

\LaTeX advanced

TikZ, etc.

20. August 2021

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

Und auch Danke an alle, die zu den Folien und zum Vortrag
beigetragen haben:

- Walter Stieben 4stieben@inf
- Ruben Felgenhauer 4felgenh@inf
- Malte Hamann 1hamann@inf
- Hauke Stieler 4stieler@inf

Referenzieren (Abschnitte)

L^AT_EX-Code:

```
\subsection{Cliquen in bipartiten Graphen}  
\label{sec:cliques}
```

% Irgendwo anders

Im Abschnitt `\ref{sec:cliques}` auf Seite
`\pageref{sec:cliques}` wurde das Finden von
Cliquen in bipartiten Graphen beschrieben.

Ergebnis: Im Abschnitt 3.2 auf Seite 7 wurde das Finden von
Cliquen in bipartiten Graphen beschrieben.

Referenzieren (Figures)

L^AT_EX-Code:

```
\begin{figure}[t]
  \includegraphics[width=7cm]{images/lichtstrahl}
  \caption{Brechung eines Lichtstrahls beim Wechsel
           des Mediums}
  \label{fig:lichtbrechung}
\end{figure}
% Irgendwo anders
```

Der Lichtstrahl wird gebrochen, wie
Abbildung `\ref{fig:lichtbrechung}` zeigt.

Ergebnis: Der Lichtstrahl wird gebrochen, wie Abbildung 3 zeigt.

- Man verwaltet eine BibT_EX-Datei (*.bib) mit Literaturangaben
- Mit `\cite[Seite X]{Referenz}` referenziert man eine solche Angabe, mit optionaler Seitenangabe.
- Mit `\footcite{Referenz}` kann die Referenz direkt in die Fußnote mit hochgestelltem Index geschrieben werden.
- Vor `pdflatex` wirft man `bibtex` an

L^AT_EX-Code:

% Im Header

```
\bibliographystyle{alpha}
```

% Beim Zitat

Für die Lösung des Travelling-Salesman-Problems
wurde ein heuristischer Algorithmus `\cite{lin19973}`
gewählt.

% An der Stelle des Literaturverzeichnis

```
\bibliography{literatur}
```

BibT_EX-Eintrag: (aus “literatur.bib”)

```
@article{lin1973,  
  author = {Shen Lin and Brian W. Kernighan},  
  title   = {An Effective Heuristic Algorithm for the  
             Travelling-Salesman Problem},  
  journal = {Operations Research},  
  volume  = {21},  
  year    = {1973},  
  pages   = {498--516},  
}
```

Ergebnis:

Für die Lösung des Travelling-Salesman-Problems wurde ein heuristischer Algorithmus [LK73] gewählt.

Literatur

- [LK73] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the travelling-salesman problem. *Operations Research*, 21:498–516, 1973.

Biber:

- Moderner Ersatz für Bib \TeX
- Guter UTF-8-Support
- Bessere Verwaltung von Styles
- Mehr Kontrolle über Sortierung
- Nicht überall verbreitet

Bib \LaTeX

- Package zum Einstellen vieler Dinge mittels \LaTeX -Code
- Einfaches wissenschaftliches Zitieren
- Für den Einsatz mit Biber entwickelt (UTF-8-Support)
- Funktioniert mittelmäßig mit Bib \TeX

- minted arbeitet mit Pygments (python-library).
- Benötigt `-shell-escape` als Parameter von `pdflatex`.

\LaTeX -Code:

```
\usepackage{minted}
% ...
\begin{minted}{java}
class MeineKlasse{
    private int meineVariable; // Deklaration

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

Ergebnis:

```
class MeineKlasse{  
    private int meineVariable; // Deklaration  
  
    public void meineMethode(){  
        meineVariable = 42; // Initialisierung  
    }  
}
```

Mit lstlisting

L^AT_EX-Code:

```
\usepackage{listings}
\lstset{...} % style-einstellungen
% ...
\begin{lstlisting}[caption=Variablen]
class MeineKlasse{
    private int meineVariable; // Deklaration

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

Ergebnis:

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

Listing 1: Variablen

L^AT_EX-Code:

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

Mit algorithmic (Pseudocode)

L^AT_EX-Code:

```
\begin{algorithmic}
  \IF{some condition is true}
    \STATE do some processing
  \ELSIF{some other condition is true}
    \STATE do some different processing
  \ENDIF
\end{algorithmic}
```

Ergebnis:

```
if some condition is true then
  do some processing
else if some other condition is true then
  do some different processing
end if
```


Eigene Befehle

L^AT_EX-Code:

```
% TeX-style
\def\heute{Heute ist der \today.}
% LaTeX-style (besser)
\newcommand{\heute}{Heute ist der \today.}
\newcommand{\TikZ}{Ti\textit{k}Z}
% Verwendung
\heute
\TikZ
```

Ergebnis: Heute ist der 20. August 2021.
TikZ

L^AT_EX-Code:

% LaTeX-style

```
\newcommand{\bus}[4]{Ein Bus der Linie #1 Richtung  
#2 fährt von der Haltestelle  
#3 um #4 ab.}
```

% Verwendung

```
\bus{181}{Sternschanze}{Informatikum}{13:37}
```

Ergebnis: Ein Bus der Linie 181 Richtung Sternschanze fährt von Informatikum um 13:37 ab.

L^AT_EX-Code:

```
\newenvironment{texttttit}  
    {\begingroup\ttfamily\itshape}  
    {\endgroup}
```

% Verwendung

```
\begin{texttttit}  
    Dies ist ein Test  
\end{texttttit}
```

Ergebnis: *Dies ist ein Test*

\LaTeX -Code (meinpackage.sty):

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{meinpackage}[2000/01/01 Mein Package]

\newcommand{\helloworld}{Hello World!}

\DeclareOption{german}{
  \renewcommand{\helloworld}{Hallo Welt!}
}

\ProcessOptions\relax

\endinput
```

Eigene Klassen

L^AT_EX-Code (meineklasse.cls):

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{meineklasse}[2000/01/01 Meine Klasse]

\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage[ngerman]{babel}
\RequirePackage{lmodern}

\DeclareOption{sans}{
  \renewcommand*\familydefault{\sfdefault}
}

\ProcessOptions\relax

\LoadClass[a4paper,10pt]{scrartcl}

\endinput
```

LaTeX-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

Alternative: Include-Dateien

`\input{<datei>}` lädt den Inhalt von `<datei>` stumpf, als ob er so im Dokument stehen würde. `\include{<datei>}` macht eine neue Seite auf und kann nicht geschachtelt werden, legt aber eigene aux-Dateien an und kann somit Seitenzahlen und Querverweise beachten.

LaTeX-Code (beispiel.inc):

Hallo `\LaTeX`!

LaTeX-Code (Hauptdokument):

`\input{beispiel.inc}`

Ergebnis:

Hallo LaTeX!

Mehr tolle Mathe-Tricks

L^AT_EX-Code:

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

Aus `\(\int_a^b \sum_a^b\)` wird `\(\displaystyle \int\limits_a^b \sum_a^b\)`

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.

Mehr Mathe mit mathtools

L^AT_EX-Code:

```

\mathsf{sinc}(x) \coloneqq \begin{dcases}
1 & \& x = 0 \\
\frac{\sin(x)}{x} & \& \text{sonst} \\
\end{dcases} \\
\underbrace{\exp(ix)}_{\text{Hier wird es hässlich}}
= \cos(x) + \underbrace{i \sin(x)}_{\mathclap{\text{Hier wird es hübsch}}}

```

Ergebnis:

$$\text{sinc}(x) := \begin{cases} 1 & x = 0 \\ \frac{\sin(x)}{x} & \text{sonst} \end{cases}$$

$$\underbrace{\exp(ix)}_{\text{Hier wird es hässlich}} = \cos(x) + \underbrace{i \sin(x)}_{\text{Hier wird es hübsch}}$$

SI-Einheiten mit siunitx

L^AT_EX-Code:

```
\sisetup{...} % Einstellungen für siunitx
```

Wissenschaftliche Notation: `\(n = \num{1.1e3}\) \\\`

Einheiten: `\(\si{J} = \si{kg\m\squared\per\s\squared}\) \\\`

Kombiniert: `\(\SI{50}{\percent} c = \SI{1.5e8}{\m\per\s}\) \\\`

Intervalle: `\(\SIrange{1.3e5}{3.6e6}{\kg\m\per\s}\) \\\`

Ergebnis:

Wissenschaftliche Notation: $n = 1,1 \cdot 10^3$

Einheiten: $J = \frac{\text{kg m}^2}{\text{s}^2}$

Kombiniert: $50 \% c = 1,5 \cdot 10^8 \frac{\text{m}}{\text{s}}$

Intervalle: $1,3 \cdot 10^5 - 3,6 \cdot 10^6 \frac{\text{kg m}}{\text{s}}$

Noch mehr Mathe mit physics

L^AT_EX-Code:

Beträge und Normen:

```
\(\norm{\vec{x}} = \sum_{k=1}^n \abs{x_k}^2\) \\\
```

Landau-Notation: `\(\order{n \cdot \log(n)}\) \\\`

Differentiale: `\(\int f(x) \dd{x};`

```
\quad \dv{f}{x}; \quad \quad \pdv[2]{f}{x}\)
```

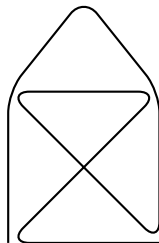
Ergebnis:

Beträge und Normen: $\|\vec{x}\| = \sum_{k=1}^n |x_k|^2$

Landau-Notation: $\mathcal{O}(n \cdot \log(n))$

Differentiale: $\int f(x) dx; \quad \frac{df}{dx}; \quad \frac{\partial^2 f}{\partial x^2}$

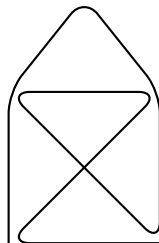
- **TikZ ist kein Zeichenprogramm.**
- Abbildungen werden mit TikZ beschrieben und durch PGF gerendert.
- Sehr umfangreiches Paket (Dokumentation: >1000 Seiten), viele Möglichkeiten.
- Hat direkte Unterstützung für Petrinetze :-)
- Overkill: Animationen in einer PDF mittels JavaScript und TikZ :o



\LaTeX -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}
```

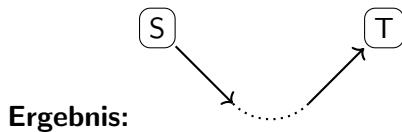
Ergebnis:



TikZ-Code:

```
\begin{tikzpicture}
  \node[shape=rectangle,draw=black,rounded corners]
    (s) at (0, 0) {S};
  \node[shape=rectangle,draw=black,rounded corners]
    (t) at (3, 0) {T};

  \draw[thick, ->]      (s)      -- (1, -1);
  \draw[thick, dotted]
    (1, -1) to [bend right = 45] (2, -1);
  \draw[thick,->]      (2, -1) -- (t);
\end{tikzpicture}
```

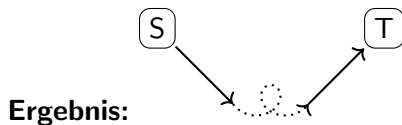


■ Hobby-Kurven mittels hobby-Paket

TikZ-Code:

```
\begin{tikzpicture}
  \node[shape=rectangle,draw=black,rounded corners]
    (s) at (0, 0) {S};
  \node[shape=rectangle,draw=black,rounded corners]
    (t) at (3, 0) {T};

  \draw[thick, ->]      (s)      -- (1, -1);
  \draw[thick, ->, dotted]
    (1, -1)
    to[curve through={(1.5, -1.1) .. (1.5,-0.75) .. (1.5, -1.1)}]
    (2, -1);
  \draw[thick, ->]      (2, -1) -- (t);
\end{tikzpicture}
```

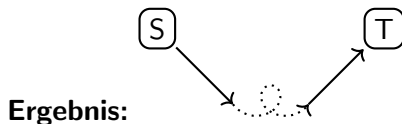



Styles für gesamtes TikZpicture

TikZ-Code:

```
\begin{tikzpicture}
[
  ->,
  thick,
  knoten/.style={shape=rectangle,draw=black,rounded corners}
]
\node[knoten] (s) at (0, 0) {S};
\node[knoten] (t) at (3, 0) {T};

\draw (s) -- (1, -1);
\draw[dotted]
  (1, -1)
  to[curve through={(1.5, -1.1) .. (1.5,-0.75) .. (1.5, -1.1)}]
  (2, -1);
\draw (2, -1) -- (t);
\end{tikzpicture}
```



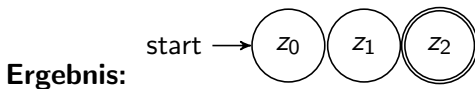
- Automaten (state-machines) per automata-Paket
- Für Positionierung positioning-Paket
- Und für Pfeile arrows-Paket

Mehr Informationen über Automaten, Pfeile, Positionierung, Optionen, etc. gibt es unter <http://hauke-stieler.de/public/tikz-for-state-machines.pdf> (im selben Ordner ist auch die *.tex Datei).

TikZ-Code:

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

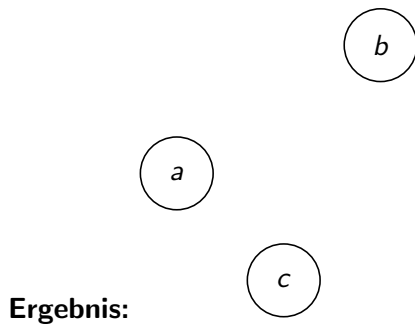
    \node[state,initial]    (0)                {$z_0$};
    \node[state]            (1) [right of=0]    {$z_1$};
    \node[state,accepting] (2) [right of=1]    {$z_2$};
\end{tikzpicture}
```



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}
```

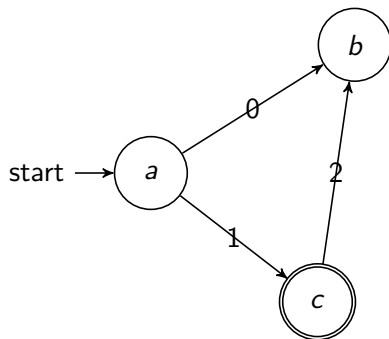


TikZ-Code:

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



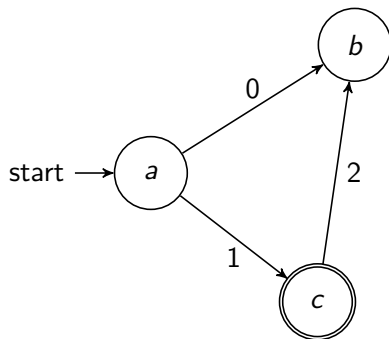
Ergebnis:

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}
```



Ergebnis:

TikZ-Code:

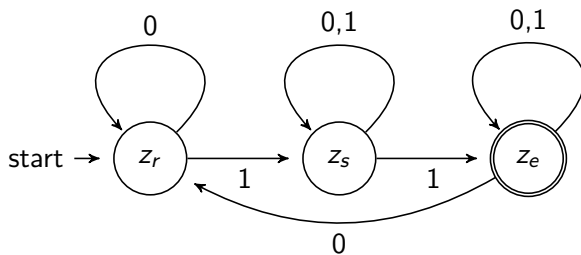
```

\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}

```



Ergebnis:

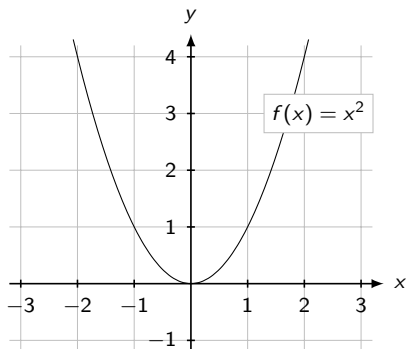
```
\usepackage{pgf}
% ...
\begin{tikzpicture}[>=latex,semithick,font=\scriptsize,scale=0.75]
  \draw[very thin,color=lightgray] (-3.2,-1.2) grid (3.2,4.2);
  \draw[->] (-3.2,0) -- (3.4,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.4) node[above] {$y$};

  \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) {$f(x)=x^2$};
\end{tikzpicture}
```

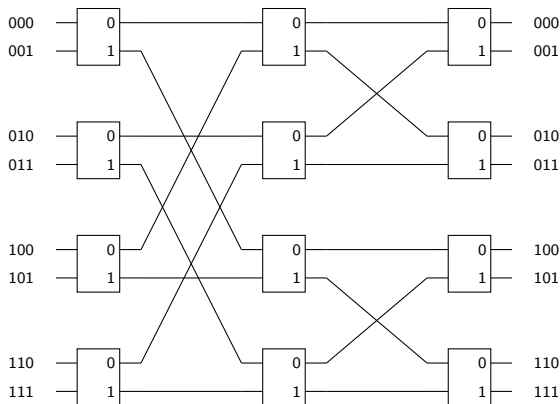

Ergebnis:



Alternative: Gnuplot lässt sich vielfältig mit \LaTeX kombinieren

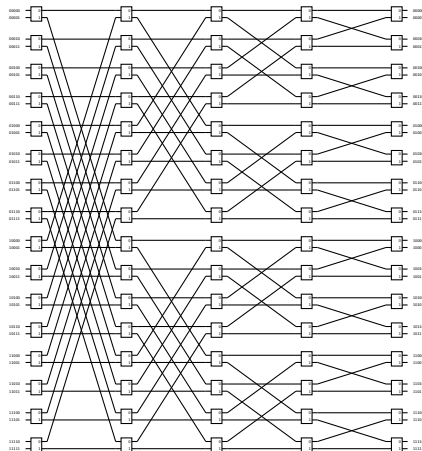
TikZ an die Grenzen getrieben

Banyan-Netz (3 Stufen)



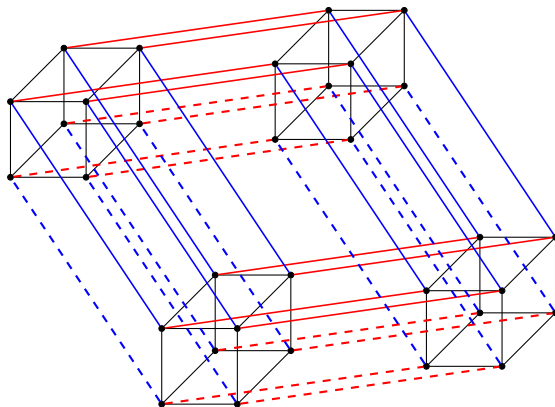
TikZ an die Grenzen getrieben

Banyan-Netz (5 Stufen)



TikZ an die Grenzen getrieben

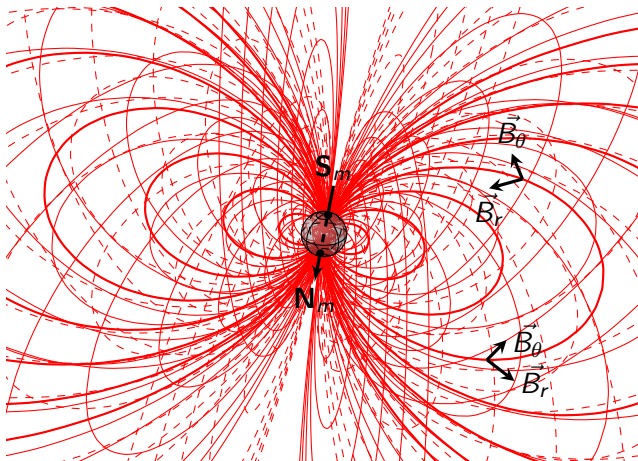
5-dimensionaler Hyperwürfel



... mehrere kaputte Kaffeemaschinen später ...

TikZ an die Grenzen getrieben

TikZ at its best



BEWARE

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

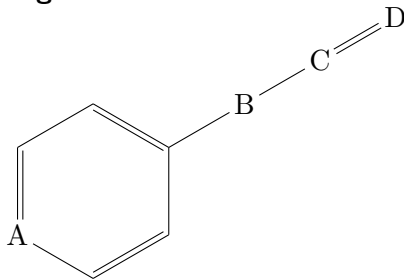
Was sonst so geht

Chemische Bilder mit chemfig

L^AT_EX-Code:

```
\chemfig{A*6(---(-B-C=D)---)}
```

Ergebnis:



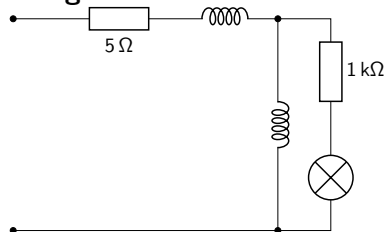
Was sonst so geht

Schaltskizzen mit CircuiTikZ

\LaTeX -Code:

```
\begin{circuitikz}[european,cute inductors]
\draw (0,0) to [short, *-] (6,0)
      to [lamp] (6,2)
      to [R, l_1=1<\kilo\ohm>] (6,4)
      to [short] (5,4)
      (0,4) to [short, *-] (1,4)
      to [R, l_1=5<\ohm>] (3,4)
      to [L] (5,4)
      to [L,*-] (5,0);
\end{circuitikz}
```

Ergebnis:



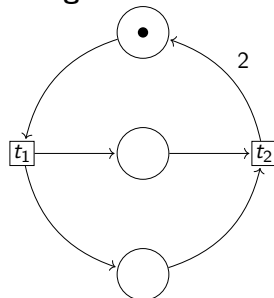
\LaTeX -Code:

```
\usetikzlibrary{petri}
\begin{tikzpicture}[node distance=1.5cm,auto]
  \node [place,tokens=1] (p1)      {};
  \node [place] (p2) [below of=p1] {};
  \node [place] (p3) [below of=p2] {};

  \node [transition] (t1) [left of=p2] {\$t_1\$}
    edge [pre,bend left] (p1)
    edge [post,bend right] (p3)
    edge [post] (p2);

  \node [transition] (t2) [right of=p2] {\$t_2\$}
    edge [pre] (p2)
    edge [pre,bend left] (p3)
    edge [post,bend right] node[swap] {2} (p1);
\end{tikzpicture}
```

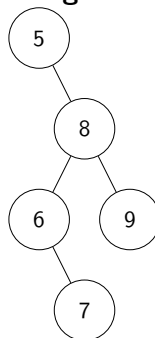
Ergebnis:



L^AT_EX-Code:

```
\begin{tikzpicture}[every node/.append style={
draw,circle,minimum width=10mm}]
\node {5}
  child[missing]
  child{node {8}
    child{node {6}
      child[missing]
      child{node {7}
        }
    }
  child{node {9}
  }
};
\end{tikzpicture}
```

Ergebnis:



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

.gitignore

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

<https://raw.githubusercontent.com/github/gitignore/master/TeX.gitignore>

.gitignore

Beispiel:

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

```
*.aux
```

```
*.log
```

```
[...]
```

```
## Intermediate documents:
```

```
*.dvi
```

```
*-converted-to.*
```

```
# *.ps
```

```
# *.pdf
```

```
[...]
```

make:

- `make` automatisiert das Bauen von Projekten
- Macht Dokumentation meistens überflüssig
- Automatischer Abgleich von Änderungsdatum von Quelle und Ziel
- Festlegen des Build-Prozesses durch Regeln in der `Makefile`
- Regelfindung per Pattern-Matching

latexmk:

- Kann \LaTeX -Dokumente vollautomatisch kompilieren
- Alternative: `latexrun`

Beispiel für Makefile:

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

```
default: all
```

```
all: Abgabe.pdf
```

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

```
latexmk -pdf $<
```

```
clean:
```

```
latexmk -C
```