Kompaktkurs zu LaTeX 2ε

Dirk Nitschke Wolfgang Ripken

2. März 1998

Inhaltsverzeichnis

1 Das Konzept von LaTeX **1**
1.1 Wer bin ich? 1
1.2 Wozu LaTeX? oder: Textsatz damals und heute 1
1.3 Logisches vs. visuelles Design 2

2 Eingabe von Texten **5**
2.1 Eingabedatei und Ausgabedateien 5
2.2 Aufbau einer Eingabedatei 5
2.3 Zulässige Zeichen 7
2.4 Befehle und Umgebungen 8
2.5 Sätze und Absätze 9
2.6 Eingabe besonderer und unzulässiger Zeichen 9
2.6.1 Besondere Zeichen 9
2.6.2 Akzente und andere Spezialitäten 10
2.6.3 Anführungszeichen und Bindestriche 11
2.6.4 Auslassungspunkte 12
2.6.5 Noch mehr Leerzeichen 12
2.6.6 Kommentare 12
2.7 Das `german`-Paket 12

3 Eigene Befehle und Umgebungen **15**
3.1 Definieren von Befehlen 15
3.2 Definieren von Umgebungen 16

4 Strukturen **17**
4.1 Titelinformationen 17
4.2 Der Abstract 18
4.3 Gliederungseinheiten 19
4.4 Das Inhaltsverzeichnis 20

5 Querverweise **21**

6 Literaturangaben **21**
6.1 Literaturverzeichnis 22
6.2 Referenzen auf Literaturangaben 23

7 Listen **23**

8 Theoreme und ähnliches **25**

9 Fußnoten **27**

10 Mathematik **29**
10.1 Eingebettete Formeln 29
10.2 Mathematische Symbole 30
10.3 Hoch- und Tiefstellungen 36
10.4 Lange Funktionen- und Operatornamen 37
10.5 Variieren der Schriftart von Buchstaben 38
10.6 Abgesetzte Formeln 38
10.7 Text in Formeln 39
10.8 Klassen mathematischer Symbole 39

11 Noch mehr Mathematik **43**
11.1 Brüche 43
11.2 Klammersymbole in verschiedenen Größen 43
11.3 Mathematische Akzente 44
11.4 Formelsysteme 46
11.4.1 Formelsysteme ohne Ausrichtung 46
11.4.2 Formelsysteme mit Ausrichtung an einer Position . . . 46
11.4.3 Formelsysteme mit Ausrichtung an mehreren Positionen 47
11.5 Formelnummern 47
11.6 Referenzieren von Formeln 48
11.7 Brechen abgesetzter Formeln 49
11.8 Satzzeichen in abgesetzten Formeln 49
11.9 Matrizen 49
11.10 Fallunterscheidungen 51
11.11 Definition weiterer Funktionen- und Operatornamen 51

12 Abbildungen und Tabellen **53**
12.1 Einbinden von Bildern 53
12.2 Zentrierungen 54
12.3 Tabellen 54
12.3.1 Text über mehrere Spalten 55
12.4 Gleitobjekte 56
12.5 Abbildungs- und Tabellenverzeichnis 57

Literatur **59**

Tabellenverzeichnis

1 Dokumentklassen 6
2 Pakete 6
3 Akzente 10
4 Sprachspezifische Zeichen 10
5 Einige Befehle des `german`-Pakets 13
6 Große griechische Buchstaben 31
7 Kleine griechische Buchstaben 31
8 Mathematische Symbole 32
9 Lange Funktionen- und Operatornamen 37
10 Klammersymbole 45
11 Mathematische Akzente 45

1 Das Konzept von LaTeX

1.1 Wer bin ich?

LaTeX (sprich: lah-tech) ist der Name eines Systems zum Setzen von Dokumenten. Es handelt sich dabei um eine von Leslie Lamport initiierte Erweiterung des Satzsystems TeX (sprich: tech), welches von Donald E. Knuth entwickelt wurde. Die erste weit verbreitete Version von LaTeX trug die Nummer 2.09. In der folgenden Zeit wurden viele Verbesserungen an dieser Version vorgenommen – leider verlief dies unkoordiniert, so daß es zu einem Durcheinander von Versionen kam. Abhilfe schafft die aktuelle und offizielle Version LaTeX 2_ε. In diese Version wurden unter anderem viele Verbesserungen zur Einbindung von Grafiken, farbigen Textteilen und zur Behandlung von unterschiedlichen Schrifttypen eingearbeitet. Wenn wir im folgenden über LaTeX sprechen, so meinen wir – solange nicht anderes erwähnt wird – stets LaTeX 2_ε.

1.2 Wozu LaTeX? oder: Textsatz damals und heute

Bevor Computer Einzug in unser Leben hielten, reichte ein Autor ein hand- oder maschinengeschriebenes Manuskript bei einem Verlag ein. *Dort* wurde über die *typographische Gestaltung* des Textes entschieden: Zeilenlänge, Abstände zwischen Absätzen, die zu verwendenden Schrifttypen und vieles mehr mußten festgelegt werden, um ein ansprechendes und ausgeglichenes Äußeres zu erzielen. Hieraus ergaben sich Anweisungen für den Setzer, mit denen er entschied, an welcher Stelle der Seite die Zeichen und Worte des Autors erschienen.

Heute werden Manuskripte mit dem Computer erstellt und dann an den Verlag geschickt. Aus Kostengründen vervielfältigt der Verlag häufig den Text nur noch, bindet und verschickt ihn. Die typographische Gestaltung, bei der es sich um ein Handwerk handelt, welches *erlernt werden muß,* liegt somit beim Autor. Da von uns nur wenige die Möglichkeit haben werden, dieses Handwerk zu erlernen, liegt es nahe, die Aufgabe dem Computer zu übertragen und sich selbst ganz auf den *Inhalt* und die *Struktur* des Textes zu konzentrieren. Die Verwendung von LaTeX bewerkstelligt genau diese Arbeitsteilung: LaTeX übernimmt die *typographische Gestaltung*, TeX erledigt den *Textsatz* und Sie sind für die *Struktur* und den *Inhalt* des Textes zuständig.

Menschliche Gestalter wissen, wovon ein Manuskript handelt. Sie *erkennen* an den Formulierungen, ob es sich beim gerade zu bearbeitenden Textteil um eine Überschrift, ein Zitat, eine mathematische Formel oder eine Auflistung handelt. Diese Informationen fließen in den Textsatz mit ein. Ein

Computerprogramm kann Ihren Text nicht verstehen und ohne Hilfe Ihrerseits nicht entscheiden, welche logische Bedeutung ein Textteil hat. Somit müssen Sie LaTeX die *logische Struktur* des Textes durch spezielle *Befehle* mitteilen. Diese Methode des Textsatzes nennt man *logisches Design*.

„Das kann ... doch auch alles – und das ist einfacher zu bedienen!“ werden Sie sagen. Dabei handelt es sich bei ... wahrscheinlich um ein sogenanntes Wysiwyg-Programm, welches Ihnen „what you see is what you get“ („man bekommt, was man sieht“ – für mich eher „man bekommt, was man verdient“) verspricht. Bei solchen Programmen haben Sie meist die Möglichkeit, Ihren Text in beliebiger Größe und Schriftart an einer beliebigen Stelle auf der Seite zu plazieren und hin und her zu schieben. Dies hat aber nichts mit der logischen Struktur des Textes zu tun, sondern mit dem eigentlichen Textsatz. Sie teilen dem Programm *nicht* mit, wie der Text aufgebaut ist, sondern wie er *aussehen* soll. Dies bezeichnet man auch als *visuelles Design*. Die Verwendung von logischem statt visuellem Design ist es, was LaTeX von Wysiwyg-Programmen unterscheidet.

1.3 Logisches vs. visuelles Design

Visuelles Design und damit Wysiwyg-Programme eignen sich durch ihre leichte Handhabbarkeit gut zum Schreiben von *kurzen* Texten, da man mit ihnen *mal schnell* etwas eingeben kann. Bei umfangreicheren Texten und insbesondere wissenschaftliche Arbeiten kann *mal schnell* leicht in erheblich mehr Arbeit ausarten. Dazu ein Beispiel: Nehmen wir an, in Ihrem Text kommt der (mathematische) Ausdruck (A, B) vor. Bei einem Wysiwyg-Programm geben Sie einfach `(A,B)` ein. In LaTeX könnten Sie das auch so eingeben, jedoch würden wir dann *visuelles* Design statt *logischem* Design betreiben. Wir müssen daher LaTeX mitteilen, daß es sich um einen mathematischen Textteil handelt. Dies wäre zum Beispiel durch die Eingabe von `$(A,B)$` zu bewerkstelligen. Wir haben die logische Struktur aber noch nicht vollständig freigelegt: Dieser Ausdruck hat auch eine besondere mathematische Bedeutung und bezeichnet häufig ein *Skalarprodukt*. Fortgeschrittene LaTeX-Benutzer werden daher einen Befehl definieren, der dies wiedergibt. Ein solcher Befehl könnte `\skp` heißen und die Eingabe wäre dann `$\skp{A}{B}$`. Taucht jetzt in Ihrem Text ein *weiteres* Skalarprodukt (C, D) auf, so können Sie den Befehl `\skp` erneut verwenden und `$\skp{C}{D}$` eingeben.

Wozu das alles? Warum soll ich nicht einfach `(A,B)` und `(C,D)` eingeben? Diese Frage zielt auf den Sinn logischen Designs. Nehmen wir das Beispiel von eben. Sollten Sie sich – nachdem Sie bereits einige Seiten Ihres Textes eingegeben haben – dazu entschließen, ein Skalarprodukt *anders* zu bezeichnen, etwa $\langle A, B\rangle$ statt (A, B), so brauchen Sie bei *logischem* Design nur an *einer* Stelle Ihrer Eingaben eine Änderung vornehmen, nämlich in der Definition des Befehls `\skp`. Bei *visuellem* Design müssen Sie bei jedem Auftreten von `(A,B)`, `(C,D)` und *allen weiteren* irgendwo eingegebenen Skalarprodukten die entsprechenden Änderungen von Hand vornehmen und jedesmal entscheiden, ob eine Korrektur nötig ist.

Ein weiterer Vorteil liegt darin, daß sich viele arbeitsintensive und fehlerträchtige Aufgaben durch logisches Design vereinfachen – d. h. dem Computer übertragen – lassen. Dazu zählt zum Beispiel die automatische Numerierung und Referenzierung von Textabschnitten, Fußnoten, mathematischen Formeln, Tabellen und Literaturangaben sowie das Erstellen von Inhalts- und Schlagwortverzeichnissen.

Das ganze hat aber auch Nachteile: Eine umfangreichere Änderung an den LaTeX bekannten Formatierungsregeln ist nicht ganz einfach und sollte erfahrenen Benutzern vorbehalten bleiben.

2 Eingabe von Texten

2.1 Eingabedatei und Ausgabedateien

LaTeX entnimmt den zu setzenden Text und alle speziellen Befehle zur logischen Struktur des Textes einer *Eingabedatei,* die man mit einem Editor seiner Wahl erstellen kann. Im allgemeinen besteht der *Dateiname* einer solchen Eingabedatei aus zwei Teilen, die durch einen Punkt voneinander getrennt werden. Der vordere Teil wird als *Grundname* und der hintere als *Erweiterung* bezeichnet. Der Grundname Ihrer Eingabedateien kann beliebig sein (bis auf die Beschränkungen, die Ihr Computersystem Ihnen vorgibt: Unter DOS darf der Grundname nur 8 Zeichen lang sein, auf dem Macintosh jedoch 31 Zeichen lang). Die Erweiterung sollte `tex` lauten (auf Systemen, die Groß- und Kleinschreibung in Dateinamen unterscheiden, soll `tex` klein geschrieben werden).

LaTeX erzeugt mehrere Ausgabedateien, deren Grundname so lautet wie der der Eingabedatei, die Sie von LaTeX bearbeiten lassen. Die wichtigste dieser Dateien ist diejenige mit der Erweiterung `dvi`. Diese Datei benötigen Sie zum Betrachten und Ausdrucken des gesetzten Textes.

2.2 Aufbau einer Eingabedatei

Eine Eingabedatei besteht aus zwei Teilen, der *Präambel* und dem *Textteil.*

In der Präambel teilt man LaTeX als erstes mit, welche *Dokumentenklasse* – d. h. welches Layout – für das folgende Dokument verwendet werden soll. Dies geschieht mit dem Befehl

`\documentclass[`⟨*Optionen*⟩`]{`⟨*Klassenname*⟩`}`.

Einige der standardmäßig zu LaTeX 2_ε gehörenden Klassen finden Sie mit einer kurzen Beschreibung in Tabelle 1.

Erweiterungen von LaTeX – wie zum Beispiel spezielle Unterstützung für deutschsprachige Texte oder zusätzliche Strukturen für mathematischen Formelsatz – werden in *Paketen* zusammengefaßt. Mit dem Befehl

`\usepackage[`⟨*Optionen*⟩`]{`⟨*Paketnamen*⟩`}`

kann man in der Präambel die Vereinbarungen aus solchen Paketen verwenden. Eine kleine Auswahl an Paketen zeigt Tabelle 2. Außerdem können in der Präambel kleinere Änderungen an den Layoutvorgaben oder eigene Befehle vereinbart werden.

Der Textteil eines Dokumentes beginnt *immer* mit dem Befehl

Tabelle 1: Dokumentklassen

article Wird für Artikel in wissenschaftlichen Zeitschriften, Vorträge, Praktikumsarbeiten, Seminararbeiten, kürzere Berichte, Anträge, Gutachten, Programmbeschreibungen, Einladungen u. s. w. verwendet.

report Für längere Berichte, die aus mehreren Kapiteln bestehen, Diplomarbeiten, Dissertationen, Skripte u. s. w.

book Wird für Bücher verwendet.

slides Speziell zur Erstellung von Folien.

letter Eine Dokumentenklasse für Briefe nach amerikanischem Muster.

proc Geeignet für *Proceedings* (Konferenzbände).

Tabelle 2: Pakete

german Spezielle Unterstützung für das Schreiben von deutschsprachigen Texten. So wird unter anderem die Eingabe von Umlauten vereinfacht.

amsmath Ergänzungen für den Satz mathematischer Texte.

amsthm Befehle zur Definition „theoremartiger“ Strukturen.

```
\begin{document}
```

und endet mit dem Befehl

```
\end{document}.
```

Zwischen diesen Befehlen steht der Text. Dieser wird von TEX automatisch in Zeilen gebrochen, wobei Worte gegebenenfalls getrennt werden.

Alles, was Sie hinter `\end{document}` eingeben, wird von LATEX ignoriert.

2.3 Zulässige Zeichen

Eine Eingabedatei darf leider nicht alle Zeichen enthalten, die Sie auf Ihrer Tastatur finden. Zulässige Zeichen sind:

Groß- und Kleinbuchstaben: `A B` ... `Z a b` ... `z`
Ziffern: `0 1` ... `9`
Satzzeichen: `. : ; ? ! ‘ ’ ( ) [ ] - / * @`
Mathematische Zeichen: `+ = | < >`
Sonderzeichen: `# % $ & _ { } ~ ^ \ "`
Unsichtbare Zeichen: ⟨*Leertaste*⟩ ⟨*Tabulatortaste*⟩ ⟨*Zeilenendetaste*⟩

Hieraus ergibt sich gleich eine (für deutschsprachige Benutzer unerfreuliche) Einschränkung:

Die deutschen Umlaute sind *nicht* zugelassen!

Welche Bedeutung haben diese Zeichen in einer Eingabedatei?

Groß- und Kleinbuchstaben, Ziffern und Satzzeichen interpretiert LATEX genau so, wie man sie eingibt. Die Eingabe

```
Heute regnet es.
```

erzeugt somit die Ausgabe

Heute regnet es.

Die fünf Zeichen

```
+ = | < >
```

sind für mathematische Formeln gedacht. `+` und `=` können aber auch im normalen Text verwendet werden (Auf den Unterschied zwischen „normalem“ und mathematischem Text wird in Abschnitt 10 eingegangen).

Die zehn Sonderzeichen

`# % $ & _ { } ~ ^ \`

haben für LaTeX eine besondere Bedeutung. Wenn man eines dieser Zeichen im laufenden Text erzeugen möchte – also zum Beispiel „Dieses Skript kostet 95 $“ schreiben will –, so muß man dies LaTeX durch *Befehle* mitteilen. Das elfte Sonderzeichen `"` hat eine besondere Bedeutung, wenn das `german`-Paket benutzt wird.

Darüber hinaus sind noch *unsichtbare* Zeichen erlaubt: Leerzeichen (durch Drücken der Leertaste), Tabulatoren (Tabulatortaste) und die Zeichen für das Zeilenende (Return oder Enter). All diese unsichtbaren Zeichen werden von TeX gleich behandelt. Mehrere Leerzeichen bewirken dasselbe wie *ein* Leerzeichen. Eine Leerzeile wird von TeX als das Ende eines Absatzes verstanden. Mehrere Leerzeilen wirken wie eine Leerzeile. Das bedeutet, daß es *nicht* möglich ist, durch mehrere Leerzeichen bzw. Leerzeilen den Abstand zwischen zwei Worten bzw. Absätzen zu vergrößern (das hatten Sie doch wohl nicht etwa vor?).

2.4 Befehle und Umgebungen

Es war eben schon von *Befehlen* die Rede. Sie bestehen entweder aus einem einzelnen Zeichen wie `~`, einem *Backslash* `\` gefolgt von genau *einem* nichtalphabetischen Zeichen (also keinem Groß- oder Kleinbuchstaben) wie zum Beispiel `\"` oder einem `\` gefolgt von einer Folge von Groß- und Kleinbuchstaben wie `\LaTeX`. Leerzeichen und ein Zeilenende nach den letztgenannten Befehlen werden ignoriert. Um nach einem solchen Befehl einen Wortzwischenraum zu erzeugen, muß man einen weiteren Befehl eingeben: `\␣`, also einen Backslash gefolgt von einem Leerzeichen. Einige Befehle haben eine *Sternform,* die man erhält, indem man direkt nach dem Befehl das Zeichen `*` eingibt.

Viele Befehle haben eines oder mehrere *Argumente,* die man zwingend angeben muß und in geschweifte Klammern `{` und `}` einfaßt. Es gibt auch Befehle, die *optionale Argumente* haben. Solche *können* angegeben werden, müssen es aber nicht. Falls sie angegeben werden, so werden sie von `[` und `]` begrenzt.

Bei *Umgebungen* handelt es sich um ein Paar

`\begin{`⟨*Name*⟩`}` … `\end{`⟨*Name*⟩`}`

von Befehlen. Eine solche Umgebung haben wir bereits kennengelernt: die `document`-Umgebung, die den Textteil einschließt.

2.5 Sätze und Absätze

TEX ignoriert die Art und Weise, wie die Eingabedatei formatiert ist und achtet nur auf die logischen Konzepte Wortende, Satzende und Absatzende. Dadurch gestaltet sich die Eingabe von einfachen Sätzen und Absätzen recht einfach, da man den Text wie auf der Schreibmaschine schreibt. Der Zeilenumbruch und die Trennung von Worten wird allerdings von TEX übernommen.

```
Mit  Leerzeichen kennzeichnet  man
das Ende eines Wortes oder Satzes.
Dabei ist es unerheblich, wieviele
Leerzeichen           man eingibt.
1 ist so gut wie              15.


Eine oder mehrere Leerzeilen zeigen
das Ende eines Absatzes an.
```

Mit Leerzeichen kennzeichnet man das Ende eines Wortes oder Satzes. Dabei ist es unerheblich, wieviele Leerzeichen man eingibt. 1 ist so gut wie 15.

Eine oder mehrere Leerzeilen zeigen das Ende eines Absatzes an.

2.6 Eingabe besonderer und unzulässiger Zeichen

2.6.1 Besondere Zeichen

Eben wurde erwähnt, daß die Zeichen

```
# % $ & _ { } ~ ^ \ "
```

für LATEX eine besondere Bedeutung haben und nicht zur Erzeugung der entsprechenden Zeichen benutzt werden können. Da zumindest die ersten sieben dieser Zeichen ab und zu in gewöhnlichen Texten auftauchen, kann man sie durch Voranstellen des `\` erzeugen:

```
\# \% \$ \& \_ \{ \}
```

% $ & _ { }

(Wenn Sie die übrigen vier Zeichen `~`, `^`, `\`, `"` wirklich erzeugen wollen, *fragen Sie jemanden, der sich damit auskennt.*)

Tabelle 3: Akzente

Eingabe	Ausgabe	Eingabe	Ausgabe
`` \`{o} ``	ò	`\u{o}`	ŏ
`\'{o}`	ó	`\v{o}`	ǒ
`\^{o}`	ô	`\H{o}`	ő
`\"{o}`	ö	`\t{oo}`	o͡o
`\~{o}`	õ	`\c{o}`	o̧
`\={o}`	ō	`\d{o}`	ọ
`\.{o}`	ȯ	`\b{o}`	o̱
`\r{o}`	o̊		

Tabelle 4: Sprachspezifische Zeichen

Eingabe	Ausgabe	Eingabe	Ausgabe
`\oe`	œ	`\OE`	Œ
`\o`	ø	`\O`	Ø
`\ae`	æ	`\AE`	Æ
`\l`	ł	`\L`	Ł
`\aa`	å	`\AA`	Å
`` ?` ``	¿	`` !` ``	¡
`\ss`	ß		

2.6.2 Akzente und andere Spezialitäten

Die deutschen Umlaute sind in der Eingabedatei nicht zulässig. Dasselbe gilt für akzentuierte Zeichen anderer Sprachen. Diese Zeichen lassen sich durch spezielle Befehle ausgeben, die in Tabelle 3 angegeben sind. Die Tabelle zeigt, wie man die entsprechenden Akzente an dem Zeichen `o` anbringt. Sie können `o` aber durch jeden anderen Buchstaben ersetzen. Bei `i` und `j` muß man allerdings beachten, daß der Punkt verschwindet, wenn sie einen Akzent erhalten sollen. Auch hierfür ist TEX gerüstet und erzeugt durch die Befehle `\i` und `\j` die punktlosen Varianten ı und ȷ.

`\'{E}l est\'{a} aqu\'{\i}.` Él está aquí.

Neben Akzenten gibt es noch Zeichen, die sich nur in bestimmten Sprachen finden. Tabelle 4 zeigt, mit welchem Befehl man welches Zeichen erzeugen kann. Leider muß man auch hier etwas beachten:

All diese Befehle können nur im Textmodus verwendet werden!

Erfreulicher verhalten sich die folgenden Befehle, die man sowohl in gewöhnlichem Text als auch im Mathematikmodus benutzen kann:

`\dag`	†	`\S`	§	`\copyright`	©
`\ddag`	‡	`\P`	¶	`\pounds`	£

2.6.3 Anführungszeichen und Bindestriche

Weitere besondere Zeichen sind Anführungszeichen und Bindestriche. Da TeX ein Programm zum *Textsatz* ist und nicht *Schreibmaschinensatz,* muß man bei diesen Zeichen etwas aufpassen. Insbesondere gibt es Unterschiede zwischen deutsch- und englischsprachigen Texten.

Im Maschinensatz verwendet man für Anführungszeichen entweder das doppelte `"` oder das einfache Anführungszeichen `'`, welches auch als Apostroph Verwendung findet. Im deutschen Schriftsatz werden vornehmlich die Zeichen „ und “ benutzt.

Der Buchsatz kennt auch drei verschiedene Bindestriche. Ihre Verwendung ist ebenfalls sprachabhängig. All diese Striche lassen sich in TeX durch die Eingabe von ein, zwei oder drei aufeinanderfolgenden `-` erzeugen.

```
Ein Bindestrich innerhalb eines
Wortes wie in O-Beine. Ein
Halbgeviertstrich -- auch
Streckenstrich genannt, wird bei
Bereichsangaben wie 4--5 Stunden
oder -- wie dieser -- als
Gedankenstrich zur Abgrenzung von
Teils"atzen benutzt. Der Geviertstrich
--- wird in deutschen Texten im
allgemeinen nicht verwendet, kann
aber in Tabellen bei fehlenden
Angaben sinnvoll sein. Im Englischen
wird er als Gedankenstrich verwendet
und ohne umgebende Leerzeichen
eingegeben.
```

Ein Bindestrich innerhalb eines Wortes wie in O-Beine. Ein Halbgeviertstrich – auch Streckenstrich genannt, wird bei Bereichsangaben wie 4–5 Stunden oder – wie dieser – als Gedankenstrich zur Abgrenzung von Teilsätzen benutzt. Der Geviertstrich — wird in deutschen Texten im allgemeinen nicht verwendet, kann aber in Tabellen bei fehlenden Angaben sinnvoll sein. Im Englischen wird er als Gedankenstrich verwendet und ohne umgebende Leerzeichen eingegeben.

Das Minuszeichen $-$ gehört *nicht* zu diesen Strichen! Es kann nur im mathematischen Modus verwendet werden.

2.6.4 Auslassungspunkte

Auslassungspunkte werden im Textsatz besonders gesetzt. Es gibt daher den Befehl `\dots`, der die passenden Punkte erzeugt.

```
Nicht Orange, Zitrone, ...,
sondern Orange, Zitrone, \dots
```

Nicht Orange, Zitrone, ..., sondern Orange, Zitrone, …

2.6.5 Noch mehr Leerzeichen

Wenn man verhindern möchte, daß LaTeX zwischen zwei Worten einen Zeilenumbruch erzeugt, so kann man dies durch den Befehl `~`, den man an Stelle eines Leerzeichens eingibt, verhindern. Ein typisches Beispiel ist

```
\dots\ finden Sie auf S.~239.
```

… finden Sie auf S. 239.

2.6.6 Kommentare

Falls TeX in Ihrer Eingabedatei auf ein `%` stößt, so wird dieses Zeichen, alle nachfolgenden der entsprechenden Zeile und alle Leerzeichen am Anfang der folgenden Zeile ignoriert. Dadurch kann man Notizen in die Eingabedatei einfügen, die gar nicht oder noch nicht ausgedruckt werden sollen. Außerdem kann man so ein Zeile beenden ohne ein Leerzeichen zu erzeugen.

```
Dies ist ein% lehrreiches
   Beispiel. Man beachte
den % nicht gewollten
fehlenden Abstand zwischen
% Das sieht niemand!
"`ein"' und "`Beispiel"'!
```

Dies ist einBeispiel. Man beachte den fehlenden Abstand zwischen „ein“ und „Beispiel“!

2.7 Das german-Paket

Wie zuvor beschrieben, ist die Eingabe deutscher Umlaute ein wenig umständlich. Hier kann Ihnen das `german`-Paket helfen (siehe Tabelle 5).

Tabelle 5: Einige Befehle des `german`-Pakets

Eingabe	Ausgabe	Eingabe	Ausgabe
"`	„	"'	“
"a	ä	"A	Ä
"e	ë	"E	Ë
"i	ï	"I	Ï
"o	ö	"O	Ö
"u	ü	"U	Ü
"s	ß		