

YOUR TITLE

Seminar Softwareverifikation
Firstname Lastname

Zusammenfassung

Dieses Dokument demonstriert die Seminarvorlage für die Ausarbeitung und dient gleichzeitig als minimale Kurzeinführung für \LaTeX .

1 \LaTeX Kurzeinführung

Die \LaTeX Kurzeinführung benutzt wichtige \LaTeX Befehle, die man für die Erstellung einer Seminarausarbeitung benötigt. Sie ist aber bei weitem nicht vollständig. Die Struktur der Einführung selbst demonstriert, wie man die verschiedenen Gliederungsebenen erstellt. Referenzen werden im Zusammenhang mit Abbildungen, Algorithmen und Tabellen gezeigt.

Um die Vorlage selber zu nutzen, löschen Sie einfach den Inhalt der Kurzeinführung. Der Inhalt beginnt mit `\begin{abstract}` und endet vor dem Befehl `\bibliographystyle{plain}`. Zusätzlich müssen Sie Ihren Titel und Namen setzen. Falls Sie in Englisch schreiben, gehen Sie die Kommentare in der Präambel durch und ersetzen sie, wenn angegeben, die Befehle durch die englische Variante.

1.1 Aufzählungen

Es gibt drei grundlegende Aufzählungsarten, deren Darstellung man mit bestimmten Paketen anpassen kann. Hier beschränken wir uns auf die drei Aufzählungsarten.

Die erste Aufzählungsart ist die Nummerierung. Diese wird mit der `enumerate` Umgebung erstellt. Das Beispiel definiert die Studienreihenfolge.

1. Bachelorstudium
2. Masterstudium

Die zweite Art sind Listen mit Aufzählungspunkten. Diese Listen werden mit der `itemize` Umgebung gesetzt. Das Beispiel nennt zwei Aufgaben der Softwareentwicklung.

- Implementierung
- Testen

Zum Schluss stellen wir noch die `description` Umgebung vor.

Syntax Die Syntax einer Programmiersprache definiert, welche Zeichenfolgen Programme der Sprache sind.

Semantik Die Semantik einer Programmiersprache definiert, die Interpretation der gültigen Zeichenfolgen (syntaktisch korrekte Programme).

1.2 Mathematische Darstellung

Dieser Abschnitt gibt einen kurzen Einblick in die mathematische Darstellung und überdeckt nur einen sehr kleinen Bruchteil der Möglichkeiten. Bevor wir mit dem Einblick beginnen, noch ein kurzer Tipp. Generell nutzt man häufig die AMS Pakete für die mathematische Darstellung. Mit den AMS Pakete kann man unter anderem viele mathematische Symbole setzen. Eine gute, kurze Einführung gibt der AMS Short Math Guide¹.

1.2.1 Einfache Formeln

Wir nutzen Zusammenhänge in rechtwinkligen Dreiecken, um die drei wichtigsten Formeltypen vorzustellen. Beginnen wir mit dem Satz des Pythagoras, der häufig durch die Gleichung $a^2 + b^2 = c^2$ in Erinnerung bleibt. Als nächstes folgen zwei Definitionen der Trigonometrie, nämlich Sinus und Kosinus.

$$\begin{aligned}\sin \alpha &= \frac{\text{Gegenkathete}}{\text{Hypotenuse}} \\ \cos \alpha &= \frac{\text{Ankathete}}{\text{Hypotenuse}}\end{aligned}\tag{1}$$

1.2.2 Theoremumgebung

Mit dem `amsthm` Paket kann man verschiedene Theoremumgebungen definieren. Die Vorlage definiert bereits Umgebungen für Theoreme, Lemmas, Propositionen, Korollare und Definitionen. Die Nutzung der Umgebungen funktioniert ähnlich. Im Folgenden werden die Definitionsumgebung und die Lemmaumgebung genutzt, um eine Definition für den größten gemeinsamen Teiler zu geben und ein Lemma anzugeben, das besagt, dass der größte gemeinsame Teiler eindeutig ist. Der Beweis des Lemmas wurde mittels der `proof` Umgebung gesetzt. Diese setzt automatisch das Q.E.D. Kästchen.

Definition 1 (Größter gemeinsamer Teiler). Seien $x, y \in \mathbb{Z}$ zwei ganze Zahlen. Der größte gemeinsame Teiler ist die natürliche Zahl $z \in \mathbb{N}$, für die gilt:

1. $x \bmod z = 0 \wedge y \bmod z = 0$ und
2. $\neg \exists z' \in \mathbb{N} : z' > z \wedge x \bmod z' = 0 \wedge y \bmod z' = 0$.

Lemma 1. *Der größte gemeinsame Teiler zweier Zahlen ist eindeutig.*

Beweis. Seien $x, y \in \mathbb{Z}$ beliebig. Weiterhin seien z, z' größte gemeinsame Teiler von x und y . Gemäß Definition gilt:

- $x \bmod z = 0 \wedge y \bmod z = 0$,
- $x \bmod z' = 0 \wedge y \bmod z' = 0$,

¹[ftp://ftp.ams.org/ams/doc/amsmath/short-math-guide.pdf](http://ftp.ams.org/ams/doc/amsmath/short-math-guide.pdf)

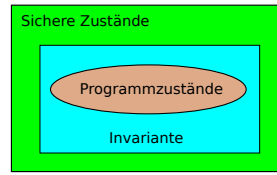


Abbildung 1: Beispielabbildung (Darstellung sichere Invariante)

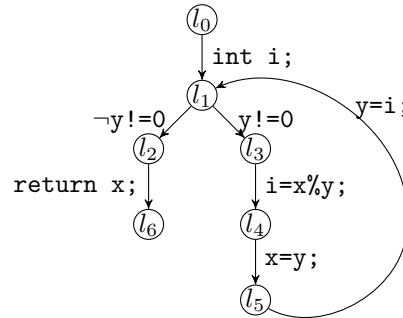


Abbildung 2: Grafik mit TikZ

- $\neg \exists z'' \in \mathbb{N} : z'' > z \wedge x \bmod z'' = 0 \wedge y \bmod z'' = 0$ und
- $\neg \exists z'' \in \mathbb{N} : z'' > z' \wedge x \bmod z'' = 0 \wedge y \bmod z'' = 0$.

Daraus folgt, dass $z \leq z'$ und $z' \leq z$. Somit gilt $z = z'$. □

1.3 Abbildungen

Abbildungen werden innerhalb von Gleitumgebungen gesetzt. Gleitumgebungen erlauben die Beschriftung und verhindern, dass die Darstellung über die Seite umgebrochen werden kann.

Die einfachste Variante eine Abbildung zu erstellen ist eine vorhandene Grafik zu integrieren. Abbildung 1 stellt so eine externe Grafik dar.

Eine weitere, häufig genutzte Variante zur Erstellung von Abbildungen ist TikZ. Hier nutzt man Zeichenbefehle innerhalb einer `tikzpicture` Umgebung. In der Regel braucht man aber etwas Übung bei der Erstellung von TikZ Grafiken. Ein Beispiel für eine mit TikZ erstellte Grafik ist Abb. 2.

1.4 Tabellen

Wie Abbildungen werden Tabellen häufig in Gleitumgebungen gesetzt. Die Gleitumgebung für Tabellen heißt `table`. Es gibt verschiedene Möglichkeiten eine Tabelle zu setzen. Zur Demonstration nutzen wir die einfachste, nämlich die `tabular` Umgebung. Tabelle 1 besteht aus einer Zeile und vier Spalten, die die jeweiligen Ausrichtungen demonstrieren.

Tabelle 1: Beispieltabelle

linksbündig	zentriert	rechtsbündig	Spalte mit fester Größe (er- mög- licht Zeile- num- bruch)

Algorithmus 1 Euklidischer Algorithmus zur Berechnung des größten gemeinsamen Teilers

Eingabe: Natürliche Zahlen $x, y \in \mathbb{N}_0^+$

▷ Eingabe

Ausgabe: $z = ggT(x, y)$

▷ z größter gemeinsamer Teiler

```

1: if  $x = 0$  then
2:    $z := y$ ;
3:   return  $z$ ;
4: while  $y \neq 0$  do
5:   if  $x > y$  then
6:      $x := x - y$ ;
7:   else
8:      $y := y - x$ ;
9:  $z := x$ ;
10: return  $z$ 
```

1.5 Algorithmen

Wie Abbildungen und Tabellen kann man auch Algorithmen in Gleitumgebungen setzen, um ihre Position im Text zu ändern. Für Algorithmen wird die `algorithm` Umgebung genutzt. Ähnlich wie für Tabellen gibt es auch für Algorithmen verschiedene Möglichkeiten. In der Vorlage schlagen wir das Setzen mit der `algorithmic` Umgebung aus dem Paket `algorithmicx` vor. Zur Demonstration setzen wir den euklidischen ggT Algorithmus 1 mit dieser Umgebung.

1.6 Quelltext

Es gibt verschiedene Varianten um Quelltext zu setzen. Diese Vorlage demonstriert zwei Möglichkeiten: die Verbatim Umgebung und das Listings-Paket.

Verbatim Umgebung Die Verbatim Umgebung ist die einfachste Art zur Darstellung von Quelltext. Innerhalb der Umgebung werden alle \LaTeX Befehle ignoriert. Außerdem übernimmt der Autor des \LaTeX Quelltexts die Verantwortung für die Formatierung, also zum Beispiel die Einrückungen. Folgender Codeausschnitt zur Berechnung des ggTs wird mittels der Verbatim Umgebung gesetzt.

```
int ggT(int x, int y)
{
    if(y==0)
        return x;
    return ggT(y, x%y);
}
```

Das Listings-Paket Im Gegensatz zur Verbatim Umgebung unterstützt das Listings-Paket Syntaxhighlighting. Syntaxhighlighting ist für eine Vielzahl von Programmiersprachen verfügbar. Außerdem bietet das Listings-Paket eine Reihe an Formatierungsoptionen. Nachfolgender Java Code ist mit dem Listings-Paket gesetzt.

```
int ggT(int x, int y)
{
    int i;
    while(y!=0)
    {
        i= x%y;
        x=y;
        y=i;
    }
    return x;
}
```

Besonders praktisch ist, dass der Programmcode nicht im \LaTeX Quellcode enthalten sein muss. Mit dem folgenden Befehl kann der Programmcode aus einer Datei, z.B. einer Java-Datei, gelesen und verwendet werden.

```
\lstinputlisting[language=Python]{sourcefile}
```

1.7 Zitieren

Manuell kann man das Literaturverzeichnis mit der **thebibliography** Umgebung erstellen. Die Vorlage nutzt standardmäßig BibTeX zur Erstellung des Literaturverzeichnisses. Die verwendete BibTeX Datei **literature.bib** zeigt zwei mögliche Einträge. Diese Einträge stammen aus Literaturdatenbanken wie DBLP. Eine Übernahme von BibTeX Einträgen aus Literaturdatenbanken ist häufig einfacher. Kommen wir nun zu den Beispielen für Zitierungen.

Das Lehrbuch Model Checking [1] gibt einen theoretischen und praktischen Einblick.

In der Informatik sind direkte Zitate eher selten. Falls man dennoch einmal ein direktes, mehrzeilige Zitate angeben möchte, gibt es die **quote** Umgebung.

As far as the fundamental science is concerned, we still certainly do not know how to prove programs correct. We need a lot of steady progress in this area, which one can foresee, and a lot of breakthroughs where people suddenly find there's a simple way to do something that everybody hitherto has thought to be far too difficult. [2]

Zum Abschluss noch ein Hinweis. Nach Änderungen an Zitaten wie auch Referenzen und Namen von Gliederungselementen (sections, etc.) muss das L^AT_EXDokument mehrfach kompiliert werden.

Literatur

- [1] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, Cambridge, MA, USA, 1999.
- [2] Len Shustek. Interview: An interview with C.A.R. hoare. *Commun. ACM*, 52(3):38–41, 2009.