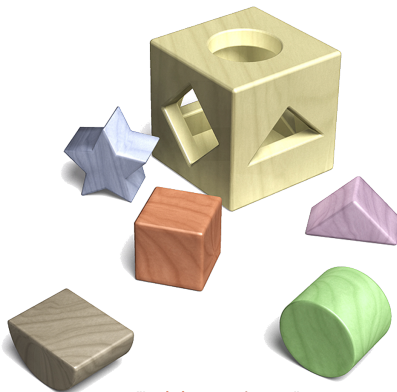


A brief introduction into TikZ

Sibylle Möhle

International Center for Computational Logic
Technische Universität Dresden
Germany

- ▶ **What is TikZ?**
- ▶ **How to draw figures?**
- ▶ **How to place text inside the figures?**



"Logic is everywhere ..."

These slides are based on slides of Tobias Pietzsch



PGF and TikZ

- ▶ **PGF (portable graphics format)**
 - ▷ package for creating graphics “inline”
 - ▷ \TeX and \LaTeX input
 - ▷ PDF, PostScript, and SVG output
 - ▷ three layers: System, Basic, and Frontend

- ▶ **TikZ (TikZ is not a drawing program)**
 - ▷ is a frontend layer for PGF (portable graphics format)
 - ▷ high-level user interface
 - ▷ this introduction refers to version 2.10
 - ▷ for features and bug-fixes: <http://www.texample.net/tikz/builds/>

- ▶ **Written by Till Tantau (author of Beamer package) and Mark Wibrow**



Setup your environment

- Use a \LaTeX document
- Use the TikZ package and required TikZ libraries

```
\usepackage{tikz}
\usetikzlibrary{...}
```

- Draw TikZ figure in the document.

```
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

```
\tikz \draw (0,0) circle (0.5);
```



Setup your environment

- Use a \LaTeX document
- Use the TikZ package and required TikZ libraries

```
\usepackage{tikz}
\usetikzlibrary{...}
```

- Draw TikZ figure in the document.

```
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

```
\tikz \draw (0,0) circle (0.5);
```



Setup your environment

- ▶ Use a \LaTeX document
- ▶ Use the `TikZ` package and required `TikZ` libraries

```
\usepackage{tikz}
\usetikzlibrary{...}
```

- ▶ Draw `TikZ` figure in the document.

```
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

```
\tikz \draw (0,0) circle (0.5);
```



Drawing Basics

- ▶ A *path* is a series of straight and curved line segments

- ▶ Syntax:

```
\path (path-operation | graphic-option)* ;
```

- ▶ There are two major operations: *draw* and *node*

- ▶ `\draw` is an abbreviation for `\path [draw]`

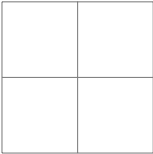
- ▶ Next, drawing figures by using `\draw` is presented

- ▶ By using `\node` text is added afterwards



Drawing with paths

► A step by step example for `\draw`:

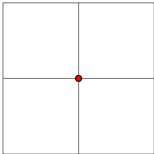


```
\begin{tikzpicture}  
  \path [draw] (0,0) circle (0.5)  
    (-0.5,-0.5) -- (0.5,0.5) ;  
\end{tikzpicture}
```



Drawing with paths

- ▶ A step by step example for `\draw`:



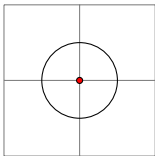
```
\begin{tikzpicture}
\path [draw] (0,0) circle (0.5)
(-0.5,-0.5) -- (0.5,0.5) ;
\end{tikzpicture}
```

- ▶ **Move-To Operation:** move to coordinate `(0,0)`
- ▶ without drawing anything



Drawing with paths

- ▶ A step by step example for `\draw`:



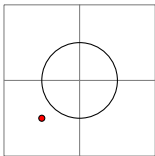
```
\begin{tikzpicture}
\path [draw] (0,0) circle (0.5)
(-0.5,-0.5) -- (0.5,0.5) ;
\end{tikzpicture}
```

- ▶ Circle Operation: draw circle with radius (0.5)
- ▶ current point remains (0,0)



Drawing with paths

- ▶ A step by step example for `\draw`:



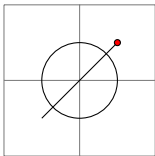
```
\begin{tikzpicture}
\path [draw] (0,0) circle (0.5)
(-0.5,-0.5) -- (0.5,0.5) ;
\end{tikzpicture}
```

- ▶ **Move-To Operation:** move to coordinate `(-0.5,-0.5)`
- ▶ without drawing anything



Drawing with paths

- ▶ A step by step example for `\draw`:



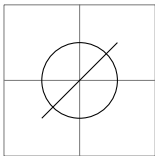
```
\begin{tikzpicture}
\path [draw] (0,0) circle (0.5)
(-0.5,-0.5) -- (0.5,0.5) ;
\end{tikzpicture}
```

- ▶ Line-To Operation: line to coordinate (0.5,0.5)



Drawing with paths

- ▶ A step by step example for `\draw`:



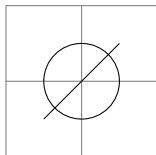
```
\begin{tikzpicture}
\path [draw] (0,0) circle (0.5)
(-0.5,-0.5) -- (0.5,0.5) ;
\end{tikzpicture}
```

- ▶ Path ends.



Graphic Options

- Add color and line styles to the picture:

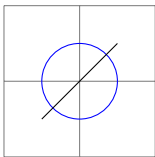


```
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```



Graphic Options

- Add color and line styles to the picture:



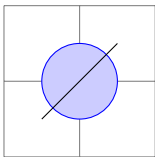
```
\begin{tikzpicture}
  \draw [color=blue] (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Add color to the line.



Graphic Options

- Add color and line styles to the picture:



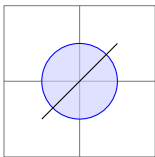
```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white] (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Fill the circle with color.



Graphic Options

- Add color and line styles to the picture:



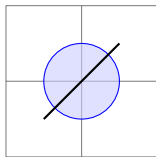
```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white, fill opacity=0.6]
    (0,0) circle (0.5);
  \draw (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Make circle transparent.



Graphic Options

- Add color and line styles to the picture:



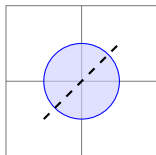
```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white,fill opacity=0.6]
    (0,0) circle (0.5);
  \draw [thick] (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Draw a thick line.



Graphic Options

- Add color and line styles to the picture:



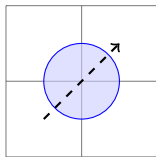
```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white,fill opacity=0.6]
    (0,0) circle (0.5);
  \draw [thick, dashed] (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Draw a dashed line.



Graphic Options

- Add color and line styles to the picture:



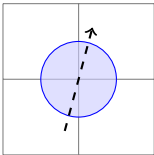
```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white,fill opacity=0.6]
    (0,0) circle (0.5);
  \draw [thick, dashed, ->] (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Draw an arrow.



Graphic Options

- Add color and line styles to the picture:



```
\begin{tikzpicture}
  \draw [color=blue, fill=blue!20!white,fill opacity=0.6]
    (0,0) circle (0.5);
  \draw [thick, dashed, ->, rotate=30]
    (-0.5,-0.5) -- (0.5,0.5);
\end{tikzpicture}
```

- Rotate the arrow.



Scoping and Grouping Graphic Options

- Grouping multiple graphic options:



```
\begin{tikzpicture}  
  \draw [orange, very thick] (0,0) circle (0.5);  
  \draw [orange, very thick] (-0.5,-0.5) -- (0.5,0.5);  
  \draw [orange, very thick] (0.5,-0.5) -- (-0.5,0.5);  
\end{tikzpicture}
```



Scoping and Grouping Graphic Options

- Grouping multiple graphic options:



```
\begin{tikzpicture} [orange, very thick]
  \draw (0,0) circle (0.5);
  \draw(-0.5,-0.5) -- (0.5,0.5);
  \draw(0.5,-0.5) -- (-0.5,0.5);
\end{tikzpicture}
```

- Define a set of options for the whole figure.



Scoping and Grouping Graphic Options

- Grouping multiple graphic options:



```
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \begin{scope} [orange, very thick]
    \draw(-0.5,-0.5) -- (0.5,0.5);
    \draw(0.5,-0.5) -- (-0.5,0.5);
  \end{scope}
\end{tikzpicture}
```

- Set a scope.



Scoping and Grouping Graphic Options

► Grouping multiple graphic options:



```
\begin{tikzpicture}
  [marked/.style = {orange, very thick}]
  \draw (0,0) circle (0.5);
  \draw [marked] (-0.5,-0.5) -- (0.5,0.5);
  \draw [marked] (0.5,-0.5) -- (-0.5,0.5);
\end{tikzpicture}
```

► Define a style.



Scoping and Grouping Graphic Options

- Grouping multiple graphic options:



```
\tikzset{marked/.style = {orange, very thick}}
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw [marked] (-0.5,-0.5) -- (0.5,0.5);
  \draw [marked] (0.5,-0.5) -- (-0.5,0.5);
\end{tikzpicture}
```

- Draw a global style.



Scoping and Grouping Graphic Options

- Grouping multiple graphic options:



```
\tikzset{marked/.style = {orange, very thick}}
\begin{tikzpicture}
  \draw (0,0) circle (0.5);
  \draw [marked,blue] (-0.5,-0.5) -- (0.5,0.5);
  \draw [marked,dashed] (0.5,-0.5) -- (-0.5,0.5);
\end{tikzpicture}
```

- Properties of styles can be overwritten.



Coordinates

- ▶ There exists multiple coordinate systems
 - ▷ Polar coordinates are not introduced here

- ▶ **Absolute coordinates**

```
\tikz \draw [thick,red] (0,0) -- (2mm, 0) -- (2mm, 5pt);
```



- ▶ **Relative coordinates**

```
\tikz \draw [thick,red] (0,0) -- +(2mm, 0) -- +(0, 5pt);
```



- ▶ **Relative coordinates**

```
\tikz \draw [thick,red] (0,0) -- ++(2mm, 0) -- +(0, 5pt);
```



- ▶ **Named points**

```
\tikz \draw [thick,red] (10mm,1mm) coordinate (c1) circle (5pt)
(0,0) -- (c1);
```



Nodes

- ▶ A node is a simple shape with some text on it.



```
\tikz \path node [shape=circle, draw, color=red] {hello};
```

- ▶ Constructed using the path-operation *node*.
- ▶ *\path node* can be abbreviated as *\node*.
- ▶ *shape=* and *color=* can be omitted if there is no confusion.



Nodes

- ▶ A node is a simple shape with some text on it.



```
\tikz \node [circle, draw, red] {hello};
```

- ▶ Constructed using the path-operation *node*.
- ▶ *\path node* can be abbreviated as *\node*.
- ▶ *shape=* and *color=* can be omitted if there is no confusion.



Node Syntax

```
\path ... node [options] (name) at (coordinate) {contents} ...;
```



Node Syntax

```
\path ... node [options] (name) at (coordinate) {contents} ...;
```

- ▶ *options* may contain node shape, color, sizes, labels, ...
- ▶ A node may get a *name* for later reference.
- ▶ A node may be placed using *at* (*coordinate*).
(Otherwise it is placed at the current path coordinate.)
- ▶ Nodes *contents* can be (almost) arbitrary \LaTeX .



Node Syntax

```
\path ... node [options] (name) at (coordinate) {contents} ...;
```

- ▶ *options* may contain node shape, color, sizes, labels, ...
- ▶ A node may get a *name* for later reference.
- ▶ A node may be placed using *at* (*coordinate*).
(Otherwise it is placed at the current path coordinate.)
- ▶ Nodes *contents* can be (almost) arbitrary \LaTeX .

TikZ is quite liberal with respect to the order of the arguments.



```
\tikz \node [circle] at (0,0) [draw] (hello) [red] {hello};
```



Placing Nodes



```
\begin{tikzpicture}
  \node [packet] (p1) at (0,0.5) {$p_1$};
  \node [packet] (p1) at (1,0.5) {$p_1$};
  \node [packet] (p1) at (4,0.5) {$p_1$};
\end{tikzpicture}
```

- Can be placed at absolute positions.



Placing Nodes



```
\begin{tikzpicture}
  \node [packet] (p1) at (0,0.5) {$p_1$};
  \node [packet,right=0.25 of p1] (p2) {$p_2$};
  \node [packet,right=2 of p2] (pn) {$p_n$};
\end{tikzpicture}
```

► **Relative placement allows to say things like:**

► “Node (p2) should be right of (p1)” (wherever (p1) happens to be).

► Requires the **positioning** TikZ library:

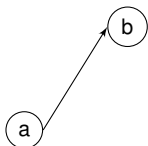
```
\usetikzlibrary{positioning}
```

► **available:** right=of, below=of, above left=of, ...



Connecting Nodes

- We can simply draw paths between node anchors.



```
\usetikzlibrary{arrows}

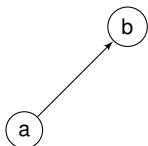
\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b}
  \draw [->] (a.east) -- (b.west);
\end{tikzpicture}
```

- Each node has anchors, e.g. **east**, **west**, **center**, ...



Connecting Nodes

- We can simply draw paths between node anchors.



```
\usetikzlibrary{arrows}

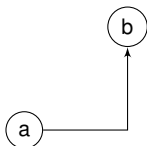
\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) -- (b);
\end{tikzpicture}
```

- Each node has anchors, e.g. **east**, **west**, **center**, ...
- If anchor specifications are left out TikZ tries to be smart about the anchor it should choose.



Connecting Nodes

- ▶ We can simply draw paths between node anchors.



```
\usetikzlibrary{arrows}

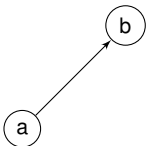
\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) -| (b);
\end{tikzpicture}
```

- ▶ Each node has anchors, e.g. **east**, **west**, **center**, ...
- ▶ If anchor specifications are left out TikZ tries to be smart about the anchor it should choose.
- ▶ Another path operation **-|**
- ▶ Again, TikZ is clever about the correct anchors



Connecting Nodes

- A very powerful path operation is `to`.



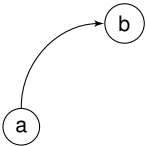
```

\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to (b);
\end{tikzpicture}
  
```



Connecting Nodes

- A very powerful path operation is **to**.

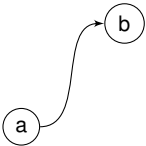


```
\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] (b);
\end{tikzpicture}
```



Connecting Nodes

- A very powerful path operation is **to**.

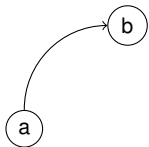


```
\begin{tikzpicture}[>=latex']
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [out=0, in=180] (b);
\end{tikzpicture}
```



Labeling Connections

- Connections can be labeled by inserting text nodes in the path.

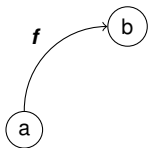


```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] (b);
\end{tikzpicture}
```



Labeling Connections

- Connections can be labeled by inserting text nodes in the path.

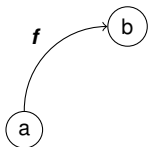


```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] node [auto] {$f$} (b);
\end{tikzpicture}
```



Labeling Connections

- Connections can be labeled by inserting text nodes in the path.



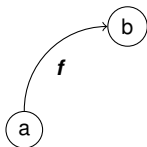
```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] node [auto] {$f$} (b);
\end{tikzpicture}
```

- The `auto` option places the label such that it is next to the path and doesn't overlap anything.



Labeling Connections

- Connections can be labeled by inserting text nodes in the path.



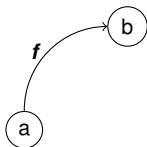
```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] node [auto, swap] {$f$}(b);
\end{tikzpicture}
```

- The **auto** option places the label such that it is next to the path and doesn't overlap anything.
- **swap** places the label on the other side of the path.



Labeling Connections

- Connections can be labeled by inserting text nodes in the path.



```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) to [bend left=45] node [left] {$f$}(b);
\end{tikzpicture}
```

- The **auto** option places the label such that it is next to the path and doesn't overlap anything.
- **swap** places the label on the other side of the path.
- Instead of **auto**, also **left**, **above**, **right** and **below** can be used



For Loop

- ▶ For drawing multiple objects a for loop can be used
- ▶ Loops can be nested



```
\begin{tikzpicture}  
  \foreach \y in {1,2,3} {  
    \draw [blue, ultra thick] (0,\y ) circle [radius=0.3];  
  }  
\end{tikzpicture}
```



Good Practice

- ▶ Place nodes relative to each other
 - ▶ Do not use too many colors and styles
 - ▶ Do not overload figures.
-
- ▶ Further information:
 - ▷ **Gnuplot** has a TikZ terminal
 - ▷ **Inkscape** can export to TikZ
 - ▷ **Dia** is another GUI that can export TikZ
 - ▷ see <http://www.texample.net/tikz/resources/>
 - ▷ **documentation**
<http://www.texample.net/media/pgf/builds/pgfmanualCVS2012-11-04.pdf>
(v2.10-CVS)
<ftp://ftp.tu-chemnitz.de/pub/tex/graphics/pgf/base/doc/pgfmanual.pdf>
(v3.01a)

