

Karl-Koch-Universität Astadt



TECHNISCHE UNIVERSITÄT CHEMNITZ

Fakultät für Bla(h)fasel

Institut für schriftliches Gelaber

Diplomarbeit

Diplomarbeitsvorlage mit \LaTeX
Basisversion, V1.20

Uwe Domaratius

Astadt, den 29. Januar 2010

Prüfer: Prof. Dr. Karl Ranseier

Betreuer: Dipl.-Inf. Max Murks

Domaratus, Uwe

uwe.domaratus@informatik.tu-chemnitz.de

Diplomarbeitvorlage mit L^AT_EX

Diplomarbeit, Fakultät für Bla(h)fasel

Karl-Koch-Universität Astadt, Januar 2010

Danksagung

Dank gebührt den Mitgliedern der Arbeitsgruppe des TeX-Stammtisches der TU Chemnitz für die ursprüngliche Vorlage, die ich als Grundlage für dieses Projekt nutzte. Weiterhin muß ich mich bei Martin Fiedler (martin.fiedler@gmx.net) bedanken, der mich beim Erstellen des Makefiles unterstützte. Schlußendlich geht ein Dankeschön an alle, die diese Vorlage benutzen und mir hoffentlich Verbesserungsvorschläge zukommen lassen.

Zusammenfassung

Das ist ein Abstract der Arbeit, welches auch ein wenig länger sein könnte. Der Inhalt des Abstracts wird in der Datei `data.tex` festgelegt.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Kurzanleitung	1
1.2	Voraussetzungen	2
1.2.1	Windows	2
1.2.2	Linux	2
2	Aufbau, Gliederung, Layout	3
2.1	Allgemeine Gliederung dieser Vorlage	3
2.2	Überschriften	3
2.2.1	Kurzbeispiel	4
2.2.1.1	Beispielsubsubsection	4
2.3	Seitenstil	4
2.3.1	Ein- und zweiseitiges Layout	4
2.3.2	Kopf- und Fußzeilen	5
3	Verweise	6
3.1	Querverweise innerhalb des Dokuments	6
3.2	Links zu externen Quellen	6
3.3	Literaturverweise und Quellenangaben	6
3.4	Fußnoten	7
4	Tabellen	8
4.1	Die tabular-Umgebung	8
4.2	Das tabularx-Package	9
5	Abbildungen	10
5.1	Grundlagen	10
5.1.1	Einbinden von Grafiken	10
5.1.2	Zwei Bilder parallel	11
5.2	Weitere Grafikpakete	12
5.2.1	Subfigure	12
5.2.2	Das pgf-Package	12
5.2.3	Gnuplot	13

6	Algorithmen	14
6.1	Die Verbatim-Umgebung	14
6.2	Listings und andere Pakete	14
7	Sonstiges	16
7.1	Listen und Aufzählungen	16
7.1.1	Bullet Lists	16
7.1.2	Aufzählungen	16
7.1.3	sonstige Listen	17
7.2	Wörtliches Zitieren	17
7.2.1	Quote	17
7.2.2	Quotation	18
7.2.3	Verse	18
7.3	Der Mathemodus	18
7.4	Theoreme und Definitionen	19
7.5	Hilfsmittel für den Autor	20
8	Glossar, Abkürzungsverzeichnis und Index	21
8.1	Index	21
8.2	Glossar	21
8.3	Abkürzungsverzeichnis	22
A	Setzen des Dokuments	23
B	Verwendete L^AT_EX-Packages	24
C	Das Makefile	27
D	Dateien der Vorlage	28
E	Hilfsdateien	29
F	Version History	30
G	Known Bugs	31

Abbildungsverzeichnis

5.1	Bild	10
5.2	gerahmtes Bild	11
5.3	zwei Bilder parallel	11
5.4	zwei Bilder parallel mit subfig	12
5.5	Kaffeetasse mit pgf	13
5.6	Beispiele für Gnuplot	13

Tabellenverzeichnis

4.1 Tabellenbeispiel mit tabular	9
4.2 Tabellenbeispiel mit tabularx	9

Algorithmenverzeichnis

6.1 Beispiel für algorithm-Umgebung	14
---	----

1 Einleitung

Diese Datei enthält die Anleitung zur Nutzung der Vorlage für verschiedene Typen von Arbeiten. Sie basiert auf einer entsprechenden Vorlage des T_EX-Stammtisches der TU Chemnitz¹ und wurde erheblich modifiziert.

In den folgenden Kapiteln dieser Anleitung wird ein Überblick über die Verwendung der Vorlage, Zweck der Dateien und typische Anwendungsfälle gegeben.

Hinweis: Diese Anleitung ist **keine** Grundlageneinführung in L^AT_EX. Wer eine solche sucht, sollte sich die Links im Wikipedia-Artikel² ansehen. Zusätzlich empfiehlt sich das Studium der Seite zur „Diplomarbeit mit L^AT_EX“³. Es ist auch nicht gesagt, daß eine mit dieser Vorlage erstellte Diplomarbeit von den Prüfern angenommen wird, die bestehen möglicherweise auf einem eigenen Layout, es sollte also vorher abgeklärt werden, ob diese Vorlage akzeptiert wird.

Im folgenden werden die gängigen Formatierungsmöglichkeiten vorgestellt. Dazu gehören Überschriften, Fußnoten, Tabellen, Abbildungen, Literaturverweise, Aufzählungen usw. usf. Zu den einzelnen Möglichkeiten gibts jeweils verschiedene Beispiele. Die Anhänge geben einen Überblick über die verwendeten L^AT_EX-Packages und weitere technische Details.

1.1 Kurzanleitung

Wer sofort loslegen möchte, muß nur die folgenden Schritte durchführen. Vorausgesetzt wird natürlich, daß eine T_EX-Installation mit allen verwendeten Packages vorhanden ist (siehe Anhang **B**):

1. Metadaten anpassen

Sämtliche Metadaten (Autor, Titel der Arbeit usw.) sind in der Datei `data.tex` hinterlegt und können direkt angepaßt werden.

2. Literaturdatenbank anpassen

Die bestehende Datei `literatur.bib` um die für die eigene Arbeit notwendigen Einträge ergänzen ist ebenso möglich wie das Austauschen der Datei gegen eine bereits bestehende BibT_EX-Datenbankdatei.

¹<http://www.tu-chemnitz.de/urz/anwendungen/tex/>

²<http://de.wikipedia.org/wiki/LaTeX>

³<http://www.dml.drzoom.ch/>

3. Includes anpassen

In der Datei `vorlage.tex` werden die einzelnen Kapitel eingebunden. Die Inhalte der Kapiteldateien können einfach gegen die eigenen Inhalte ausgetauscht werden, im schlimmsten Fall ist das Einbinden zusätzlicher Packages notwendig.

1.2 Voraussetzungen

1.2.1 Windows

Unter Windows hat sich die MikTeX-Distribution bewährt. Diese liegt gegenwärtig in Version 2.5 vor. Ein Installer kann von der Homepage⁴ heruntergeladen werden, fehlende L^AT_EX-Packages können bequem per Updateprogramm nachinstalliert werden.

Als Editor empfiehlt sich z.B. das TeXnicCenter⁵, welches umfangreiche Bearbeitungsmöglichkeiten, Syntaxhighlighting und einiges mehr unterstützt. Zudem erlaubt es das einfache Nachinstallieren von L^AT_EX-Paketen, sobald diese erstmals benötigt werden.

1.2.2 Linux

Die gängigen Distributionen verfügen über vorgefertigte Pakete zur Installation aller notwendigen Programme für L^AT_EX. Ich habe das ganze unter Ubuntu mit TeXLive getestet, für andere Distributionen sollte es ähnlich funktionieren. Das gravierendste Problem ist, daß die L^AT_EX-Pakete teilweise nicht auf dem allerneuesten Stand sind, es ist daher vermutlich notwendig, einige L^AT_EX-Pakete per Hand zu aktualisieren. Beim Testen betraf dies die Packages `pgf` und `KOMA-Script`.

Als Editor kann natürlich jeder Texteditor benutzt werden, speziell auf L^AT_EX zugeschnitten sind, zum Beispiel `kile` und `texmaker`.

⁴<http://www.miktex.org/>

⁵<http://www.toolscenter.org/>

2 Aufbau, Gliederung, Layout

Die Vorlage umfaßt mehrere *.tex-Dateien. Die Hauptdatei ist `vorlage.tex`, von dieser Datei werden alle weiteren Teile eingebunden, angefangen bei der Titelseite `titel.tex` über die einzelnen Kapitel mit dem eigentlichen Inhalt der Arbeit. Grundlegende Daten zur Arbeit (Autor, Titel, Name der Hochschule usw.) sind in der Datei `data.tex` abgelegt. Die ebenfalls eingebundene Datei `user.sty` definiert hilfreiche Befehle, die in späteren Kapiteln erklärt werden.

2.1 Allgemeine Gliederung dieser Vorlage

Eine wissenschaftliche Arbeit gliedert sich grundsätzlich in drei Teile, Vorspann, Hauptteil und Anhänge. Dies ist auch im Manual zum KOMA-Script¹ prima erklärt.

Achtung: Beim Testen unter Linux mußte das KOMA-Script manuell aktualisiert werden, da die TexLive-Installation eine zu alte Version beinhaltete.

Der Vorspann umfaßt die Titelseite, gefolgt von Abstract, Danksagung und den automatisch generierten Auflistungen von Inhaltsverzeichnis, Liste der Abbildungen, Tabellen und Algorithmen. Sollte eine dieser Auflistungen, z.B. die Liste der Algorithmen, nicht notwendig sein, so kann diese in der Datei `vorlage.tex` auskommentiert werden. Ebenso wird der eigentliche Inhalt der Arbeit über diese Datei eingebunden.

Die Anhänge umfassen das Literaturverzeichnis, Glossar, Index und ähnliches. Sollten diese Teile nicht benötigt werden, ist die Datei `vorlage.tex` entsprechend abzuändern. Bei Verwendung des `makefiles` bzw. des Übersetzungsprofils für TeXnicCenter werden diese Teile automatisch aus dem Text des Hauptteils generiert, ansonsten sind manuelle Aufrufe von BibTeX und Makeindex notwendig (siehe Anhang A).

2.2 Überschriften

Die Gliederung in Kapitel, Abschnitte usw. erfolgt mit den Standard-L^AT_EX-Befehlen. Falls eine Überschrift sehr lang ist, hat man die Möglichkeit, mittels `\section[kurz]{lang}` eine Kurzbeschreibung zu generieren, die dann für das Inhaltsverzeichnis genutzt wird.

¹ [ftp://ftp.ctan.org/tex-archive/macros/latex/contrib/koma-script/scrguide.pdf](http://ftp.ctan.org/tex-archive/macros/latex/contrib/koma-script/scrguide.pdf)

2.2.1 Beispiel für eine subsection mit Kurztext

Im Inhaltsverzeichnis wird diese subsection als „Kurzbeispiel“ aufgeführt.

2.2.1.1 Beispielsubsubsection

Die Formatierung der Überschriften wird in der Datei `vorlage.tex` festgelegt. Die Anweisung `\secnumdepth{\subsectionlevel}` sagt aus, daß alle Überschriften bis einschließlich subsection eine Nummer kriegen. Bei Bedarf läßt sich das natürlich einfach ändern.

Ähnlich verhält es sich mit der Anweisung `\tocdepth{\subsectionlevel}`, die steuert, bis zu welcher Gliederungsebene Einträge ins Inhaltsverzeichnis geschrieben werden.

Paragraphbeispiel Ich bin ein Paragraph.

Subparagraphbeispiel Ich bin ein Subparagraph.

Minisec

Zusätzlich bietet die Vorlage noch den Befehl `\minisec{}` an. Damit kann man z.B. Paragraphen mit einer abgesetzten Überschrift versehen, so wie diesen hier.

2.3 Seitenstil

Die Standardeinstellungen für Seitengröße und Ränder der Vorlage entsprechen den Standards im KOMA-Script und sollten typographischen Grundsätzen genügen. Man kann diese Einstellungen natürlich ändern, sollte sich aber im Klaren sein, daß sich die Entwickler beim Festlegen der Standardwerte durchaus etwas gedacht haben. Eine vollgequetschte Seite mit minimalen Rändern mag ein paar Seiten einsparen, sieht aber vergleichsweise bescheiden aus und liest sich dank überlanger Zeilen erheblich schwieriger. Gerade letzteres kann auch unbewußt in die Bewertung einer Arbeit einfließen! Man sollte sich also im Zweifel auf das Anpassen von Kopf- und Fußzeilen beschränken.

Wichtig ist ansonsten noch die eventuell notwendige Anpassung für die Bindekorrektur. Falls das fertige Dokument später professionell gebunden werden kann es sein, daß dafür ein größerer Rand notwendig ist. Diese Korrektur ist als Option der Dokumentenklasse anzugeben, und zwar als `BCOR=10mm` (oder mit entsprechend anderen Zahlenwerten).

2.3.1 Ein- und zweiseitiges Layout

Standardmäßig wird die Vorlage in doppelseitigem Layout gesetzt. Sollte dies nicht erwünscht werden, so muß in der Datei `vorlage.tex` in den Optionen der Dokumentenklasse die option `twoside` gegen `oneside` getauscht werden.

Abgesehen von den Randeinstellungen und der Formatierung der Kopf- und Fußzeilen gibt es keine Unterschiede zwischen ein- und doppelseitigem Layout.

2.3.2 Kopf- und Fußzeilen

Beim doppelseitigen Layout entsprechen die Kopf- und Fußzeilen dem KOMA-Script Standard, d.h. Kopfzeile mit einer Linie vom Text getrennt, die Fußzeile ohne Trenner. In der Kopfzeile befinden sich jeweils am äußeren Rand die Bezeichnung der aktuellen `chapter` (auf geraden Seiten) bzw. `section` (ungerade Seiten). Die Seitenzahl wird in der Fußzeile außen gedruckt. Auf Seiten, auf denen ein neues Kapitel begonnen wird, bleibt die Kopfzeile leer.

Im einseitigen Layout wurden die Standardeinstellungen verändert. In der Kopfzeile befinden sich links die aktuelle `chapter` und rechts die `section`. Die Seitenzahl ist in der Fußzeile zentriert. Auf Seiten, auf denen ein neues Kapitel begonnen wird, bleibt auch hier die Kopfzeile leer.

In den Anhängen ist die Kopfzeile einfacher definiert, dort ist grundsätzlich nur die Bezeichnung des aktuellen Anhangs linksbündig dargestellt. Dies gilt sowohl für ein- als auch doppelseitiges Layout.

Es ist unter Umständen notwendig, die Einstellungen für die Kopf- und Fußzeilen anzupassen. So ist es beim Schreiben von Artikeln für die Veröffentlichung in Sammelbänden sinnvoll, auf jeder Seite den Autor und den Titel des Artikels zu vermerken. Die Anpassung des Layouts geschieht in der Datei `user.sty`. Die zur Verfügung stehenden Befehle und Möglichkeiten kann man der Anleitung zum KOMA-Script entnehmen, dort ist das sehr ausführlich erklärt.

3 Verweise

3.1 Querverweise innerhalb des Dokuments

Man kann in \LaTeX sehr einfach Querverweise erzeugen. Dazu wird an die Stelle, auf die verwiesen werden soll ein `\label{11}` gesetzt, auf welches dann mit `\ref{11}` und `\pageref{11}` verwiesen werden kann. Die Vorlage verwendet das Package `hyperref`. Somit werden beim Erstellen eines PDF mittels `pdflatex` alle Querverweise im Dokument zu klickbaren Links. Als Beispiel gibt es einen Verweis auf das Kapitel 5.1 auf Seite 10. Das Aussehen der Verweise (Farbe, Schriftart, ...) läßt sich in den Optionen des Packages `hyperref` einstellen.

Sofern das Dokument gedruckt werden soll, machen sich farbige Textabschnitte ziemlich schlecht. In diesem Fall ist zum Package `hyperref` die Option `draft=true` hinzuzufügen. Damit wird die gesamte Hyperlinkfunktionalität ausgeschaltet und der Text kann problemlos auch s/w gedruckt werden.

3.2 Links zu externen Quellen

Über die eingebundenen Packages lassen sich auch Links zu externen Quellen wie z.B. Internetseiten einbinden. Dafür gibt es den Befehl `\url{}`. Ein Link zu <http://www.google.de> sei als Beispiel angegeben, der zumindest korrekt formatiert und im PDF auch anklickbar ist.

3.3 Literaturverweise und Quellenangaben

Die Vorlage geht davon aus, daß BibTeX für das Literaturverzeichnis genutzt wird. Der verwendete BibTeX-Stil beinhaltet ebenfalls ein Feld für Hyperlinks zum Paper bzw. zur Homepage des Autors. Diese Formatdatei heißt `alphurl.bst`¹, man kann auch andere Stile verwenden, je nach Vorlieben. Für das Erstellen eigener Formate sei auf das Package `custom-bib` verwiesen.

Die Formatdatei beschreibt dabei sowohl die Formatierung des Literaturverzeichnisses, als auch der Verweise im Dokument. In diesem Fall werden Verweise auf Quellen mit einem Kürzel aus Autor und Jahreszahl dargestellt.

¹Mittels `urlbst` aus dem Standardstil `alpha` erzeugt, siehe <http://www.tug.org/tex-archive/biblio/bibtex/contrib/urlbst/urlbst.html>

Alle zitierten Literaturwerke stehen in der Datei `literatur.bib`. Die Einträge haben alle ein bestimmtes Format, welches in zahlreichen BibTeX-Seiten im Netz erklärt wird (als Basis möge der Wikipedia-Artikel² dienen). Wichtig ist hierbei die Beachtung der Regeln für Umlaute.

Um nun einen Verweis auf eine dort verzeichnete Quelle gibt es den Befehl `\cite{label}`. Das Label steht im BibTeX-Eintrag der Datenbank an erster Stelle. Der Verweis zur Literaturangabe sieht dann beispielsweise so aus: „Mittelbach et al [GMS00] haben ein empfehlenswertes Buch über \LaTeX geschrieben“. Diese Vorlage [Dom06] steht als Beispiel auch schon drin.

Alle im Text irgendwo zitierten Quellen werden automatisch ins Literaturverzeichnis eingebunden und alphabetisch sortiert. BibTeX bietet auch noch massenhaft andere Möglichkeiten, aber dafür muß man die entsprechenden Bücher lesen.

3.4 Fußnoten

Fußnoten³ lassen sich sehr leicht definieren. Man setzt an die Stelle, an der die Fußnote erscheinen soll ein `\footnote{text}`. Man kann auch sehr lange Fußnoten erzeugen⁴, ebenso besteht die Möglichkeit, URLs als Fußnote anzugeben⁵.

²<http://de.wikipedia.org/wiki/BibTeX>

³Ich bin eine Fußnote

⁴Ich bin eine sehr lange Fußnote. Ich bin eine sehr lange Fußnote.

⁵<http://www.google.de>

4 Tabellen

Tabellen sind in \LaTeX zugegebenermaßen ziemlich ekelhaft. Die Vorlage benutzt das `tabularx`-Package, damit kann man zumindest einigermaßen vernünftig formatierte Tabellen erstellen. Ein Blick in die Anleitung oder ein gutes Buch zum Thema kann aber nicht schaden.

Tabellen werden automatisch ins Tabellenverzeichnis aufgenommen, solange sie im Rahmen von `\begin{table} \end{table}` stehen und eine `caption`, also eine Beschriftung, aufweisen.

4.1 Die tabular-Umgebung

Die wichtigsten Dinge finden sich in der Beispieltabelle. Der Quelltext zeigt den prinzipiellen Aufbau. Wichtig ist die Zeile `\begin{tabular}{|c|lr}`, die setzt drei Spalten, die erste centered, die nächsten links- bzw. rechtsbündig. Die Pipes erzeugen senkrechte Striche zwischen den Spalten. Eine horizontale Linie in einer Tabelle erzeugt man mittels `\hline`, es gibt auch Befehle für Linien nur an bestimmten Spalten und Möglichkeiten, Zellen zu verknüpfen, aber das geht ohne Anleitung quasi immer schief.

Die Zeile `\begin{table}[htbp]` positioniert die Tabelle auf der Seite, und zwar zuerst **here**, danach in der Reihenfolge **top** (ganz oben auf der Seite), **bottom** (ganz unten auf der Seite) und **page** (neue Seite). Letzteres passiert unter Umständen auch, wenn man eine zu breite oder zu große Tabelle hat.

Die Tabelle wird automatisch so breit wie der Inhalt. Es gibt jedoch in diesem Beispiel keinen Zeilenumbruch oder ähnlichen Komfort. Aus diesem Grund kann eine Tabelle auch sehr schnell zu breit werden.

```
\begin{table}[htbp]
  \begin{tabular}{|c|lr}
    Zeile 1, Spalte 1 & Zeile 1, Spalte 2 & Zeile 1, Spalte 3 \\
    Zeile 2, Spalte 1 & Zeile 2, Spalte 2 & Zeile 2, Spalte 3 \\
    \hline
    Zeile 3, Spalte 1 & Zeile 3, Spalte 2 & Zeile 3, Spalte 3 \\
  \end{tabular}
  \caption[Tabellenbeispiel mit tabular]{Einfache Tabelle mit tabular.}
\end{table}
```

Dieser Quelltext erzeugt eine einfache dreispaltige Tabelle, die die wesentlichen Möglichkeiten demonstriert. Im Einzelfall kann es sinnvoll sein, die Tabelle zu zentrieren, dies ist durch

eine Klammerung mittels `\begin{center} ... \end{center}` innerhalb der `table`-Umgebung möglich.

Zeile 1, Spalte 1	Zeile 1, Spalte 2	Zeile 1, Spalte 3
Zeile 2, Spalte 1	Zeile 2, Spalte 2	Zeile 2, Spalte 3
Zeile 3, Spalte 1	Zeile 3, Spalte 2	Zeile 3, Spalte 3

Tabelle 4.1: Einfache Tabelle mit `tabular`.

4.2 Das tabularx-Package

Mit dem Package `tabularx` hat man ein paar mehr Formatierungsmöglichkeiten. Man kann eine Zielbreite der Tabelle angeben, z.B. mit `\begin{tabularx}{8cm}{|c|X|X}`. Das `x` bezeichnet dabei eine Tabellenspalte mit anpassbarer Breite. Wie man sieht, klappt dann auch plötzlich der Zeilenumbruch in Tabellenzellen.

```
\begin{table}[htbp]
  \begin{tabularx}{10cm}{|c|X|X}
    Zeile 1, Spalte 1 & Zeile 1, Spalte 2 & Zeile 1, Spalte 3 \\
    Zeile 2, Spalte 1 & Zeile 2, Spalte 2 & hier ist Text,
    der den Rest der Tabelle beeinflusst \\
  \end{tabularx}
  \caption{Tabellenbeispiel mit tabularx}
\end{table}
```

Zeile 1, Spalte 1	Zeile 1, Spalte 2	Zeile 1, Spalte 3
Zeile 2, Spalte 1	Zeile 2, Spalte 2	hier ist Text, der den Rest der Tabelle be- einflusst

Tabelle 4.2: Tabellenbeispiel mit `tabularx`

5 Abbildungen

5.1 Grundlagen

L^AT_EX ist bei den Grafikformaten recht wählerisch, um es mal vorsichtig zu formulieren. Grundsätzlich versteht latex nur Grafiken im Encapsulated-PostScript Format (*.eps). Zur Erstellung von PDF-Dateien kann und sollte pdflatex verwendet werden. Dieses Programm wiederum versteht nur Grafiken in den Formaten *.pdf, *.jpg und *.png. Für universelle Benutzbarkeit bedeutet das, daß alle Grafiken sowohl als *.eps als auch in einem für pdflatex verständlichen Format vorliegen müssen.

Das dieser Vorlage beiliegende Makefile kümmert sich um die Konvertierung sämtlicher Formate, genaueres findet sich in Anhang C.

5.1.1 Einbinden von Grafiken

Die Vorlage definiert mehrere fertige Befehle zum Einfügen von Bildern, die für die meisten Anwendungen ausreichen sollten. Die Syntax für den einfachsten dieser Befehle lautet

```
\pic[shortcaption]{dateiname}{width}{caption}{label}
```

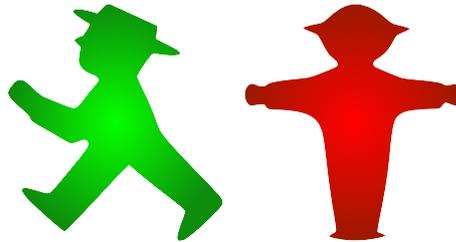


Abbildung 5.1: Eine einfache Abbildung.

`shortcaption` ist ein optionaler Kurztitel für das Abbildungsverzeichnis. Der `dateiname` ist ohne Endung anzugeben, damit sowohl latex als auch pdflatex ohne Quelltextänderungen funktionieren. `width` ist die Breite des Bildes und ist mitsamt Einheit anzugeben (z.B. „3cm“). `caption` ist die Bildunterschrift, die auch im Abbildungsverzeichnis erscheint, sofern keine `shortcaption` angegeben ist. Das `label` fungiert als Ziel für Querverweise auf das Bild. Falls

man einen Rahmen um das Bild haben will, gibt es den Befehl `\fpic` mit den gleichen Parametern wie `\pic`.

Für die Positionierung von Grafiken gelten die gleichen Parameter `htbp` wie bei Tabellen. Sollte die Grafik zu breit für eine Seite sein, landet sie ganz am Ende! Ein Eintrag im Abbildungsverzeichnis wird ebenfalls automatisch erstellt. Da die `caption` üblicherweise eine längere Bildbeschreibung enthält, sollte man unbedingt eine sinnvolle `shortcaption` setzen, damit das Abbildungsverzeichnis nicht zugemüllt wird.



Abbildung 5.2: Eine einfache gerahmte Abbildung.

Querverweise zu Bildern lassen sich selbstverständlich auch mittels `\ref{}` realisieren, so z.B. dieser Verweis zu Bild [5.1](#) oder zur gerahmten Abbildung [5.2](#).

5.1.2 Zwei Bilder parallel



Abbildung 5.3: Beispiel für zwei parallele Bilder.

Für zwei Bilder parallel (z.B. für Vergleiche von Meßwerten) gibt es analog den Befehl

```
\twopics[shortcaption]{datei1}{datei2}{caption}{label}
```

Die Bildgröße wird dabei immer auf die halbe Textbreite gesetzt, so daß die zwei Bilder genau die Textbreite einnehmen. Der Querverweis führt dann nur zu der übergeordneten Umgebung, die beide Bilder beinhaltet, z.B. zu [5.3](#). Es ist nicht möglich, auf eines der beiden Bilder zu verweisen.

5.2 Weitere Grafikpakete

5.2.1 Subfigure

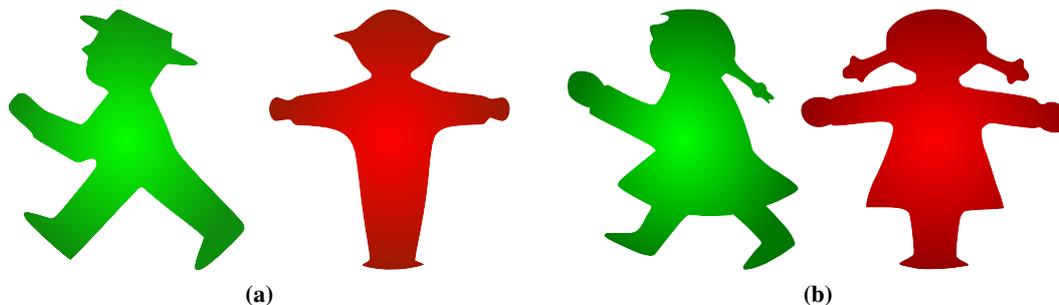


Abbildung 5.4: Zwei Bilder mit subfig (a) das linke Bild, (b) das rechte Bild

Das Package `subfig` erlaubt die Erstellung von Untergrafiken (1a, 1b, ...). Auch dafür gibts ein vorgefertigtes Skript, allerdings nur für zwei Unterbilder. Für mehr als zwei solche Bilder muß die ganze Sache per Hand eingetragen werden. Das Skript `\subfigs` arbeitet exakt wie `\twopics` und nimmt auch die gleichen Parameter. Das sieht dann aus wie in [Abbildung 5.4](#). Querverweise in die Subfigs funktionieren natürlich auch, wie dieser Verweis zu [Abbildung 5.4b](#).

Man kann auch manuell solche Unterabbildungen erzeugen, dafür kann man sich am Code des `subfigs`-Makros in `user.sty` orientieren.

5.2.2 Das pgf-Package

Dieses Package wurde entwickelt, um Graphen und diverse andere mathematische Dinge direkt in \LaTeX zu setzen. Der Funktionsumfang ist gewaltig, die Anleitung hat mal eben schlappe 330 Seiten, enthält aber eine gut verständliche Kurzeinführung. Man kann damit also so ziemlich alles Zeichnen, was sich als Prinzipskizze in einer Arbeit einbauen läßt, z.B. die Struktur eines Programms, Erklärungen und Skizzen zu geometrischen Sachverhalten und und und... Man kann natürlich auch einfach nur eine Tasse Milchkaffee mit Zucker basteln, wie in [Abbildung 5.5](#) zu sehen.

Achtung: Beim Testen unter Linux mußte dieses Paket manuell aktualisiert werden, da die TexLive-Installation nicht die notwendige Version 1.10 beinhaltete.



Abbildung 5.5: Beispiel für die Verwendung des pgf-Packages

5.2.3 Gnuplot

Mit Gnuplot lassen auf einfache Weise Funktionsgraphen oder Meßkurven darstellen. Die Formatierung der Kurven erfolgt entweder über ein Kommandozeilen-Interface oder mittels einfacher Skriptdateien. Das Makefile beinhaltet Routinen zur Erstellung von Diagrammen aus derartigen Skriptdateien, genaueres ist in Anhang C erläutert. Abbildung 5.6 zeigt Beispiele für mit Gnuplot erstellte Graphen.

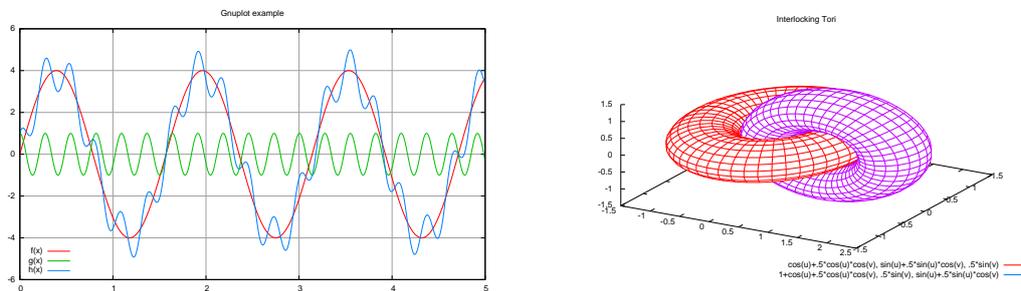


Abbildung 5.6: Zwei mit Gnuplot erstellte Graphen

Um einen Plot als Datei zu exportieren, bieten sich die folgenden Einstellungen an:

```
set terminal postscript eps enhanced color solid
set output "dateiname.eps"
```

Diese eps-Grafik kann dann weiter nach pdf konvertiert werden.

Bei der Verwendung des Makefiles darf weder `terminal` noch `output` gesetzt sein, das würde die Einstellungen des Makefiles überschreiben.

6 Algorithmen

In vielen Fällen ist es notwendig, Programmquelltexte in einer Arbeit zu setzen. Dabei kommen sich die vielen Sonderzeichen oftmals mit dem Interpreter von \LaTeX ins Gehege, weswegen es dafür spezielle Umgebungen zur korrekten Formatierung gibt.

6.1 Die Verbatim-Umgebung

Für kurze Quelltextausschnitte, die exakt wie getippt dargestellt werden sollen, gibt es die `verbatim`-Umgebung. Damit kann man längere Code-Blöcke schreiben, allerdings ohne weitere Formatierung.

Das ist ein Beispiel für einen `verbatim`-Block.
Hier werden auch \LaTeX -Sonderzeichen ignoriert!
Mit der Zeilenlänge sollte man tierisch aufpassen

Ebenfalls interessant ist die Formatierung im Fließtext mittels `\verb|`. Alles was auf die erste Pipe folgt, wird als `verbatim` behandelt, bis eine weitere Pipe folgt. Anstatt der Pipes kann auch ein beliebiges anderes Zeichen als Trenner genutzt werden.

6.2 Listings und andere Pakete

```
#include <stdio.h>
int main(int argc, char **argv) {
    printf("Hallo_Welt!\n");
    return 0;
}
```

Algorithm 6.1: Beispiel für die `algorithm`-Umgebung mit `listings`. Die `Algorithm`-Umgebung besorgt die Positionierung (wie bei `figures`, `tables` usw), das `lstlisting` innendrin die Formatierung.

Für das Setzen von Quelltexten gibt es zahlreiche spezialisierte Pakete, die beispielsweise für bestimmte Programmiersprachen oder das Setzen von Pseudocode bestimmt sind, z.B. die Packages `algorithm2e` und `algorithmicx`. Die Vorlage verwendet das `listings`-Package.

Für das `listings`-Package wird am Anfang einmal definiert, welche Programmiersprachen innerhalb der Arbeit unterstützt werden sollen. Dafür steht in der Datei `vorlage.tex` die Zeile `\lstset{optionen}` drin. Die Voreinstellungen für Syntax-Highlightung und ähnliches lassen sich ebenfalls ändern, sofern das gewünscht wird.

Für die Code-Blöcke wurde die Umgebung `algorithm` definiert, das funktioniert dann analog zu `table`, `figure` usw. Ein Beispiel findet sich in Algorithmus 6.1.

7 Sonstiges

7.1 Listen und Aufzählungen

Es gibt im wesentlichen drei Sorten Listen, bullet lists, Aufzählungslisten und sonstige Listen. Diese können beliebig geschachtelt werden. Das Aussehen der Bullets bzw. das Format der Aufzählung kann mit diversen Parametern verändert werden, genaueres ist in [GMS00] nachlesbar.

7.1.1 Bullet Lists

Diese werden in der `itemize`-Umgebung erstellt. Jeder Eintrag wird mit `\item` eingeleitet. Und so siehts aus

- Item 1
- Item 2
 - Schachtelung 1
 - Schachtelung 2
 - * und noch mehr Schachteln
- Item 3

7.1.2 Aufzählungen

Diese werden genauso definiert, nur daß das Stichwort hier `enumerate` heißt.

1. Item 1
2. Item 2
 - a) Schachtelung 1
 - b) Schachtelung 2
 - i. und noch mehr Schachteln
3. Item 3

7.1.3 sonstige Listen

Diese werden mittels `description` definiert und enthalten jeweils ein Stichwort, welches den Platz der Aufzählungsnummer einnimmt:

A ein Buchstabe

B noch ein Buchstabe

Text viele Buchstaben

7.2 Wörtliches Zitieren

In vielen Fällen wird man in wissenschaftlichen Arbeiten auch wörtlich zitieren müssen, auch dafür gibt es in \LaTeX geeignete Umgebungen. Die Vorlage bietet zusätzlich den Befehl `attrib`, um Quellen für Zitate anzugeben. Innerhalb dieser Angaben können Verweise ins Literaturverzeichnis sowie Fußnoten gesetzt werden. Beispiele dafür sind angegeben.

7.2.1 Quote

Diese Umgebung ist insbesondere für kurze Zitate, Sätze u.ä. vorgesehen. Die Zitate werden links und rechts eingerückt. Die Vorlage verwendet eine leicht umdefinierte Version der Umgebung, die optional die Angabe des Zitaturnehers erlaubt, wenn man den `attrib`-Befehl nicht nutzen möchte.

„ \LaTeX kann alles, außer Kaffee kochen.“

Uwe Domaratius

Diese Zeile wird mit dem folgenden Block erzeugt:

```
\begin{quote}[\textit{Uwe Domaratius}]
"\"LaTeX{} kann alles, außer Kaffee kochen.""
\end{quote}
```

Mit der Benutzung von `attrib` sieht das Beispiel ganz genauso aus:

„ \LaTeX kann alles, außer Kaffee kochen.“

Uwe Domaratius

Hier sieht der Quellcode folgendermaßen aus:

```
\begin{quote}
"\"LaTeX{} kann alles, außer Kaffee kochen.""
\attrib{\textit{Uwe Domaratius}}
\end{quote}
```

7.2.2 Quotation

Für längere Ausschnitte mit mehreren Abschnitten bietet sich die `quotation`-Umgebung an. Auch hier erfolgt an beiden Rändern eine Einrückung, ebenso wurde die Umgebung umdefiniert um eine optionale Zitatquelle (inklusive Textauszeichnung) angeben zu können. Als Beispiel kann man sich auch mal selbst zitieren:

„Diese Datei enthält die Anleitung zur Nutzung der Vorlage für verschiedene Typen von Arbeiten. Sie basiert auf einer entsprechenden Vorlage der TU Chemnitz und wurde erheblich modifiziert.

In den folgenden Kapiteln dieser Anleitung wird ein Überblick über die Verwendung der Vorlage, Zweck der Dateien und typische Anwendungsfälle gegeben.“

(Diplomarbeitvorlage für \LaTeX , [Dom06])

Die Benutzung des optionalen Parameters bzw. des `\attrib`-Befehls erfolgt analog zu den Beispielen der `quote`-Umgebung.

7.2.3 Verse

Diese Umgebung formatiert den Text als Ausschnitt aus einem Gedicht oder Liedtext, also bei Quellen, bei denen der Zeilenumbruch und die Aufteilung in Strophen beachtet werden muß. Allerdings erlaubt die \LaTeX -Standardumgebung `verse` nur sehr wenig, was die gestalterische Freiheit angeht. Für ein sehr ausgefeiltes Layout empfiehlt sich die Benutzung des Packages `verse`, mit dem sich das Aussehen derartiger Zitate schön steuern läßt.

Ein Limerick

Ein Hacker hat boshaft gelacht,
und Chaos per Virus entfacht.
Der Wurm ist gemein,
Und hat obendrein,
den Rechner zum Absturz gebracht.

Wikipedia¹

7.3 Der Mathemodus

Der Formelsatz ist ja die große Stärke von \LaTeX , es gibt auch massenhaft Packages dafür, das kann man aber alles im \LaTeX -Begleiter [GMS00] nachlesen.

¹[http://de.wikipedia.org/wiki/Limerick_\(Gedicht\)#weitere_Beispiele](http://de.wikipedia.org/wiki/Limerick_(Gedicht)#weitere_Beispiele)

Die Matheumgebung sollte klar sein, mit $\$$ wird im Fließtext gewechselt und man kann so nette Sachen wie $a^2 + b^2 = c^2$ schreiben. Für komplexe Formeln gibt es die `equation`-Umgebung und noch vieles anderes mehr, eine einfache abgesetzte Formel wird in `\[` und `\]` geklammert, so wie die folgende:

$$\beta = \arcsin\left(\frac{n_1}{n_2} \sin(\alpha)\right)$$

Eine etwas komplexere Gleichung mit Arrays sieht dann zum Beispiel so aus wie Gleichung 7.1:

$$R_s = \left[\frac{\sin(\alpha - \beta)}{\sin(\alpha + \beta)}\right]^2 \quad R_t = \left[\frac{\tan(\alpha - \beta)}{\tan(\alpha + \beta)}\right]^2 \quad R(0) = \frac{(n_1 - n_2)^2}{(n_1 + n_2)^2} \quad (7.1)$$

7.4 Theoreme und Definitionen

\LaTeX stellt als Umgebung für wissenschaftliche Arbeiten die `\theorem`-Umgebung als Basis für eigene Klassen zur Verfügung. Diese Grundvariante kann man undefinieren und erweitern, z.B. für Definitionen, Beweise oder ähnliches. Die Vorlage benutzt das Paket `ntheorem` und definiert folgende Umgebungen (die je nach gewählter Sprache passend angezeigt werden):

Theorem 1 *Ich bin ein Theorem, welches durch die Umgebung `theorem` definiert wird.*

Lemma 1 *Ich bin ein Lemma, welches durch die Umgebung `lemma` definiert wird. Und wenn man zwei Lemmata hat, ergibt sich unter Umständen ein Dilemma.* □

Satz 1 *Ich bin ein Satz, welches durch die Umgebung `prop` definiert wird. Falls die Sprache auf Englisch gesetzt wird, heiße ich Proposition.* □

Korollar 1 *Ich bin ein Korollar, welches durch die Umgebung `coroll` definiert wird. Falls die Sprache auf Englisch gesetzt wird, heiße ich Corollary.* □

Definition 1 *Ich bin eine Definition, welche durch die Umgebung `definition` definiert wird. Und als Beispiel definieren wir tatsächlich noch was: Sei $\varepsilon < 0$.* □

Beispiel 1 *Ich bin ein Beispiel, welches durch die Umgebung `example` definiert wird. Falls die Sprache auf Englisch gesetzt wird, heiße ich Example.* □

Anmerkung 1 *Ich bin eine Anmerkung, welche durch die Umgebung `remark` definiert wird. Falls die Sprache auf Englisch gesetzt wird, heiße ich Remark.* □

BEWEIS Ich bin ein Beweis, welcher durch die Umgebung `proof` definiert wird. Falls die Sprache auf Englisch gesetzt wird, heie ich Proof. ■

Man kann auch Verweise auf diese Umgebungen erzeugen, indem man innerhalb der Umgebung ein entsprechendes Label setzt, so wie in Theorem 1 geschehen. Ebenso ist es möglich, eine Liste aller Definitionen etc. auszugeben. Die vielen weiteren Möglichkeiten sind im Manual zum Package erklärt. Sämtliche Sätze, Lemmata usw. werden automatisch durchnummeriert, man kann auch eine Liste von Sätzen analog zum Abbildungsverzeichnis erzeugen lassen.

Achtung: Unter Umständen erzeugt L^AT_EX eine Warnung bezüglich fehlender Fonts, das stört aber nicht weiter.

7.5 Hilfsmittel für den Autor

→ In der Vorlage sind zwei Befehle definiert, namentlich `\missing{Beispiel fehlt}`, das Ergebnis ist ein Pfeil am Rand mit dem Hinweistext. (Beispiel fehlt)

Die zweite Sache ist ein farbiger Kasten, in den man Kommentare reinschreiben kann, was noch so gemacht werden muß. Der Befehl dazu lautet `\comment{text}`.

Das ist der Kommentarkasten, da kann man Anmerkungen reinschreiben. Durch die Farbe fällt er beim Drüberblättern auf, so daß man merkt, wenn irgendwo noch was fehlt.

8 Glossar, Abkürzungsverzeichnis und Index

Die Vorlage enthält alles, um einen Stichwortindex, ein Glossar sowie ein extra Abkürzungsverzeichnis zu generieren. Dazu sind neben dem eigentlichen Setzen des Dokuments aber eine Handvoll Extraschritte notwendig. Die Befehlsfolge lautet:

```
(pdf)latex vorlage
# Index bauen
makeindex vorlage
# Abkürzungsverzeichnis bauen
makeindex -s vorlage.ist -t vorlage.alg -o vorlage.acn vorlage.acr
# Glossar bauen
makeindex -s vorlage.ist -t vorlage.glg -o vorlage.gls vorlage.glo
(pdf)latex vorlage
```

Das zur Vorlage gehörende Makefile kümmert sich um diese Abhängigkeiten, so daß man sich das ständige Tippen ersparen kann. Genaueres dazu steht im Anhang [C](#).

8.1 Index

Einen Index kann man sehr leicht realisieren. Dazu wird nur der Befehl `\index{schlagwort}` benötigt. Im Index erscheinen dann sämtliche auf diese Art indizierten Schlagwörter mitsamt Seitenzahlen. Es gibt umfangreiche Möglichkeiten, den Index zu formatieren, ein Nachschlagen in [\[GMS00\]](#) kann dabei nicht schaden. Die Vorlage enthält einen umfangreichen Index, die entsprechenden `\index{}`-Befehle finden sich im gesamten Quelltext verstreut.

8.2 Glossar

Das Glossar wird auf eine ähnliche Art und Weise erstellt. Wann immer man an einer Stelle von einem bestimmten Begriff spricht und einen Eintrag im Glossar dazu haben will, nutzt man den Befehl `\xglossary`. Die Syntax dazu zeigt das Beispiel.

```
\xglossary{name=Beispiel,
description=dient dem Verständnis}
{Beispiel}
```

8.3 Abkürzungsverzeichnis

Im Abkürzungsverzeichnis, auch Liste der Akronyme genannt, werden alle in der Arbeit verwendeten Abkürzungen gelistet und auf die Seite verwiesen, auf der sie das erste Mal benutzt und somit definiert werden. Ein neues Akronym wird mit dem Befehl `\newacronym` definiert:

```
\newacronym[DBA]{DBA}
{dreibuchstabile Abkürzung}
{name={DBA},description={Dreibuchstabile Abkürzung}}
```

Das definiert das Akronym DBA und einen Eintrag fürs Abkürzungsverzeichnis (erzeugt aber keine Ausgabe direkt im Quelltext). Nun kann man, wann immer notwendig, direkt das Akronym `\DBA` benutzen. Bei der ersten Benutzung wird die Langfassung aus der ersten geschweiften Klammer genutzt, bei allen weiteren Benutzungen wird das Akronym genutzt. Der folgende Absatz enthält mehrere Beispiele für Akronyme, welche hier im Quelltext des Dokumentes definiert wurden.

Hier wird das Akronym für dreibuchstabile Abkürzung (DBA) (gesunder Menschenverstand (GMV) kennt das auch als three letter word (TLW)) das erste mal genutzt, und hier erfolgt die zweite Nutzung von DBA (mit GMV auch TLW genannt). Im Abkürzungsverzeichnis taucht dann das Akronym, die Erklärung und die Seitenzahl auf, an der es mittels `\newacronym` definiert wurde. Im Fließtext sollte man dem Akronym geschweifte Klammern gönnen, damit dahinter auch ein Leerraum bleibt, also `\DBA{ }` in diesem Beispiel. Die Einzelheiten hierzu sind in der Anleitung zum `glossary`-Paket erklärt.

A Setzen des Dokuments

Zum korrekten Setzen der Vorlage sind mehrere Arbeitsschritte notwendig. Es empfiehlt sich, dafür eine Batchdatei zu schreiben, sofern man nicht ohnehin das mitgelieferte Makefile nutzt.

1. Grafiken in geeignete Formate konvertieren

`pdflatex` kann Grafiken in den Formaten `jpg`, `png` und `pdf` verarbeiten, `latex` kennt nur `eps`, es müssen also alle Grafiken in den jeweils benötigten Formaten vorhanden sein.

2. \LaTeX laufen lassen

```
latex vorlage.tex oder pdflatex vorlage.tex
```

Dies muß mehrmals gemacht werden, da erst beim zweiten Mal Inhaltsverzeichnis, Abbildungsverzeichnis etc. korrekt eingebunden werden. Diese werden im ersten Durchlauf lediglich erstellt. Durch das Einbinden im zweiten Durchlauf kann es jedoch sein, daß sich Seitenzahlen für Querverweise etc. erneut ändern.

3. Erzeugen des Literaturverzeichnisses

```
bibtex vorlage.aux
```

4. Index erstellen

```
makeindex vorlage
```

5. Glossar erstellen

```
makeindex -s vorlage.ist -t vorlage.glg -o vorlage.gls vorlage.glo
```

6. Abkürzungsverzeichnis erstellen

```
makeindex -s vorlage.ist -t vorlage.alg -o vorlage.acn vorlage.acr
```

7. Fertiges Dokument inklusive aller Verzeichnisse, Index, usw. setzen

```
latex vorlage.tex oder pdflatex vorlage.tex
```

Dies wird solange wiederholt, bis keine Warnungen „Rerun to get...“ mehr angezeigt werden.

B Verwendete L^AT_EX-Packages

amsmath

AMS mathematical facilities for L^AT_EX:

<http://www.ams.org/tex/amslatex.html>

babel

Multilingual support for Plain T_EX or L^AT_EX:

www.ctan.org/tex-archive/macros/latex/required/babel/babel.pdf

breakurl

Line-breakable `\url`-like links in hyperref when compiling via dvips/ps2pdf:

www.ctan.org/tex-archive/macros/latex/contrib/breakurl/breakurl.pdf

calc

Simple arithmetic in L^AT_EX commands:

www.ctan.org/tex-archive/macros/latex/required/tools/calc.pdf

caption

Customising captions in floating environments:

www.ctan.org/tex-archive/macros/latex/contrib/caption/anleitung.pdf

color

Colour control for L^AT_EX documents:

www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.pdf

float

Improved interface for floating objects:

www.ctan.org/tex-archive/macros/latex/contrib/float/README

fontenc

Standard package for selecting font encodings

glossary

Create a glossary:

www.ctan.org/tex-archive/macros/latex/contrib/glossary/glossary.pdf

hyperref

Extensive support for hypertext in L^AT_EX:

www.ctan.org/tex-archive/macros/latex/contrib/hyperref/doc/manual.pdf

ifthen

Conditional commands in L^AT_EX documents

inputenc

Accept different input encodings

KOMA-Script

A bundle of versatile classes and packages:

<http://www.komascript.de>

listings

Typeset source code listings using L^AT_EX:

www.ctan.org/tex-archive/macros/latex/contrib/listings/listings-1.3.pdf

makeidx

Standard LaTeX package for creating indexes

ntheorem

Enhanced theorem environment

www.ctan.org/tex-archive/macros/latex/contrib/ntheorem/ntheorem.pdf

pgf

Create PostScript and PDF graphics in T_EX: <http://www.ctan.org/tex-archive/graphics/pgf/doc/generic/pgf/version-for-pdftex/en/pgfmanual.pdf>

pslatex

Use PostScript fonts by default

subfig

Figures broken into subfigures:

www.ctan.org/tex-archive/macros/latex/contrib/subfig/subfig.pdf

tabularx

Tabulars with adjustable-width columns

url

Verbatim with URL-sensitive line breaks

verse

Aids for typesetting simple verse:

www.ctan.org/tex-archive/macros/latex/contrib/verse/verse.pdf

Allgemeine Package-Informationen

Eine Auflistung aller verfügbaren Packages gibt es unter <http://texcatalogue.sarovar.org/alpha.html>.

C Das Makefile

Da es etwas mühsam ist, immer wieder \LaTeX , BibTeX, Makeindex usw. aufzurufen und sämtliche Parameter im Kopf zu behalten, habe ich dafür ein makefile geschrieben. Wer keine Ahnung hat, was das ist, benutzt vermutlich Windows und muß daher sowieso drauf verzichten (und braucht nur noch den folgenden Absatz lesen).

Unter Windows empfiehlt sich als Editor für das Schreiben von \LaTeX das TeXnicCenter¹. Für diesen Editor kann man Ausgabeprofile definieren. Ein solches Ausgabeprofil kann dabei sämtliche notwendigen Aufrufe von BibTeX und Co. beinhalten. Diese Profile können als XML exportiert und importiert werden.

Das Makefile bietet natürlich erheblich mehr Möglichkeiten. Es erzeugt auf Wunsch alle drei gängigen Ausgabeformate (PDF, PS, DVI), konvertiert Bilder automatisch in die korrekten Formate, und beinhaltet diverse Checks. Da nur in den seltensten Fällen alles von Grund auf neu durchgearbeitet werden muß, ist die Arbeit mit dem Makefile auch wesentlich produktiver. Die einzelnen Targets des Makefiles sind in diesem mittels `make help` nachlesbar, ebenso ist das gesamte Makefile umfangreich dokumentiert.

¹<http://www.texniccenter.org/>

D Dateien der Vorlage

figs/	Verzeichnis für Vektorgrafiken in den Formaten fig und svg
pics/	Verzeichnis für Pixelgrafiken in den Formaten jpg und png
plots/	Verzeichnis für Gnuplot-Steuerdateien
Makefile	Das Makefile zum Setzen der gesamten Vorlage
literatur.bib	Die Literatureinträge für BibTeX
alphaurl.bst	Der verwendete BibTeX-Stil
vorlage.dvi	Die fertige Vorlage als DVI
logo_*.*	Schriftzug der TU Chemnitz in Deutsch und Englisch, als eps und pdf
vorlage.pdf	Die fertige Vorlage als PDF
vorlage.ps	Die fertige Vorlage als PS
user.sty	definiert zahlreiche zusätzliche Befehle
anhaenge.tex	die Anhänge
data.tex	grundlegende Metadaten
kapitel*.tex	die Inhalte der einzelnen Kapitel der Vorlage
titel.tex	das Layout der Titelseite
vorlage.tex	Die Hauptdatei, von hier werden alle weiteren Dateien eingebunden
figs/ampelmann.fig	Ampelmännchen im fig-Format
figs/ampelfrau.svg	Ampelfrau im svg-Format
pics/ampel*.*	Ampelfrau und -männchen als eps und pdf
plots/sinus.eps	Beispielplot Sinuskurven
plots/tori.eps	Beispielplot Interlocked Tori
plots/sinus.pdf	Beispielplot Sinuskurven
plots/tori.pdf	Beispielplot Interlocked Tori
plots/sinus.plt	Gnuplot Skript
plots/tori.plt	Gnuplot Skript

Die Verzeichnisse sind nur relevant bei der Benutzung des Makefiles! Sollte dieses nicht genutzt werden, müssen die Grafiken und Plots vor dem Setzen in ein für (pdf)latex verwertbares Format konvertiert werden. Die Vektorgrafiken sollten dabei keinesfalls rasterisiert, sondern direkt nach EPS und/oder PDF konvertiert werden.

E Hilfsdateien

Die folgenden Dateien werden von den verschiedenen Programmen beim Setzen der Vorlage erzeugt und können danach gefahrlos gelöscht werden:

vorlage.acn	Acronyme
vorlage.acr	Acronyme
vorlage.alg	makeindex logfile
*.aux	auxiliary files
vorlage.bbl	Bibliographie
vorlage.blg	BibTeX logfile
vorlage.glg	Glossar logfile
vorlage.glo	Glossar
vorlage.gls	Glossar
vorlage.idx	Index
vorlage.ilg	Index
vorlage.ind	Index
vorlage.ist	Index
vorlage.loa	Algorithmenverzeichnis
vorlage.lof	Abbildungsverzeichnis
vorlage.log	Logfile
vorlage.lot	Tabellenverzeichnis
vorlage.out	PDF-Bookmarks
vorlage.thm	Verweise auf Definitionen etc.
vorlage.toc	Inhaltsverzeichnis

F Version History

1.20 29.01.2010

Anpassung auf das pgf-Package 2.0

1.10 04.04.2007

Anpassung auf das pgf-Package 1.10, einige kleinere Korrekturen und Ergänzungen, Zusammenfassung beider Vorlagen in eine Gesamtvorlage ohne pstricks

1.00 07.10.2006

erste Veröffentlichung

G Known Bugs

- \LaTeX bringt möglicherweise die Warnung „no bold math fonts“.
- je nach benutzter Version von KOMA-Script kann es zu Warnungen wegen Verwendung veralteter Optionen kommen, das sind Probleme in anderen benutzten Packages, nicht in dieser Vorlage

Literaturverzeichnis

- [Dom06] Uwe Domaratus. Diplomarbeitvorlage mit \LaTeX . Diplomarbeit, Chemnitz University of Technology, 2006.
- [GMS00] Michael Goossens, Frank Mittelbach, and Alexander Samarin. *Der \LaTeX -Begleiter*. Pearson Studium, 2000. Available from: <http://www.pearson-studium.de/main/main.asp?page=bookdetails&ProductID=106617&SID=%7B0EF7E587%2D7DC6%2D4CAA%2D969C%2DB07A1E07F1EE%7D>.

Index

Abbildungen, 10–13

Formate, 10

`\fpic`, 11

Gnuplot, 13

Konvertierung, 10

`pgf`, 12

`\pic`, 10

`\subfigs`, 12

`\twopics`, 11

Abkürzungen

Abkürzungsverzeichnis, 22

`\newacronym`, 22

`\caption`, 8

`\comment`, 20

Dateien, 28

Formelsatz, 18–19

Fußnoten, 7

Gliederung, 3–4

`\minisec`, 4

Glossar, 21

`\xglossary`, 21

Hilfsdateien, 29

Index, 21

`\index`, 21

Inhaltsverzeichnis, 3–4

`\secnumdepth`, 4

`\tocdepth`, 4

L^AT_EX-Packages, 24

Listen, 16

description, 17

enumerate, 16

itemize, 16

Literatur

BibTeX, 6, 7

`\cite`, 7

Literaturverzeichnis, 6, 7

Makefile, 27

`\missing`, 20

PDF, 10

pdfLaTeX, 6, 10

Quelltexte, 14–15

listings, 14

`\verb`, 14

`\verbatim`, 14

Tabellen, 8–9

Positionierung, 8

Tabellenverzeichnis, 8

`\table`, 8

`\tabular`, 8

`\tabularx`, 9

theorem, 19

Verweise

`\hyperref`, 6

`\label`, 6

`\pageref`, 6

`\ref`, 6

URLs, 6

in Fußnoten, 7
zu externen Zielen, 6

Zitate

attrib, 17
quotation, 18
quote, 17
verse, 18

Selbständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendeten Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Astadt, den 29. Januar 2010

Uwe Domaratus