

L^AT_EX Crashkurs



Teil 3: Aufbaupaket MINT

Anja Wolffgramm
Freie Universität Berlin
Institut für Informatik

16. November 2018

Präsentationsfolien Diesen Foliensatz findet ihr online unter:
[http://www.mi.fu-berlin.de/stud/mentoring/
veranstaltungen-gesamt/veranstaltungen_fuev](http://www.mi.fu-berlin.de/stud/mentoring/veranstaltungen-gesamt/veranstaltungen_fuev) oder
<https://bit.ly/2qBSajG>

Livecoding-Dokument <https://bit.ly/2SZF6Sz>

- 1 Mathematikmodus
- 2 Kommandos & Umgebungen definieren
- 3 Quellcode einbinden
- 4 Anfertigung von Grafiken mit tikz
- 5 Abspann

Ist ein Modus, der dafür optimiert ist, mathematische Formeln und Symbole darzustellen.[14, 3, 23, 10, 22] Benötigte Pakete:

```

\usepackage{amsmath}      % vor fontspec laden!
\usepackage{mathtools}   % modifiziert amsmath
\usepackage{amssymb}     % mehr mathematische symbole
\usepackage{amsthm}      % für proof
\usepackage{mathrsfs}    % für \mathscr
\usepackage{latexsym}
\usepackage{marvosym}    % zusätzl. Zeichen (\Lightning)
\usepackage{cancel}      % zum Durchstreichen (\cancel)
  
```

**inline**

Im Fließtext wird der betreffende Ausdruck mit Dollarzeichen **\$** umgeben.

Beispiel

```
Mitten im Text steht $E = mc^2$ (Einstein)
```

Mitten im Text steht $E = mc^2$ (Einstein)

abgesetzt

Um einzeilige Formeln abzusetzen, bedient man sich der Mathematik-Umgebung: `\[... \]`.

Beispiel

```
\[ E = mc^2 \]
```

$$E = mc^2$$

Die **align**-Umgebung

- ▶ schaltet den Mathematikmodus ein,
- ▶ zentriert die Formel,
- ▶ erlaubt Zeilenumbrüche (`\\`),
- ▶ nummeriert die Zeilen und
- ▶ ermöglicht die Ausrichtung der Zeilen zueinander (mittels **&**).

Beispiel

```
\begin{align}
(a+b)^2 &=(a+b)(a+b) \\
&=a^2+2ab+b^2
\end{align}
```

$$(a+b)^2 = (a+b)(a+b) \quad (1)$$

$$= a^2 + 2ab + b^2 \quad (2)$$

Hinweis

Will man keine Nummerierung, nutzt man **align***



Dieser gibt normalerweise Probleme. Mit folgendem Trick[1] klappt es:

```
\subsubsection{\texorpdfstring{Something with  $\beta$ 
in it.}{Something with beta in it.}}
```



Möchte man Text in Mathematikmodus schreiben, geht dies wie folgt:

```
\[n=2k\implies n\text{ gerade}\]
```

$n = 2k \implies n \text{ gerade}$

Manche Funktionalitäten unterscheiden sich im Text- und Mathemodus:

Befehl	Textmodus	Mathematikmodus
Unterstrich	<code>_</code>	<code>_</code>
Dach	<code>\^{}</code>	<code>\^{}</code>
hochstellen	<code>\textsuperscript</code>	<code>^</code>
Tabelle	<code>tabular</code>	<code>array</code>
Bold	<code>\textbf{}</code>	<code>\mathbf{}</code>
Backslash	<code>\textbackslash</code>	<code>\backslash</code>

Exponent

```
a^2 \cdot x^{n+1}
```

$$a^2 \cdot x^{n+1}$$

Indicés

```
x_1 + \ldots + x_{k+1}
```

$$x_1 + \dots + x_{k+1}$$

Quantoren

```
\forall x \in K  
\exists y \in K:  
x \cdot y = x
```

$$\forall x \in K \exists y \in K : x \cdot y = x$$

Log. Operat.

```
\neg x \wedge (y \vee z)  
\implies x
```

$$\neg x \wedge (y \vee z) \implies x$$

Ungleichungen

```
x \leq \sqrt[4]{x} + 42
```

$$x \leq \sqrt[4]{x} + 42$$

Funktion

```
f\colon\mathbb{R}^2
\to\mathbb{N}
```

 $f: \mathbb{R}^2 \rightarrow \mathbb{N}$

Sinus

```
\sin\alpha
```

 $\sin \alpha$

Bruch

```
\frac{x+1}{2}
```

 $\frac{x+1}{2}$

Summe

```
\sum_{i=0}^{n+1} q^i
```

 $\sum_{i=0}^{n+1} q^i$

Produkt

```
\prod\limits_{i=1}^n i=n!
```

 $\prod_{i=1}^n i = n!$

Integral

```
\int\limits_0^{\infty} e^x dx
```

$$\int_0^{\infty} e^x dx$$

Vereinigung

```
\bigcup\limits_{i \in I} A_i
```

$$\bigcup_{i \in I} A_i$$

Limes

```
\lim_{n \to \infty} \frac{1}{n} = 0
```

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

Vektor

```
\vec{v}
```

$$\vec{v}$$

Binom

```
\binom{n}{k}
```

$$\binom{n}{k}$$

Pfeil `x \rightarrow y` $x \rightarrow y$

langer Pfeil `x \longrightarrow y` $x \longrightarrow y$

Doppelpfeil `x \Rrightarrow y` $x \rightrightarrows y$

Abbild `x \mapsto y` $x \mapsto y$

overset `x \overset{(1)}{\Longrightarrow} y` $x \overset{(1)}{\Longrightarrow} y$

underset `x \underset{(1)}{\Longrightarrow} y` $x \underset{(1)}{\Longrightarrow} y$

Beispiel

$$\left(\int_1^9 \frac{x}{2} dx\right)^2 \rightarrow \text{schlecht}$$

$$\left(\int_1^9 \frac{x}{2} dx\right)^2 \rightarrow \text{gut}$$

Darum werden Klammern wie folgt geschrieben:

`\left(... \right)` (...)

`\lceil ... \rceil` [...]

`\left[... \right]` [...]

`\langle ... \rangle` <...>

`\lfloor ... \rfloor` [...]

`\left\{ ... \right\}` {...}

Beispiel

$$a_0 + (x - x_0)(a_1 + (x - x_1)(\dots (a_{n-1} + (x - x_{n-1})a_n) \dots)) \rightarrow \text{schlecht}$$

$$a_0 + (x - x_0) \left(a_1 + (x - x_1) \left(\dots (a_{n-1} + (x - x_{n-1}) a_n) \dots \right) \right) \rightarrow \text{gut}$$

Darum kann man Klammern manuell skalieren:

`(\big(\Big(\bigg(\Bigg(`



Eine Fallunterscheidung erhält man mit der **cases** Umgebung:

```
\[ f(x) =
  \begin{cases}
    1, & \text{falls } x \bmod 2 = 0 \\
    0, & \text{sonst} \\
  \end{cases} \quad \text{for all } x \in \mathbb{R}
\]
```

$$f(x) = \begin{cases} 1, & \text{falls } x \bmod 2 = 0 \\ 0, & \text{sonst} \end{cases} \quad \forall x \in \mathbb{R}$$

► Array

```

\begin{array}{l|cr}
& x_0 & x_1 & \\\ \hline
y_0 & 1 & 0 & \\\
y_1 & 0 & 1 & \\\
\end{array}

```

	x_0	x_1
y_0	1	0
y_1	0	1

► Matrizen

```

\begin{pmatrix}
x_{1,1} & \dots & x_{1,n} \\
\vdots & \ddots & \vdots \\
x_{m,1} & \dots & x_{m,n}
\end{pmatrix}

```

$$\begin{pmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{pmatrix}$$

Weitere Matrixtypen sind:

bmatrix $\left[\dots \right]$

Bmatrix $\left\{ \dots \right\}$

vmatrix $\left| \dots \right|$

Vmatrix $\left\| \dots \right\|$

Hinweis

Mit Hilfe des **mathtools** Pakets können unter Angabe des * die Spaltenausrichtung modifiziert werden. Beispiel:

```
\begin{pmatrix*}[r]
  x_1 & x_2 \\
  0 & 1
\end{pmatrix*}
```

```
\begin{theorem}[<Name>]
...
\end{theorem}
```

```
\begin{corollary}
...
\end{corollary}
```

```
\begin{lemma}
...
\end{lemma}
```

Theorem (Pythagoras)

This is a theorem about right triangles and can be summarised in the next equation

$$x^2 + y^2 = z^2$$

Corollary

There's no right rectangle whose sides measure 3cm, 4cm, and 6cm.

Lemma

Given two line segments whose lengths are a and b respectively there is a real number r such that $b = r \cdot a$.

```
\begin{proof}
...
\end{proof}
```

```
\begin{definition}
...
\end{definition}
```

Beweis.

Mit $r = \frac{b}{a}$ gilt stets $b = r \cdot a$. □

Definition

$$fak(n) := \begin{cases} 1, & n = 0 \\ n \cdot fak(n-1), & \text{sonst} \end{cases} \quad \forall n \in \mathbb{N}$$

Wie kann man Einheiten darstellen?

Beispiel

$$v = 3.14km/h \longrightarrow \text{schlecht}$$

$$v = 3,14 \frac{\text{km}}{\text{h}} \longrightarrow \text{gut}$$

Außerdem möchte man leicht zwischen deutscher und englischer Zahlendarstellung umstellen.

Das Paket **siunitx**[24][18] bietet eine angenehme Handhabung von Zahlen und Einheiten.

Vorab stellen wir es auf deutsche Verwendung ein:

```
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {,},      % Dezimalzeichen
  per-mode = fraction,              % km/s als Bruch
  list-final-separator = { und },   % Trennungszeichen
  list-pair-separator = { und },    % ... in Listen
  range-phrase = { bis },
}
```



Zahlen können wie folgt dargestellt werden:

<code>\num{.1234}</code>	0,1234
<code>\num{3,141529}</code>	3,141 529
<code>\num{3.45d-4}</code>	$3,45 \times 10^{-4}$
<code>\num{-e10}</code>	-10^{10}

Es können auch Zahlenlisten oder Bereiche angegeben werden:

<code>\numlist{10;30;50;70}</code>	10, 30, 50 und 70
<code>\numrange{10}{30}</code>	10 bis 30

Zahlen mit Einheiten lassen sich wie folgt darstellen:

```
\SI{<ZAHL>}{<EINHEIT>}
```

Es gibt u.a. folgende vordefinierte Einheiten und Präfixe:

<code>\ampere</code>	A	<code>\metre</code>	m
<code>\coulomb</code>	C	<code>\micro</code>	μ
<code>\degree</code>	°	<code>\mega</code>	M
<code>\degreeCelsius</code>	°C	<code>\metre, \milli</code>	m
<code>\hertz</code>	Hz	<code>\mole</code>	mol
<code>\hour</code>	h	<code>\nano</code>	n
<code>\giga</code>	G	<code>\newton</code>	N
<code>\kilo</code>	k	<code>\ohm</code>	Ω
<code>\kilogram</code>	kg	<code>\second</code>	s

Es lassen sich auch Bereiche mit Einheit leicht angeben:

```
\SIlist{10;30;45}{\micro\second} 10  $\mu$ s, 30  $\mu$ s und 45  $\mu$ s
```

```
\SIrange{10}{30}{\micro\second} 10  $\mu$ s bis 30  $\mu$ s
```


Es lassen sich Einheiten auch selbst definieren:

```
\DeclareSIUnit{\NAME}{DEFINITION}
```

Beispiel

Die Definition:

```
\DeclareSIUnit{\kmh}{\kilo\meter\per\hour}
```

Die Anwendung:

```
\SI{1,079e+9}{\kmh}
```

So sieht es aus:

$$1,079 \times 10^9 \frac{\text{km}}{\text{h}}$$

Möchte man im Dokument zwischen verschiedenen Sprachen hin- und her schalten, gibt es die folgende Möglichkeit für **siunitx**:

```

\documentclass[english, ngerman]{scrartcl}
\usepackage{babel}% Sprachen global laden
\usepackage{translator}
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {,},% Dezimalzeichen
  list-final-separator = { \translate{and} },
  list-pair-separator   = { \translate{and} },
  range-phrase         = { \translate{to (numerical range)} },
}

```

Anschließend wechselt man wie folgt:

```

\selectlanguage{english}
\SIlist[locale=UK]{3.2;4.4;5.5}{\kmh}

\selectlanguage{ngerman}
\SIlist{3.2;4.4;5.5}{\kmh}% locale=DE voreingestellt
  
```

siunitx bietet folgende Sprachen: UK, US, DE (Germany), FR (French) and ZA (South Africa)

- ▶ $\backslash\mathrm{normal}\{ABCDEF abcdef 0123456789\}$
- ▶ $\backslash\mathrm{rm}\{ABCDEF abcdef 0123456789\}$
- ▶ $\backslash\mathrm{mathit}\{ABCDEF abcdef 0123456789\}$
- ▶ $\backslash\mathrm{mathbf}\{\mathbf{ABCDEF abcdef 0123456789}\}$
- ▶ $\backslash\mathrm{mathsf}\{ABCDEF abcdef 0123456789\}$
- ▶ $\backslash\mathrm{mathtt}\{ABCDEF abcdef 0123456789\}$
- ▶ $\backslash\mathrm{mathfrak}\{\mathfrak{ABCDEF abcdef 0123456789}\}$
- ▶ $\backslash\mathrm{mathcal}\{\mathcal{ABCDEF}\}$
- ▶ $\backslash\mathrm{mathbb}\{\mathbb{ABCDEF}\}$
- ▶ $\backslash\mathrm{mathscr}\{\mathscr{ABCDEF}\}$

(benötigt **mathrsfs**)

Im Mathemodus sind die Abstände[4] nicht immer optimal. Dies lässt sich manuell korrigieren:

<code>\!</code>	$f(x) = x^2 + 3x + 2$	(= -3 mu)
	$f(x) = x^2 + 3x + 2$	(= 1 mu)
<code>\,</code>	$f(x) = x^2 + 3x + 2$	(= 3 mu)
<code>\:</code>	$f(x) = x^2 + 3x + 2$	(= 4 mu)
<code>\;</code>	$f(x) = x^2 + 3x + 2$	(= 5 mu)
<code>\</code>	$f(x) = x^2 + 3x + 2$	[Backslash+Space] (= Space)
<code>\quad</code>	$f(x) = x^2 + 3x + 2$	(= 18 mu)
<code>\quad\quad</code>	$f(x) = x^2 + 3x + 2$	(= 36 mu)

Hinweis

1mu ist die Längeneinheit des Mathemodus und ist äquivalent zu 1/18 em, wessen Breite von der Mathesymbolfamilie abhängt.



Man kann sich seine eigenen Operatoren erstellen. Dazu gibt es zwei Möglichkeiten:

```
\DeclareMathOperator*{\meineOp}{\r{a}} % wird in der
% Präambel definiert, ermöglicht \limits_{..}
\newcommand{\meineOp}{\operatorname{\r{a}}}
% kann überall definiert werden
```

Beispiel

```
\meineOp(x) = 2^x + 1
\meineOp_{i=0} x_i
```

So sieht es aus

$$\mathring{a}(x) = 2^x + 1 \quad (3)$$

$$\mathring{a}_{i=0}^N x_i \quad (4)$$

Man kann sich eigene Kommandos[9] erstellen, um z. B. Schreibaufwand oder ständiges Kopieren einzusparen.

```
\newcommand{\<Name>}{<Was es tun soll>}
```

Beispiel

```
\newcommand{\zz}{\ensuremath{\raisebox{+0.25ex}{Z}}%
  \kern-0.4em\raisebox{-0.25ex}{Z}}%
  \; \xspace}%
}
```

Und so sieht es aus: \mathbb{Z}



Kommandos können auch Argumente erhalten:

```
\newcommand{\<Name>}[<Argumentzahl>]{<Was es mit dem  
Argument #1 tun soll>}
```

Beispiel

```
\newcommand{\Notiz}[1]{%  
  \begin{center}\begin{minipage}{0.75\linewidth}  
    \setlength{\fboxrule}{1.5pt}% Rahmenbreite  
    \fcolorbox{green!40!yellow}{green!40!yellow!40}%  
      {\parbox{\dimexpr\linewidth-2\fboxsep}{#1}}%  
  \end{minipage}\end{center}%  
}
```

meine Notiz

Es wird ein Default-Wert angegeben:

```
\newcommand{\<Name>}[<Argumentzahl>][<Defaultwert>]{%
  was mit den Argumenten #1,#2,... getan werden soll}
```

Beispiel

```
\newcommand{\binomPlusExp}[3][2]{\ensuremath{
  \left(#2 + #3 \right)^{#1}}}
```

In der Anwendung:

$$\backslash\text{binomialPlus}\{x\}\{3\} \implies (x + 3)^2$$

$$\backslash\text{binomialPlus}[5]\{x\}\{3\} \implies (x + 3)^5$$

Achtung

Das Optionale Argument muss das erste sein: **#1**



Man kann eine eigene Umgebung[12] erstellen – oder eine bereits vorhandene modifizieren:

```
\newenvironment{<Name>}  
  {<Befehlsbeginn>}  
  {<Befehlsende>}
```

Beispiel

```
\newenvironment{Aufgabenstellung}  
  {\textbf{Aufgabenstellung:}\begin{itshape}}  
  {\end{itshape}}
```

So sieht es aus:

Aufgabenstellung: *Schauen Sie sich die Referenzen an!*



Auch Umgebungen können (optionale) Argumenten erhalten.

Achtung

Allerdings kann das Befehlssende nicht auf Argumente zugreifen:

```
\newenvironment{<Name>}[<Argumentzahl>]%
  {some code #1} % allowed
  {some code #1} % not allowed
```

Ein Trick ist es, ein globales Kommando als Platzhalter zu benutzen:

```
\newcommand{\Listenart}{itemize}
\newenvironment{meineListe}[2][
  {\renewcommand{\Listenart}{#2}}
  \begin{\Listenart}[leftmargin=2em, labelsep=0.5em,
  itemindent=0em, labelwidth=1.5em, #1]}
  {\end{\Listenart}}
```

Wenn Text nicht vom \LaTeX -Compiler interpretiert werden soll, kann man ihn in ein **verb**-Befehl setzen:

```
\verb| nicht interpretierter Text |
```

Dabei muss dem Befehl mitgeteilt werden, wann dies endet. Dies geschieht durch ein Zeichen, das den Bereich einleitet und abschließt.

Das Paket **verbatim** bietet eine Umgebung für mehrere Zeilen:

```
\usepackage{verbatim}
\begin{document}
  \begin{verbatim}
    diese Zeilen sollen nicht interpretiert
    werden # _ & \ -- % Kommentar
  \end{verbatim}
\end{document}
```

Um Programmcode in das Dokument einzubinden, möchte man sprachspezifisches Syntax-Highlighting und Zeilennummerierungen haben. Dafür gibt es das Paket **listings**[5]:

```
\begin{lstlisting}[language=<Sprache>,  
caption={<Beschriftung>}]  
<Quellcode>  
\end{lstlisting}
```

Es werden sehr viele Programmiersprachen unterstützt. Einige wenige davon sind:

- | | | | |
|-----------|----------|----------|--------------------|
| ▶ bash | ▶ HTML | ▶ Octave | ▶ R |
| ▶ C, C++ | ▶ Java | ▶ PHP | ▶ SQL |
| ▶ Haskell | ▶ Matlab | ▶ Python | ▶ T _E X |

Nun müssen noch weitere Einstellungen vorgenommen werden:

```

\lstset{% in der Präambel
  basicstyle=\small\selectfont,           % Schriftgröße
  backgroundcolor = \color{lightgray},% Farbbox
  commentstyle=\color{blue}\ttfamily,    % Kommentarstil
  frame=single,                           % Rahmen
  keywordstyle=\bfseries\color{blue},    % Keyword Stil
  numbers=left,                            % Zeilennummern
  numberstyle=\tiny\color{green},        % Zeilennr. Stil
  stringstyle=\color{orange}\ttfamily,% String Stil

```

Beispiel:

```
\begin{lstlisting}[language=Haskell,
  caption={meine Haskell Funktion}]
fak :: [Integer] -> [Integer]
fak 0 = 1
fak n = n * fak(n-1)
\end{lstlisting}% lstlisting
```

Und so sieht der Haskell-Code aus:

```
1 fak :: [Integer] -> [Integer]
2 fak 0 = 1
3 fak n = n * fak(n-1)
```

Source Code 1: meine Haskell Funktion

In der Regel programmiert man in einer Datei und möchte Teile dieser im \LaTeX -Dokument einbinden. Dies geht mittels **lstinputstring**:

```
\lstinputstring [language=<Sprache>,
caption={<Beschriftung>}] {<Pfad>/<Dateiname>.<Typ>}
```

Wenn nur einige Zeilen eingebunden werden sollen:

linerange $\langle z_1 \rangle$ - $\langle z_2 \rangle$ Bereich im Quellcode

firstline $\langle z \rangle$ Beginn des einzubindenden Quellcodes

lastline $\langle z \rangle$ Ende des einzubindenden Quellcodes

firstnumber $\langle z \rangle$ Beginn der Zeilen-Nummerierung



Eine Liste der Listings (LOL) wird mit dem Befehl `\lstlistoflistings`

Das Paket **tikz**[2] ermöglicht das Zeichnen. Allerdings muss man sich an eine teilweise neue Syntax und einige neue Begriffe gewöhnen.

Eine **tikz** Grafik ist wie folgt aufgebaut:

```
\begin{tikzpicture}[<Optionen>
  <Befehle zum Zeichnen>
\end{tikzpicture}
```

Hinweis

Besonders schön machen sich Grafiken in einer **figure**-Umgebung.



```
\draw[line width=5pt, draw=red]
(0,0) -- (2,0);
```



```
\draw[line width=1.5pt, ->]
(0, 0) -- (1.5, 0);
```



```
\draw[line width=1.5pt, fill=green]
(0, 0) rectangle (2, 1);
```



```
\draw[line width=1.5pt]
(0, 0) circle (0.5);
```



```
\draw[line width=1.5pt]
(0, 0) ellipse (1 and 0.5);
```



```
\draw[line width=1.5pt]
(0, 0) arc (45:270:0.5);
```



```
\draw[line width=1.5pt] (0:2ex) --
(60:2ex) -- (120:2ex) -- (180:2ex) --
(240:2ex) -- (300:2ex) -- cycle;
```



```
\draw[line width=1.5pt]% relative Koord.
(0, 0) -- ++( 1, 0) -- ++(0, 1)
-- ++(-1, 0) -- ++(0, -1);
```



```
\draw[line width=1.5pt]
(0, 0) -- ++( 45:1) -- ++(135:1)
-- ++(225:1) -- ++(315:1);
```

$$l \frac{a}{b} r$$

```
\draw[line width=1pt] (1, 0)
  node[left] {$l$} --
  node[above]{$a$}
  node[below]{$b$} (2, 0)
  node[right]{$r$};
```

Nachfolgend wird das **shapes** Paket benötigt:

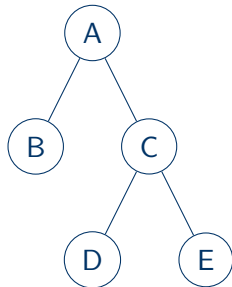


```
\node[ellipse, fill=yellow]
  (abc) at (0, 1) {Abc};
```

Ein Beispiel für einen Baum[7]

```
\begin{tikzpicture}[% Optionen
every node/.style={shape=circle,
draw}]
  \node {A} % Wurzel
    child {node {B}}
    child {node {C}
      child {node {D}}
      child {node {E}}
    };
\end{tikzpicture}
```

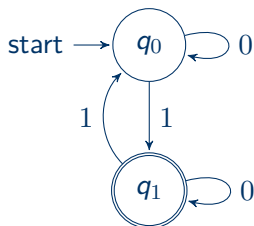
Abbildung 1: Ein Baum



Für Automaten[20, 6] benötigt man zusätzlich die Pakete **automata** und **arrows**.

```
\begin{tikzpicture}[>=stealth',
shorten >=1pt, transform shape,
node distance=5em, auto]
% zeichnet die Zustände
\node[state,initial] (A){$q_0$};
\node[state,accepting] (B)
[below of=A] {$q_1$};
\path[->] % zeichnet die Pfeile
(A)edge node{$1$} (B)
(A)edge[loop right] node{$0$}(A)
(B)edge[loop right] node{$0$}(B)
(B)edge[bend left=45]
node[left]{$1$}(A);
\end{tikzpicture}
```

Abbildung 2: Ein Automat



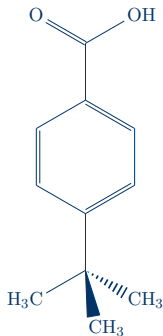
Für Strukturformeln benötigt man zusätzlich das Paket **chemfig**[8].

```
\begin{tikzpicture}
\chemfig{O=[:-30](-[:30]OH)-[:90]
*6(---(-[:90](-[:210]H_3C)
(<[:80]CH_3)<[:30]CH_3)---)}
\end{tikzpicture}
```

Hinweis

Wenn die Grafik zu groß wird, eignet sich die `\scalebox{<Zahl>}{<Objekt>}`, wobei die Zahl größer als Null sein muss.

Abbildung 3: PTBBA
Strukturformel



Templates Um euch den Einstieg etwas zu erleichtern, haben wir für euch Templates erstellt. Dieses könnt ihr für die Abgabe von Übungszetteln oder zum Schreiben von Skripten benutzen.

▶ **Mathe/(Bio-)Informatik**

<https://www.overleaf.com/read/xrxnqqnjccd>

▶ **Pyhsik** <https://www.overleaf.com/read/pvmmgzchwysx>

▶ **Präsentationsfolien**

<https://www.overleaf.com/read/mpspghtzsfns>

Weitere Templates[13] für verschiedene Anwendungen gibt es online zu finden.

- ▶ Die Freie Universität Berlin bietet einen L^AT_EX-ABV-Kurs an:
<http://latex.userpage.fu-berlin.de/>
- ▶ Des weiteren gibt es eine Einstiegshilfe in L^AT_EX[11] von der Studienberatung der Freien Universität Berlin.
- ▶ Weiterführende Informationen für Fortgeschrittene gibt es auf den Seiten der TU Graz.[18]
- ▶ Grundlegendes zum Schreiben von Bachelor- und Masterarbeiten hat Matthias Pospiech[16] bereit gestellt.

- [1] **Samuel Albert.** *How do I put special characters in a section title?*. Okt. 2012. URL:
<https://tex.stackexchange.com/questions/77657/how-do-i-put-special-characters-in-a-section-title> (besucht am 14. 11. 2018).

- [2] **Jacques Crémer.** *A very minimal introduction to TikZ**. Techn. Ber. März 2011. URL:
<http://cremeronline.com/LaTeX/minimaltikz.pdf> (besucht am 15. 11. 2018).

- [3] **Matthias Heinkenschloss.** *L^AT_EX Mathematical Symbols*. URL:
<http://www.caam.rice.edu/~heinken/latex/symbols.pdf> (besucht am 14. 11. 2018).

- [4] **Matthias Heinkenschloss.** *Spacing in math mode.* 2018. URL: https://www.overleaf.com/learn/latex/Spacing_in_math_mode (besucht am 13. 11. 2018).
- [5] **Jobst Hoffmann, Brooks Moses und Carsten Heinz.** *The Listings Package.* Techn. Ber. Version 1.7. Sep. 2018. URL: <ftp://ftp.fu-berlin.de/tex/CTAN/macros/latex/contrib/listings/listings.pdf> (besucht am 14. 11. 2018).
- [6] **Ichibann und Stefan Kottwitz.** *Drawing Graphs in L^AT_EX.* Juni 2012. URL: <http://tex.stackexchange.com/questions/45734/drawing-graphs-in-latex> (besucht am 14. 11. 2018).

- [7] **Stefan Kottwitz.** *Example: A simple Tree.* Aug. 2015. URL: <http://www.texample.net/tikz/examples/tree/> (besucht am 14. 11. 2018).
- [8] **o.A.** *L^AT_EX/Chemical Graphics.* Aug. 2018. URL: https://en.wikibooks.org/wiki/LaTeX/Chemical_Graphics (besucht am 14. 11. 2018).
- [9] **o.A.** *Commands.* URL: <https://www.overleaf.com/learn/latex/Commands> (besucht am 14. 11. 2018).
- [10] **o.A.** *T_EX Cookbook.* Techn. Ber. Nov. 1989. URL: <https://www2.stat.duke.edu/~fei/Tutorials/TeX%2520cookbook.pdf> (besucht am 14. 11. 2018).

- [11] **o.A.** *Einstiegshilfe in L^AT_EX*. Sep. 2018. URL:
https://de.wikibooks.org/wiki/LaTeX-W%C3%B6rterbuch:_Silbentrennung (besucht am 09. 11. 2018).
- [12] **o.A.** *Environments*. URL:
<https://www.overleaf.com/learn/latex/Environments> (besucht am 14. 11. 2018).
- [13] **o.A.** *LaTeX Templates*. 2018. URL:
<https://www.latextemplates.com> (besucht am 01. 11. 2018).
- [14] **o.A.** *L^AT_EX/Mathematics*. Okt. 2018. URL:
<https://en.wikibooks.org/wiki/LaTeX/Mathematics> (besucht am 14. 11. 2018).
- [15] **o.V. o.T..** Webseite. TikZ examples. URL:
<http://www.texample.net> (besucht am 12. 05. 2017).

- [16] **Matthias Pospiech.** *Erstellung von Bachelor und Masterarbeiten mit \LaTeX .* Webseite. Dez. 2011. URL: <http://www.matthiaspospiech.de/blog/2011/12/09/erstellung-von-bachelor-und-masterarbeiten-mit-latex/> (besucht am 09. 11. 2018).
- [17] **Matthias Pospiech.** *\LaTeX Variablen, If Abfragen und Schleifen.* Apr. 2008. URL: <http://www.matthiaspospiech.de/blog/2008/04/13/latex-variablen-if-abfragen-und-schleifen/> (besucht am 15. 11. 2018).
- [18] **Thomas Quaritsch.** *Anwendung für Fortgeschrittene.* Nov. 2013. URL: <http://latex.tugraz.at/latex/fortgeschrittene> (besucht am 01. 11. 2018).

- [19] **Andrew Stacey.** *The tikzmark package.* Webseite. Version 1.2. Apr. 2016. URL: <http://ctan.mackichan.com/graphics/pgf/contrib/tikzmark/tikzmark.pdf> (besucht am 12. 05. 2017).
- [20] **Hauke Stieler.** *TikZ für Automaten.* Techn. Ber. Version 0.2de_DE. Apr. 2015. URL: <http://hauke-stieler.de/public/tikz-for-state-machines.pdf> (besucht am 14. 11. 2018).
- [21] **Kresten Krab Thorup, Frank Jensen und Chris Rowley.** *The calc package – Infix notation arithmetic in \LaTeX .* Techn. Ber. Mai 2017. URL: <https://mirror.informatik.hs-fulda.de/tex-archive/macros/latex/required/tools/calc.pdf> (besucht am 15. 11. 2018).

- [22] **Herbert Voß.** *LaTeX Referenz der Umgebungen, Makros, Längen und Zähler.* Bd. 1. Dez. 2013. URL: <https://www.lehmanns.de/page/latexreferenz> (besucht am 14. 11. 2018).
- [23] **Herbert Voß.** *Math mode.* Techn. Ber. Jan. 2014. URL: <http://tug.ctan.org/obsolete/info/math/voss/mathmode/Mathmode.pdf> (besucht am 14. 11. 2018).
- [24] **Joseph Wright.** *siunitx – A comprehensive (SI)units package.* Techn. Ber. Mai 2018. URL: <ftp://ftp.mpi-sb.mpg.de/pub/tex/mirror/ftp.dante.de/pub/tex/macros/latex/contrib/siunitx/siunitx.pdf> (besucht am 13. 11. 2018).