwierig sein kann den eigentlichen Auslöser einer Fehlermeldung zu finden. Manchmal sieht das Dokument trotz Fehlermeldung oder Warnung korrekt aus, der Fehler macht sich dann aber später bemerkbar. Die Meldungen sind leider oft nicht sehr aussagekräftig, weshalb es am einfachsten ist direkt nach dem Auftreten eines Fehlers den Teil der Arbeit anzuschauen, der als letztes geändert wurde.

2.2 Unterkapitel

Unterkapitel sollten ein abgeschlossenes Thema behandeln. Einzelne Unterkapitel in einem Kapitel sind zu vermeiden, also z. B. in Kapitel 2 das Unterkapitel 2.1, aber kein weiteres Unterkapitel. In diesem Fall ist es besser entweder den Inhalt von 2.1 direkt in Kapitel 2 zu schreiben, oder falls 2 und 2.1 thematisch zu weit voneinander entfernt sind, aus Unterkapitel 2.1 ein eigenes Kapitel 3 zu machen.

2.3 Grafiken

In der Informatik sind die häufigsten Grafiken entweder Diagramme oder Plots. Beide Arten von Grafiken lassen sich gut als Vektorgrafiken erstellen und einbinden. Der Vorteil von Vektor- gegenüber Pixelgrafiken ist, dass beliebig weit in eine Grafik hereingezoomt werden kann, ohne dass sie unscharf wird. Zudem benötigen Vektorgrafiken meistens weniger Speicherplatz.

Grafiken können direkt in LaTeX mit dem TikZ Paket [2] erstellt werden. Die Verwendung ist etwas gewöhnungsbedürftig, da Grafiken mit Code beschrieben werden, bietet aber viele Freiheiten. Außerdem werden die so erstellten Grafiken direkt in LaTeX gerendert und verwenden die selbe Schriftart wie im Text und eine konsistente Schriftgröße im gesamten Dokument. Ein weiteres beliebtes Programm zum Erstellen von Vektorgrafiken ist Inkscape [3]. Zudem bieten viele Programme die Möglichkeit eine Grafik z. B. als PDF zu exportieren, was in LaTeX als Vektorgrafik eingebunden werden kann.

Pixelgrafiken lassen sich nicht immer vermeiden, z. B. wenn eine Foto in die Arbeit eingebunden werden soll. In diesem Fall sollte darauf geachtet werden, dass die Grafik über eine ausreichende Auflösung verfügt. Eine Auflösung von 300 dpi ist ein guter Richtwert, um beim Drucken ein gutes Ergebnis zu erhalten.

Abbildungen 2.1 und 2.2 zeigen beide den Aufbau des IEEE Floating Point Formats. Abbidlung 2.2 ist eine Pixelgrafik, während Abbildung 2.1 mit TikZ erstellt wurde. Der Unterschied wird beim hereinzoomen deutlich.

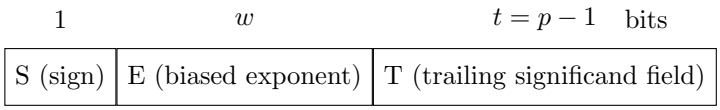

Abbildung 2.1: Aufbau des IEEE Floating Point Formats als Vektorgrafik mit TikZ erzeugt.

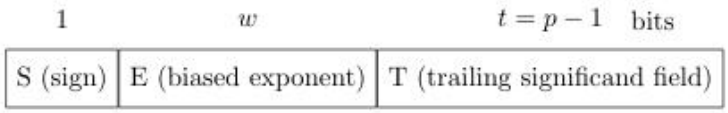

Abbildung 2.2: Aufbau des IEEE Floating Point Formats als Pixelgrafik.

2.4 Tabellen

Tabellen können in LaTeX direkt erstellt werden. Tabelle 2.1 zeigt ein Beispiel dafür. Einfache Tabellen lassen sich schnell erstellen, bei komplizierteren Tabellen ist es manchmal einfacher zusätzliche Pakete zu verwenden. Mit dem Paket multirow [4]

können z. B. einfacher Tabellen erstellt werden, bei denen einzelne Zeilen oder Spalten zusammengefasst sind.

Parameter	binary16	binary32	binary64	binary128
k, storage width in bits	16	32	64	128
w, exponent field width in bits	5	8	11	15
t, significand field width in bits	10	23	52	112
emax, maximum exponent e	15	127	1023	16383
bias, $E - e$	15	127	1023	16383

Tabelle 2.1: IEEE 754-2019 Floating Point Formate als Beispiel für das Einbinden einer Tabelle.

2.5 Quellcode

Um Quellcode in die Arbeit einzubinden, können in LaTeX Listings verwendet werden. Es gibt für populäre Sprachen vorgefertigte Umgebungen, welche die Syntax farblich hervorheben. Quellcode sollte eingebunden werden, wenn eine konkrete Implementierung in einer Sprache erläutert wird. Für die Erklärung eines Algorithmus ist es oft übersichtlicher ein Schaubild oder Pseudocode zu verwenden. Es sollten nur kurze Codeabschnitte eingebunden werden, die für den Leser einfach nachvollziehbar sind und nur den für die Erklärung relevanten Code enthalten. Längere Codeabschnitte können im Anhang stehen. Der komplette Code, der für die Arbeit geschrieben wurde, sollte in einem Repository abgelegt werden.

Listing 2.1 zeigt ein Beispiel für ein Codelisting in der Programmiersprache C. Algorithmus 2.1 zeigt einen Routing Algorithmus als Pseudocode. Der Code wurde mit dem Paket algorithm2e [5] erstellt.

```
1 #include <stdio.h>
2 // comments are highlighted in green
3 void main() {
4     // keywords of the language are highlighted in blue
5     for (int i = 0; i <= 42; ++i) {
6         printf("%d\n", i);
7     }
8     // strings are highlighted in red
9     printf("Hello World!");
```

```
}
```

Listing 2.1: Beispiel für ein Codelisting in der Sprache C.

```
if destination in west direction then
    go West;
else if destination in same column then
    if destination in north direction then
        go North;
    else
        go South;
    end
else if destination in north east direction then
    go North or go East
else if destination in south east direction then
    go South or go East
else if destination in same row and in east direction then
    go East;
else if at destination then
    done;
```

Algorithmus 2.1: West First-Routing Algorithm.

2.6 Literatur

Ein Literaturverzeichnis kann z. B. mit dem Bibtex Paket [6] erstellt werden. Dazu wird für jede Quelle ein Eintrag in der Datei `references.bib` angelegt. An der passenden Stelle im Text können diese Einträge mit dem `\cite{}` Befehl zitiert werden. Für jede Quelle die zitiert wird, legt LaTeX im Literaturverzeichnis einen Eintrag an.

Beschreibungen der Quellen im Bibtex-Format müssen meistens nicht selbst erstellt werden, sondern können direkt bei vielen Verlagen und Bibliotheken direkt generiert werden. Wissenschaftliche Softwareprojekte geben ebenfalls oft auf ihrer Website eine Beschreibung im Bibtex-Format an.

2.7 Abkürzungen

Für jede verwendete Abkürzung kann ein Eintrag in der Datei `acronyms.tex` angelegt werden. Wenn diese Abkürzung im Text zum ersten Mal auftaucht, sollte der Begriff ausgeschrieben werden mit der Abkürzung in Klammern dahinter. Bei weiteren Vorkommen im Text kann dann die eigentliche Abkürzung verwendet werden. In LaTeX gibt es dafür spezielle Befehle. Beispiel für ausgeschriebene Abkürzung (siehe Quelltext des Dokuments für die entsprechenden Befehle): Network Interface Controller (NIC). Beispiel für das Verwenden der Abkürzung: NIC. Die verwendeten Abkürzungen werden automatisch im Abkürzungsverzeichnis aufgelistet.

3 Schlussbetrachtung

Im letzten Kapitel sollte die Arbeit zusammengefasst und ein Fazit gezogen werden. Außerdem sollte beschrieben werden, wie es mit dem Projekt weitergehen kann und welche Punkte vielleicht interessant wären aber im Rahmen der Arbeit nicht bearbeitet werden konnten.

A Anhang

Der Anhang kann Teile der Arbeit enthalten, die im Hauptteil zu weit führen würden, aber trotzdem für manche Leser interessant sein könnten. Das können z. B. die Ergebnisse weiterer Messungen sein, die im Hauptteil nicht betrachtet werden aber trotzdem durchgeführt wurden. Es ist ebenfalls möglich längere Codeabschnitte anzuhängen. Jedoch sollte der Anhang kein Ersatz für ein Repository sein und nicht einfach den gesamten Code enthalten.

B Abbildungsverzeichnis

C Tabellenverzeichnis

D Listings

E Literaturverzeichnis

[1] Overleaf c/o Digital Science, "Overleaf." https://www.overleaf.com/project, accessed on 2022-01-13.

[2] T. Tantau, "The tikz and pgf packages." https://github.com/pgf-tikz/pgf, accessed on 2022-01-13.

[3] Inkscape Project, "Inkscape." https://inkscape.org, accessed on 2022-01-13.

[4] P. van Oostrum, "Multirow package." https://github.com/pietvo/multirow, accessed on 2022-01-13.

[5] C. Fiorio, "algorithm2e package." https://ctan.space-pro.be/tex-archive/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf, accessed on 2022-01-17.

[6] O. Patashnik, "Multirow package." https://tug.org/bibtex/, accessed on 2022-01-13.