

Universität des Saarlandes
FR 6.1, Mathematik
Bachelorseminar Zahlentheorie



Tipps und Tricks zu L^AT_EX

Handout zur Sitzung vom 04.06.2009

erstellt von Ute Gebhardt und Alice Keller

Inhaltsverzeichnis

1	Die Hauptdatei	3
2	Kapitel anlegen	3
3	Mathematische Umgebungen	4
4	Einiges zu Abständen	5
5	Verweise	6
6	Packages erstellen und einbinden	6
7	Allgemeines zu Verzeichnissen	9
8	Das Inhaltsverzeichnis	10
9	Der Index	11
10	Das Symbolverzeichnis	13
11	Das Literaturverzeichnis	14
12	Weiterführende Literatur	15

Dieses Handout ist eine sehr lange Stichwortliste zu dem Vortrag, nicht mehr — aber auch nicht weniger.

1 Die Hauptdatei

Dokumentklasse
Packages german epic amsmath epsfig amssymb <i>eigenes</i>
Lege Verzeichnisse an: Index Symbolverzeichnis Literaturverzeichnis
<code>\begin{document}</code>
Inhaltsverzeichnis anlegen und ausführen
Lese ein und schreibe Textdateien: Kapitel 1 : Kapitel n
Schreibe Verzeichnisse: Symbolverzeichnis Index Literaturverzeichnis
<code>end{document}</code>

2 Kapitel anlegen

Man legt für jedes Kapitel eine separate Datei mit aussagekräftigem Namen an.

- Die einzelnen Kapitel-Dateien sind keine eigenständigen \LaTeX -Dateien und brauchen daher keinen Kopf, `\documentclass` o.ä.
- Jede enthält

- einen *Arbeitstitel* (wird später durch die Überschrift ersetzt),
 - evtl. *Unterüberschriften* (je nach Feinheit der vorliegenden Gliederung),
 - eine kurze *Zusammenfassung* des geplanten Inhaltes, evtl. eine *Überleitung* zum nächsten Kapitel.
- Kapitel werden mit `\input` in die Hauptdatei eingebunden (an der Stelle zwischen `\begin{document}` und `\end{document}`, wo das Kapitel später stehen soll).

Gliederungsbefehle und automatische Numerierung

Kapitelüberschriften etc. werden *nicht* manuell durch Umschalten auf Fettschrift o. ä. erstellt!

Man verwendet die bereitstehenden Gliederungsbefehle:

<code>\documentclass{...}</code>	article	report	book	amsbook
<code>\part</code>	I	I	I	1
<code>\chapter</code>	—	1	1	1
<code>\section</code>	1	1.1	1.1	1.
<code>\subsection</code>	1.1	1.1.1	1.1.1	1.1.
<code>\subsubsection</code>	1.1.1			1.1.1.
<code>\paragraph</code>		ohne Nummern		
<code>\subparagraph</code>				

- Überschriften *ohne Nummern* z. B. durch

$$\backslash\text{section*}\{Überschrift\}$$
 Diese werden nicht ins Inhaltsverzeichnis aufgenommen und setzen den Zähler nicht hoch.
- Überschriften mit *abweichendem Eintrag* im Inhaltsverzeichnis z. B. durch

$$\backslash\text{section}[EintragInhalt]\{Überschrift\}$$

3 Mathematische Umgebungen

Standard sind die Umgebungen `equation`, `displaymath` und `eqnarray`. Das Package `amsmath` bietet noch weitere wie `align`, `gather` usw. an (vgl. BspPaket).

4 Einiges zu Abständen

Es gibt drei verschiedene Maßangaben für Abstände:

- feste Maße
- feste Maße in absoluten Größen wie
`\hspace{1cm}` `\vspace{1cm}` *Nicht verwenden!!*
- Schriftgrößen abhängige feste Maße wie
`\hspace{1em}` `\vspace{1ex}` *besser*
em bezeichnet die Breite eines m, ex die Höhe eines x.
- elastische Maße wie
`\setlength{\parskip}{1ex plus0.5ex minus0.2ex}`
Dabei kann der **Sollwert** um den *Dehnwert* erhöht bzw. um den *Schrumpfwert* erniedrigt werden.

Globale Veränderungen

Bei ständig wiederkehrenden Abstandsänderungen zu überlegen:

→ Globale Änderung dieses Abstandes

Beispiele

- Zeilenabstand
- Einzug der ersten Zeile eines Absatzes
- zwischen Absätzen
- vor und nach gesonderten Umgebungen
- in Tabellen oder arrays

Schriftgrößenabhängige Abstände

Horizontale Abstände

<code>\hspace</code>	Breite variabel
<code>\quad</code>	normaler Zwischenraum in Schriftgröße
<code>\qquad</code>	das Doppelte
<code>\quad\quad</code>	normales Leerzeichen
<code>\;</code>	etwas weniger als ein normales Leerzeichen
<code>\,</code>	noch weniger

Der Abstand `\quad` wird eingefügt

- nach Aufruf von Befehlen wie `\LaTeX` in einem Text
- in abgesetzten mathematischen Formeln vor dem Satzzeichen.

vertikale Abstände

<code>\vspace</code>	Länge variabel
<code>\bigskip</code>	eine Zeile
<code>\medskip</code>	die Hälfte einer Zeile
<code>\smallskip</code>	ein Viertel einer Zeile

Mit `\renewcommand {\arraystretch}{x}` wird der horizontale Zellenabstand auf x ·(normaler Abstand) gesetzt.

Mit `\renewcommand {\baselinestretch}{x}` wird der Zeilenabstand auf x ·(normaler Abstand) gesetzt.

5 Verweise

Alle Verweise und jede Verwaltung von Verzeichnissen sollten so weit wie möglich von L^AT_EX gemacht werden!

Labels setzen

Syntax: `\label{Name}`

- `\label` steht typischerweise nach Gliederungsbefehlen oder nach Beginn einer Umgebung
- Das gilt insbesondere für `\begin{equation}`, aber nicht für `$$... $$`
- man sollte sich bei der Namensgebung Gedanken machen und aussagekräftige Namen verwenden

Labels aufrufen

Syntax: `\ref{Name}` oder `\pageref{Name}`

- `\ref` liefert die Gliederung- bzw. Formelnummer von *Name*.
- `\pageref` liefert die Seitenzahl, auf der *Name* steht.
- Um Layout (z. B. umgebende Klammern) muß man sich selbst kümmern.

6 Packages erstellen und einbinden

Generelles

- Packages sind erkennbar an der Endung `.sty`.
- Sie enthalten zusätzliche Befehle, Umgebungen, Symbole etc.
- Um auf die zusätzlichen Befehle etc. zugreifen zu können, muß zunächst das entsprechende Package im Vorspann eingebunden werden.

Ein eigenes .sty-File anlegen

Die eigene .sty-Datei enthält selbstdefinierte Stilvorlagen, Makros etc. und wird mit `\usepackage` in das Hauptdokument eingebunden.

Vorüberlegungen:

- Für welche Objekte brauche ich Makros?
 - häufig vorkommende Befehle mit umständlicher Syntax (um das Tippen zu vereinfachen)
 - solche Objekte, bei denen ich noch nicht sicher bin, wie ich sie nennen werde (um nachträgliches Ändern zu vereinfachen)
 - häufig vorkommende Umgebungen für Sätze, Beweise, Beispiele etc. (für einheitliches Layout)
- Wie sollen die Bezeichner für diese Objekte lauten?

Makros für Befehle definieren

Für einzelne Befehle gibt es folgende Möglichkeiten:

- `\newcommand{\Name}[anz]{Definition}`
Der optionale Parameter *anz* gibt die Anzahl der zu übergebenden Parameter an. Auf diese kann mit `#1, ..., #anz` zugegriffen werden.
- `\renewcommand`, Syntax: wie bei `\newcommand`
wird verwendet, falls `\Name` bereits existiert und überschrieben werden soll (mit *Vorsicht* zu genießen).

Umgebungen definieren

Für Umgebungen hat man:

- `\newenvironment{Name}[anz]{zuerst}{zuletzt}`
In *zuerst* stehen die Befehle, die bei `\begin{Name}` ausgeführt werden sollen. Hier kann mit `#1, ..., #anz` auf evtl. übergebene Parameter zugegriffen werden.
In *zuletzt* stehen die Befehle, die bei `\end{Name}` ausgeführt werden sollen. Hier kann *nicht* auf die übergebenen Parameter zugegriffen werden.

Sinnvolles Definieren von Umgebungen

Anfängerfehler beim Gebrauch von `\newenvironment` und `\newcommand`:

- alles und jedes wird neu- bzw. undefiniert,
- zu viele verschiedene Schriftarten/-größen werden verwandt,
- weitere lustige Symbole aus kuriosen Packages, die man zufällig entdeckt, werden benutzt.

Das ist fehleranfällig, unübersichtlich, läßt sich auf anderen Computern oft nicht mehr \LaTeX en ...und kostet unnötig viel Zeit!

\LaTeX gibt für viele Situationen bereits Standardumgebungen vor, die sich in der Praxis bewährt haben. Wo möglich, sollte man sie benutzen, und nicht selbst zu kreativ werden!

Der `\newtheorem`-Befehl

Syntax (im `.sty`-File):

```

\newtheorem{Name}{Überschrift}
oder \newtheorem{Name}{Überschrift}[KapZähler]
oder \newtheorem{Name}[ThmZähler]{Überschrift}

```

Aufruf im Text: `\begin{Name} ... \end{Name}`

Dies erzeugt eine abgesetzte Umgebung mit der Überschrift *Überschrift* und einem Zähler. Das Verhalten des Zählers hängt von den optionalen Parametern *KapZähler* bzw. *ThmZähler* ab.

Die unterschiedlichen Zählerarten

- Ohne Verwendung der optionalen Zähler, werden alle Vorkommen von *Name*-Umgebungen der Reihe nach durchnummeriert.
- Bei Angabe eines Parameters *KapZähler* (z. B. `chapter`, `section`,...) wird die Nummer der *Name*-Umgebung zurückgesetzt, wenn der entsprechende Gliederungsbefehl aufgerufen wird.
- Bei Angabe eines Parameters *ThmZähler* (das ist der Name *Name* einer vorher definierten Theorem-Umgebung) erhält die neue Umgebung keine eigene Numerierung, sondern wird mit der angegebenen Umgebung mitgezählt.

Stil der Theorem-Umgebung ändern

benötigtes Package: `amsthm`

Das `amsthm`-Paket stellt drei Stil-Varianten zur Verfügung:

`plain`, `definition`, `remark`

(Das genaue Aussehen hängt von der Dokumentklasse ab.)

Der Befehl `\theoremstyle{Variante}` setzt für alle nachfolgenden `\newtheorem`-Befehle den Stil auf *Variante*.

(Es gibt auch `\newtheoremstyle`, mit dem man leicht eigene Stil-Varianten definieren kann.)

Die Beweis-Umgebung

Umgebungen für Beweise werden ebenfalls vom `amsthm`-Paket zur Verfügung gestellt

```
\begin{proof}[abweichende Überschrift] ... \end{proof}
```

Dies erzeugt eine abgesetzte Umgebung, die

- mit *Proof*. eingeleitet wird (bzw. mit *abweichende Überschrift*, falls angegeben),
- und mit \square endet.

Beweisüberschrift und -endezeichen lassen sich einfach ändern:

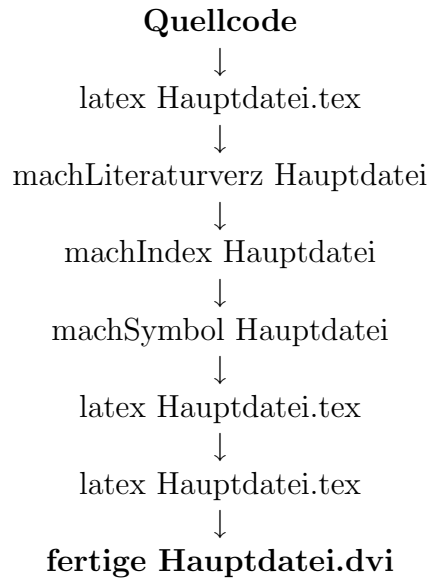
```
\renewcommand{\proofname}{Beweisüberschrift}
\renewcommand{\qedsymbol}{hübsches Symbol}
```

7 Allgemeines zu Verzeichnissen

Die automatisierte Erstellung und Verwaltung von Verzeichnissen jeder Art besteht aus bestimmten *Bausteinen*. Welche Bausteine benötigt werden, variiert von Verzeichnis zu Verzeichnis, aber ein grobes Gerüst läßt sich wie folgt beschreiben.

- Einbinden gewisser *Pakete*
Sie enthalten die Befehle, die den gesamten Ablauf intern steuern.
(Darum muß man sich in der Regel nicht kümmern.)
- Spezieller *Befehl* fordert \LaTeX auf, Verzeichnis anzulegen.
Dabei wird eine Steuerungsdatei erstellt, die später von einem anderen Programm eingelesen wird.
- Im Text: *Anweisungen*, was \LaTeX in die Steuerungsdatei schreiben soll.
`\[machEintrag][optionen]{eintrag}`
- Der Aufruf `latex Hauptdatei.tex` führt zum Anlegen der Steuerungsdatei.
- Meistens: weiteres Programm aufrufen. `machVerzeichnis Hauptdatei` erzeugt neue Datei, die für \LaTeX lesbar ist.
- Erneuter Aufruf `latex datei.tex` führt an der Stelle `printVerzeichnis` zum Schreiben des Verzeichnisses.

Vom Quellcode zum Dokument



Arbeitserleichterung

Vergleiche hierzu auch BspPaket, hier ist eine solche Datei geschrieben.

- In Datei [tudues] schreiben: Befehle in einzelnen Zeilen, am Schluß return!
- [tudues] zur Batch-Datei machen: mit `chmod 700 [tudues]` auf “ausführen“ setzen

8 Das Inhaltsverzeichnis

benötigtes Package: —

Aufruf in der Hauptdatei: `\tableofcontents`

(an der Stelle, an der das Inhaltsverzeichnis erscheinen soll)

Durch `latex Hauptdatei.tex` werden folgende Schritte bewirkt:

1. Falls `Hauptdatei.toc` existiert, wird das Inhaltsverzeichnis angelegt.
2. `Hauptdatei.toc` wird (neu) geschrieben. Dabei werden alle Gliederungsbefehle ohne `*` aufgenommen.

Deshalb wichtig:

Änderungen am Inhaltsverzeichnis werden erst nach dem *zweiten Aufruf* von `latex Hauptdatei.tex` wirksam.

Wo muß man nachhelfen?

- Gliederungsbefehle mit *, die trotzdem ins Inhaltsverzeichnis sollen, müssen manuell eingetragen werden.
- Ebenso: Literatur-, Symbolverzeichnis, Index.

allgemeine Syntax: (die Position dieses Befehls im Text der Hauptdatei bestimmt die Seitenzahl, die ins Inhaltsverzeichnis eingetragen wird)

```
\addcontentsline{Verzeichnis}{Gliederungsstil}{Überschrift}
```

Dabei ist *Verzeichnis* das Verzeichnis, in das man eintragen will (hier toc).

9 Der Index

Vorüberlegung: Wieviel Index brauche ich?

- Schachtelungstiefe?
so Gruppe oder so Faktorgruppe
, Faktorgruppe Gruppe
-ntafel Gruppentafel
- Nur die Seite mit der Definition oder jedes Auftreten oder etwas dazwischen?
- Wie wichtig muß ein Begriff sein, um im Index zu stehen?
- Querverweise?
Stellen, *siehe* Bewertungen
- Sortieren nach Substantiven?
so Gruppe, endliche oder so endliche Gruppe

Arbeiten in der Hauptdatei

- Pakete

```
\usepackage{makeidx}  
\usepackage{showidx}
```

Dabei ist `showidx` ein "Kontrollpaket".

Einbinden → Am äußeren Seitenrand werden die Einträge im Index an der Stelle aufgeführt, an der sie im Quelltext definiert werden

→ erleichtert Korrekturen

- Aufforderung an L^AT_EX, Index anzulegen und zu schreiben

```
\makeindex  
:
```

```

\begin{document}
  :
\printindex
  :
\end{document}

```

Syntax im Quelltext

Im Quelltext wird ein Eintrag gemacht, der im Dokument auf Seite p steht.

Quelltext (irgendwo)

```

\index{Indexeintrag}
\index{Eintrag|see{anderswo}}
\index{Eintrag1!Eintrag2}

\index{Eintrag1} und
\index{Eintrag1!Eintrag2}

```

Dokument (Index)

```

Indexeintrag, p
Eintrag, siehe anderswo
Eintrag1
  Eintrag2, p
Eintrag1, p1
Eintrag2, p2

```

✚ sehr viele andere Optionen

Warnung: Sehr exakt arbeiten!

Taucht `Eintrag1` mehrfach auf und variieren dabei z.B. die Leerzeichen oder Zeilenumbrüche, so führt das zu verschiedenen Einträgen!

Einbinden ins Dokument

Unser Gerüst heißt hier `Haupt.tex`.

Aufrufe

```

latex Haupt.tex

makeindex Haupt.idx
makeindex -s myindex.ist Haupt.idx

latex Haupt.tex

```

erzeugte Dateien

```

Haupt.dvi (alter Index)
Haupt.idx (Steuerdatei)
Haupt.ind (Indexdatei)
Haupt.ilg (Kommentar)
Haupt.dvi (neuer Index)

```

Die *Kommentardatei* enthält die Fehlermeldungen von `makeindex`.

Datei `myindex.ist` enthält Umdefinierungen von Befehlen für `makindex`.

10 Das Symbolverzeichnis

Faustregel: Ein Symbol, das über den Abschnitt, in dem es definiert wird, hinaus verwendet wird, sollte im Symbolverzeichnis stehen.

Arbeiten in der Hauptdatei

- Paket: `\usepackage[german]{nomencl}`
- in der Hauptdatei:

```
\makenomenclature
:
\begin{document}
:
\printnomenclature[Abstand]
:
\end{document}
```

Syntax im Quelltext

Quelltext

```
\nomenclature[spezi]{Symbol}{Bedeutung \refpage}
```

Symbolverzeichnis

```
Symbol Bedeutung, Seite p
```

Einbinden ins Dokument

Aufrufe

```
latex Haupt.tex
makeindex -s nomencl.ist -o Haupt.glo
latex Haupt.tex
```

erzeugte Dateien

Haupt.dvi (<i>altes Symbolverzeichnis</i>)	<i>Die Angabe “-s nomencl.ist“ ist nicht optional!</i>
Haupt.gls (<i>Steuerdatei</i>)	
Haupt.glo (<i>Glossardatei</i>)	<i>Hier wird das nomencl-Paket eingebunden.</i>
Haupt.ilg (<i>Kommentar</i>)	
Haupt.dvi (<i>neues Symbolverzeichnis</i>)	<i>Zwingend ist auch die Ausgabumleitung!</i>

11 Das Literaturverzeichnis

[Hart77] Robin Hartshorne, *Algebraic geometry*, Springer-Verlag, New York, 1977, Graduate Texts in Mathematics, No. 52

Wie läßt sich das mit L^AT_EX realisieren?

Anlegen der Datenbank LitList.bib

Wichtig ist die Endung .bib.

Einträge sind von der Form:

```
@book {kuerzel,
  AUTHOR   = {Hartshorne, Robin},
  TITLE    = {Algebraic geometry},
  NOTE     = {Graduate Texts in Mathematics, No. 52},
  PUBLISHER = {Springer-Verlag},
  ADDRESS  = {New York},
  YEAR     = {1977},
}
```

Korrekte Einträge über MathSciNet,

(Fachrichtungs-Seite → Bibliothek; Literaturrecherche → Uni Bielefeld oder MPI Bonn)

- Buch suchen,
- Darstellung BibTeX wählen,
- Eintrag mit copy & paste in die eigene Datei kopieren.

Arbeiten in der Hauptdatei

- Paket: nicht notwendig
- Aufforderung an L^AT_EX

```
\begin{document}
  :
\bibliography{litlist}
\bibliographystyle{amsalpha}
  :
\end{document}
```

Syntax im Quelltext

Quelltext

```
\cite[Satz 2.3.1]{hart}
```

Eintrag in litlist.bib

```
@book {hart,  
  AUTHOR   = {Hartshorne, Robin},  
  TITLE    = {Algebraic geometry},  
  ...
```

im Dokument

[Har77, Satz 2.3.1]

im Literaturverzeichnis

[Har77] Robin Hartshorne, *Algebraic geometry*, Springer-Verlag, New York, 1977, Graduate Texts in Mathematics, No. 52.

Einbinden ins Dokument

Aufrufe	erzeugte Dateien	Wirkung
latex Haupt.tex	Haupt.dvi (<i>altes Lit., alte Ref.</i>), Haupt.aux (<i>Steuerdatei</i>)	<code>\cite[xx]{kuerzel}</code> erzeugt Verweise in Haupt.aux.
bibtex Haupt	Haupt.bbl (<i>Lit.-angaben</i>), Haupt.blg (<i>Kommentar</i>)	bibtex schaut in litlist.bib die einzelnen Verweise nach. Aus <code>\bibliographystyle{amsalpha}</code> wird Stildatei amsalpha.bst gewählt.
latex Haupt.tex	Haupt.dvi (<i>neues Lit., alte Ref.</i>)	Haupt.bbl wird eingelesen.
latex Haupt.tex	Haupt.dvi (<i>neues Lit., neue Ref.</i>)	Auflösen aller Referenzen

12 Weiterführende Literatur

- Kopka, Helmut *L^AT_EX*, Band 1 bis 3, Addison Wesley 2002 (ca. 40 Euro pro Buch)
- Goosens u.a., *Der L^AT_EX Begleiter*, Addison Wesley 2002 (ca. 40 Euro)
- Goosens u.a., *The L^AT_EX Graphics Companion* Addison Wesley 1997 (ca. 55 Euro)

Aus dem Netz

- LaTeX Primer (mit einer Einführung in die ams-Pakete)
- The Not So Short Introduction to L^AT_EX
- Das L^AT_EX-Kochbuch (nur online)
- Dante-Server (wichtigste FAQ-Liste, Ergänzungspakete und Fonts)