

Tegnekurs i TikZ

Veronika Heimsbakk
veronahe@ulrik.uio.no

1. oktober 2014

Innhold

| | | |
|----------|-----------------------------------|-----------|
| 1 | The Basics | 2 |
| 1.1 | Linjer | 2 |
| 1.2 | Kvadrat | 2 |
| 1.3 | Sirkel | 3 |
| 1.4 | Pynte litt | 3 |
| 1.5 | Farger | 4 |
| 2 | Koordinatsystemer | 5 |
| 3 | Trær | 8 |
| 3.1 | Rød-svarte trær | 9 |
| 3.2 | Bygge et tre | 10 |
| 4 | Grafer | 11 |
| 4.1 | Tegne grafen | 12 |
| 5 | Automater | 13 |
| 5.1 | Automatens tilstander | 13 |
| 5.2 | Stien gjennom automaten | 14 |
| 6 | Resursser | 15 |

1 The Basics

For å kunne bruke pakken TikZ må man først inkludere `\usepackage{tikz}` i dokumentet sitt. Da kan vi begynne å lage illustrasjoner ved å lage `tikzpicture`.

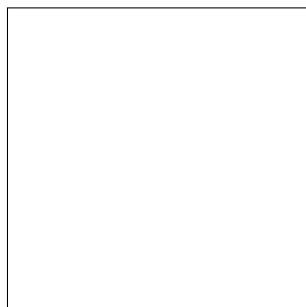
1.1 Linjer

En av de mest brukte TikZ kommandoene er `\draw`. For å tegne ei rett linje sier man hvor man vil tegne *fra* og *til*:

```
\begin{tikzpicture}
  \draw (0,0) -- (4,0);
\end{tikzpicture}
```

1.2 Kvadrat

Videre kan vi bygge på og lage et kvadrat:

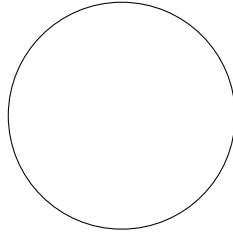


```
\begin{tikzpicture}
  \draw (0,0) -- (4,0) -- (4,4) -- (0,4) -- (0,0);
\end{tikzpicture}
```

Vi kan også bruke nøkkelordet `rectangle`, og lage en kortversjon som gjør akkurat det samme:

```
\begin{tikzpicture}
  \draw (0,0) rectangle (4,4);
\end{tikzpicture}
```

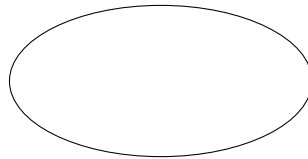
1.3 Sirkel



Den første koordinaten er sirkelens sentrum, og lengden vi oppgir til slutt er sirkelens radius.

```
\begin{tikzpicture}
  \draw (2,2) circle (1.5cm);
\end{tikzpicture}
```

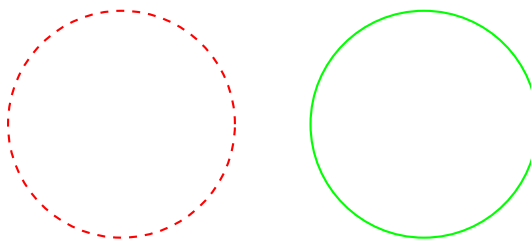
Ellipser tegnes ved at vi oppgir radiusen i x- og y-retningene:



```
\begin{tikzpicture}
  \draw (2,2) ellipse (2cm and 1cm);
\end{tikzpicture}
```

1.4 Pynte litt

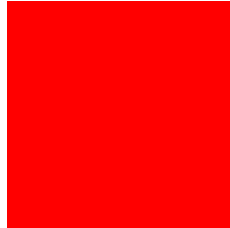
For å pynte litt på sirkelen vår, kan vi legge til noen ekstra argumenter til `\draw`-kommandoen. For eksempel slik:



```
\begin{tikzpicture}
  \draw[red, thick, dashed] (2,2) circle (1.5cm);
  \draw[green, thick] (6,2) circle (1.5cm);
\end{tikzpicture}
```

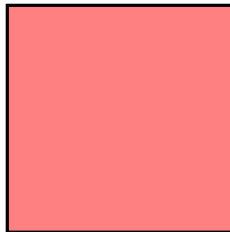
1.5 Farger

Vi kan også fylle formene våre ved å bruke kommandoen `\fill`.



```
\begin{tikzpicture}
  \fill[red] (0,0) rectangle (3,3);
\end{tikzpicture}
```

Om vi ønsker å legge til en kant rundt kvadratet, kan vi bruke kommandoen `\filldraw`. Her fyller vi kvadratet rødt med gjennomsiktighet på 50% og en tykk sort strek som kant.



```
\begin{tikzpicture}
  \filldraw[red!50, draw=black, very thick] (0,0) rectangle (3,3);
\end{tikzpicture}
```

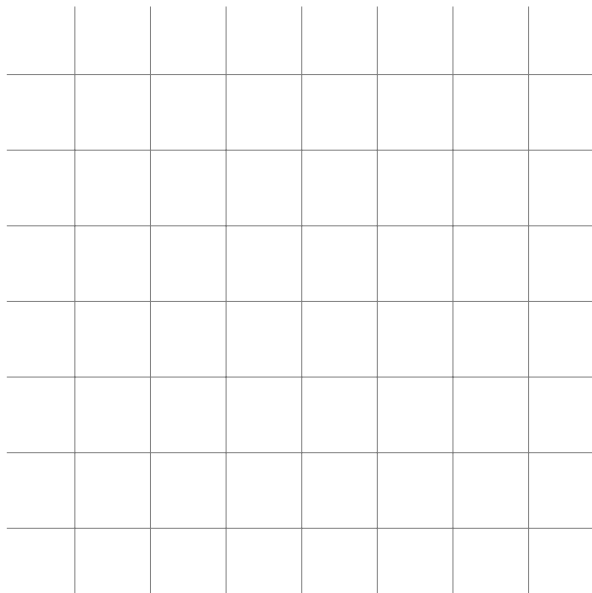
Vi har også gradient i TikZ, og det kan se slik ut:



```
\begin{tikzpicture}
  \shade[left color=black, right color=red] (0,0) rectangle (3,3);
\end{tikzpicture}
```

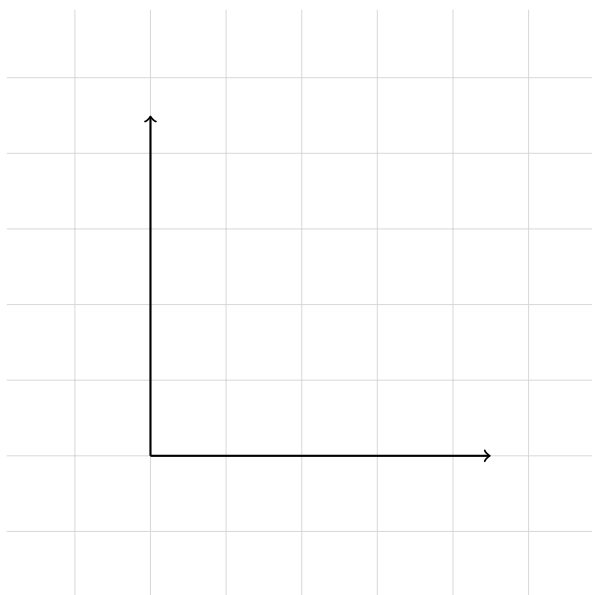
2 Koordinatsystemer

Dette eksempelet krever et rutenett, piler, noder og plassering av tall og bokstaver. Vi starter med et rutenett:



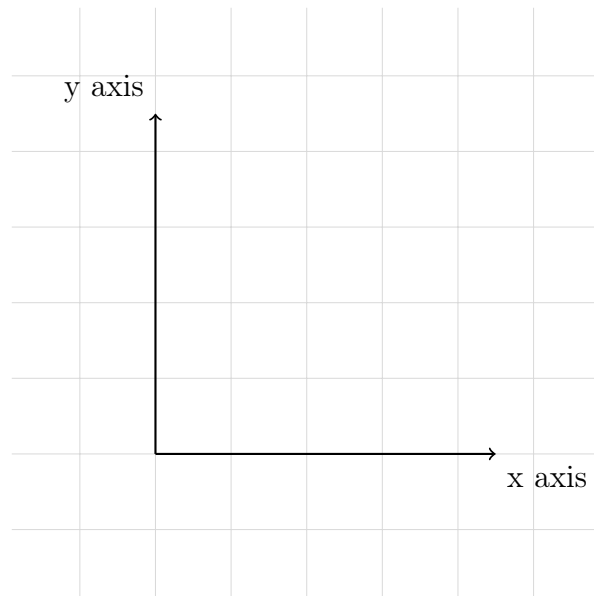
```
\draw[step=1cm,gray,very thin] (-1.9,-1.9) grid (5.9,5.9);
```

Videre trenger vi x-aksen og y-aksen. Dette er to linjer med piler i enden.



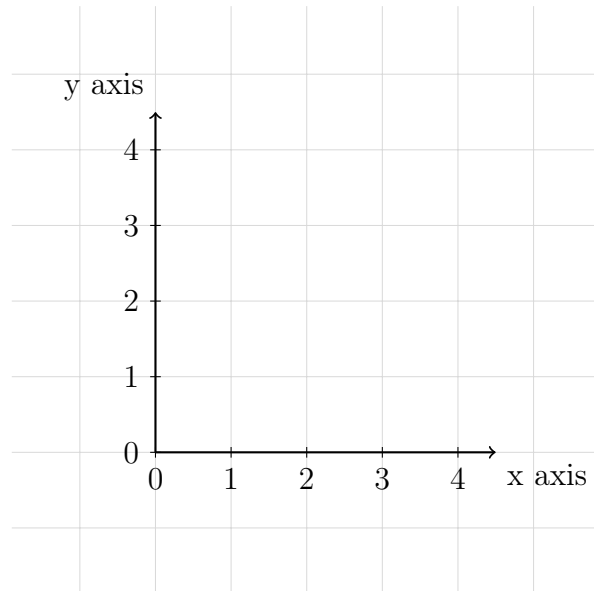
```
\draw[thick, ->] (0,0) -- (4.5,0);  
\draw[thick, ->] (0,0) -- (0,4.5);
```

Vi kan legge på tekst (*label*) ved å bruke nøkkelordet **node**. Nodene i TikZ har et sett med anker-punkter. Så når vi forteller at vi vil ha tekst ved **anchor = north west** for x-aksen vår, så plasserer den der i henhold til linjen vi allerede har tegnet.



```
\draw[thick, ->] (0,0) -- (4.5,0) node[anchor=north west] {x axis};
\draw[thick, ->] (0,0) -- (0,4.5) node[anchor=south east] {y axis};
```

Vi kan fortsette med tallene som skal gå langs aksene ved å bruke løkker:

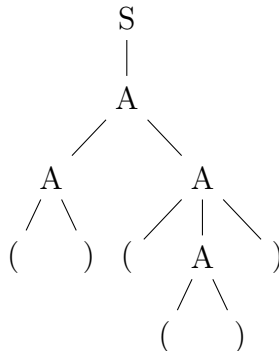


Denne løkken går over linjene vi allerede har tegnet, og setter en liten strek for hver centimeter. Og på dette anker-punktet, så skriver vi også et tall.

```
\begin{tikzpicture}
  \draw[step=1cm,gray!30,very thin] (-1.9,-1.9) grid (5.9,5.9);
  \draw[thick, ->] (0,0) -- (4.5,0) node[anchor=north west] {x axis};
  \draw[thick, ->] (0,0) -- (0,4.5) node[anchor=south east] {y axis};

  \foreach \x in {0,1,2,3,4}
    \draw (\x cm, 2pt) -- (\x cm, -2pt) node[anchor=north] {$\x$};
  \foreach \y in {0,1,2,3,4}
    \draw (2pt, \y cm) -- (-2pt, \y cm) node[anchor=west] {$\y$};
\end{tikzpicture}
```

3 Trær

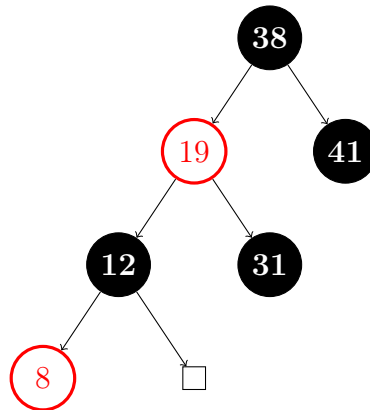


```
\begin{tikzpicture}[every node/.style={},
                    level 2/.style={sibling distance=20mm},
                    level 3/.style={sibling distance=10mm},
                    level distance=30pt]
\node {S}
  child { node{A}
    child { node {A}
      child { node {(} }
      child { node {)}} }
    }
  child { node {A}
    child { node {(} }
    child { node {A}
      child { node {(} }
      child { node {)}} }
    }
  child { node {}} }
}
;
\end{tikzpicture}
```

Vi begynner med å fortelle hvordan vi vil treet skal se ut. Argumentene til `tikzpicture` sier at vi ikke har noen felles stil for alle nodene, men vi kunne skrevet `every node/.style={red}` og alle nodene hadde blitt røde.

Videre forteller vi hvor stor avstand vi vil ha mellom søskennoder i de forskjellige lagene, og til slutt hvor stort rom vi vil ha mellom lagene.

3.1 Rød-svarte trær



Å tegne trær på denne måten krever ingen tilleggssbiblioteker fra TikZ. Dette er et eksempel på tegning med *parametre*. Utseende til nodene i treet er definert som parametre som har et sett med *.style*-opsjoner.

```
\tikzset{
  treenode/.style = {align=center, inner sep=0pt},

  % Sorte noder
  node_black/.style = {treenode, circle, white,
    font=\bfseries, draw=black,
    fill=black, text width=0.8cm},

  % Røde noder
  node_red/.style = {treenode, circle, red, draw=red,
    text width=0.8cm, very thick},

  % Null-pekere
  node_null/.style = {treenode, rectangle, draw=black,
    minimum width=0.3cm, minimum height=0.3cm}
}
```

Starter med å definere `treenode`, som er felles for alle typer noder. Røde og sorte noder tegnes som `circle`, hvor sorte noder har `fill=black` og tekstfarge `white`, mens røde noder har rødt omriss med `draw=red`, og tekstfarge `red`. Null-nodene sier vi skal være sorte `rectangle`. Tegnes som små kvadrater på $0.3\text{ cm} \times 0.3\text{ cm}$.

3.2 Bygge et tre

```
\begin{tikzpicture}[->,level/.style={ sibling distance = 2cm,
                                   level distance = 1.5cm }]
\node [node_black] {38}
  child {node [node_red] {19}
    child {node [node_black] {12}
      child {node [node_red] {8} }
      child {node [node_null] {} }
    }
    child {node [node_black] {31} }
  }
  child { node [node_black] {41} }
;
\end{tikzpicture}
```

Setter forskjellige opsjoner med:

```
{tikzpicture}[->, level/.style={sibling distance=2cm, level
                                distance)=1.5cm}]
```

Her sier vi at treet skal tegnes med piler (->), og at stilen (.style) for distansen mellom søskennoder skal være 2 cm, og distansen mellom barn og foreldre skal være 1.5 cm.

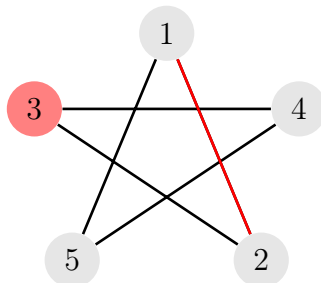
Videre så tegnes treet ved å definere roten:

```
\node [opt.] {node value} ;
```

Så kan man bygge treet ved å legge inn barna til roten, osv.:

```
\node [opt.] {node value}
  child {node [opt.] {node value} } ;
```

4 Grafer



Det fins enklere måter å tegne grafer på enn dette, men jeg syns denne måten er fin. Den krever heller ingen andre biblioteker eller pakker enn TikZ selv. Eksempel på en veldig mye enklere måte kommer på slutten av denne seksjonen.

Vi starter med å definere de forskjellige elementene til en graf.

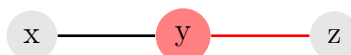
```
\begin{tikzpicture}
  \tikzstyle{vertex} = [circle,fill=black!10]
  \tikzstyle{selected vertex} = [vertex, fill=red!50]

  \tikzstyle{selected edge} = [draw,line width=1pt,-,red!100]
  \tikzstyle{edge} = [-,black,line width=1pt]
\end{tikzpicture}
```

Her fortelle vi at **vertexer** (eller noder), skal være sirkler som er fylt med sort med en gjennomsiktighet på 10%. Markerte noder skal også være fylt, da med en annen farge.



Kanter skal tegnes som sorte linjer (`[-, black ...]`). Og markerte kanter skal være røde.



4.1 Tegne grafen

For å plassere nodene rundt om på arket sier man hvor man vil de skal være. For eksempelet på toppen (grafene som har stjerne-form), er TikZ-koden som følger:

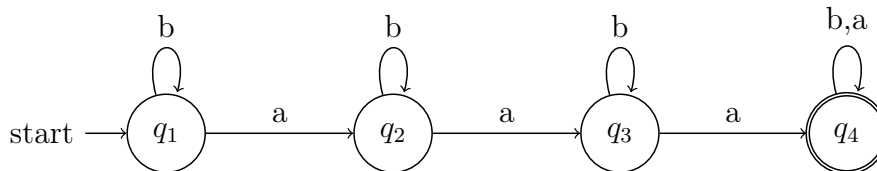
```
\begin{tikzpicture}[scale=5]
  \tikzstyle{vertex}          = [circle,fill=black!10]
  \tikzstyle{selected vertex} = [vertex, fill=red!50]

  \tikzstyle{selected edge}   = [draw,line width=1pt,-,red!100]
  \tikzstyle{edge}            = [-,black,line width=1pt]

  \node[vertex]          (v1) at (1.25,1.7) {1};
  \node[vertex]          (v2) at (1.5,1.1)  {2};
  \node[selected vertex] (v3) at (0.9,1.5)  {3};
  \node[vertex]          (v4) at (1.6,1.5)  {4};
  \node[vertex]          (v5) at (1,1.1)    {5};

  \draw[edge]            (v1)--(v2)--(v3)--(v4)--(v5)--(v1);
  \draw[selected edge]   (v1)--(v2);
\end{tikzpicture}
```

5 Automater



```
\begin{tikzpicture}[->,auto,node distance=3cm,line width=0.2mm]
  \node[initial,state] (A) {\q_1$};
  \node[state] (B) [right of=A] {\q_2$};
  \node[state] (C) [right of=B] {\q_3$};
  \node[state,accepting] (D) [right of=C] {\q_4$};

  \path (A) edge [loop above] node {b} (A)
        edge node {a} (B)
        (B) edge [loop above] node {b} (B)
        edge node {a} (C)
        (C) edge [loop above] node {b} (C)
        edge node {a} (D)
        (D) edge [loop above] node {b,a} (D);
\end{tikzpicture}
```

For denne måten å tegne automater på, så settes alle parametre som beskriver automaten i definisjonen til `tikzpicture`. Man må også inkludere `\usetikzlibrary{automata}`. Her har automaten følgende egenskaper:

```
{tikzpicture}[->, auto, node distance=3cm, line width=0.2mm]
```

Dette forteller oss at automaten skal tegnes med piler (`->`), nodene skal ha avstand på 3 cm, og linjene en tykkelse på 0,2 mm. Auto stiller teksten *over* linjene, i stedet for *på* linjene.

5.1 Automatens tilstander

En automat har tre typer tilstander: starttilstanden, vanlig tilstand(er), og aksepterende tilstand(er).

```
\node[state] (node-name) {name of state};
```

I tillegg til `[state]`, så kan man ha med opsjonen `[initial, state]` for starttilstanden, eller `[state, accepting]` for aksepterende tilstand.

5.2 Stien gjennom automaten

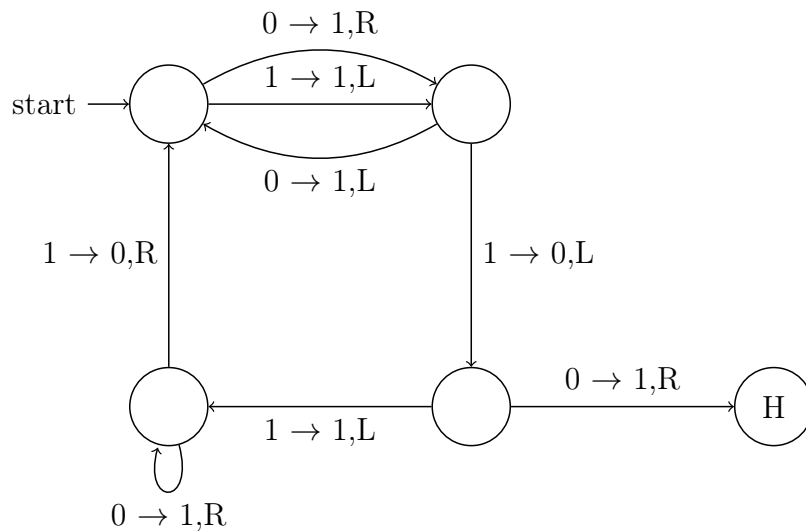
Stien tegnes gjennom en path. Denne konstrueres på følgende vis:

```
\path (from-node) edge [opt.] node {weight} (to-node).
```

Her kan [opt] være loop above/below, bend left/right.

Flittig bever

Her er en flittig 4-bever. Denne automaten dekker de fleste opsjoner.



```

\begin{tikzpicture}[->,auto,node distance=4cm,line width=0.2mm]
  \node[initial,state] (A) {};
  \node[state] (B) [below of=A] {};
  \node[state] (C) [right of=A] {};
  \node[state] (D) [below of=C] {};
  \node[state] (E) [right of=D] {H};

  \path (A) edge node {1 $\rightarrow$ 1,L} (C)
        (A) edge [bend left] node {0 $\rightarrow$ 1,R} (C)
        (C) edge [bend left] node {0 $\rightarrow$ 1,L} (A)
        (B) edge node {1 $\rightarrow$ 0,R} (A)
        (B) edge [loop below] node {0 $\rightarrow$ 1,R} (B)
        (D) edge node {1 $\rightarrow$ 1,L} (B)
        (C) edge node {1 $\rightarrow$ 0,L} (D)
        (D) edge node {0 $\rightarrow$ 1,R} (E);
\end{tikzpicture}

```

6 Resursser

Gøyale eksempler

- Enderman
- Dartboard
- India map

Lære mer?

- Introduksjon til Circuitikz
- Tankekart med TikZ
- Generere TikZ-kode fra GeoGebra