

L^AT_EX advanced

TikZ, etc.

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

4. Januar 2018

Code-Highlighten

Mit lstlisting

Ergebnis:

```
1 class MeineKlasse{
2     private int meineVariable; //
        Deklaration
3
4     public void meineMethode(){
5         meineVariable = 42; //
            Initialisierung
6     }
7 }
```

Listing 1: Variablen

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
```


Math advanced

Mehr tolle Mathe-Tricks

L^AT_EX-Code:

Große Klammern gehen auch: `\(\left(\frac{n^2 + 1}{3} \right)^2\)` `\`

Aus $\left(\int_a^b \sum_{k=1}^n f_k(x) dx\right)$ wird $\left(\displaystyle \int\limits_a^b \sum_{k=1}^n f_k(x) dx\right)$

Ergebnis:

Große Klammern gehen auch: $\left(\frac{n^2+1}{3}\right)^2$

Aus $\int_a^b \sum_a^b$ wird $\int_a^b \sum_a^b$

Anmerkung: Über `\everymath{\displaystyle}` kann jeder Mathe-Modus automatisch im `displaystyle` sein.

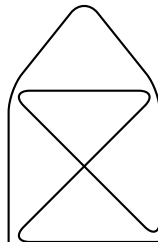
Grundlagen

TikZ

L^AT_EX-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}
```

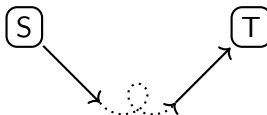
Ergebnis:



Grundlagen

Styles für gesamtes TikZpicture

Ergebnis:



Automaten

Zustände

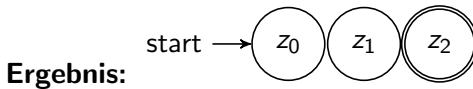
TikZ-Code:

```
\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:

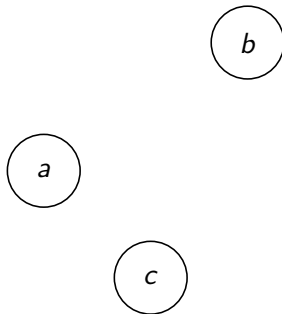
```
\usetikzlibrary{
    automata,
    arrows,
    positioning}
% ...
\begin{tikzpicture}[->,
    >=stealth',
    semithick,
    node distance=2cm]

    \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

Ergebnis:



Automaten

Pfeile

TikZ-Code:

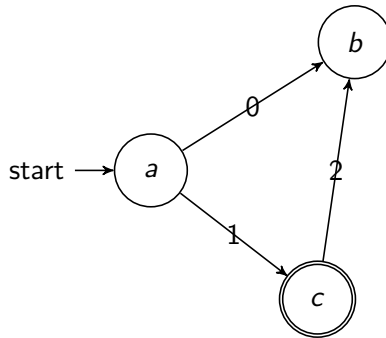
```
\begin{tikzpicture}[->,
  >=stealth',
  semithick,
  node distance=2cm]

\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:

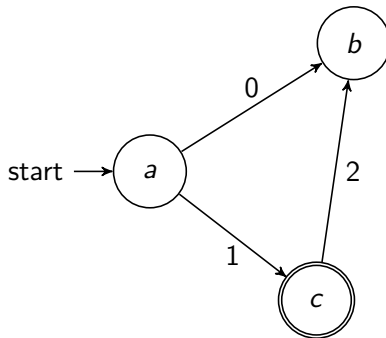
```
\begin{tikzpicture}[->,
  >=stealth',
  semithick,
  node distance=2cm]

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

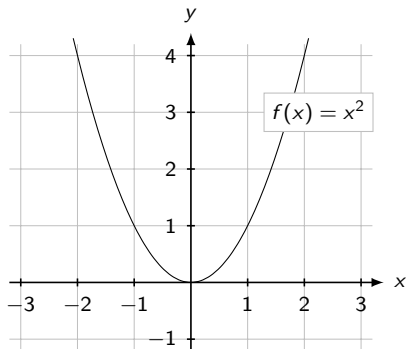
  \foreach \y/\ytext in {-1/-1, 1/1, 2/2, 3/3, 4/4}
  \draw[shift={(0,\y)}] (2pt,0pt) -- (-2pt,0pt) node[left] {$\ytext$};

  \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

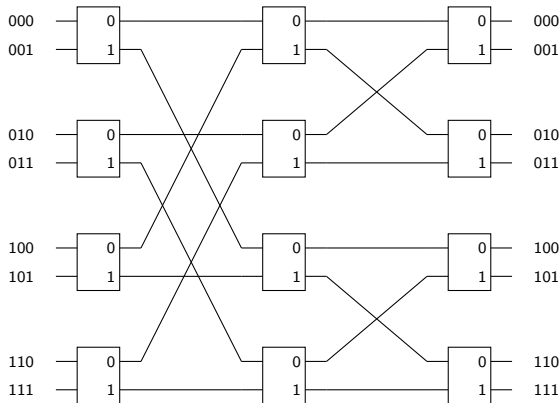
Ergebnis:



Alternative: Gnuplot lässt sich vielfältig mit L^AT_EX kombinieren

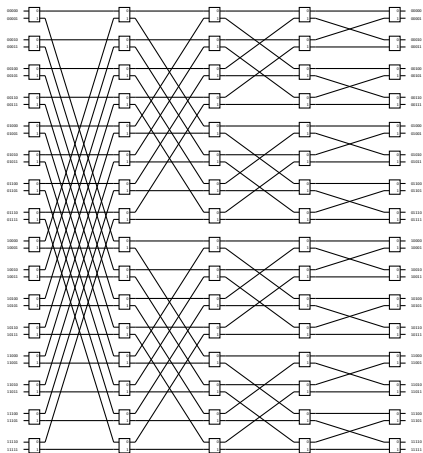
TikZ an die Grenzen getrieben

Banyan-Netz (3 Stufen)



TikZ an die Grenzen getrieben

Banyan-Netz (5 Stufen)



Versionsverwaltung

Version control in a nutshell

Versionsverwaltungs-Software (VCS):

- Ermöglicht es, an vielen Rechnern Dateien kollaborativ zu bearbeiten
- Über Branches könnt ihr parallel an den gleichen Dateien arbeiten
- Branches können zusammengeführt werden (merge)
- Vllt. seht ihr noch ein KBS dazu ;)

Git:

- Das populärste VCS von Linus Torvalds
- Unter Linux meist vorinstalliert, unter Windows z.B. via git-scm
- Dezentral, vielseitig, einfach, mächtig

Versionsverwaltung

.gitignore

- Die .gitignore verwaltet automatisch ignorierte Dateien im Repository
- L^AT_EX erzeugt viele temporäre Dateien
- The lazy way: <https://github.com/github/gitignore>

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

Versionsverwaltung

make & latexmk

Beispiel für Makefile:

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

```
default: all
```

all: Abgabe.pdf

%.pdf: %.tex

```
latexmk -pdf $<
```

clean:

latexmk -C