

Die alg-script-Klasse

Paul Ebermann*

23. März 2006

Diese Dokumentation gehört zu alg-script v0.13 vom 2006/03/23.

Dank an die Regulars in `de.comp.text.tex`, die bei meinen Problemen immer geduldig weitergeholfen haben.

*Paul-Ebermann@gmx.de

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ziel	4
1.2	Urheberrecht und Benutzungsbedingungen	4
2	Benutzer-Dokumentation	6
2.1	Anwendung	6
2.2	Optionen	6
2.3	Umgebungen	6
2.3.1	Listen	7
2.3.2	Sätze	7
2.3.3	Unnummerierte Sätze	8
2.3.4	Definitionen	9
2.3.5	Beispiele, Fragen, etc.	9
2.3.6	Beweise	10
2.4	Befehle	11
2.4.1	Diverse Symbole für den mathematischen Modus. . .	11
2.4.2	Definitionen von mathematischen Symbolen und Begriffen	12
2.4.3	Abkürzungen bzw. Textbausteine	13
3	Der Code für den Dokumentations-Treiber	14
4	Der Code der Klasse	15
4.1	Identifikation	15
4.2	Optionen	15
4.2.1	doc	15
4.2.2	ϕ statt φ : oldphi	15
4.3	Geladene Pakete	16
4.3.1	Basisklasse	16
4.3.2	doc	16
4.3.3	\mathcal{AMS} -Pakete	16
4.3.4	Eingabe	18
4.3.5	Deutsch	18
4.3.6	xspace	18
4.3.7	Neues <code>enumerate</code>	18
4.3.8	Index	19

Inhaltsverzeichnis

4.4	Einige geänderte Zähler	19
4.5	Neue Umgebungen	19
4.5.1	Neue Aufzählungen	19
4.5.2	Sätze im Kasten	19
4.5.3	Definitionen, Beispiele, Fragen	22
4.5.4	Beweise	24
4.6	Mathematische Symbole	24
4.6.1	Alternatives Symbol für kartesisches Produkt	24
4.6.2	Anderes <code>\phi</code>	24
4.6.3	Schönere Namen für ν und μ	25
4.6.4	Einige Pfeile	25
4.6.5	Bekannte Zahlenmengen	25
4.6.6	Weitere Mengen	26
4.6.7	Klammern & Co.	27
4.6.8	Operatoren mit Namen	28
4.6.9	Weitere Symbole	29
4.7	Hilfsmittel für Definitionen	29
4.8	Textbausteine	31
4.9	Vermischtes	31
4.10	Der Schluss	32
A	The L^AT_EX Project Public License	33
A.1	Preamble	33
A.2	Definitions	33
A.3	Conditions on Distribution and Modification	34
A.4	No Warranty	37
A.5	Maintenance of The Work	37
A.6	Whether and How to Distribute Works under This License	39
A.6.1	Choosing This License or Another License	39
A.6.2	A Recommendation on Modification Without Distribution	39
A.6.3	How to Use This License	39
A.6.4	Derived Works That Are Not Replacements	40
A.6.5	Important Recommendations	40
B	Liste der Änderungen	42
C	Index	44

1 Einleitung

1.1 Ziel

Die Klasse `alg-script` habe ich geschrieben, um damit – wie der Name andeutet – meine Algebra-Skripte zu formatieren. Bisher (am 10.2.2006) gibt es – neben dieser Dokumentation¹ sowie einigen kleineren Testdokumenten – genau zwei mit dieser Klasse gesetzte Dokumente.

Das erste ist meine Mitschrift der Vorlesung LAAG I, Kapitel 0, „Einführung in die Mengenlehre“. Ursprünglich hatte ich geplant, diese Mitschrift in Gemeinschaftsarbeit weiterzuführen, aber daraus ist nichts geworden.

Das zweite Dokument ist noch nicht fertig – es ist eine Zusammenfassung des Stoffes, den ich für meine Spezialisierungsprüfung lernen muss. Im Laufe des Abschreibens der Vorlesungsskripte habe ich entsprechend auch die Klasse mit weiteren Befehlen ergänzt.

Als ich neulich meinem Kommilitonen Tobi davon erzählte, war er ganz begeistert ... Er bat mich, das Skript doch online zu stellen (zumindest habe ich das so verstanden). Danke für die Ermutigung, Tobi.

Ich habe jetzt zumindest die Klasse (und die anderen von meinem Skript verwendeten selbstgeschriebenen Pakete) soweit zurechtgeputzt, dass ich sie vorzeigen kann, und alles online gestellt ...

Solange ich mein Diplom noch nicht habe, werde ich noch den Computer-Account am Institut für Mathematik der HU-Berlin haben, und entsprechend wird das Skript in der aktuellsten veröffentlichten Version genauso wie die Klasse und die Packages bei <http://www.math.hu-berlin.de/~ebermann/alg-script/> zu finden sein.

Nachtrag: Inzwischen habe ich noch ein drittes Dokument auf meiner Festplatte gefunden - *Grundbegriffe der Kategoriellen Algebra*, eine Mitschrift einer Mini-Vorlesung von 2003. Es war noch nicht fertig, und ich integriere es jetzt in meinen Spezialisierungs-Stoff, da passt es gut hinein.

1.2 Urheberrecht und Benutzungsbedingungen

Ich erlaube die (und ermuntere sogar zur) Verwendung der Klasse, um damit eigene \LaTeX -Dokumente zu erstellen. Natürlich gebe ich keine Garantie

¹Diese Dokumentation ist kein verlässliches Beispiel für die Klasse, da hier einige der Einstellungen etwas verbogen wurden. Beispielsweise ist der breitere Rand immer links und auch noch etwas breiter als normal, weil hier immer die Namen der Befehle erscheinen.

1 Einleitung

für das Funktionieren in irgendeiner Form. Außerdem gibt es noch eine Einschränkung:

Die Klasse `alg-script` (d.h. die Quelltext-Datei `alg-script.dtx` und das Installationsscript `alg-script.ins` sowie die daraus generierte Klassendatei `alg-script.cls` und Dokumentation – dieses Dokument –) stehen unter der LPPL (L^AT_EX Project Public License), Version 1.3b oder später.

Der „maintenance-status“ ist „author-maintained“, d.h. in etwa, nur ich (Paul Ebermann) darf Änderungen vornehmen und dann geänderte Versionen unter selbem Namen weitergeben. Wer sonst etwas ändert, sollte auch den Namen der Klasse ändern und seine Änderungen klar markieren.

Eine Kopie der LPPL (auf Englisch, die Version, die beim Erstellen der Doku gefunden wurde) ist im Anhang A, ab Seite 33 zu finden.

2 Benutzer-Dokumentation

Meine `alg-script`-Klasse lädt eine Reihe weiterer Pakete und definiert einige nützliche Befehle.

Viele Befehle sind maßgeschneidert für die Verwendung in meinem Mengenlehre-Skript – diese werden in zukünftigen Versionen vielleicht in ein extra-Paket verschoben.

Hier eine kurze Übersicht – leider (noch) nicht vollständig, der Rest ist bei der Implementation (Kapitel 4) zu finden.

2.1 Anwendung

Um die Klasse zu verwenden, schreibe folgendes an den Anfang deines Dokumentes:

```
\documentclass{alg-script}
```

Vor dem Klassennamen können (in eckigen Klammern) noch einige Optionen stehen - diese gelten dann auch für alle geladenen Packages, und werden (sofern nicht selbst verwendet) an die Basisklasse `scrbook` weitergegeben. Beispiel:

```
\documentclass[oldphi,draft]{alg-script}
```

2.2 Optionen

Diese Klasse einige Optionen.

`doc` `doc` ist eigentlich nur für die Erzeugung dieser Dokumentation geeignet. Es bewirkt, dass das Package `doc` direkt nach dem Laden der Basisklasse `scrbook`, jedenfalls aber noch vor dem Laden des Paketes `amscd` geladen wird. Dies ist leider notwendig, da sich `doc` und `amscd` nur in dieser Reihenfolge vertragen.

`oldphi` `oldphi` ist für Leute, die – anders als ich – die ursprüngliche L^AT_EX-Version von `\phi`, nämlich ϕ mehr mögen als φ . Standardverhalten meiner Klasse ist es, φ zu verwenden, `oldphi` schaltet das ab.

2.3 Umgebungen

Die Klasse definiert verschiedene neue Umgebungen. Außerdem kann man natürlich die von `amsmath`, `amsthm`, `amscd` etc. definierten Umgebungen

nutzen.

2.3.1 Listen

Zusätzlich zu der L^AT_EX-Standard-Listenumgebung `enumerate`, welche ja je nach Schachtelungstiefe anders zählt, habe ich einige weitere `enum*`-Umgebungen definiert.¹ `enumabc` nummeriert per (a), (b), (c):

- (a) Erster Eintrag.
- (b) Zweiter Eintrag.
- (c) Dritter Eintrag.

`enumnum` `enumnum` nummeriert per (1), (2), (3):

- (1) Erster Eintrag.
- (2) Zweiter Eintrag.
- (3) Dritter Eintrag.

`enumiii` `enumiii` nummeriert per (i), (ii), (iii):

- (i) Erster Eintrag.
- (ii) Zweiter Eintrag.
- (iii) Dritter Eintrag.

2.3.2 Sätze

`satz` Sätze werden in einem Kasten dargestellt, die Nummerierung erfolgt automatisch pro Abschnitt (`\section`).

```
\begin{satz}
  Dies ist ein Test-Satz ohne jeden Sinn.
\end{satz}
```

führt zu

SATZ 2.3.1. Dies ist ein Test-Satz ohne jeden Sinn.
--

2.3.3 Unnummerierte Sätze

satzon

```
\begin{satzon}
  Der vorige Satz hatte keinen Sinn.
\end{satzon}
```

für Sätze, Bemerkungen, etc., denen Herr Kleinert keine Nummer zugewiesen hatte:

SATZ OHNE NUMMER.	Der vorige Satz hatte keinen Sinn.
-------------------	------------------------------------

korollar
behauptung

Außerdem habe ich einige spezielle dieser nichtnummerierten Sätze – die Anwendung erfolgt genauso.

BEHAUPTUNG.	Das ist allgemein in diesem Abschnitt mit den Beispielsätzen so der Fall.
-------------	---

KOROLLAR.	Auch dieser Korollar ist sinnlos.
-----------	-----------------------------------

eigsatz

Weiterhin habe ich noch eine Umgebung, um Eigenschaften eines eben definierten Objektes in Satzform (aber ohne Nummer) festzustellen.

```
\begin{eigsatz}[iii]
  \item $a + 0 = a$
  \item $1 \cdot a = a$
  \item $0 \cdot a = 0$
\end{eigsatz}
```

ergibt

EIGENSCHAFTEN.

(i) $a + 0 = a$

(ii) $1 \cdot a = a$

(iii) $0 \cdot a = 0$

Mit dem optionalen Parameter – möglich sind `num`, `abc`, `iii`, voreingestellt ist `num` – kann man die Art der Liste (aus Abschnitt 2.3.1) auswählen.

satzbem

Gelegentlich werden mehrere Sätze als „Bemerkung“ formuliert und nur *lokal* nummeriert. Das geht mit der Umgebung `satzbem`:

¹Dafür verwende ich das `enumerate`-Paket von David Carlisle, welches auch das übliche `enumerate` um einen optionalen Parameter erweitert – eine Zusammenfassung der Möglichkeiten findet sich im Abschnitt 4.3.7 im Implementationsteil, Details in der Dokumentation des Packages.

BEMERKUNG 1. Diese Bemerkung wird nur innerhalb des Unterkapitels nummeriert, zeigt dessen Nummer aber nicht mit an.

lemma Lemmata werden ebenfalls nicht automatisch nummeriert. Mit Hilfe des optionalen Parameters kann man allerdings Namenszusätze unterbringen. Ansonsten steht da nur LEMMA.

```
\begin{lemma}[ von Paul]
  Dieses Lemma wird für den Beweis des obigen
  Satzes nicht viel nützen.
\end{lemma}
```

LEMMA VON PAUL. Dieses Lemma wird für den Beweis des obigen Satzes nicht viel nützen.
--

satzgen Sollte man einmal einen Satz mit speziellen Namen benötigen, kann dafür die `satzgen`-Umgebung erhalten.²

```
\begin{satzgen}{Spezialsatz}
  Dieser Satz dient nur als Beispiel.
\end{satzgen}
```

SPEZIALSATZ. Dieser Satz dient nur als Beispiel.

2.3.4 Definitionen

defn Her Kleinert hatte die Definitionen zwar nicht nummeriert, aber ich habe einfach einmal eine Nummerierung (pro Kapitel) eingeführt.

```
\begin{defn}
  Ein Satz heißt \textbf{sinnlos}, wenn man keine
  vernünftige Anwendung dafür finden kann.
\end{defn}
```

Definition 2.1. Ein Satz heißt *sinnlos*, wenn man keine vernünftige Anwendung dafür finden kann.

Weitere Hilfsmittel für Definitionen sind in Abschnitt 2.4.2 zu finden.

vereinbarung Vereinbarungen funktionieren genauso, und verwenden sogar den selben Zähler wie die Definitionen.

2.3.5 Beispiele, Fragen, etc.

bsp

```
\begin{bsp}
  Beispiele werden je Abschnitt durchnummeriert.
\end{bsp}
```

²Damit wurden auch die bisher genannten Satz-Umgebungen definiert.

wird zu

Beispiel 2.3.1. Beispiele werden je Abschnitt durchnummeriert.

frage

```
\begin{frage}
  Waren die Beispiel-Sätze und -Definitionen
  in diesem Abschnitt sinnvoll?
\end{frage}
```

...führt zu ...

Frage. Waren die Beispiel-Sätze und -Definitionen in diesem Abschnitt sinnvoll?

fragen Mit `fragen` kann man mehrere Fragen zusammen auflisten.

```
\begin{fragen}
  \item erste Frage?
  \item zweite Frage?
  \item dritte Frage?
\end{fragen}
```

Fragen. (1) erste Frage?

(2) zweite Frage?

(3) dritte Frage?

antwort Es gibt natürlich auch Antworten, welche genauso formatiert werden. (Oft ist es sinnvoller einen Satz zu verwenden.), Fazits, Probleme, Ziele,

fazit Lösungen, Lösungsideen und „Achtung“s – alle werden genauso angewandt
problem und ergeben ähnliche Ergebnisse.

ziel

loesung **2.3.6 Beweise**

loesungsidee

achtung

beweis

```
\begin{beweis}
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
  Da wir noch keinen wirklichen Satz
  aufgestellt haben, hier nur ein Dummy-Beweis.
\end{beweis}
```

Man beachte die Einleitung ...

Beweis.

Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. Da wir noch keinen wirklichen Satz aufgestellt haben, hier nur ein Dummy-Beweis. \square

... und das QED-Zeichen am Schluss.

2.4 Befehle

2.4.1 Diverse Symbole für den mathematischen Modus.

Die Aufzählung hier ist leider nicht vollständig - für eine komplette Liste schaue man in Abschnitt 4.6 im Implementationsteil.

Symbol	Befehl	Bedeutung
\emptyset	<code>\leer</code>	Leere Menge. Statt $\emptyset = \emptyset$
$\xrightarrow{\sim}$	<code>\bijekt</code>	Pfeil für Bijektionen
\mathbb{C}	<code>\C</code>	Menge der komplexen Zahlen
\mathbb{R}	<code>\R</code>	Menge der reellen Zahlen
\mathbb{N}	<code>\N</code>	Menge der natürlichen Zahlen ohne 0
\mathbb{N}_0	<code>\No</code>	$\mathbb{N}_0 := \mathbb{N} \cup \{0\}$
\mathbb{Q}	<code>\Q</code>	Menge der rationalen Zahlen
\mathbb{Z}	<code>\Z</code>	Menge der ganzen Zahlen
\mathfrak{P}	<code>\Pot</code>	Potenzmenge einer Menge
\mathfrak{S}	<code>\Perm</code>	Permutationsmenge
\mathcal{P}	<code>\Pol</code>	Menge der ganzzahligen Polynome n. Grades: $\mathcal{P}_n^{\mathbb{Z}}$
\mathfrak{C}	<code>\CF</code>	Menge der Cauchy-Folgen
\mathfrak{N}	<code>\NF</code>	Menge der Nullfolgen
$\mathbb{P}_{\mathbb{R}}^1$	<code>\Proj{1}</code>	Projektive reelle Gerade
$\mathbb{P}_{\mathbb{R}}^2$	<code>\Proj{2}</code>	Projektive reelle Ebene
\mathfrak{Z}_f	<code>\Zul{f}</code>	Menge der f -zulässigen Teilmengen
$\langle x \rangle$	<code>\erzUG{x}</code>	die von x erzeugte Untergruppe
$\langle S \rangle$	<code>\erzUG{S}</code>	dito

Fortsetzung nächste Seite.

Fortsetzung von der vorigen Seite.

Symbol	Befehl	Bedeutung
\bar{z}	<code>\konj{x}</code>	konjugiert komplexe Zahl zu z
$ x $	<code>\abs{x}</code>	Betrag von x (ändert die Größe automatisch)
$(a + b)$	<code>\klamm{a+b}</code>	runde Klammern (ändern die Größe automatisch)
$\{0\}$	<code>\menge{0}</code>	Mengenklammern (ändern die Größe automatisch)
$[\pi]$	<code>\eckKl{\pi}</code>	größtes Ganzes von π
∞	<code>\oo</code>	unendlich
Kor	<code>\Kor</code>	Menge der Korrespondenzen / Korrelationen
Abb	<code>\Abb</code>	Menge der Abbildungen
incl	<code>\incl</code>	Inklusionsabbildung
const	<code>\const</code>	konstante Abbildung
Inj	<code>\Inj</code>	Menge der Injektionen
Surj	<code>\Surj</code>	Menge der Surjektionen
Bij	<code>\Bij</code>	Menge der Bijektionen
Trans	<code>\Trans</code>	Menge der transzendenten Zahlen: $\text{Trans}_{\mathbb{Z}}(R)$
Alg	<code>\Alg</code>	Menge der algebraischen Zahlen: $\text{Alg}_{\mathbb{Z}}(R)$
Spec	<code>\Spec</code>	Primspektrum eines Ringes: $\text{Spec}(\mathbb{Z})$.
Ext	<code>\Ext</code>	Menge der Extrempunkte (siehe Bourbaki)

2.4.2 Definitionen von mathematischen Symbolen und Begriffen

`\idef` Frisch definierte mathematische Symbole sollen oft auch in den Index, `\mdef` und/oder am Rand der aktuellen Seite erscheinen, damit sie leichter zu `\defx` finden sind. Deswegen habe ich die Befehle:

```
\idef{\Pot}
\mdef{\Pot(A)}
\defx{\Pot(A)}
```

für die Darstellung des Symbols `\Pot` im Index (alle Symbole sind am Anfang, unter „“ einsortiert), am Rand, oder an der aktuellen Stelle definiert. `\defx{...}` ist eigentlich überflüssig, nur der Vollständigkeit halber hier enthalten.

`\imdef` Dann gibt es noch die Kombinationen
`\xmdef`
`\xidef`
`\ximdef`

```

\imdef{}    % Index + Rand
\xmdef{}    % hier + Rand
\xidef{}    % hier + Index
\ximdef{}   % hier + Index + Rand

```

`\index` Um Text (keine Symbole) in den Index aufzunehmen, verwendet man den Befehl `\index{Das hier kommt in den Index}`, der bei \LaTeX mitgeliefert wird. Für Definitionen habe ich auch noch `\defi{Wort}` definiert, was *Wort* (das können auch mehrere Wörter sein) an der aktuellen Stelle fett hervorhebt und zusätzlich in den Index aufnimmt. Falls im Index eine andere Form zu finden sein soll (etwa wegen gramatischer Unterschiede), ist noch ein optionaler Parameter möglich: `\defi[Index-Eintrag]{Text}` nimmt *Index-Eintrag* in den Index auf und hebt *Text* hervor.

2.4.3 Abkürzungen bzw. Textbausteine

Dann habe ich noch für einige häufig benutzte Textstücke eigene Befehle definiert:

- `\ubung{7}{3}` – Übungsaufgabe 3 in Serie 7
- `\dsfBa` – dann sind folgende Bedingungen äquivalent:
- `\DsfBa` – Dann sind folgende Bedingungen äquivalent:
- `\oBdA` – ohne Beschränkung der Allgemeinheit
- `\OBdA` – Ohne Beschränkung der Allgemeinheit

3 Der Code für den Dokumentations-Treiber

Der folgende Code sorgt dafür, dass beim Ausführen von latex mit der Datei `alg-script.dtx` diese Dokumentation erzeugt wird.

```
1 <*driver>
2 \documentclass[draft,doc,oneside]{alg-script}
```

Wir machen die in der Klasse vorgenommene Einschränkung der Numerierung rückgängig (für die Doku). Außerdem passen wir die Index-Erzeugung bei `\idef` an den durch `doc` veränderten Index-Mechanismus an. Die Angaben auf dem Rand werden angepasst, damit sie sich nicht mit den Makro-Namen überlappen.

```
3 \setcounter{secnumdepth}{4}
4 \renewcommand{\idef}[1] {\index{##1$\encapchar usage}}
5 \renewcommand{\mdef}[1] {\marginpar{\boxed{#1}}}
```

Diverse Einstellungen für die Doku sind im Paket `pauldoc` getroffen. Damit die richtige Lizenz eingebunden wird, definieren wir den Dateinamen.

```
6 \usepackage[chapter]{pauldoc}[2006/02/14]
7 \newcommand{\lpplfilename}{lppl.tex}
```

Wir stellen noch einige Längen ein, um das Seiten-Layout von `ltxdoc` nachzuahmen – das Layout von `alg-script` taugt nichts für die Dokumentation, insbesondere, da `doc` die Makronamen immer links schreiben will.

```
8 \setlength{\textwidth}{355.0pt}
9 \setlength{\marginparwidth}{126.0pt}
10 \setlength{\oddsidemargin}{82.0pt}
11 \setlength{\evensidemargin}{82.0pt}
12 \setlength{\marginparpush}{0.0pt}
```

Um die ziemlich lange Tabelle der Symbole in Abschnitt 2.4.1 zu setzen, dient das Package `longtable`.

```
13 \usepackage{longtable}
```

Hier wird die Dokumentation schließlich erzeugt. (Listfiles dekommentiere ich bei Bedarf, um mir die Liste der verwendeten Dateien anzeigen zu lassen.)

```
14 %\listfiles
15 \begin{document}
16   \DocInput{alg-script.dtx}
17 \end{document}
18 </driver>
```

4 Der Code der Klasse

4.1 Identifikation

Wir deklarieren zunächst die benötigte \LaTeX -Version und identifizieren diese Klasse.

```
19 <*class>
20 \NeedsTeXFormat{LaTeX2e}[2001/06/01]
21 \ProvidesClass{alg-script}
22     [2006/03/23 v0.13 Pauls Algebra-Skript-Klasse (PE)]
23 </class>
```

4.2 Optionen

4.2.1 doc

Falls die `doc`-Option angegeben wird, soll das `doc`-Package (in 4.3.2) geladen werden.

```
24 \DeclareOption{doc}{%
25     \renewcommand{\vielleicht@doc}{\RequirePackage{doc}[2006/02/02]}%
26 }%
27 \newcommand{\vielleicht@doc}{}%
```

4.2.2 ϕ statt φ : `oldphi`

`\@defphi` Normalerweise tauschen wir `\phi` gegen `\varphi` aus – die Option `oldphi` schaltet das ab. Wenn die Option gegeben ist, wird `\@defphi` als `{}` definiert, andernfalls als `\@newphi`.

```
28 \newcommand{\@defphi}{\@newphi}
29
30 \DeclareOption{oldphi}{
31     \renewcommand{\@defphi}{}
32 }
```

Später (4.6.2) wird dann `\@newphi` definiert und `\@defphi` aufgerufen, um die eventuelle Vertauschung vorzunehmen.

4.3 Geladene Pakete

4.3.1 Basisklasse

Als Basisklasse laden wir `scrbook`, die KOMA-Script-Buchklasse. KOMA-Script ist in aktuellen \TeX -Distributionen enthalten – ich empfehle die Lektüre des beiliegenden `scrguide`.

Wir übergeben hier die Optionen `11pt` (für die Schriftgröße), `a4paper` (für die Papiergröße). Weiterhin werden alle Optionen übergeben, die diese Klasse bekommen hat, außer die selbst behandelten – diese werden hier gleich ausgeführt.

```

33 \DeclareOption*{
34   \PassOptionsToClass{\CurrentOption}{scrbook}
35 }
36 \PassOptionsToClass{11pt,a4paper,leqno}{scrbook}
37 \ProcessOptions{}
38 \LoadClass{scrbook}[2004/09/16]
```

4.3.2 doc

Falls die `doc`-Klassenoption gegeben war, laden wir hier das `doc`-Paket – das ist ein Hack, der leider notwendig war, um die Dokumentation für diese Klasse mit der Klasse selbst zu erstellen, da sich `amscd` (siehe 4.3.3.5) und `doc` nur beim Laden in dieser Reihenfolge vertragen. Wenn jemand eine bessere Lösung findet, immer her damit.

```

39 \vielleicht@doc
```

4.3.3 \mathcal{AMS} -Pakete

Diverse \LaTeX -Pakete werden von der American Mathematical Society bereitgestellt, sie sind insgesamt unter dem Namen $\mathcal{AMS}\text{-}\LaTeX$ bekannt. Sie sind in jeder vernünftigen \TeX -Distribution enthalten.

Mehr Infos zu den einzelnen Paketen gibt es in den Doku-Dateien `amsl.doc.dvi`, `amsth.doc.dvi` und `instr-1.dvi` (sollten in `ams*`-Verzeichnissen unter `texmf/doc/latex/` zu finden sein).

4.3.3.1 \mathcal{AMS} -Math

`amsmath` definiert eine Reihe weiterer mathematischer Umgebungen, welche einige Vorteile gegenüber den in \LaTeX eingebauten haben.

Wir übergeben `namelimits`, um bei Operatoren wie `sin` und `lim` die mit `^` und `_` anzugebenden Indizes/Grenzen im Display-Math-Modus darüber bzw. darunter zu schreiben, nicht daneben.

```

40 \PassOptionsToPackage{namelimits,centertags}{amsmath}
41 \RequirePackage{amsmath}
```


4.3.3.2 \mathcal{AMS} -thm

Das Package `amsthm` erweitert den (später verwendeten) `\newtheorem`-Befehl, sowie definiert die `proof`-Umgebung, welche wir für unsere `beweis`-Umgebung (siehe 4.5.4) benutzen.

```
42 \RequirePackage{amsthm}
```

4.3.3.3 \mathcal{AMS} -Symbole

Das Package `amssymb` definiert einige nützliche mathematische Symbole. Sie sind in der „Comprehensive L^AT_EX Symbol List“ (`symbols-a4`) enthalten – zusätzlich ist in „Users Guide to AMSFonts“ (`amsfndoc`) ab Seite 20 (in Version 2.2d, Januar 2002) eine Liste enthalten.

`\mathbb` Außerdem werden von diesem Package die Befehle `\mathbb`, für „schöne“
`\mathfrak` Mengensymbole wie \mathbb{R} , \mathbb{C} , \mathbb{N} , und `\mathfrak` für Frakturbuchstaben wie \mathfrak{B} ,
 \mathfrak{V} und \mathfrak{a} definiert.

```
43 \RequirePackage{amssymb}
```

4.3.3.4 Euler-Schriftart

`\mathcal` Wir wollen die Euler-Skript-Schrift für `\mathcal` verwenden.

```
44 \RequirePackage{eucal}
```

Beispiel der Anwendung:

Das \mathcal{R} -Integral ist nicht
 so allgemein wie das \mathcal{L} -Integral.

Ergebnis:

Das \mathcal{R} -Integral ist nicht so allgemein wie das \mathcal{L} -Integral.

4.3.3.5 Kommutative Diagramme

Das `amscd`-Package erlaubt einfache kommutative Diagramme ohne diagonale Pfeile.

```
45 \RequirePackage{amscd}
```

`CD` Dafür wird die `CD`-Umgebung genutzt:

```
\begin{CD}
  A @>f>> B @\\
  @VgVV @& @AA{h}A @\\
  C @<<k<< D @
```

wird zu

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ g \downarrow & & \uparrow h \\ C & \xleftarrow{k} & D \end{array}$$

Genauer ist in `amsl.doc.dvi` auf Seite 20 erläutert.

4.3.4 Eingabe

Um die direkte Eingabe von Sonderzeichen (z.B. ä, ö ü, ß) möglich zu machen, laden wir das `inputenc`-Paket mit der Option `latin1`. Das heißt insbesondere, dass die zu verarbeitenden Dateien in ISO-8859-1 kodiert sein sollten, sonst kommt dabei nur Unfug heraus.

```
46 \RequirePackage[latin1]{inputenc}
```

4.3.5 Deutsch

Wir wollen deutschen Text (und vor allem deutsche Bezeichnungen etwa fürs Inhaltsverzeichnis etc.) und die Silbentrennung der neuen deutschen Rechtschreibung nutzen.

```
47 \RequirePackage[ngerman]{babel}
```

4.3.6 xspace

`\xspace` Das `xspace`-Paket hilft uns bei den Definitionen einiger Befehle, indem dort nach der Verwendung des Makros folgender Leerraum beachtet (und nicht verschluckt) wird.

```
48 \RequirePackage{xspace}
```

4.3.7 Neues enumerate

`enumerate` Wir laden mit diesem Package eine Erweiterung der `enumerate`-Umgebung.

```
49 \RequirePackage{enumerate}
```

Es kann jetzt mit `\begin{enumerate}[(a)] ... \end{enumerate}` die Art der Aufzählungszeichen (Standard: (1)) geändert werden. Folgende Zeichen werden durch den Zähler angepasst:

a	a b c d ...
A	A B C D ...
1	1 2 3 4 ...
i	i ii iii iv ...
I	I II III IV ...

Einige Beispielumgebungen werden wir später (Abschnitt 4.5.1) damit noch definieren.

4.3.8 Index

`\printindex` Um einen Index setzen zu können, verwenden wir dieses Standard-Package.
 50 `\RequirePackage{makeidx}`

4.4 Einige geänderte Zähler

Wir wollen die Nummerierung nur für Kapitel (`\chapter`) und Abschnitt (`\section`), kleinere Unterteilungen nicht. (Für diese Dokumentation wurde das rückgängig gemacht, daher ist der Effekt hier nicht zu beobachten.)

51 `\setcounter{secnumdepth}{1}`

Ins Inhaltsverzeichnis sollen dagegen Kapitel, Abschnitte und Unterabschnitte (`\subsection`) aufgenommen werden.

52 `\setcounter{tocdepth}{2}`

4.5 Neue Umgebungen

4.5.1 Neue Aufzählungen

Hier einige häufig verwendete Anwendungen des vorhin (Abschnitt 4.3.7) geladenen `enumerate`-Packages.

`enumabc` Diese drei Umgebungen funktionieren exakt identisch – sie rufen jeweils
`enumnum` die `enumerate`-Umgebung auf, mit jeweils anderem optionalen Parameter.
`enumiii` Dieser legt die Nummerierungsweise fest.

```
53 \newenvironment{enumabc}
54 {\begin{enumerate}[(a)]}
55 {\end{enumerate}}
56 \newenvironment{enumnum}
57 {\begin{enumerate}[(1)]}
58 {\end{enumerate}}
59 \newenvironment{enumiii}
60 {\begin{enumerate}[(i)]}
61 {\end{enumerate}}
```

4.5.2 Sätze im Kasten

Unsere `satz`-Umgebung soll die Sätze in einem Kasten anzeigen. Das ist etwas aufwendiger als die üblichen Satz-Umgebungen.

`\satz@box` In `\satz@box` werden wir unsere Sätze zwischenspeichern.
 62 `\newsavebox{\satz@box}`

satzgen Wir definieren eine generische Satz-Umgebung, um später einfach weitere satzähnliche Konstrukte bauen zu können. Sie nimmt einen Parameter. Der ganze Rest wird in einer LR-Box untergebracht und diese dann in `\satz@box` gespeichert. Dann lassen wir etwas Rand und beginnen eine `minipage` von 80% der Spaltenbreite. Dahinein kommt jetzt zunächst der Text des Argumentes – in einem Extra-Rahmen und in `SMALLCAPS`.

```

63 \newenvironment{satzgen}[1]
64  {%
65   \begin{lrbox}{\satz@box}%
66   \par\par\ \ %
67   \begin{minipage}{0.8\linewidth}%
68   \fbox{\textsc{#1.}}%
69  }

```

Dann folgt der Text, der im Inneren der Umgebung auftaucht. Am Ende schließen wir die `Minipage` ab, lassen etwas Platz, und beenden die `LR-Box` (die, wir erinnern uns, in `\satz@box` gespeichert wird). Danach kommt die eigentliche Ausgabe: Etwas Platz, ein Rahmen mit unserer gespeicherten `Box`, und wieder etwas Platz.

```

70  {%
71 \end{minipage}\end{lrbox}%
72   \par\medskip%
73   \fbox{\usebox{\satz@box}}%
74   \par\medskip%
75  }

```

Das Ganze sieht jetzt in etwa so aus:

```

\begin{satzgen}{Beispiel 1}
  Dies ist ein Beispieltext
\end{satzgen}

```

wird zu:

BEISPIEL 1.	Dies ist ein Beispieltext
-------------	---------------------------

Aus dieser generischen Satz-Umgebung basteln wir uns jetzt das, was wir benötigen.

satz Die `satz`-Umgebung hat eine eigene Zählung innerhalb jedes Abschnittes. (`\numberswithin` wird durch `amsmath` definiert.) Es wird jeweils die Nummer mit ausgegeben.

```

76 \newcounter{satz}
77 \numberwithin{satz}{section}
78 \newenvironment{satz}
79  {
80   \refstepcounter{satz}
81   \begin{satzgen}{Satz \thesatz}
82  }
83  { \end{satzgen} }

```

Ein Beispiel der Anwendung ist

```
\begin{satz}
  Dies ist ein Test-Satz.
\end{satz}
```

mit der Ausgabe:

SATZ 4.5.1.	Dies ist ein Test-Satz.
-------------	-------------------------

satzbem Die `satzbem`-Umgebung hat eine eigene Zählung innerhalb jedes Unterabschnittes, gibt dessen Nummer aber nicht mit aus.

```
84 \newcounter{satzbem}[subsection]
85 \@addtoreset{satzbem}{section}
86 \newenvironment{satzbem}
87 {%
88   \refstepcounter{satzbem}%
89   \begin{satzgen}{Bemerkung \thesatzbem}%
90 }
91 {\end{satzgen}}
```

satzon Ein „Satz ohne Nummer“.

```
92 \newenvironment{satzon}
93 { \begin{satzgen}{Satz ohne Nummer} }
94 { \end{satzgen} }
```

Herr Kleinert hat neben seinen eigentlichen (durchnummerierten) Sätzen noch diverse weitere Korollare, Bemerkungen, etc. aufgeschrieben, die ich – im Mengenlehre-Skript – alle als „Satz ohne Nummer“ ins Skript aufnehme.

Ein paar spezielle „Sätze ohne Nummer“ habe ich für die späteren Vorlesungen aber trotzdem noch:

behauptung Eine Behauptung.

```
95 \newenvironment{behauptung}
96 { \begin{satzgen}{Behauptung} }
97 { \end{satzgen} }
```

eigsatz `eigsatz` ist die Kombination von `satzgen` mit einer der `enumXXX`-Umgebungen. Den genauen Namen dieser Umgebung müssen wir uns merken, damit wir die Umgebung nachher wieder beenden können.

```
98 \newenvironment{eigsatz}[1][num]
99 {%
100   \renewcommand*{\@temp}{#1}%
101   \begin{satzgen}{Eigenschaften}%
102   \begin{enum\@temp}%
103 }
104 {
105   \end{enum\@temp}%
```

```
106     \end{satzgen}%
107 }
```

korollar Ein Korollar.

```
108 \newenvironment{korollar}
109 { \begin{satzgen}{Korollar} }
110 { \end{satzgen} }
```

awa Das Auswahlaxiom wird speziell markiert.¹

```
111 \newenvironment{awa}
112 { \begin{satzgen}{Auswahlaxiom} }
113 { \end{satzgen} }
```

lemma Lemmata sind ja vom Prinzip auch Sätze, sollten also genauso formatiert werden. Ein optionaler Parameter kann etwa eine Nummer oder ein zusätzlicher Name sein, welcher nach „Lemma“ ausgegeben wird.

```
114 \newenvironment{lemma}[1] []
115 { \begin{satzgen}{Lemma#1} }
116 { \end{satzgen} }
```

4.5.3 Definitionen, Beispiele, Fragen

defn Definitionen werden kursiv gedruckt und innerhalb jedes Kapitels durchnummeriert.

```
117 \theoremstyle{plain}
118 \newtheorem{defn}{Definition}[chapter]
```

vereinbarung Eine Vereinbarung ist ähnlich wie eine Definition zu handhaben, und wird daher auch genauso formatiert und nummeriert.

```
119 \newtheorem{vereinbarung}[defn]{Vereinbarung}
```

bsp Ein Beispiel wird dagegen wie ganz normaler Text formatiert. Die Nummerierung erfolgt je Abschnitt.

```
120 \theoremstyle{remark}
121 \newtheorem{bsp}{Beispiel}[section]
```

frage Eine Frage wird wie normaler Text formatiert, und bekommt keine Nummerierung.

```
122 \newtheorem*{frage}{Frage}
```

fragen Das ist eine Kombination einer frage-artigen Umgebung mit `enumnum`.

```
123 \newtheorem*{fragen@impl}{Fragen}
124 \newenvironment*{fragen}
```

¹Diesen Satz braucht man üblicherweise nicht (und auch ich nur ein einziges Mal). Ich überlege daher, ihn rauszustreichen bzw. in ein eigenes Paket für die entsprechende Vorlesung zu verschieben.

```

125 {%
126   \begin{fragen@impl}%
127     \begin{enumnum}%
128   }
129 {
130   \end{enumnum}%
131   \end{fragen@impl}%
132 }

antwort Zu einer Frage gehört natürlich auch eine Antwort.
133 \newtheorem*{antwort}{Antwort}

problem Analog zur Frage.
134 \newtheorem*{problem}{Problem}

loesung Gehört zu einem Problem dazu.
loesungsidee 135 \newtheorem*{loesung}{Lösung}
136 \newtheorem*{loesungsidee}{Lösungsidee}

achtung Gelegentlich gibt es kritische Stellen, auf die hingewiesen werden muss ... Im
warnung Prinzip ist das nichts anderes als eine Bemerkung.
137 \newtheorem*{achtung}{Achtung}
138 \newtheorem*{warnung}{Warnung}

fazit Ein Fazit wird wie normaler Text formatiert, und bekommt keine Numme-
bemerkung rierung, genauso wie eine Bemerkung und eine Erinnerung.
erinnerung 139 \newtheorem*{fazit}{Fazit}
140 \newtheorem*{bemerkung}{Bemerkung}
141 \newtheorem*{erinnerung}{Erinnerung}

satzfazit Ein Fazit kann gelegentlich auch die Form eines Satzes haben, sollte dann
auch wie ein solcher dargestellt werden.
142 \newenvironment{satzfazit}
143 {\begin{satzgen}{Fazit}}
144 {\end{satzgen}}

ziel Gelegentlich ist es für die Motivation ganz hilfreich, wenn man
gesagt bekommt, wo man eigentlich hin will. Das geht mit
\begin{ziel}...\end{ziel}.
145 \newenvironment*{ziel}
146 {%
147   \noindent\textbf{Ziel:}%
148   \begin{quotation}%
149   }
150   {%
151   \end{quotation}%
152   }

```

4.5.4 Beweise

`beweis` Für Beweise passe ich die `proof`-Umgebung aus `amsthm` geeignet an.

```
153 \newenvironment{beweis}[1] []
154 {
155   \begin{proof}[Beweis#1]%
156     ~\par%
157 }
158 {\end{proof}}
```

4.6 Mathematische Symbole

Diverse Symbole werden in den Vorlesungen verwendet. Teilweise habe ich (für vorhandene Symbole) neue Namen erfunden, teilweise die Symbole erst zusammengesetzt.

4.6.1 Alternatives Symbol für kartesisches Produkt

`\varprod` Wir wollen neben dem normalen Symbol \prod für das kartesische Produkt auch das Symbol \times verwenden können. Dazu kopiere ich hier den passenden Code aus `txfonts`. Es wird die Schriftart `txexa` benötigt. (Falls das nicht einzeln geht, müssen die TX Fonts komplett installiert werden.)

```
159 \DeclareSymbolFont{largesymbolsA}{U}{txexa}{m}{n}
160 \DeclareMathSymbol{\varprod}{\mathop}{largesymbolsA}{"10}
```

Anwendungsbeispiel: $\varprod_{i=1}^n M_i$ wird zu

$$\times_{i=1}^n M_i.$$

4.6.2 Anderes `\phi`

`\@newphi` Ich tausche das übliche `\phi` (ϕ) gegen das mir besser gefallende `\varphi` (φ) aus: `\@newphi` sichert das alte `\phi` (ϕ) als `\oldphi`, und redefiniert `\phi` dann `\phi`, um `\varphi` (φ) anzuzeigen.

```
161 \newcommand{\@newphi}{%
162   \newcommand{\oldphi}{\phi}%
163   \let\oldphi\phi%
164   \renewcommand{\phi}{\varphi}%
165 }
```

Dann rufen wir noch `\@defphi` auf, was (je nachdem, ob die Option `oldphi` gegeben wurde oder nicht, siehe 4.2.2) entweder `\@newphi` aufruft oder nichts tut.

```
166 \@defphi
```


4.6.3 Schönere Namen für ν und μ

μ
 ν
 M
 N

`\my` Die englischen Namen der griechischen Buchstaben μ (mu) und ν (nu) passen mir nicht so sehr, daher definiere ich mir eben noch als Aliase `\my` und `\ny` sowie die Großbuchstaben M und N (welche identisch mit den entsprechenden lateinischen Buchstaben sind) dazu.

```
167 \newcommand{\my}{\mu}
168 \newcommand{\My}{M}
169 \newcommand{\ny}{\nu}
170 \newcommand{\Ny}{N}
```

4.6.4 Einige Pfeile

\rightsquigarrow

`\bijekt` Ein Pfeil mit einer Tilde drüber, wie er für Bijektionen (bzw. Isomorphismen) verwendet wird.

```
171 \DeclareRobustCommand*\bijekt}
172 { \tilde{\longrightarrow}\hspace{0.3em} }
```

\twoheadrightarrow

`\surj` Ein Pfeil mit doppelter Spitze, wie er für Surjektionen (bzw. Epimorphismen) verwendet wird.

```
173 \DeclareRobustCommand*\surj}
174 { \twoheadrightarrow }
```

\hookrightarrow

`\inj` Ein Pfeil mit einem Haken am Ende, wie er für Injektionen (bzw. Monomorphismen) verwendet wird.

```
175 \DeclareRobustCommand*\inj}
176 { \hookrightarrow }
```

\Leftrightarrow

`\gdw` Das ist ein „genau-dann-wenn“-Pfeil. Der Name `\gdw` ist einfach schneller zu schreiben und ein Leser des Quelltextes weiß gleich die Bedeutung.

```
177 \DeclareRobustCommand*\gdw}
178 { \ensuremath{\Leftrightarrow}\xspace }
```

\Rightarrow
 \Rightarrow

`\folgt` Das ist ein Pfeil für Folgerungen – `\folgt` kann im Text- oder im Mathe-Modus verwendet werden. Für die `align`-Umgebung ist `\folgtx` geeignet (am Anfang einer Zeile), dahinter wird viel Platz gelassen.

```
179 \newcommand{\folgt}{\ensuremath{\Rightarrow}}
180 \newcommand{\folgtx}{\Rightarrow\quad}
```

4.6.5 Bekannte Zahlenmengen

\mathbb{C}

`\C` Die Menge der komplexen Zahlen.

```
181 \DeclareRobustCommand*\C{\mathbb{C} }
```

\mathbb{R}

`\R` Die Menge der reellen Zahlen.

```
182 \DeclareRobustCommand*\R{\mathbb{R}}
```

\mathbb{N} $\backslash\mathbb{N}$ Die Menge der natürlichen Zahlen (ohne 0).183 `\DeclareRobustCommand*\N{\mathbb{N}}` \mathbb{N}_0 $\backslash\mathbb{N}_0$ Die Menge der natürlichen Zahlen (mit 0). Dieses Symbol kann auch im Textmodus benutzt werden.184 `\DeclareRobustCommand*\No { \ensuremath{\mathbb{N}_0} }` \mathbb{Q} $\backslash\mathbb{Q}$ Die Menge der rationalen Zahlen.185 `\DeclareRobustCommand*\Q { \mathbb{Q} }` \mathbb{Z} $\backslash\mathbb{Z}$ Die Menge der ganzen Zahlen.186 `\DeclareRobustCommand*\Z{\mathbb{Z}}`

4.6.6 Weitere Mengen

 \mathbb{A}_n^k $\backslash\mathbb{A}_n^k$ Der affine Raum bestimmter Dimension zu einem Körper $\backslash\mathbb{A}_n^k$ wird mit $\backslash\mathbb{A}$ gebildet.187 `\DeclareRobustCommand*\Aff{\mathbb{A}}` \mathbb{P} $\backslash\mathbb{P}$ Die Potenzmenge einer Menge.188 `\DeclareRobustCommand*\Pot { \mathfrak{P} }` \mathfrak{S} $\backslash\mathfrak{S}$ Die Menge aller Permutationen einer Menge.189 `\DeclareRobustCommand*\Perm { \mathfrak{S} }` $\backslash\mathcal{P}_n^{\mathbb{Z}}$ Polynomengen. Als Beispiel: $\backslash\mathcal{P}_n^{\mathbb{Z}}$ wird zu $\mathcal{P}_n^{\mathbb{Z}}$ (die Menge aller Polynome mit ganzzahligen Koeffizienten und Grad n).190 `\DeclareRobustCommand*\Pol { \mathcal{P} }` \mathfrak{C} $\backslash\mathfrak{C}$ Die Menge der Cauchy-Folgen in einem normierten Raum191 `\DeclareRobustCommand*\CF { \mathfrak{C} }` \mathfrak{N} $\backslash\mathfrak{N}$ Die Menge der Nullfolgen in einem normierten Raum.192 `\DeclareRobustCommand*\NF { \mathfrak{N} }` $\mathbb{P}_{\mathbb{R}}^1$ $\backslash\mathbb{P}_{\mathbb{R}}^1$ Projektive reelle Gerade – $\backslash\mathbb{P}_{\mathbb{R}}^1$ – oder Ebene – $\backslash\mathbb{P}_{\mathbb{R}}^2$. $\mathbb{P}_{\mathbb{R}}^2$ 193 `\DeclareRobustCommand*\Proj[1]{\mathbb{P}_{\mathbb{R}}^{\#1}}` \mathfrak{Z}_f $\backslash\mathfrak{Z}_f$ Menge der f -zulässigen Teilmengen des Definitionsbereiches von f : $\backslash\mathfrak{Z}_f$. (Siehe Beweis vom Fixpunktlema von Bourbaki.)194 `\DeclareRobustCommand*\Zul[1]{\mathfrak{Z}_{\#1}}`

4.6.7 Klammern & Co.

Diese Makros nehmen alle ein Argument, welches dann irgendwie in die Ausgabe eingebaut wird. Die Größe passt sich dabei an, wenn nötig.

$\langle a \rangle$	<code>\verzUG</code>	Die von einem Element a oder einer Menge S in einer Gruppe erzeugte Untergruppe <code>\verzUG{a}</code> bzw <code>\verzUG{S}</code> .
$\langle S \rangle$	195	<code>\DeclareRobustCommand*\verzUG}[1</code>
	196	<code>{\left\langle #1 \right\rangle}</code>
\overline{x}	<code>\konj</code>	Die zu einer komplexen Zahl x konjugiert komplexe Zahl <code>\konj{x}</code> .
	197	<code>\DeclareRobustCommand*\konj}[1] { \overline{#1} }</code>
$ x $	<code>\abs</code>	Der absolute Betrag einer (reellen oder komplexen) Zahl x ist <code>\abs{x}</code> .
	198	<code>\DeclareRobustCommand*\abs}[1] { \left\lvert #1 \right\rvert }</code>
(x)	<code>\klamm</code>	Einfach nur runde Klammern um einen Term x : <code>\klamm{x}</code>
	199	<code>\DeclareRobustCommand*\klamm}[1] { \left(#1 \right) }</code>
$\{1\}$	<code>\menge</code>	Für Mengen brauchen wir entsprechend Mengenklammern: <code>\menge{1}</code> . Etwas aufwendiger: <code>\menge{x \in N \mid x > 2}</code> wird zu $\{x \in N \mid x > 2\}$.
	200	<code>\DeclareRobustCommand*\menge}[1]{\left\{ #1 \right\} }</code>
$\{1, \dots, n\}$	<code>\einsbis</code>	Häufig benötigter Spezialfall: <code>\einsbis{n}</code> .
	201	<code>\newcommand*\einsbis}[1]{\menge{1, \dots, #1}}</code>
	<code>\menget</code>	Falls in den Mengenklammern nur einfacher Text stehen soll (also nichts mathematisches), ist <code>\menget</code> geeignet.
	202	<code>\newcommand*\menget}[1]{\menge{\text{#1}}}</code>
	<code>\rightmid</code>	<code>\rightmid</code> malt einen senkrechten Strich in ausreichender Höhe rechts neben das Argument, <code>\leftmid</code> links daneben. <code>\mengemid</code> zeigt zwei Argumente in einer (ausreichend großen) Mengenklammer sowie einen ausreichend großen senkrechten Strich dazwischen.
	203	<code>\DeclareRobustCommand*\rightmid}[1]{\left. #1 \, \, \right }</code>
	204	<code>\DeclareRobustCommand*\leftmid}[1]{\left \, \, #1 \right. }</code>
	205	<code>\DeclareRobustCommand*\mengemid}[2]{%</code>
	206	<code>\menge{ \left. #1 \vphantom{#2} \, \, \right #2}</code>
	207	<code>}</code>
$\lceil x \rceil$	<code>\eckKl</code>	Für verschiedene Zwecke (etwa das größte Ganze einer reellen Zahl, oder eine Äquivalenzklasse einer Ä-Relation) werden eckige Klammern benötigt: <code>\eckKl{x}</code> .
	208	<code>\DeclareRobustCommand*\eckKl}[1]{\left[#1 \right] }</code>

4.6.8 Operatoren mit Namen

Einige Operatoren, bei denen üblicherweise der Name hingeschrieben wird, für die Verwendung im mathematischen Modus. Diese Namen werden so nicht wie üblich in Schräg-Schrift, sondern in Roman angezeigt. Bei der Verwendung kommen oft noch Indizes und/oder Klammern mit „Parametern“ dazu.

Für die Mathematische Bedeutung bitte ich im Skript nachzulesen, die kurzen Bemerkungen und der Index sollten ausreichen, die Stelle zu finden.

<code>\Kor</code>	Menge der Korrespondenzen/Korrelationen	209	<code>\DeclareMathOperator{\Kor}{Kor}</code>
<code>\Abb</code>	Menge der Abbildungen	210	<code>\DeclareMathOperator{\Abb}{Abb}</code>
<code>\incl</code>	Inklusionsabbildung	211	<code>\DeclareMathOperator{\incl}{incl}</code>
<code>\const</code>	Konstante Abbildung	212	<code>\DeclareMathOperator{\const}{const}</code>
<code>\Inj</code>	Menge der Injektionen	213	<code>\DeclareMathOperator{\Inj}{Inj}</code>
<code>\Surj</code>	Menge der Surjektionen	214	<code>\DeclareMathOperator{\Surj}{Surj}</code>
<code>\Bij</code>	Menge der Bijektionen	215	<code>\DeclareMathOperator{\Bij}{Bij}</code>
<code>\Trans</code>	Menge der transzendenten Zahlen	216	<code>\DeclareMathOperator{\Trans}{Trans}</code>
<code>\Alg</code>	Menge der algebraischen Zahlen.	217	<code>\DeclareMathOperator{\Alg}{Alg}</code>
<code>\Spec</code>	Primspektrum eines Ringes	218	<code>\DeclareMathOperator{\Spec}{Spec}</code>
<code>\Specm</code>	Maximalspektrum eines Ringes	219	<code>\DeclareMathOperator{\Specm}{Specm}</code>
<code>\Ext</code>	Menge der Extrempunkte. (Siehe Beweis zum Fixpunktlema von Bourbaki.)	220	<code>\DeclareMathOperator{\Ext}{Ext}</code>

`\Mod` Die Menge der Modul-Strukturen auf einer Gruppe (über einem Ring)
 221 `\DeclareMathOperator{\Mod}{Mod}`

`\Hom` Menge von Homomorphismen (einer Gruppe, eines Ringes, eines Vektorraumes, etc.)
 222 `\DeclareMathOperator{\Hom}{Hom}`

`\End` Menge von Endomorphismen einer algebraischen Struktur.
 223 `\DeclareMathOperator{\End}{End}`

`\Fix` Menge der Fixpunkte.
 224 `\DeclareMathOperator{\Fix}{Fix}`

`\Transl` Menge von Translationen.
 225 `\DeclareMathOperator{\Transl}{Transl}`

4.6.9 Weitere Symbole

(Doku fehlt noch.)

226 `\DeclareRobustCommand*\ohne}{\setminus}`
 227 `\DeclareRobustCommand*\oo}{\infty}`
 228 `\DeclareRobustCommand*\leer}{\varnothing}`

$f \circ g$

`\kringel` Die Verkettung zweier Funktionen $f \circ g$ wird „eff Kringel geh“ ausgesprochen, also wollen wir sie auch `f \kringel g` schreiben können.
 229 `\DeclareRobustCommand*\kringel}{\circ}`

$A \dot{\cup} B$

`\disjU` Die Vereinigung zweier disjunkter Mengen: $A \dot{\cup} B$.
 230 `\DeclareRobustCommand*\disjU}{\mathop{\dot{\cup}}}`

4.7 Hilfsmittel für Definitionen

In einer Vorlesung werden regelmäßig neue Begriffe und Symbole definiert. Dafür habe ich ein paar zusätzliche Befehle entwickelt.

`\mdef` `\mdef{x}` zeigt ein Symbol am Rand (engl. „margin“) einem Kästchen an. Dabei wird (durch das verwendete `\boxed`) automatisch in den mathematischen Modus gewechselt. Das ganze wird jeweils zur Seite hin geschoben (am linken Rand nach rechts, am rechten Rand nach links), damit nicht so eine große Lücke vorhanden ist.

231 `\newcommand*\mdef}[1]`
 232 `{ \marginpar[\hfill\boxed{#1}]{\boxed{#1}\hfill} }`

Demonstrationen davon waren auf den vorhergehenden Seiten zu sehen.

4 Der Code der Klasse

`\defx` `\defx` sollte eigentlich `\xdef` heißen, aber das war schon für irgend etwas anderes vergeben. Der Befehl wechselt einfach nur in den mathematischen Modus (oder bleibt da) und gibt das Argument aus. Der Befehl ist nur aus Vollständigkeitsgründen (siehe die Benennung der folgenden Befehle) vorhanden.

```
233 \newcommand*{\defx}[1] {\ensuremath{ #1 }}
```

`\idef` `\idef` nimmt sein Argument in den Index auf. Die Symbole tauchen dort ganz am Anfang auf, ziemlich willkürlich geordnet.

```
234 \newcommand*{\idef}[1]{\index{ @$#1$}}
```

Die nächsten Befehle sind einfach nur Kombinationen dieser drei Befehle.

`\xmdef` `\xmdef` zeigt das Symbol an der aktuellen Stelle und am Rand.

```
235 \newcommand*{\xmdef}[1] {\defx{#1}\mdef{#1}}
```

`\ximdef` Hiermit wird ein Symbol an der aktuellen Stelle, am Rand und im Index gezeigt.

```
236 \newcommand*{\ximdef}[1]
237 {%
238     \defx{#1}%
239     \mdef{#1}%
240     \idef{#1}%
241 }
```

`\imdef` `\imdef` zeigt das Symbol im Index und am Rand.

```
242 \newcommand*{\imdef}[1]{\idef{#1}\mdef{#1}}
```

`\xidef` Was fehlt noch? – Richtig, dieser Befehl zeigt das Symbol hier und im Index.

```
243 \newcommand*{\xidef}[1]{\defx{#1}\idef{#1}}
```

`\defi` [*indextext*]{*text*} Wir geben zunächst *text* fettgedruckt aus.

```
244 \newcommand*{\defi}[2][\empty]%
245 {%
246     \textbf{#2}%
```

Danach überprüfen wir, ob *indextext* angegeben wurde. Falls ja, nehmen wir diesen in den Index auf, sonst *text*.

```
247     \ifx#1\empty%
248         \index{#2}%
249     \else%
250         \index{#1}%
251     \fi%
252 }
```

`\defin` Eine leichte Abwandlung davon ist `\defin`, welches den ersten Parameter fett druckt und den zweiten in den Index aufnimmt.

Inzwischen ist dieser Befehl überflüssig, weil `\defi` einen optionalen Parameter versteht – entsprechend verwendet die Definition von `\defin` auch `\defi`.

```
253 \newcommand*{\defin}[2]{%\textbf{#1}\index{#2}
254     \defi[#2]{#1}%
255 }
```

4.8 Textbausteine

Für einige häufig verwendete Texte habe ich mir Bausteine definiert.

`\ubung` Verweise auf Übungsaufgaben werden (z.B.) per `\ubung{2}{3}` erledigt, das Ergebnis ist

Übungsaufgabe 3 in Serie 2

```
256 \newcommand*{\ubung}[2]{Übungsaufgabe #2 in Serie #1}
```

`\DsfBa` Während Herr Kleinert an der Tafel nur „D.s.f.B.ä.“ anscrieb, gönnen wir uns hier den Luxus, das auszuschreiben: `\DsfBa` ergibt „Dann sind folgende Bedingungen äquivalent.“.

```
257 \newcommand*{\DsfBa}
258 {Dann sind folgende Bedingungen äquivalent:\xspace}
```

`\dsfBa` Wir haben auch eine Variante, falls die Klausel mitten im Satz vorkommt: dann sind folgende Bedingungen äquivalent:

```
259 \newcommand*{\dsfBa}
260 {dann sind folgende Bedingungen äquivalent:\xspace}
```

`\oBdA` Ein (besonders in Beweisen) häufig verwendeter Begriff ist `\oBdA`, also ohne Beschränkung der Allgemeinheit.

```
261 \newcommand*{\oBdA}{ohne Beschränkung der Allgemeinheit\xspace}
```

`\OBdA` Falls das am Satzanfang vorkommt, muss es natürlich großgeschrieben werden.

```
262 \newcommand*{\OBdA}{Ohne Beschränkung der Allgemeinheit\xspace}
```

4.9 Vermischtes

`\Vek` Vektoren und Vektorräume werden bei Herrn Kleinert immer in altdeutscher Schrift geschrieben. Daher hier z.B. `\Vek V` für einen Vektorraum. Die gleiche Formatierung gilt für Ideale in Ringen.

```
263 \DeclareRobustCommand*{\Vek}[1]{\mathfrak{#1}}
264 \DeclareRobustCommand*{\ideal}[1]{\mathfrak{#1}}
```

`\thesection` `\thesection` wird von diversen Befehlen für die Ausgabe der Nummerierung der Abschnitte benutzt. Wir wollen hier die Kapitelnummer mit ausgeben, deswegen definieren wir diesen Befehl hier neu.

```
265 \renewcommand{\thesection} {\thechapter.\arabic{section}}
```

Da die standardmäßige Silbentrennung von L^AT_EX bei einigen Wörtern versagt, hier die Abhilfe:

```
266 \hyphenation{Schroe-der}
```

`\tild` Um auch im Textmodus bequem eine Tilde verwenden zu können, wechseln wir einfach in den mathematischen Modus und nehmen da das passende Symbol.

```
267 \DeclareRobustCommand*\tild{\sim\hspace}
```

`\Aussage` Dieses Makro wird in Kapitel 0 bei den logischen Symbolen verwendet.² Es wird in spitzen Klammern und schrägedruckt das Wort „Aussage“ und dann das Argument (1 oder 2) gedruckt. Beispiel: `$$\Aussage{1}$$` wird zu *`Aussage1`*.

```
268 \DeclareRobustCommand*\Aussage[1]
269     {\erzUG{\text{\textsl{Aussage#1}}}}
```

Beim Setzen der Wörter soll lieber etwas mehr Luft gelassen werden, als dass die Wörter über den Rand ragen (und T_EX sich dann beschwert).

```
270 \sloppy
```

Die einzelnen Seiten sollen lieber mit unterschiedlich hohen unteren Enden des Textes ausgestattet werden, als die Lücken dazwischen zu dehnen.

```
271 \raggedbottom
```

Wir wollen einen Index aller Symbole, definierten Wörter, etc. anlegen.

```
272 \makeindex
```

4.10 Der Schluss

Am Ende sagen wir T_EX, dass die Eingabe-Datei zu Ende ist.

```
273 \endinput
274 </class>
```

Damit ist dann unsere Klasse zu Ende.

²Auch dieses Makro wird nur an einer Stelle verwendet und ist damit ein Abschlusskandidat.

A The L^AT_EX Project Public License

LPPL Version 1.3b 2006-01-07

Copyright 1999 2002–2006 L^AT_EX3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

A.1 Preamble

The L^AT_EX Project Public License (LPPL) is the primary license under which the the L^AT_EX kernel and the base L^AT_EX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T_EX-related (such as a L^AT_EX package), but you may use it with small modifications even if your work is unrelated to T_EX.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L^AT_EX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘`cfgguide.tex`’ and ‘`modguide.tex`’ in the base L^AT_EX distribution for suggestions.

A.2 Definitions

In this license document the following terms are used:

Work Any work being distributed under this License.

Derived Work Any work that under any applicable law is derived from the Work.

Modification Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify To apply any procedure that produces a Derived Work under any applicable law.

Distribution Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun’s Network File System (NFS).

Compiled Work A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the ‘Copyright Holder’ under any applicable law.

Base Interpreter A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a ‘L^AT_EX-Format’ or in the case of files belonging to the ‘L^AT_EX-format’ a program implementing the ‘T_EX language’.

A.3 Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.

A *The L^AT_EX Project Public License*

2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
 - a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
 - b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

- c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
- d) You distribute at least one of the following with the Derived Work:
 - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
 - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
- 7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
- 8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
- 9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
- 10.
 - a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
 - b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor

the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

A.4 No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

A.5 Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication

for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
 - a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
 - b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L^AT_EX work, this could be done, for example, by posting to `comp.text.tex`.)
3.
 - a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
 - b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

A.6 Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

A.6.1 Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, the discussion in ‘`modguide.tex`’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A.6.2 A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

A.6.3 How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

A The L^AT_EX Project Public License

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status ‘maintained’.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘pig.dtx’, ‘pig.ins’, and ‘pig.sty’ (the last being generated from ‘pig.dtx’ using ‘pig.ins’), the ‘Base Interpreter’ referring to any ‘L^AT_EX-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

A.6.4 Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

A.6.5 Important Recommendations

A.6.5.1 Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee

A The L^AT_EX Project Public License

to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.

B Liste der Änderungen

v0.00	Allgemein: Erste Test-Fassung . . .	1		
v0.01	Allgemein: Diverse <code>\usepackage-</code> Anweisungen eingefügt.	18		
v0.02	Allgemein: Der Treiber-Code ist jetzt auch dokumentiert.	13		
v0.03	<code>satz</code> : Erste Version der Satz- Umgebung.	20		
v0.04	Allgemein: Das Package wurde zu einer Klasse umgebaut.	14		
	Der Treiber wird umgestellt, um jetzt mit der Klasse zu ar- beiten.	13		
v0.05	Allgemein: Ich habe die Be- nutzerdokumentation aus <code>Anleitung.tex</code> eingefügt (etwas abgeändert)	12		
v0.06	Allgemein: Fertig	31		
v0.07	<code>\varprod</code> : Ich habe das <code>\varprod</code> -Symbol aus den TX Fonts eingefügt.	23		
v0.08	Allgemein: <code>enumabc</code> , <code>enumnum</code> , <code>enumiii</code> aus <code>aufgaben.cls</code> kopiert und Dokumentation dafür geschrieben.	18		
v0.09	<code>bsp</code> : Tippfehler in „Beispiel“ zu „Beispiel“ korrigiert.	21		
v0.10	<code>\defphi</code> : Neu: Option <code>oldphi</code> , um die Erset- zung von <code>\varphi</code> durch <code>\varphi</code> abzuschalten.	14		
	Allgemein: Hinzufügungen für die Verwendung für Prüfungs-Stoff.	1		
	<code>\defin</code> : Neu hinzugekommen	29		
	<code>\folgt</code> : Neu aus <code>aufgaben.cls</code> ko- piert (und angepasst)	24		
	<code>\folgtx</code> : Neu aus <code>aufgaben.cls</code> kopiert (und angepasst)	24		
	<code>\my</code> : Neu aus <code>aufgaben.cls</code> ko- piert (und angepasst)	24		
	<code>\My</code> : Neu aus <code>aufgaben.cls</code> ko- piert (und angepasst)	24		
	<code>\ny</code> : Neu aus <code>aufgaben.cls</code> ko- piert (und angepasst)	24		
	<code>\Ny</code> : Neu aus <code>aufgaben.cls</code> ko- piert (und angepasst)	24		
	<code>\phi</code> : Neu: Ersetzung durch <code>\varphi</code>	23		
	<code>\ximdef</code> : Überflüssige Leerzei- chen in der Definition (und damit in der Ausgabe) ent- fernt.	29		
v0.11	Allgemein: <code>scrbook</code> statt <code>ams-</code> <code>book</code> als Basisklasse, dafür <code>amsmath</code> , <code>amsthm</code> und <code>makei-</code> <code>dx</code> als einzelne Packages.	15		
	Benutzungsbedingungen + LPPL eingefügt	3		
	Diverse Symbol-Kommandos sind jetzt robust, damit sie die Index-Erstellung überle- ben.	24		
	Package <code>babel</code> statt <code>ngerman</code>	17		
	<code>satzfazit</code> : Neu hinzugekom- men	22		
	<code>fazit</code> : Neu hinzugekommen	22		
	<code>\defi</code> : Jetzt mit optionalem Ar- gument, daher <code>\defin</code> über-			

B Liste der Änderungen

<p>flüssig. 29</p> <p><code>\defin</code>: Jetzt überflüssig (da <code>\defi</code> jetzt einen optionalen Parameter hat). 29</p> <p><code>\disjU</code>: Jetzt als <code>\mathop</code>, daher mit richtigem Abständen. . . 28</p> <p><code>behauptung</code>: Neu eingefügt . . . 20</p> <p><code>bemerkung</code>: Neu hinzugekommen 22</p> <p><code>lemma</code>: Jetzt kleingeschrieben (wie alle Umgebungen hier) 21</p> <p>Jetzt mit optionalem Parameter 21</p> <p><code>beweis</code>: Jetzt mit optionalem Argument 23</p> <p><code>\gdw</code>: Jetzt mit <code>\ensuremath</code>, geht also auch im Textmodus. 24</p> <p><code>\ideal</code>: neu 30</p> <p><code>ziel</code>: Neu hinzugekommen . . . 22</p> <p><code>eigsatz</code>: Neu hinzugekommen . . . 20</p> <p><code>\inj</code>: Neu hinzugefügt (auch in <code>aufgaben.cls</code> vorhanden) . . 24</p> <p><code>\kringel</code>: Neu aus <code>aufgaben.cls</code> kopiert. 28</p> <p><code>\mengemid</code>: Neu 26</p> <p><code>\menget</code>: Neu aus <code>aufgaben.cls</code> kopiert. 26</p> <p><code>antwort</code>: Neu 22</p> <p><code>korollar</code>: Neu eingefügt 21</p> <p><code>fragen</code>: Neu hinzugekommen . . 21</p> <p><code>erinnerung</code>: Neu hinzugekommen 22</p> <p><code>\Specm</code>: Neu dazu 27</p> <p><code>\surj</code>: Neu hinzugefügt (auch in <code>aufgaben.cls</code> vorhanden) . . 24</p> <p><code>\Vek</code>: jetzt robust. 30</p>	<p>v0.11a</p> <p>Allgemein: Index und Änderungen vertauscht. 12</p> <p>Verwendung der aktuellen Version der Lizenz (mit Label) und entsprechend von <code>paul-doc.dtx</code> 12</p> <p>v0.12</p> <p><code>satz</code>: Jetzt <code>\refstepcounter</code> statt <code>\stepcounter</code> verwendet, so das Referenzen auf den Satz nicht nur die Abschnittsnummer geben. . . . 19</p> <p><code>satzbem</code>: Neu 20</p> <p>Zählerdefinition vereinfacht. . . 20</p> <p><code>satzgen</code>: Jetzt 80% statt 75% der Seitenbreite, und etwas weniger gezwungener White-space am Ende. 19</p> <p><code>achtung</code>: Neu 22</p> <p><code>\einsbis</code>: Neu 26</p> <p><code>beweis</code>: Jetzt doch einfach <code>\par</code> mit Versuch, einen Seitenumbruch zu verhindern. 23</p> <p><code>loesung</code>: Neu 22</p> <p><code>loesungsidee</code>: Neu 22</p> <p><code>problem</code>: Neu 22</p> <p><code>\tild</code>: Jetzt mit <code>\xspace</code>, damit bei Bedarf ein Leerzeichen gelassen wird. 31</p> <p>v0.13</p> <p><code>warnung</code>: Neu 22</p> <p><code>satzbem</code>: Zähler wird jetzt auch bei <code>\section</code> zurückgesetzt. . . 20</p> <p><code>satzgen</code>: <code>\medskip</code> davor und dahinter eingefügt. 19</p> <p>Jetzt wird <code>\linewidth</code> anstatt <code>\columnwidth</code> verwendet. . . 19</p>
--	---

C Index

Schräggedruckte Nummern verweisen auf die Seite, auf der der Eintrag beschrieben ist, unterstrichene Nummern zeigen auf die Zeilennummer der Definition, sonstige Zahlen auf die Zeilennummer einer Verwendung.

Symbols		
M	24	\Rightarrow 24
N	24	φ 23
A_n^k	25	\times 23
\mathbb{C}	24	<code>\@addtoreset</code> 85
\mathfrak{C}	25	<code>\@defphi</code> <u>28</u> , 166
\mathbb{N}	25	<code>\@newphi</code> 28, <u>161</u>
\mathfrak{N}	25	<code>\@temp</code> 100, 102, 105
\mathbb{N}_0	25	<code>\{</code> 200
\mathfrak{S}	25	<code>\}</code> 200
\mathfrak{B}	25	\sqcup 66
$\mathbb{P}_{\mathbb{R}}^1$	25	
$\mathbb{P}_{\mathbb{R}}^2$	25	A
\mathbb{Q}	25	<code>\Abb</code> <u>210</u>
\mathbb{R}	24	<code>\abs</code> <u>198</u>
\Rightarrow	24	<code>achtung</code> (environment) 9, <u>137</u>
\mathfrak{B}	30	<code>\Aff</code> <u>187</u>
\mathbb{Z}	25	<code>\Alg</code> <u>217</u>
\mathfrak{Z}_f	25	<code>antwort</code> (environment) 9, <u>133</u>
$ x $	26	<code>\Aussage</code> <u>268</u>
$\xrightarrow{\sim}$	24	<code>awa</code> (environment) <u>111</u>
\dot{U}	28	
$[x]$	26	B
\Rightarrow	24	<code>behauptung</code> (environment) 7, <u>95</u>
$\langle S \rangle$	26	<code>bemerkung</code> (environment) <u>139</u>
$\langle a \rangle$	26	<code>beweis</code> (environment) 9, <u>153</u>
\Leftrightarrow	24	<code>\Bij</code> <u>215</u>
\mathfrak{a}	30	<code>\bijekt</code> <u>171</u>
\hookrightarrow	24	<code>bsp</code> (environment) 8, <u>120</u>
(x)	26	
\bar{x}	26	C
\circ	28	<code>\C</code> <u>181</u>
$\{1, \dots, n\}$	26	<code>CD</code> (environment) 16
$\{1\}$	26	<code>\CF</code> <u>191</u>
μ	24	<code>\circ</code> 229
ν	24	<code>\const</code> <u>212</u>
ϕ	23	<code>\cup</code> 230
		<code>\CurrentOption</code> 34

C Index

D	
<code>\DeclareMathSymbol</code>	160
<code>\DeclareOption</code>	24, 30, 33
<code>\DeclareRobustCommand</code>	171, 173, 175, 177, 181–195, 197–200, 203–205, 208, 226–230, 263, 264, 267, 268
<code>\DeclareSymbolFont</code>	159
<code>\defi</code>	12, <u>244</u> , 254
<code>\defin</code>	<u>253</u>
<code>defn</code> (environment)	8, <u>117</u>
<code>\defx</code>	11, 233, 235, 238, 243
<code>\defx{}</code>	<u>233</u>
<code>\disjU</code>	<u>230</u>
<code>\doc</code>	5
<code>\dot</code>	230
<code>\dots</code>	201
<code>\DsfBa</code>	<u>257</u>
<code>\dsfBa</code>	<u>259</u>
<code>frage</code>	9, <u>122</u>
<code>fragen</code>	9, <u>123</u>
<code>korollar</code>	7, <u>108</u>
<code>lemma</code>	8, <u>114</u>
<code>loesung</code>	9, <u>135</u>
<code>loesungsidee</code>	9, <u>135</u>
<code>problem</code>	9, <u>134</u>
<code>satz</code>	6, <u>76</u>
<code>satzbem</code>	7, <u>84</u>
<code>satzfazit</code>	<u>142</u>
<code>satzgen</code>	8, <u>63</u>
<code>satzon</code>	7, <u>92</u>
<code>vereinbarung</code>	8, <u>119</u>
<code>warnung</code>	<u>137</u>
<code>ziel</code>	9, <u>145</u>
<code>erinnerung</code> (environment)	<u>139</u>
<code>\erzUG</code>	<u>195</u> , 269
<code>\evensidemargin</code>	11
<code>\Ext</code>	<u>220</u>
E	
<code>\eckKl</code>	<u>208</u>
<code>eigsatz</code> (environment)	7, <u>98</u>
<code>\einsbis</code>	<u>201</u>
<code>\else</code>	249
<code>\empty</code>	244, 247
<code>\encapchar</code>	4
<code>\End</code>	<u>223</u>
<code>enumabc</code> (environment)	6, <u>53</u>
<code>enumerate</code> (environment)	<u>49</u>
<code>enumiii</code> (environment)	6, <u>53</u>
<code>enumnum</code> (environment)	6, <u>53</u>
environments:	
<code>achtung</code>	9, <u>137</u>
<code>antwort</code>	9, <u>133</u>
<code>awa</code>	<u>111</u>
<code>behauptung</code>	7, <u>95</u>
<code>bemerkung</code>	<u>139</u>
<code>beweis</code>	9, <u>153</u>
<code>bsp</code>	8, <u>120</u>
<code>CD</code>	16
<code>defn</code>	8, <u>117</u>
<code>eigsatz</code>	7, <u>98</u>
<code>enumabc</code>	6, <u>53</u>
<code>enumerate</code>	<u>49</u>
<code>enumiii</code>	6, <u>53</u>
<code>enumnum</code>	6, <u>53</u>
<code>erinnerung</code>	<u>139</u>
<code>fazit</code>	9, <u>139</u>
F	
<code>fazit</code> (environment)	9, <u>139</u>
<code>\fi</code>	251
<code>\Fix</code>	<u>224</u>
<code>\folgt</code>	<u>179</u>
<code>\folgtx</code>	<u>179</u>
<code>frage</code> (environment)	9, <u>122</u>
<code>fragen</code> (environment)	9, <u>123</u>
G	
<code>\gdw</code>	<u>177</u>
H	
<code>\hfill</code>	232
<code>\Hom</code>	<u>222</u>
<code>\hookrightarrow</code>	176
I	
<code>\ideal</code>	<u>263</u>
<code>\idef</code>	4, 11, <u>234</u> , 240, 242, 243
<code>\ifx</code>	247
<code>\imdef</code>	11, <u>242</u>
<code>\incl</code>	<u>211</u>
<code>\index</code>	4, 12, 234, 248, 250, 253
<code>\infty</code>	227
<code>\Inj</code>	<u>213</u>
<code>\inj</code>	<u>175</u>
K	
<code>\klamm</code>	<u>199</u>

C Index

<p><code>\konj</code> 197</p> <p><code>\Kor</code> 209</p> <p><code>korollar</code> (environment) 7, 108</p> <p><code>\kringel</code> 229</p> <p style="text-align: center;">L</p> <p><code>\langle</code> 196</p> <p><code>\leer</code> 228</p> <p><code>\leftmid</code> 203</p> <p><code>\Leftrightarrow</code> 178</p> <p><code>lemma</code> (environment) 8, 114</p> <p><code>\let</code> 163</p> <p><code>\linewidth</code> 67</p> <p><code>\listfiles</code> 14</p> <p><code>\LoadClass</code> 38</p> <p><code>loesung</code> (environment) 9, 135</p> <p><code>loesungsidee</code> (environment) . 9, 135</p> <p><code>\longrightarrow</code> 172</p> <p><code>\lpplfilename</code> 7</p> <p style="text-align: center;">M</p> <p><code>\makeindex</code> 272</p> <p><code>\marginparpush</code> 12</p> <p><code>\marginparwidth</code> 9</p> <p><code>\mathbb</code> 43</p> <p><code>\mathcal</code> 44</p> <p><code>\mathfrak</code> 43</p> <p><code>\mathop</code> 160, 230</p> <p><code>\mdef</code> 5, 11, 231, 235, 239, 242</p> <p><code>\medskip</code> 72, 74</p> <p><code>\menge</code> 200, 201, 202, 206</p> <p><code>\mengemid</code> 203</p> <p><code>\menget</code> 202</p> <p><code>\Mod</code> 221</p> <p><code>\mu</code> 167</p> <p><code>\My</code> 167</p> <p><code>\my</code> 167</p> <p style="text-align: center;">N</p> <p><code>\N</code> 183, 184</p> <p><code>\NF</code> 192</p> <p><code>\No</code> 184</p> <p><code>\noindent</code> 147</p> <p><code>\nu</code> 169</p> <p><code>\numberwithin</code> 77</p> <p><code>\Ny</code> 167</p> <p><code>\ny</code> 167</p> <p style="text-align: center;">O</p> <p><code>\OBdA</code> 262</p>	<p><code>\oBdA</code> 261</p> <p><code>\oddsidemargin</code> 10</p> <p><code>\ohne</code> 226</p> <p><code>\oldphi</code> 5, 161</p> <p><code>\oo</code> 227</p> <p style="text-align: center;">P</p> <p><code>\PassOptionsToPackage</code> 40</p> <p><code>\Perm</code> 189</p> <p><code>\phi</code> 161</p> <p><code>\Pol</code> 190</p> <p><code>\Pot</code> 188</p> <p><code>\printindex</code> 50</p> <p><code>problem</code> (environment) 9, 134</p> <p><code>\ProcessOptions</code> 37</p> <p><code>\Proj</code> 193</p> <p style="text-align: center;">Q</p> <p><code>\Q</code> 185</p> <p><code>\quad</code> 180</p> <p style="text-align: center;">R</p> <p><code>\R</code> 182, 193</p> <p><code>\raggedbottom</code> 271</p> <p><code>\rangle</code> 196</p> <p><code>\refstepcounter</code> 80, 88</p> <p><code>\RequirePackage</code> 25, 41–50</p> <p><code>\Rightarrow</code> 179, 180</p> <p><code>\rightmid</code> 203</p> <p style="text-align: center;">S</p> <p><code>satz</code> (environment) 6, 76</p> <p><code>\satz@box</code> 62, 65, 73</p> <p><code>satzbem</code> (environment) 7, 84</p> <p><code>satzfazit</code> (environment) 142</p> <p><code>satzgen</code> (environment) 8, 63</p> <p><code>satzon</code> (environment) 7, 92</p> <p><code>\setcounter</code> 3, 51, 52</p> <p><code>\setlength</code> 8–12</p> <p><code>\setminus</code> 226</p> <p><code>\Spec</code> 218</p> <p><code>\Specm</code> 219</p> <p><code>\Surj</code> 214</p> <p><code>\surj</code> 173</p> <p style="text-align: center;">T</p> <p><code>\textbf</code> 147, 246, 253</p> <p><code>\textsc</code> 68</p> <p><code>\textsl</code> 269</p> <p><code>\textwidth</code> 8</p>
--	--

