10 Mathematik

In diesem Abschnitt möchten wir Ihnen einen ersten Einblick in das Setzen mathematischer Formeln mit LaTeX 2_ε geben. Wir möchten dies anhand des folgenden Beispieltextes tun:

> Wir sagen, eine Folge $(a_n)_{n\in\mathbb{N}}$ reeller Zahlen *konvergiert* gegen eine reelle Zahl a, wenn es zu jedem $\epsilon > 0$ einen Index n_ϵ gibt, so daß
>
> $$|a_n - a| < \epsilon \quad \text{für alle } n > n_\epsilon$$
>
> ist. Wir schreiben dann $a_n \to a$ oder auch $\lim_{n\to\infty} a_n = a$.
>
> Eine Reihe $\sum_{k=1}^{\infty} a_k$ bezeichnen wir als *konvergent*, wenn die Folge $(\sum_{k=1}^{n} a_k)_{n\in\mathbb{N}}$ ihrer Teilsummen konvergiert.

10.1 Eingebettete Formeln

Ein mathematischer Ausdruck, der im laufenden Text erscheint, wird als *eingebettete Formel* bezeichnet. Wir heben einmal alle eingebetteten Formeln in unserem Beispieltext hervor:

> Wir sagen, eine Folge $(a_n)_{n\in\mathbb{N}}$ reeller Zahlen *konvergiert* gegen eine reelle Zahl a, wenn es zu jedem $\epsilon > 0$ einen Index n_ϵ gibt, so daß
>
> $$|a_n - a| < \epsilon \quad \text{für alle } n > n_\epsilon$$
>
> ist. Wir schreiben dann $a_n \to a$ oder auch $\lim_{n\to\infty} a_n = a$.
>
> Eine Reihe $\sum_{k=1}^{\infty} a_k$ bezeichnen wir als *konvergent*, wenn die Folge $(\sum_{k=1}^{n} a_k)_{n\in\mathbb{N}}$ ihrer Teilsummen konvergiert.

Beachten Sie bitte, daß auch der einzelne Buchstabe a in der zweiten Zeile eine eingebettete Formel ist.

LaTeX-Code, der eingebettete Formeln erzeugen soll, wird in `$`-Zeichen eingeschlossen. Das `$`-Zeichen schaltet zwischen dem Text- und dem mathematischen Modus von TeX (dem Programm, auf dem LaTeX basiert) um.

Um ein Gefühl für TeXs mathematischen Modus zu bekommen, schauen wir mal, wie TeX die folgenden Eingaben umsetzt:

`$1+1=2$`	$1+1=2$
`$1-1=0$`	$1-1=0$

Das klappt wie erwartet.

`$1 + 1 = 2$`	$1+1=2$

Leerzeichen, die im mathematischen Modus eingegeben werden, wirken sich also nicht auf die Ausgabe aus. Leerzeilen sind nicht erlaubt. (Warum sollte man auch mitten in einer Formel einen neuen Absatz beginnen wollen?)

`$g ist parallel zu h$`	$g ist parallel zu h$

Uups. Was ist hier passiert? Wenn im mathematischen Modus ein Buchstabe auftritt, so wird er von TEX als Variablenname aufgefaßt und in einem kursiven Schrifttyp in der normalen Schriftstärke gesetzt. TEX hat also unsere Eingabe als das Produkt der Variablen g, i, s, t, p, a, r, a, l, l, e, l, z, u und h interpretiert.

`$1+1=2:$ Das ist leicht!`	$1+1=2:$ Das ist leicht!

Der Doppelpunkt ist zu weit nach rechts gerückt. Das liegt daran, daß TEX den Doppelpunkt im mathematischen Modus als Relation und nicht als Satzzeichen auffaßt und dementsprechend etwas mehr Platz um ihn herum freiläßt. Besser ist es also, den Doppelpunkt außerhalb der `$`-Zeichen zu setzen:

`$1+1=2$: Das ist leicht!`	$1+1=2$: Das ist leicht!

Auch `.`, `!` und `?` werden von TEX im mathematischen Modus nicht als Satzzeichen verstanden.

☺ Ein guter Merksatz ist: „Satzzeichen gehören nicht in eine eingebettete Formel.“

10.2 Mathematische Symbole

Zurück zum (Beispiel-) Text: Wenn wir uns die eingebetteten Formeln, die darin vorkommen, noch einmal ansehen, stellen wir fest, daß wir nur von einer einzigen bisher wissen, wie wir sie erzeugen können.

a	`$a$`

Tabelle 6: Große griechische Buchstaben

A	`\mathrm{A}`	N	`\mathrm{N}`
B	`\mathrm{B}`	Ξ	`\Xi`
Γ	`\Gamma`	O	`\mathrm{O}`
Δ	`\Delta`	Π	`\Pi`
E	`\mathrm{E}`	P	`\mathrm{P}`
Z	`\mathrm{Z}`	Σ	`\Sigma`
H	`\mathrm{H}`	T	`\mathrm{T}`
Θ	`\Theta`	Υ, Y	`\Upsilon`, `\mathrm{Y}`
I	`\mathrm{I}`	Φ	`\Phi`
K	`\mathrm{K}`	X	`\mathrm{X}`
Λ	`\Lambda`	Ψ	`\Psi`
M	`\mathrm{M}`	Ω	`\Omega`

Diese Befehle funktionieren nur im mathematischen Modus.

Tabelle 7: Kleine griechische Buchstaben

α	`\alpha`	ν	`\nu`
β	`\beta`	ξ	`\xi`
γ	`\gamma`	o	`o`
δ	`\delta`	π, ϖ	`\pi`, `\varpi`
ϵ, ε	`\epsilon`, `\varepsilon`	ρ, ϱ	`\rho`, `\varrho`
ζ	`\zeta`	σ, ς	`\sigma`, `\varsigma`
η	`\eta`	τ	`\tau`
θ, ϑ	`\theta`, `\vartheta`	υ	`\upsilon`
ι	`\iota`	ϕ, φ	`\phi`, `\varphi`
κ	`\kappa`	χ	`\chi`
λ	`\lambda`	ψ	`\psi`
μ	`\mu`	ω	`\omega`

Diese Befehle funktionieren nur im mathematischen Modus.

Alle übrigen Formeln enthalten Symbole, die auf unserer Tastatur nicht vorhanden sind.

Aber keine Panik: Zum Erzeugen von Symbolen, für die es keine Taste gibt, gibt es in TEX und LATEX Befehle.

Zum Beispiel existieren Befehle zum Setzen griechischer Buchstaben. Diese stellen wir in den Tabellen 6 und 7 zusammen.

Mit Hilfe des Befehls `\epsilon` aus Tabelle 7 können wir nun schon eine weitere Formel aus unserem Beispieltext erzeugen:

$\epsilon > 0$	`$\epsilon>0$`

Alle übrigen mathematischen Symbole, die in LATEX 2_ε definiert sind, listen wir zusammen mit den Befehlen, durch die Sie diese erzeugen können, in Tabelle 8 auf (was es mit der dritten Spalte dieser Tabelle auf sich hat, erklären wir in Abschnitt 10.8).

Tabelle 8: Mathematische Symbole

Symbol	**Befehl**	**Klasse**
$.$	`.`	Ord
$\ldotp$	`\ldotp`	Punct
$\cdot$	`\cdot`	Bin
$\cdotp$	`\cdotp`	Punct
$/$	`/`	Ord
$\backslash$	`\backslash`	Ord
$\setminus$	`\setminus`	Bin
$\vert$	`|` oder `\vert`	Ord
$\mid$	`\mid`	Rel
$\Vert$	`\\|` oder `\Vert`	Ord
$\parallel$	`\parallel`	Rel
$\emptyset$	`\emptyset`	Ord
$\aleph$	`\aleph`	Ord
$\imath$	`\imath`	Ord
$\jmath$	`\jmath`	Ord
ℓ	`\ell`	Ord
$\wp$	`\wp`	Ord
$\Re$	`\Re`	Ord
$\Im$	`\Im`	Ord
∂	`\partial`	Ord
∞	`\infty`	Ord
$\prime$	`\prime`	Ord
$\top$	`\top`	Ord
$\bot$	`\bot`	Ord

Mathematische Symbole (Forts.)

Symbol	Befehl	Klasse
$\perp$	`\perp`	Rel
$\angle$	`\angle`	Ord
$\triangle$	`\triangle`	Ord
∇	`\nabla`	Ord
$\forall$	`\forall`	Ord
$\exists$	`\exists`	Ord
$\neg$	`\neg`	Ord
$\flat$	`\flat`	Ord
$\natural$	`\natural`	Ord
$\sharp$	`\sharp`	Ord
$\clubsuit$	`\clubsuit`	Ord
$\diamondsuit$	`\diamondsuit`	Ord
$\heartsuit$	`\heartsuit`	Ord
$\spadesuit$	`\spadesuit`	Ord
¶	`\mathparagraph`	Ord
§	`\mathsection`	Ord
$\$$	`\mathdollar`	Ord
$\sum$	`\sum`	Op
$\int$	`\int`	Op
$\oint$	`\oint`	Op
$\prod$	`\prod`	Op
$\amalg$	`\amalg`	Bin
$\coprod$	`\coprod`	Op
$\vee$	`\vee`	Bin
$\bigvee$	`\bigvee`	Op
$\wedge$	`\wedge`	Bin
$\bigwedge$	`\bigwedge`	Op
$\cap$	`\cap`	Bin
$\bigcap$	`\bigcap`	Op
$\cup$	`\cup`	Bin
$\bigcup$	`\bigcup`	Op
$\uplus$	`\uplus`	Bin
$\biguplus$	`\biguplus`	Op
$\sqcap$	`\sqcap`	Bin
$\sqcup$	`\sqcup`	Bin
$\bigsqcup$	`\bigsqcup`	Op
$\otimes$	`\otimes`	Bin
$\bigotimes$	`\bigotimes`	Op
$\oplus$	`\oplus`	Bin
$\bigoplus$	`\bigoplus`	Op
$\odot$	`\odot`	Bin

Mathematische Symbole (Forts.)

Symbol	Befehl	Klasse
$\bigodot$	`\bigodot`	Op
$\oslash$	`\oslash`	Bin
$\ominus$	`\ominus`	Bin
$\surd$	`\surd`	Op
$+$	`+`	Bin
$-$	`-`	Bin
$\pm$	`\pm`	Bin
$\mp$	`\mp`	Bin
$\div$	`\div`	Bin
$\times$	`\times`	Bin
$\ast$	`*` oder `\ast`	Bin
$\star$	`\star`	Bin
$\wr$	`\wr`	Bin
$\triangleleft$	`\triangleleft`	Bin
$\triangleright$	`\triangleright`	Bin
$\bigtriangleup$	`\bigtriangleup`	Bin
$\bigtriangledown$	`\bigtriangledown`	Bin
$\ddagger$	`\ddagger`	Bin
$\dagger$	`\dagger`	Bin
$\diamond$	`\diamond`	Bin
$\bullet$	`\bullet`	Bin
$\circ$	`\circ`	Bin
$\bigcirc$	`\bigcirc`	Bin
$<$	`<`	Rel
$>$	`>`	Rel
$\leq$	`\leq` oder `\le`	Rel
$\geq$	`\geq` oder `\ge`	Rel
$\gg$	`\gg`	Rel
$\ll$	`\ll`	Rel
$=$	`=`	Rel
$\neq$	`\neq` oder `\ne`	Rel
$\equiv$	`\equiv`	Rel
$\doteq$	`\doteq`	Rel
$\approx$	`\approx`	Rel
$\simeq$	`\simeq`	Rel
$\cong$	`\cong`	Rel
$\sim$	`\sim`	Rel
$\asymp$	`\asymp`	Rel
$:$	`:`	Rel
$\colon$	`\colon`	Punct
$\in$	`\in`	Rel

Mathematische Symbole (Forts.)

Symbol	**Befehl**	**Klasse**
$\ni$	`\ni`	Rel
$\notin$	`\notin`	Rel
$\supset$	`\supset`	Rel
$\subset$	`\subset`	Rel
$\supseteq$	`\supseteq`	Rel
$\subseteq$	`\subseteq`	Rel
$\sqsubseteq$	`\sqsubseteq`	Rel
$\sqsupseteq$	`\sqsupseteq`	Rel
$\propto$	`\propto`	Rel
$\dashv$	`\dashv`	Rel
$\vdash$	`\vdash`	Rel
$\models$	`\models`	Rel
$\bowtie$	`\bowtie`	Rel
$\smile$	`\smile`	Rel
$\frown$	`\frown`	Rel
$\succ$	`\succ`	Rel
$\prec$	`\prec`	Rel
$\succeq$	`\succeq`	Rel
$\preceq$	`\preceq`	Rel
$\leftarrow$	`\leftarrow` oder `\gets`	Rel
$\longleftarrow$	`\longleftarrow`	Rel
$\hookleftarrow$	`\hookleftarrow`	Rel
$\leftrightarrow$	`\leftrightarrow`	Rel
$\longleftrightarrow$	`\longleftrightarrow`	Rel
$\rightarrow$	`\rightarrow` oder `\to`	Rel
$\longrightarrow$	`\longrightarrow`	Rel
$\hookrightarrow$	`\hookrightarrow`	Rel
$\mapsto$	`\mapsto`	Rel
$\longmapsto$	`\longmapsto`	Rel
$\Leftarrow$	`\Leftarrow`	Rel
$\Longleftarrow$	`\Longleftarrow`	Rel
$\Leftrightarrow$	`\Leftrightarrow`	Rel
$\Longleftrightarrow$	`\Longleftrightarrow`	Rel
$\Rightarrow$	`\Rightarrow`	Rel
$\Longrightarrow$	`\Longrightarrow`	Rel
$\uparrow$	`\uparrow`	Rel
$\updownarrow$	`\updownarrow`	Rel
$\downarrow$	`\downarrow`	Rel
$\Uparrow$	`\Uparrow`	Rel
$\Updownarrow$	`\Updownarrow`	Rel
$\Downarrow$	`\Downarrow`	Rel

Mathematische Symbole (Forts.)

Symbol	**Befehl**	**Klasse**
$\nearrow$	`\nearrow`	Rel
$\searrow$	`\searrow`	Rel
$\nwarrow$	`\nwarrow`	Rel
$\swarrow$	`\swarrow`	Rel
$\leftharpoonup$	`\leftharpoonup`	Rel
$\leftharpoondown$	`\leftharpoondown`	Rel
$\rightleftharpoons$	`\rightleftharpoons`	Rel
$\rightharpoonup$	`\rightharpoonup`	Rel
$\rightharpoondown$	`\rightharpoondown`	Rel
$($	`(`	Open
$)$	`)`	Close
$\langle$	`\langle`	Open
$\rangle$	`\rangle`	Close
$\{$	`\{` oder `\lbrace`	Open
$\}$	`\}` oder `\rbrace`	Close
$[$	`[` oder `\lbrack`	Open
$]$	`]` oder `\rbrack`	Close
$\lceil$	`\lceil`	Open
$\rceil$	`\rceil`	Close
$\lfloor$	`\lfloor`	Open
$\rfloor$	`\rfloor`	Close
$!$	`!`	Close
$?$	`?`	Close
$,$	`,`	Punct
$;$	`;`	Punct

Diese Befehle funktionieren nur im mathematischen Modus.

Einige Pakete (wie z. B. `amssymb` oder `latexsym`) stellen Ihnen zusätzlich zu den in Tabelle 8 aufgeführten weitere mathematische Symbole zur Verfügung. Wenn Sie also in Tabelle 8 ein gewisses Symbol vermissen, könnte es in einem dieser Pakete definiert sein.

10.3 Hoch- und Tiefstellungen

In den Formeln unseres Beispieltextes tauchen Ausdrücke auf, die in einem kleineren Schrifttyp etwas höher oder tiefer gesetzt werden. Beispiele sind die Summationsgrenzen $k = 1$ und n sowie der Index k im Ausdruck $\sum_{k=1}^{n} a_k$.

Derartige Hoch- und Tiefstellungen werden in der Eingabedatei in der Form

⟨*Symbol*⟩`^{`⟨*Hochstellung*⟩`}`

bzw.

⟨*Symbol*⟩`_{`⟨*Tiefstellung*⟩`}`

angegeben.

Dies funktioniert nur im mathematischen Modus.

Wir sind nun in der Lage, weitere Formeln aus unserem Beispieltext zu erzeugen:

n_{ϵ}	`$n_{\epsilon}$`
$a_n \rightarrow \infty$	`$a_{n}\rightarrow\infty$`
$\sum_{k=1}^{\infty} a_k$	`$\sum_{k=1}^{\infty} a_{k}$`

10.4 Lange Funktionen- und Operatornamen

Versuchen wir, noch eine Formel unseres Beispieltextes zu setzen:

`$lim_{n\rightarrow\infty} a_{n}$` $lim_{n\rightarrow\infty}a_n$

Moment, das ging schief (vergleichen Sie das Ergebnis mit den Ausdruck $\lim_{n\rightarrow\infty} a_n$ aus unserem Beispieltext).

Es ist üblich, Funktionen- und Operatornamen, die aus mehreren Buchstaben bestehen, in einem aufrechten Schrifttyp zu setzen. Dadurch sollen Verwechslungen mit Produkten mehrerer Variabler vermieden werden.

In $\LaTeX\,2_\varepsilon$ sind eine Reihe langer Funktionen- und Operatornamen vordefiniert (siehe Tabelle 9). Wie Sie weitere Funktionen- und Operatornamen definieren können, erfahren Sie in Unterabschnitt 11.11.

Tabelle 9: Lange Funktionen- und Operatornamen

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\marg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

Diese Befehle funktionieren nur im mathematischen Modus.

Indem wir nun den Befehl `\lim` aus Tabelle 9 benutzen, erhalten wir:

`$\lim_{n\rightarrow\infty} a_{n}$` $\lim_{n\rightarrow\infty} a_n$

10.5 Variieren der Schriftart von Buchstaben

Wie können wir das Symbol $\mathbb{N}$ für die Menge der natürlichen Zahlen erzeugen? In LaTeX 2ε gibt es eine einfache Möglichkeit: Wir verändern die Schriftart des Buchstabens `N`.

Der Befehl

`\mathbb{`⟨*Großbuchstabe*⟩`}`

aus dem **`amsfonts`**-Paket setzt ⟨*Großbuchstabe*⟩ in einem „Blackboard Bold" Font. Taucht dabei ⟨*Großbuchstabe*⟩ in einem hoch- oder tiefgestellten Formelteil auf, so wird die Schriftgröße automatisch angepaßt.

Dieser Befehl funktioniert nur im mathematischen Modus.

Die beiden verbleibenden eingebetteten Formeln in unserem Beispieltext können wir also so erzeugen:

$(a_n)_{n\in\mathbb{N}}$	`$(a_{n})_{n\in\mathbb{N}}$`
$(\sum_{k=1}^{n} a_k)_{n\in\mathbb{N}}$	`$(\sum_{k=1}^{n} a_{k})_{n\in\mathbb{N}}$`

Es gibt noch eine ganze Reihe weiterer Befehle zum Verändern der Schriftart von Buchstaben im mathematischen Modus (siehe beispielsweise L. Lamport [9, S. 61f. oder S. 221]).

10.6 Abgesetzte Formeln

Formeln, die in einer eigenen Zeile stehen, werden als *abgesetzte Formeln* bezeichnet. In unserem Beispieltext gibt es nur eine abgesetzte Formel. Wir heben sie durch einen grauen Hintergrund hervor:

> Wir sagen, eine Folge $(a_n)_{n\in\mathbb{N}}$ reeller Zahlen *konvergiert* gegen eine reelle Zahl a, wenn es zu jedem $\epsilon > 0$ einen Index n_ϵ gibt, so daß
>
> $$|a_n - a| < \epsilon \quad \text{für alle } n > n_\epsilon$$
>
> ist. Wir schreiben dann $a_n \to a$ oder auch $\lim_{n\to\infty} a_n = a$.
>
> Eine Reihe $\sum_{k=1}^{\infty} a_k$ bezeichnen wir als *konvergent*, wenn die Folge $(\sum_{k=1}^{n} a_k)_{n\in\mathbb{N}}$ ihrer Teilsummen konvergiert.

Das `amsmath`-Paket stellt Ihnen die Umgebungen `equation` und `equation*` zum Setzen abgesetzter Formeln zur Verfügung. Die `*`-Form dieser Umgebung erzeugt im Gegensatz zur Form ohne `*` keine Formelnummer. Einfache Beispiele:

```
\begin{equation*}
  a^{2}+b^{2}=c^{2}
\end{equation*}
```

$$a^2 + b^2 = c^2$$

```
\begin{equation}
  a^{2}+b^{2}=c^{2}
\end{equation}
```

$$a^2 + b^2 = c^2 \qquad (1)$$

10.7 Text in Formeln

Die abgesetzte Formel enthält neben mathematischen Symbolen auch Text. Das `amsmath`-Paket stellt Ihnen den Befehl

`\text{`⟨*Text*⟩`}`

zum Setzen von Text in mathematischen Formeln zur Verfügung. Beispiel:

```
\begin{equation*}
  f(x)>0
  \quad
  \text{fast "uberall}
\end{equation*}
```

$$f(x) > 0 \quad \text{fast überall}$$

In diesem Beispiel ist der Befehl `\quad` für den Zwischenraum zwischen „0" und „fast" verantwortlich. Ohne diesen Befehl erhielte man:

$$f(x) > 0\text{fast überall}$$

Die Schriftgröße des Textes, den der Befehl `\text{`⟨*Text*⟩`}` erzeugt, wird automatisch angepaßt:

```
\begin{equation*}
  \sum_{\text{$n$ ungerade}}2^{-n}
\end{equation*}
```

$$\sum_{\text{n ungerade}} 2^{-n}$$

10.8 Klassen mathematischer Symbole

Jetzt ist es an der Zeit, Ihnen endlich zu erzählen, welche Bewandtnis es mit der dritten Spalte von Tabelle 8 hat. TEX teilt alle mathematischen Symbole in die folgenden Klassen ein:

Ord gewöhnliche Symbole;

Op große Operatorsymbole wie $\sum$;

Bin zweistellige Operatoren wie $+$ oder $-$;

Rel Relationen wie $<$, $>$ oder $=$;

Open öffnende Klammern;

Close schließende Klammern;

Punct Trennzeichen wie zum Beispiel die Kommas in (a, b, c).

Aufgrund der Zugehörigkeit eines Symbols zu einer dieser Klassen entscheidet TeX beispielsweise über den Abstand dieses Symbols zu den umgebenden Formelteilen, oder ob eine eingebettete Formel an einer bestimmten Stelle umgebrochen werden darf.

Wenn TeX in einer mathematischen Formel ausnahmsweise einmal völlig falsche Abstände erzeugt hat, sollten Sie überprüfen, ob die Befehle, die Sie verwendet haben, Symbole aus den richtigen Klassen erzeugen. Ist dies nicht der Fall, können Sie entweder in Tabelle 8 nachsehen, ob es nicht vielleicht einen Befehl gibt, der das gleiche Symbol in der richtigen Klasse liefert; oder Sie teilen TeX explizit mit, zu welcher Klasse das Symbol, das Sie setzen möchten, gehören soll. Hierzu dienen die Befehle

`\mathord{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als gewöhnliches Symbol zu setzen),

`\mathop{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als großes Operatorsymbol zu setzen),

`\mathbin{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als zweistelligen Operator zu setzen),

`\mathrel{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als Relation zu setzen),

`\mathopen{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als öffnende Klammer zu setzen),

`\mathclose{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als schließende Klammer zu setzen),

`\mathpunct{`⟨*Symbol*⟩`}` (um ⟨*Symbol*⟩ als Trennzeichen zu setzen).

Ein Beispiel: In der abgesetzten Formel in unserem Beispieltext kommt der Ausdruck

$$|a_n - a| < \epsilon$$

vor. Nehmen wir an, wir würden zum Setzen der Betragsstriche den Befehl `\mid` verwenden. Dann würden wir

$$\mid a_n - a \mid < \epsilon$$

erhalten (nicht schön, oder?). Der Befehl `\mid` dient zum Setzen des senkrechten Strichs in Formeln des Typs $a \mid b$ („a teilt b“). Dieser senkrechte Strich ist eine Relation, gehört also zur Klasse Rel. Für Betragsstriche könnten wir `|` direkt eingeben (das geht in diesem Fall gut, obwohl der durch `|` erzeugte Strich zur Klasse Ord gehört). Besser ist es allerdings, den linken Betragsstrich durch `\mathopen{|}` und den rechten durch `\mathclose{|}` zu setzen. Zum Setzen der Formel $\mathopen{|}-x\mathclose{|}=\mathopen{|}+x\mathclose{|}$ *müssen* wir `\mathopen{|}` und `\mathclose{|}` verwenden: Vergleichen Sie

```
$\mathopen{|}-x\mathclose{|}
  =\mathopen{|}+x\mathclose{|}$
```

$\mathopen{|}-x\mathclose{|}=\mathopen{|}+x\mathclose{|}$

und

```
$|-x|=|+x|$
```

$|-x|=|+x|$

Zum guten Schluß: Die abgesetzte Formel in unserem Beispieltext können wir folgendermaßen erzeugen:

$$\mathopen{|} a_{n}-a\mathclose{|} <\epsilon \quad \text{für alle } n>n_{\epsilon}$$

```
\begin{equation*}
  \mathopen{|} a_{n}-a\mathclose{|}
    <\epsilon
  \quad
  \text{f"ur alle $n>n_{\epsilon}$}
\end{equation*}
```

11 Noch mehr Mathematik

11.1 Brüche

Brüche kann man auf zwei Arten schreiben: mit einem schrägen Bruchstrich wie in 2/3 oder einem waagerechten Bruchstrich wie in $\frac{2}{3}$. In eingebetteten Formeln und in Hoch- und Tiefstellungen ist der schräge Bruchstrich zu bevorzugen.

Den schrägen Bruchstrich erzeugt man in LaTeX einfach durch das Zeichen `/`. Beispiel:

```
$2/3$
```

$2/3$

Obwohl ein schräger Bruchstrich aus mathematischer Sicht eigentlich zur Klasse Bin gehört, `/` aber einen Schrägstrich der Klasse Ord liefert, sollten Sie nicht `\mathbin{/}` schreiben. Andernfalls wird zuviel Leerraum um den Bruchstrich gesetzt (vgl. D. Knuth [7, S. 132]).

Einen Bruch mit einem waagerechten Bruchstrich erhält man mit dem Befehl `\frac{`⟨*Zähler*⟩`}{`⟨*Nenner*⟩`}`. Beispiel:

```
\begin{equation*}
  \frac{2}{3}
\end{equation*}
```

$$\frac{2}{3}$$

11.2 Klammersymbole in verschiedenen Größen

Ausdrücke wie

$$||x| - |y|| \leq |x - y|$$

oder

$$-((x(x+1) - (x+1))(x+1)),$$

die aus mehreren ineinander geschachtelten Formelteilen bestehen, sind oft schwer zu verstehen, da es nicht leicht ist, öffnende und schließende Klammern einander zuzuordnen. Hier ist es sinnvoll, Klammern unterschiedlicher Größe zu verwenden: Die Formeln

$$\bigl||x| - |y|\bigr| \leq |x - y|$$

und

$$-\Bigl(\bigl(x(x+1)-(x+1)\bigr)(x+1)\Bigr)$$

sehen schon sehr viel einfacher aus.

Um Klammersymbole zu erhalten, die größer als die standardmäßig erzeugten sind, stellt man öffnenden Klammern einfach einen der Befehle `\bigl`, `\Bigl`, `\biggl` oder `\Biggl` und schließenden Klammern `\bigr`, `\Bigr`, `\biggr` oder `\Biggr` voran.

```
\begin{equation*}
  \Biggl( \biggl( \Bigl( \bigl( (
  ) \bigr) \Bigr) \biggr) \Biggr)
\end{equation*}
```

$$\Biggl(\biggl(\Bigl(\bigl(( ) \bigr) \Bigr) \biggr) \Biggr)$$

Diese Befehle sorgen dann auch gleich dafür, daß die Klammersymbole, auf die sie wirken, zu den richtigen Klassen gehören (`\bigr|` bedeutet also dasselbe wie `\mathclose{\bigr|}`).

Es kann noch einen anderen Grund dafür geben, daß man größere Klammern erzeugen möchte: Die normalerweise erzeugten Klammern sind zu klein, um den Formelteil, der eingeklammert werden soll, zu umschließen. Klammern, die (mindestens) so groß sind wie der Formelteil, den sie umschließen, kann man erzeugen, indem man der öffnenden Klammer ein `\left` und der schließenden Klammer ein `\right` voranstellt.

```
\begin{equation*}
  2\left(
      \frac{
        \sum_{k=0}^{\infty}A^k
      }{
        \sum_{k=0}^{\infty}B^k}
      +1
   \right)
\end{equation*}
```

$$2\left(\frac{\sum_{k=0}^{\infty}A^k}{\sum_{k=0}^{\infty}B^k}+1\right)$$

Die Klammersymbole, auf die Sie die in diesem Unterabschnitt vorgestellten Befehle anwenden können, sind in Tabelle 10 zusammengestellt.

Der schräge Bruchstrich wirkt mitunter etwas dürftig, wenn er neben einer großen Klammer erscheint. Größere Bruchstriche erhalten Sie durch die Befehle `\big/`, `\Big/`, `\bigg/` und `\Bigg/`.

11.3 Mathematische Akzente

Mathematische Akzente (wie beispielsweise Tilden oder Dächer) können Sie durch die Befehle in Tabelle 11 erzeugen. Als Beispiel ist dort der Buchstabe a gewählt worden. Die Befehle funktionieren aber genauso mit jedem anderen Symbol.

Tabelle 10: Klammersymbole

<table>
<tr><td>(</td><td>$($</td><td>\lceil</td><td>$\lceil$</td></tr>
<tr><td>)</td><td>$)$</td><td>\rceil</td><td>$\rceil$</td></tr>
<tr><td>\{ oder \lbrace</td><td>$\{$</td><td>\lfloor</td><td>$\lfloor$</td></tr>
<tr><td>\} oder \rbrace</td><td>$\}$</td><td>\rfloor</td><td>$\rfloor$</td></tr>
<tr><td>\langle</td><td>$\langle$</td><td>\uparrow</td><td>$\uparrow$</td></tr>
<tr><td>\rangle</td><td>$\rangle$</td><td>\updownarrow</td><td>$\updownarrow$</td></tr>
<tr><td>| oder \vert</td><td>$\vert$</td><td>\downarrow</td><td>$\downarrow$</td></tr>
<tr><td>\| oder \Vert</td><td>$\Vert$</td><td>\Uparrow</td><td>$\Uparrow$</td></tr>
<tr><td>[oder \lbrack</td><td>$[$</td><td>\Updownarrow</td><td>$\Updownarrow$</td></tr>
<tr><td>] oder \rbrack</td><td>$]$</td><td>\Downarrow</td><td>$\Downarrow$</td></tr>
</table>

Diese Befehle funktionieren nur im mathematischen Modus.

Tabelle 11: Mathematische Akzente

$\widetilde{a}$	\widetilde{a}	$\check{a}$	\check{a}	$\ddot{a}$	\ddot{a}
$\tilde{a}$	\tilde{a}	$\acute{a}$	\acute{a}	$\breve{a}$	\breve{a}
$\widehat{a}$	\widehat{a}	$\grave{a}$	\grave{a}	$\bar{a}$	\bar{a}
$\hat{a}$	\hat{a}	$\dot{a}$	\dot{a}	$\vec{a}$	\vec{a}

Diese Befehle funktionieren nur im mathematischen Modus.

Wenn Sie einem i oder j einen Akzent aufsetzen wollen, stört der Punkt. Die Befehle `\imath` und `\jmath` erzeugen ein (punktloses) $\imath$ bzw. $\jmath$.

Sofern ihr Argument höchstens etwa drei Zeichen lang ist, liefern die Befehle `\widetilde{`⟨*Argument*⟩`}` und `\widehat{`⟨*Argument*⟩`}` eine Tilde bzw. ein Dach, das sich über die gesamte Breite von ⟨*Argument*⟩ erstreckt (beispielsweise ergibt `$\widetilde{abc}$` den Ausdruck $\widetilde{abc}$).

11.4 Formelsysteme

Zum Setzen von Formelsystemen, bei denen die einzelnen Formeln in mehreren Zeilen untereinander stehen, stellt Ihnen das `amsmath`-Paket u. a. die Umgebungen `gather`, `align` und `alignat` sowie `gather*`, `align*` und `alignat*` zur Verfügung. Die `*`-Formen dieser Umgebungen erzeugen im Gegensatz zu den Formen ohne `*` keine Formelnummern.

Im folgenden möchten wir diese Umgebungen anhand einfacher Beispiele vorstellen.

11.4.1 Formelsysteme ohne Ausrichtung

Die Umgebungen `gather` und `gather*` erzeugen Formelsysteme, bei denen die einzelnen Formeln zentriert in mehrereren Zeilen untereinander stehen. Zum Trennen der Zeilen dient der Befehl `\\`.

```
\begin{gather}
  a^{2}+b^{2}=c^{2}\\
  a<1
\end{gather}
```

$$a^2 + b^2 = c^2 \tag{2}$$
$$a < 1 \tag{3}$$

```
\begin{gather*}
  a^{2}+b^{2}=c^{2}\\
  a<1
\end{gather*}
```

$$a^2 + b^2 = c^2$$
$$a < 1$$

11.4.2 Formelsysteme mit Ausrichtung an einer Position

Die Umgebungen `align` und `align*` erzeugen Formelsysteme, bei denen die einzelnen Formeln in mehreren Zeilen untereinander stehen und zueinander an einer Position ausgerichtet werden. Wie bei der `gather`-Umgebung werden die Zeilen durch den Befehl `\\` getrennt. Die Position, an der die Zeilen ausgerichtet werden sollen, wird durch den Tabulator `&` markiert.

```
\begin{align}
  a^{2}+b^{2}&=c^{2}\\
  a&<1
\end{align}
```

$$\begin{aligned} a^2 + b^2 &= c^2 && (4)\\ a &< 1 && (5) \end{aligned}$$

```
\begin{align*}
  a^{2}+b^{2}&=c^{2}\\
  a&<1
\end{align*}
```

$$\begin{aligned} a^2+b^2 &= c^2 \\ a &< 1 \end{aligned}$$

11.4.3 Formelsysteme mit Ausrichtung an mehreren Positionen

Die Umgebungen `alignat` und `alignat*` erzeugen Formelsysteme, bei denen die einzelnen Formeln in mehreren Zeilen untereinander stehen und an mehreren Positionen zueinander ausgerichtet werden. Dies ist vor allem für Gleichungssysteme interessant.

Die Benutzung von `alignat` und `alignat*` ist relativ kompliziert und läßt sich wohl am einfachsten anhand eines Beispiels erklären:

```
\begin{alignat}{3}
    a&={}  &  3216b&+{}  &     13c& \\
  17a&={}  &      b&-{}  &   2353c&
\end{alignat}
```

$$\begin{alignedat}{3} a &= {} & 3216b &+ {} & 13c & \qquad (6) \\ 17a &= {} & b &- {} & 2353c & \qquad (7) \end{alignedat}$$

```
\begin{alignat*}{3}
    a&={}  &  3216b&+{}  &     13c& \\
  17a&={}  &      b&-{}  &   2353c&
\end{alignat*}
```

$$\begin{alignedat}{3} a &= {} & 3216b &+ {} & 13c & \\ 17a &= {} & b &- {} & 2353c & \end{alignedat}$$

Hier wird das Gleichungssystem durch den zweiten und vierten Tabulator in drei Blöcke aufgeteilt. Der erste Block wird am ersten Tabulator ausgerichtet, der zweite Block am dritten Tabulator und der dritte Block am fünften Tabulator.

Möchte man ein Formelsystem an n Positionen ausrichten, so teilt man es in n Blöcke auf. Die Umgebungen `alignat` und `alignat*` bekommen die Zahl n als Argument. Man darf dann insgesamt $2n-1$ Tabulatoren verwenden ($n-1$ Tabulatoren zum Trennen der Blöcke und in jedem Block einen, um daran auszurichten).

Im obigen Beispiel sind die `{}` wichtig, damit die Abstände um =, + und − richtig gesetzt werden. Ohne `{}` erhielte man:

$$\begin{alignedat}{3} a &= & 3216b &+ & 13c & \\ 17a &= & b &- & 2353c & \end{alignedat}$$

11.5 Formelnummern

Die Umgebungen `equation`, `gather`, `align` und `alignat` versehen jede Formel mit einer Formelnummer. Möchten Sie, daß eine bestimmte Formel nicht numeriert wird, so schreiben Sie

```
\notag
```

in den LaTeX-Code zur Erzeugung dieser Formel.

```
\begin{align}
  a^{2}+b^{2}&=c^{2}\notag\\
  a&<1
\end{align}
```

$$\begin{aligned} a^2 + b^2 &= c^2 \\ a &< 1 \qquad (8) \end{aligned}$$

Soll eine Formel nicht mit einer Nummer, sondern mit einer anderen Markierung (beispielsweise einem Stern) versehen werden, so schreiben Sie

`\tag{`⟨*Markierung*⟩`}`

in den LaTeX-Code zur Erzeugung dieser Formel.

```
\begin{equation}
  a^{2}+b^{2}=c^{2}\tag{$*$}
\end{equation}
```

$$a^2 + b^2 = c^2 \tag{$*$}$$

11.6 Referenzieren von Formeln

Mit Hilfe der Befehle

`\label{`⟨*Schlüssel*⟩`}`

und

`\eqref{`⟨*Schlüssel*⟩`}`

können Sie automatisch auf Formeln verweisen.

```
\begin{equation}
  a^{2}+b^{2}=c^{2}\tag{$\ast$}
  \label{eq:pythagoras}
\end{equation}
In \eqref{eq:pythagoras}
bezeichnen $a$ und $b$
die Katheten und $c$ die
Hypotenuse eines
rechtwinkligen Dreiecks.
```

$$a^2 + b^2 = c^2 \tag{$*$}$$

In $(*)$ bezeichnen a und b die Katheten und c die Hypotenuse eines rechtwinkligen Dreiecks.

11.7 Brechen abgesetzter Formeln

Anders als eingebettete Formeln, kann LaTeX 2_ε abgesetzte Formeln nicht selbsttätig brechen. Dies muß der/die Schreiber(in) des Textes selber tun. Das `amsmath`-Paket stellt ihm/ihr dazu die `split`-Umgebung zur Verfügung. Diese Umgebung funktioniert ähnlich wie die `align`-Umgebung: Die Stellen an denen eine Formel gebrochen werden soll, werden innerhalb der `split`-Umgebung durch den Befehl `\\` angegeben; und die entstehenden Formelzeilen werden an genau einer Position zueinander ausgerichtet, die durch das Tabulator-Zeichen `&` angegeben wird. Anders als die `align`-Umgebung kann die `split`-Umgebung jedoch nur innerhalb einer anderen Umgebung benutzt werden, die eine abgesetzte Formel oder ein Formelsystem erzeugt.

```
\begin{equation}
  \begin{split}
    f(x)&=a(a+1)x\\
        &=(a^{2}+a)x
  \end{split}
\end{equation}
```

$$\begin{split} f(x) &= a(a+1)x \\ &= (a^2+a)x \end{split} \tag{9}$$

Die `split`-Umgebung erzeugt selbst keine Formelnummer. Dies wird gegebenenfalls von der äußeren Umgebung (im Beispiel: `equation`) übernommen.

11.8 Satzzeichen in abgesetzten Formeln

Während man in eingebetteten Formeln keine Satzzeichen verwenden sollte, wird man dies in abgesetzten Formeln nicht umgehen können. Sie sollten dabei darauf achten, daß die Symbole, die Sie als Satzzeichen verwenden wollen, zur Klasse Punct gehören (Benutzen Sie also `\ldotp` statt `.` und `\colon` statt `:`).

11.9 Matrizen

Vom `amsmath`-Paket werden u. a. die folgenden Umgebungen zur Angabe einfacher Matrizen zur Verfügung gestellt: `matrix`, `pmatrix` und `vmatrix`. Mit diesen Umgebungen ist es möglich, Matrizen mit bis zu zehn Spalten und beliebig vielen Zeilen darzustellen.

Die Umgebungen unterscheiden sich in der Art der umschließenden Klammern, die sie erzeugen.

```
\begin{matrix}
  1 & 2\\
  3 & 4
\end{matrix}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

```
\begin{pmatrix}
  1 & 2\\
  3 & 4
\end{pmatrix}
```

$$\begin{pmatrix} 1 & 2\\ 3 & 4 \end{pmatrix}$$

```
\begin{vmatrix}
  1 & 2\\
  3 & 4
\end{vmatrix}
```

$$\begin{vmatrix} 1 & 2\\ 3 & 4 \end{vmatrix}$$

Der Tabulator `&` trennt die einzelnen Einträge einer Zeile. Zum Trennen der einzelnen Zeilen dient (wie üblich) der Befehl `\\`.

Neben diesen Umgebungen gibt es noch die `smallmatrix`-Umgebung, die eine kleine Version der Matrix erzeugt, die man durch die `matrix`-Umgebung erhalten würde.

```
\begin{smallmatrix}
  1 & 2\\
  3 & 4
\end{smallmatrix}
```

$$\begin{smallmatrix} 1 & 2\\ 3 & 4 \end{smallmatrix}$$

Die Umgebungen `matrix`, `pmatrix`, `vmatrix` und `smallmatrix` funktionieren nur im mathematischen Modus.

Die einzelnen Einträge der Matrizen werden horizontal und vertikal zentriert. Sollen sie links- oder rechtsbündig gesetzt werden, müssen sie Ihnen ein `\hfill` nach- bzw. voranstellen:

```
\begin{pmatrix}
  1.1 & -1\\
  1   & 1
\end{pmatrix}
```

$$\begin{pmatrix} 1.1 & -1\\ 1 & 1 \end{pmatrix}$$

```
\begin{pmatrix}
  1.1     & -1\\
  1\hfill & \hfill 1
\end{pmatrix}
```

$$\begin{pmatrix} 1.1 & -1\\ 1\hfill & \hfill 1 \end{pmatrix}$$

Zum Kennzeichnen von Auslassungen dienen die Befehle `\hdots`, `\vdots` und `\ddots`:

```
\begin{pmatrix}
  1      & 0      & \hdots & 0     \\
  0      & \ddots & \ddots & \vdots\\
  \vdots & \ddots & \ddots & 0     \\
  0      & \hdots & 0      & 1
\end{pmatrix}
```

$$\begin{pmatrix} 1 & 0 & \hdots & 0 \\ 0 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0 \\ 0 & \hdots & 0 & 1 \end{pmatrix}$$

Blockmatrizen können Sie setzen, indem sie die gerade vorgestellten Umgebungen ineinander schachteln:

```
\begin{pmatrix}
  \begin{matrix}
    2 & 1\\
    0 & 2
  \end{matrix}   & 0\\
  0              & \begin{matrix}
                     3 & 1\\
                     0 & 3
                   \end{matrix}
\end{pmatrix}
```

$$\begin{pmatrix} \begin{matrix} 2 & 1\\ 0 & 2 \end{matrix} & 0\\ 0 & \begin{matrix} 3 & 1\\ 0 & 3 \end{matrix} \end{pmatrix}$$

11.10 Fallunterscheidungen

Für Fallunterscheidungen stellt Ihnen das `amsmath`-Paket die Umgebung `cases` zur Verfügung.

```
\begin{equation*}
  f(x)
  =
  \begin{cases}
    1, &\text{falls $x>0$}\\
    0  &\text{sonst}
  \end{cases}
\end{equation*}
```

$$f(x) = \begin{cases} 1, & \text{falls } x>0\\ 0 & \text{sonst} \end{cases}$$

11.11 Definition weiterer Funktionen- und Operatornamen

Zusätzlich zu den in Tabelle 9 angegebenen langen Funktionen- und Operatornamen können Sie weitere Funktionen- und Operatornamen definieren. Am einfachsten funktioniert dies, wenn Sie das `amsmath`-Paket benutzen. Um einen neuen Befehl ⟨*Befehl*⟩ zu definieren, der den Funktionen- bzw. Operatornamen ⟨*Name*⟩ erzeugt, brauchen Sie dann nur

`\DeclareMathOperator{`⟨*Befehl*⟩`}{`⟨*Name*⟩`}`

in die Präambel der Eingabedatei zu schreiben.

```
\DeclareMathOperator{\arccot}{arccot}
...
\begin{equation*}
  \arccot^{\prime} x
  =
  -\frac{1}{1+x^{2}}
\end{equation*}
```

$$\operatorname{arccot}' x = -\frac{1}{1+x^2}$$

Bei einigen Operatoren sollen Hoch- und Tiefstellungen in abgesetzten Formeln nicht neben, sondern wie in

$$\lim_{n\to\infty}, \quad \max_{n=1}^{10}$$

über bzw. unter dem jeweiligen Operator erscheinen. Einen Befehl ⟨*Befehl*⟩, der einen derartigen Operatornamen ⟨*Name*⟩ erzeugt, können Sie mit dem Befehl

\DeclareMathOperator*{⟨*Befehl*⟩**}{**⟨*Name*⟩**}**

definieren.

```
\DeclareMathOperator*{\essinf}{essinf}
...
\begin{equation*}
  \essinf_{x>0}f(x)
\end{equation*}
```

$$\operatorname*{essinf}_{x>0} f(x)$$

Beachten Sie bitte, daß Sie die Befehle

\DeclareMathOperator{⟨*Befehl*⟩**}{**⟨*Name*⟩**}**

und

\DeclareMathOperator*{⟨*Befehl*⟩**}{**⟨*Name*⟩**}**

nur in der Präambel verwenden dürfen.