

# L<sup>A</sup>T<sub>E</sub>X advanced

TikZ, etc.

Walter Stieben 4stieben@inf  
Hauke Stieler 4stieler@inf  
Ruben Felgenhauer 4felgenh@inf

4. Januar 2018

Danke Henning (8pridoeh) dass wir deine Folien aus dem  
WS14/15 benutzen dürfen :D



















# Mit minted

```
class MeineKlasse{
    private int meineVariable; // Deklaration

    public void meineMethode(){
        meineVariable = 42; // Initialisierung
    }
}
```







# Code-Highlighten

# Mit verbatim

### Ergebnis:

```
# ~/.ssh/config
Host fbi
    User 7nachnam
    ForwardX11 yes
    HostName rzssh1.informatik.uni-hamburg.de
    DynamicForward 7777
    #LocalForward 6631 linuxprint.informatik.uni-hamburg.de:631
```















# Packages & Klassen

# Benutzung

**L<sup>A</sup>T<sub>E</sub>X-Code:**

```
\documentclass[sans]{meineklasse}
```

```
\usepackage[german]{meinpackage}
```

```
\begin{document}
  \helloworld
\end{document}
```

### Ergebnis:

Hallo Welt!

**Mehr Infos:** <http://ctan.mirrors.hoobly.com/macros/latex/doc/clsguide.pdf>  
[https://en.wikibooks.org/wiki/LaTeX/Creating\\_Packages](https://en.wikibooks.org/wiki/LaTeX/Creating_Packages)















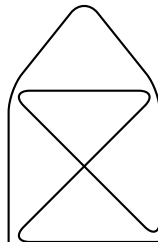
# Grundlagen

## TikZ

**L<sup>A</sup>T<sub>E</sub>X-Code:**

```
\begin{tikzpicture}
  \draw[thick,rounded corners=8pt]
    (0,0) -- (0,2) -- (1,3.25) --
    (2,2) -- (2,0) -- (0,2) --
    (2,2) -- (0,0) -- (2,0);
\end{tikzpicture}
```

### Ergebnis:











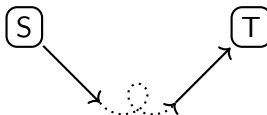




# Grundlagen

## Styles für gesamtes TikZpicture

### Ergebnis:





# Automaten

## Zustände

### TikZ-Code:

```
\usetikzlibrary{
    automata,
    arrows}

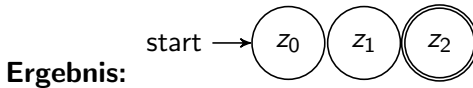
% ...

\begin{tikzpicture}[->,
    >=stealth',
    semithick]

    \node[state,initial]      (0)                {$z_0$};
    \node[state]              (1) [right of=0]    {$z_1$};
    \node[state,accepting]    (2) [right of=1]    {$z_2$};
\end{tikzpicture}
```

# Automaten

## Zustände



# Automaten

## Positionierung

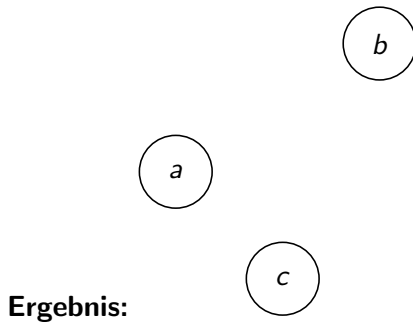
### TikZ-Code:

```
\usetikzlibrary{
    automata,
    arrows,
    positioning}
% ...
\begin{tikzpicture}[->,
    >=stealth',
    semithick,
    node distance=2cm]

    \node [state] (a)                                {$a$};
    \node [state] (b) [above right=1cm and 2cm of a] {$b$};
    \node [state] (c) [below right of = a]             {$c$};
\end{tikzpicture}
```

# Automaten

## Positionierung





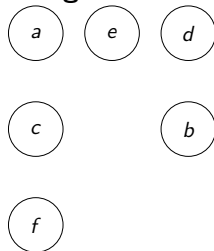
# Automaten

## Genauere Positionierung

**L<sup>A</sup>T<sub>E</sub>X-Code:**

```
\usetikzlibrary{calc}
% ...
\begin{tikzpicture}
  \node [state] (a) {} {$a$};
  \node [state] (b) [below right=1cm and 2cm of a] {$b$};
  \node [state] (c) [at=(a|-b)] {$c$};
  \node [state] (d) [at=(b|-a)] {$d$};
  \node [state] (e) [at=($(a)!1/2!(d)$)] {$e$};
  \node [state] (f) [at=($(a)!2!(c)$)] {$f$};
\end{tikzpicture}
```

### Ergebnis:



# Automaten

## Pfeile

### TikZ-Code:

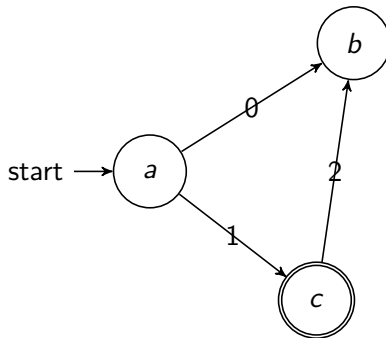
```
\begin{tikzpicture}[->,
  >=stealth',
  semithick,
  node distance=2cm]

\node [state,initial]    (a)           {$a$};
\node [state]            (b)
  [above right=1cm and 2cm of a]       {$b$};
\node [state,accepting] (c)
  [below right = 1cm and 1.5cm of a] {$c$};

\path (a) edge node {0} (b)
      (a) edge node {1} (c)
      (c) edge node {2} (b);
\end{tikzpicture}
```

# Automaten

## Pfeile



Ergebnis:

# Automaten

## Pfeile

### TikZ-Code:

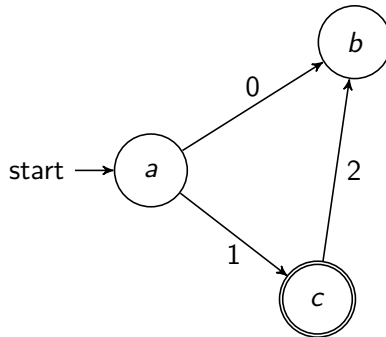
```
\begin{tikzpicture}[->,
  >=stealth',
  semithick,
  node distance=2cm]

\node [state,initial]    (a)           {$a$};
\node [state]            (b)
  [above right=1cm and 2cm of a]       {$b$};
\node [state,accepting] (c)
  [below right = 1cm and 1.5cm of a]   {$c$};

\path (a) edge[above] node {0} (b)
        edge[below] node {1} (c)
        (c) edge[right] node {2} (b);
\end{tikzpicture}
```

# Automaten

## Pfeile



**Ergebnis:**

# Automaten

## Pfeile

### TikZ-Code:

```
\begin{tikzpicture}[->,>=stealth',
  shorten >=5pt,
  node distance=2.5cm,
  semithick]

\node[initial,state]      (R)                {$z_r$};
\node[state]              (S) [right of=R]    {$z_s$};
\node[state,accepting]    (E) [right of=S]    {$z_e$};

\path (R) edge [loop,above] node {0} (R)
      (R) edge [below] node {1} (S)
      (S) edge [loop,above] node {0,1} (S)
      (S) edge [below] node {1} (E)
      (E) edge [bend left,below] node {0} (R)
      (E) edge [loop,above] node {0,1} (E);

\end{tikzpicture}
```



# Funktionen Zeichnen

## TikZ

```
\usepackage{pgf}
% ...
\begin{tikzpicture}[>=latex,semithick,font=\scriptsize,scale=0.75]
  \draw[very thin,color=lightgray] (-3.2,-1.2) grid (3.2,4.2);
  \draw[->] (-3.2,0) -- (3.4,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.4) node[above] {$y$};

  \foreach \x/\xtext in {-3/-3, -2/-2, -1/-1, 1/1, 2/2, 3/3}
  \draw[shift={(\x,0)}] (0pt,2pt) -- (0pt,-2pt) node[below] {$\xtext$};

  \foreach \y/\ytext in {-1/-1, 1/1, 2/2, 3/3, 4/4}
  \draw[shift={(0,\y)}] (2pt,0pt) -- (-2pt,0pt) node[left] {$\ytext$};

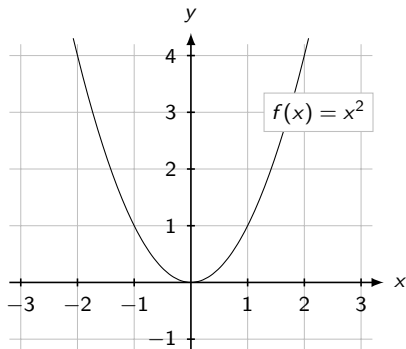
  \draw[thin,domain=-2.075:2.075,smooth,variable=\x,black]
    plot ({\x},{\x*\x});
  \draw[thin] node[inner sep=1mm,
    fill=white,
    draw=lightgray] at (2.25,3) {$f(x)=x^2$};
  \draw[thin] node[inner sep=1mm,
    fill=white,
    draw=lightgray] at (-2.25,3) {$f(x)=x^2$};
```



# Funktionen Zeichnen

## TikZ

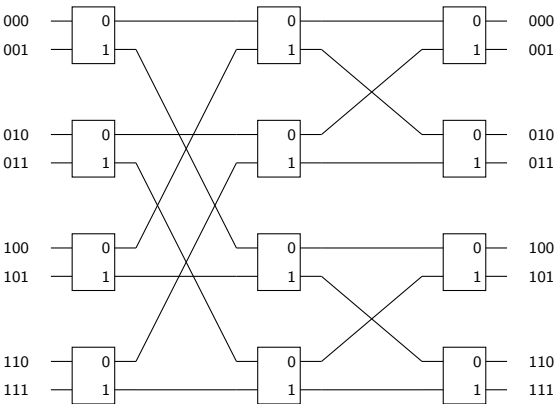
**Ergebnis:**



**Alternative:** Gnuplot lässt sich vielfältig mit L<sup>A</sup>T<sub>E</sub>X kombinieren

TikZ an die Grenzen getrieben

Banyan-Netz (3 Stufen)







# TikZ an die Grenzen getrieben

... mehrere kaputte Kaffeemaschinen später ...



# TikZ an die Grenzen getrieben

## BEWARE

Das Kompilieren dieses Dokumentes dauert auf einem 4 GHz-Quad Core knapp eine halbe Minute!













# Versionsverwaltung

## .gitignore

- Die .gitignore verwaltet automatisch ignorierte Dateien im Repository
- L<sup>A</sup>T<sub>E</sub>X erzeugt viele temporäre Dateien
- The lazy way: <https://github.com/github/gitignore>

`https://raw.githubusercontent.com/github/gitignore/master/TeX.gitignore`

# Versionsverwaltung

# .gitignore

### Beispiel:

```
## Core latex/pdflatex auxiliary files:
```

\* .aux

$$*\log$$

[...]

## ## Intermediate documents:

\*.dvi

\*-converted-to.\*

# \*.ps

# \*.pdf

[...]



# Versionsverwaltung

## make & latexmk

### Beispiel für Makefile:

```
.PHONY: default all clean
```

```
default: all
```

```
all: Abgabe.pdf
```

```
%.pdf: %.tex
```

```
latexmk -pdf $<
```

```
clean:
```

```
latexmk -C
```