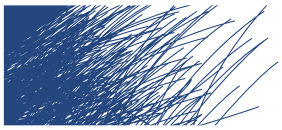


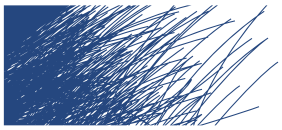
UNIVERSITEIT
GENT

Workshop: Ti*k*Z

PRIME

Pieter Belmans
Universiteit Antwerpen

24 april 2014




UNIVERSITEIT GENT

Deel 1

Inleiding

UNIVERSITEIT GENT

Waarom figuren maken in LaTeX?

\+ aanpasbaarheid

1. je document opnieuw builden maakt de afbeelding opnieuw: wijzigingen aanbrengen is dus even gemakkelijk als LaTeX-code aanpassen
2. je hebt geen externe software nodig

\+ consistentie

lettertypes, kleuren, tekstgroottes, ... zijn overal hetzelfde

\+ integratie met externe tools

als je toch externe software gebruikt is er integratie mogelijk

\+ vector graphics

geen problemen met lage resoluties
er kan oneindig ingezoomd worden

– de leercurve...

UNIVERSITEIT GENT

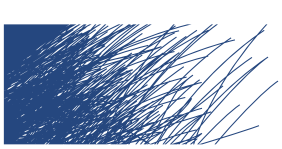
Waarom figuren maken in TikZ?

```
Making Greek letters is as easy as $\pi $.
```
Making Greek letters is as easy as π.

—Leslie Lamport, LaTeX: A Document Preparation System

```
Drawing an orange circle is as easy as
\tikz \fill [orange] (1ex,1ex) circle (1ex);.
```
Drawing an orange circle is as easy as  .

—Pieter Belmans




UNIVERSITEIT GENT

Wat is Ti*k*Z?

Ti*k*Z is een recursief acroniem voor

Ti*k*Z ist *kein* Zeichenprogramm

Dit betekent vooral dat Ti*k*Z niet bedoeld is om in te tekenen zoals in Paint, maar om "vector graphics" te produceren: we "beschrijven" de tekening.

PGF

Eigenlijk is Ti*k*Z een frontend voor PGF, Portable Graphics Format, soms wordt de combinatie PGF/Ti*k*Z genoemd.

Enkele maanden geleden is versie 3 uitgekomen, maar deze workshop is nog geschreven voor 2.10 (de versie die geïnstalleerd is).

Wat gaan we vandaag vooral *niet* doen?

Dit is *geen* grondige Ti*k*Z-cursus:

1. ik ben verre van een expert
2. de tijd is beperkt
3. Ti*k*Z is *gigantisch*
4. ik ben verre van een expert (en kan enkel uitleggen wat ik zelf ken en gebruik)

UNIVERSITEIT GENT

Wat gaan we vandaag dan *wel* doen?

1. korte introductie tot de syntax
2. waar kunnen we informatie vinden?
3. wat is er allemaal mogelijk?

Ik wil dus vooral laten zien *wat* er mogelijk is, niet *hoe* je het moet doen. Daarvoor dienen handleidingen en de vele beschikbare voorbeelden.
Achteraf is er tijd voor specifieke vragen.

UNIVERSITEIT GENT

Deel 2

Waar informatie te vinden?

UNIVERSITEIT GENT

Documentatie

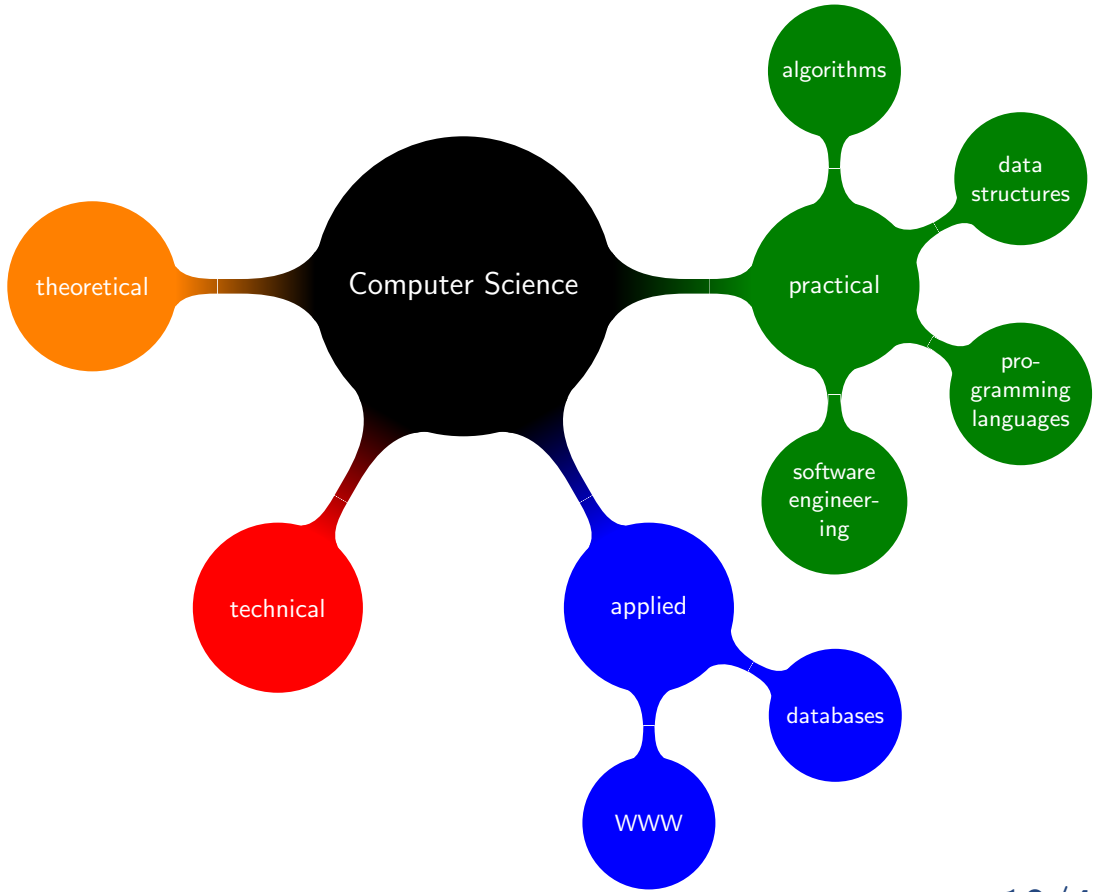
Op `http://ctan.org/pkg/pgf`:

1. de manual (voor versie 3.0.0), 1100+ pagina's
2. A very minimal introduction to Ti*k*Z

UNIVERSITEIT GENT

http://TeXample.net

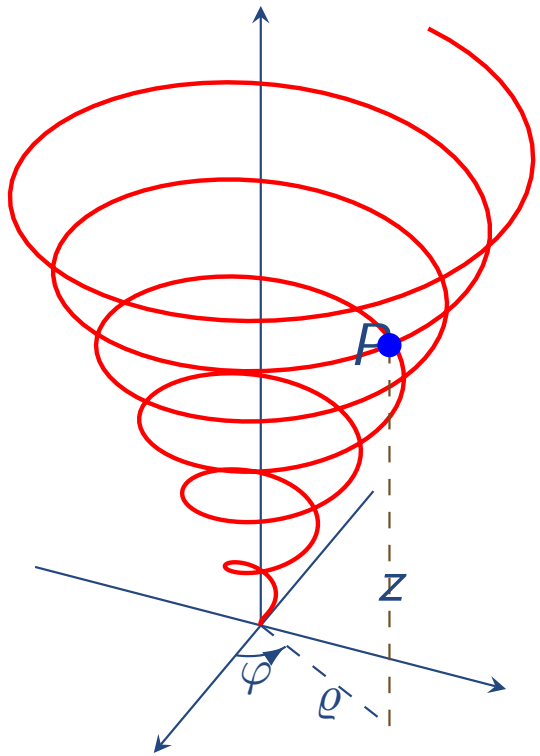
1. 350+ voorbeelden
2. gesorteerd in categorieën
3. bestaat al 8 jaar
4. heeft ook een lijst van interessante LaTeX blogs



UNIVERSITEIT GENT

http://PGFPlots.net

1. 50+ voorbeelden
2. gesorteerd in categorieën
3. bestaat nu een maand

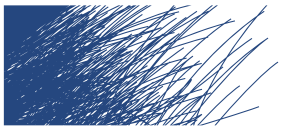


UNIVERSITEIT GENT

TEX—LATEX Stack Excange

Een vraag-en-antwoordforum voor alle LATEX-gerelateerde zaken:

1. 60000+ vragen over LATEX, `http://tex.stackexchange.com/questions?sort=votes` is een goede manier om snel veel over LATEX te leren
2. 7000+ vragen over Ti*k*Z, voor een overzicht zie `http://tex.stackexchange.com/questions/tagged/tikz-pgf`




Deel 3

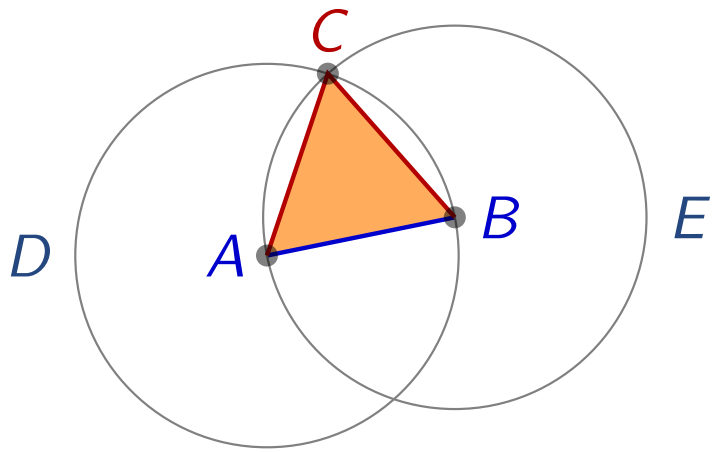
PGF/TikZ aan de hand van een voorbeeld

UNIVERSITEIT GENT

Uitgewerkt voorbeeld: een gelijkzijdige driehoek

In de *Elementen van Euclides* (3e eeuw voor Christus) worden vele constructies in vlakke meetkunde beschreven. We gaan nu een voorbeeld uit de Ti*k*Z manual bespreken:

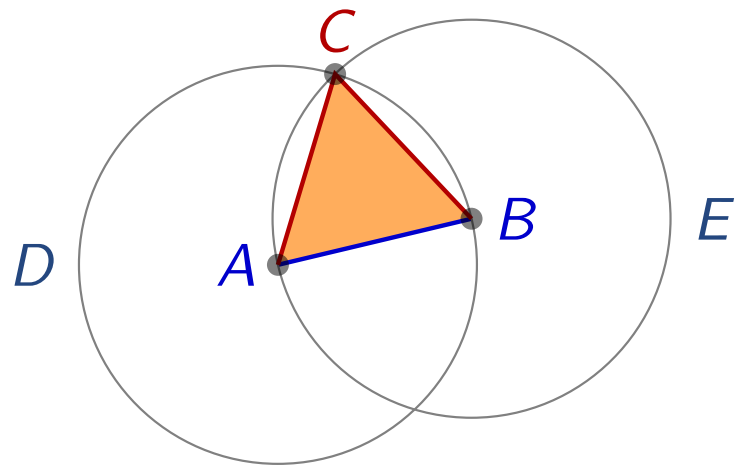
> *Hoe construeren we een gelijkzijdige driehoek op een gegeven lijnstuk?*

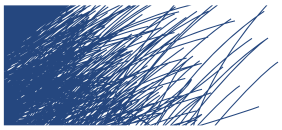


UNIVERSITEIT GENT

Stappenplan

1. het lijnstuk AB
2. de cirkels rond A en B
3. het snijpunt
4. de driehoek






UNIVERSITEIT GENT

Euclides: de configuratie

```
\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{calc,intersections,through,backgrounds}

\begin{document}
  % add picture
\end{document}
```

TikZ bevat vele libraries die bepaalde zaken gemakkelijker maken. In dit voorbeeld zullen we er vier gebruiken.

UNIVERSITEIT GENT

Euclides: het lijnstuk AB (1)

```
\begin{tikzpicture}
  \coordinate (A) at (0,0);
  \coordinate (B) at (1.25,0.25);

  \draw[blue] (A) -- (B);
\end{tikzpicture}
```

1. `\draw` is het commando om iets te tekenen
2. `--` is een gewone lijn tussen twee punten
3. we konden ook `\draw[blue] (0,0) -- (1.25,0.25);` gebruiken, maar nu kunnen we de coördinaten herbruiken

Volgende stap

Hoe voegen we namen van punten toe?

Euclides: het lijnstuk AB (2)

```
\begin{tikzpicture}
  \coordinate[label=left:\textcolor{blue}{$A$}]
    (A) at (0,0);
  \coordinate[label=right:\textcolor{blue}{$B$}]
    (B) at (1.25,0.25);

  \draw[blue] (A) -- (B);
\end{tikzpicture}
```

1. we gebruiken puntkomma's voor het einde van een commando
2. we mogen nieuwe regels beginnen als dat beter leest (of als de slides te smal zijn)

Volgende stap

Hoe kunnen we A en B een willekeurige positie laten aannemen?

Euclides: het lijnstuk AB (3)

```
\begin{tikzpicture}
  \coordinate[label=left:\textcolor{blue}{$A$}]
    (A) at ($ (0,0) + .1*(rand,rand) $);
  \coordinate[label=right:\textcolor{blue}{$B$}]
    (B) at ($ (1.25,0.25) + .1*(rand,rand) $);

  \draw[blue] (A) -- (B);
\end{tikzpicture}
```

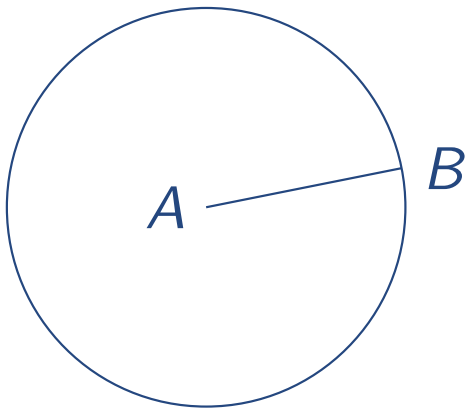
1. de library `calc` laat ons toe om heel flexibel met coördinaten te rekenen: we gebruiken dollartekens om dit aan te duiden
2. meestal is `calc` zelfs niet nodig (!)

Volgende stap

Nu de cirkels.

Euclides: de cirkels rond A en B (1)

```
\begin{tikzpicture}
  \coordinate[label=left:$A$]  (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (1.25,0.25);
  \draw (A) -- (B);

  \draw (A) let
              \p1 = ($ (B) - (A) $)
            in
              circle ({veclen(\x1,\y1)});
\end{tikzpicture}
```



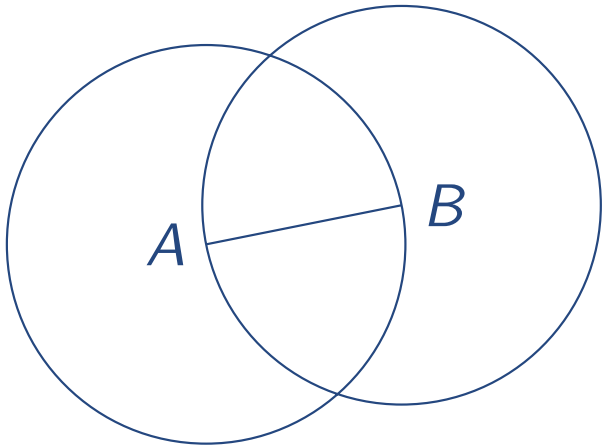
1. een cirkel tekenen `\draw (0,0) circle (1);`
2. we berekenen de coördinaten voor $B - A$ (`let ... in` is met variabelen werken)

Volgende stap

Nu nog de tweede cirkel.

Euclides: de cirkels rond A en B (2)

```
\begin{tikzpicture}
  \coordinate[label=left:$A$]  (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (1.25,0.25);
  \draw (A) -- (B);

  \draw let \p1 = ($ (B) - (A) $),
            \n2 = {veclen(\x1,\y1)}
        in
          (A) circle (\n2)
          (B) circle (\n2);
\end{tikzpicture}
```



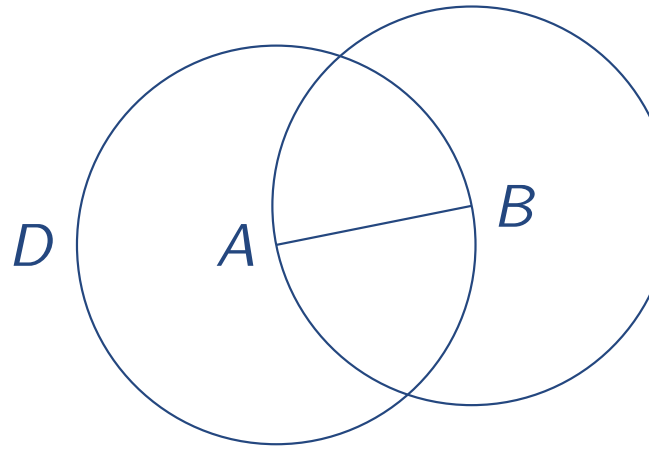
1. `\p1` zijn coördinaten, `\n1` is een getal
2. één `\draw` kan meerdere zaken tekenen

Volgende stap

We kunnen echter ook meteen cirkels door punten tekenen.

Euclides: de cirkels rond A en B (3)

```
\begin{tikzpicture}
  \coordinate[label=left:$A$]  (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (1.25,0.25);
  \draw (A) -- (B);

  \node[draw,circle through=(B),label=left:$D$]
    at (A) {};
  \node[draw,circle through=(A),label=right:$E$]
    at (B) {};
\end{tikzpicture}
```



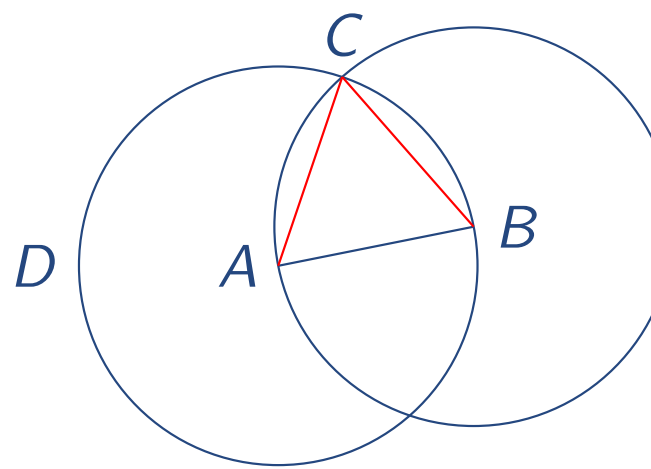
1. `\node` is een combinatie van `\coordinate` en `\draw`
2. de lege accolades zouden gebruikt kunnen worden voor tekst

Volgende stap

Het snijpunt bepalen.

Euclides: het snijpunt (1)

```
\begin{tikzpicture}
  \coordinate[label=left:$A$]  (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (1.25,0.25);
  \draw (A) -- (B);

  \node (D) [name path=D,draw,circle through=(B),
             label=left:$D$]  at (A) {};
  \node (E) [name path=E,draw,circle through=(A),
             label=right:$E$] at (B) {};

  \path[name intersections={of=D and E}];

  \coordinate[label=above:$C$] (C) at (intersection-1);

  \draw[red] (A) -- (C);
  \draw[red] (B) -- (C);
\end{tikzpicture}
```



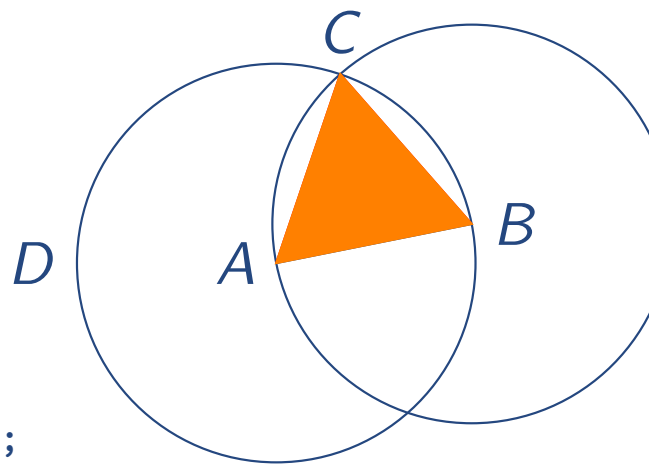
Euclides: het snijpunt (2)

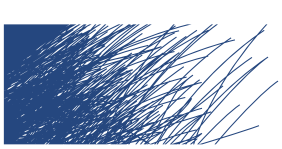
1. we berekenen een `\path` (= een reeks lijnsegmenten), dus in dit geval het (unieke) lijnstuk tussen de doorsnedes
2. we hebben de variabelen `intersection-1` en `intersection-2` die aangemaakt worden door de `intersections` library
3. we introduceren de naam `C` en tekenen twee lijnstukken

UNIVERSITEIT GENT

Euclides: de driehoek

```
\begin{tikzpicture}
  \coordinate[label=left:$A$]  (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (1.25,0.25);
  \draw (A) -- (B);

  \node (D) [name path=D,draw,circle through=(B),
             label=left:$D$]  at (A) {};
  \node (E) [name path=E,draw,circle through=(A),
             label=right:$E$] at (B) {};

  \path[name intersections={of=D and E}];

  \coordinate[label=above:$C$] (C) at (intersection-1);

  \draw[red] (A) -- (C);
  \draw[red] (B) -- (C);

  \draw[orange, fill] (A) -- (B) -- (C) -- cycle;
\end{tikzpicture}
```






UNIVERSITEIT GENT

Euclides: perfectioneren

1. we gebruiken variabelen voor de kleuren:

```
\colorlet{input}{blue!80!black}
\colorlet{triangle}{orange!80!white}
\colorlet{output}{red!70!black}
```

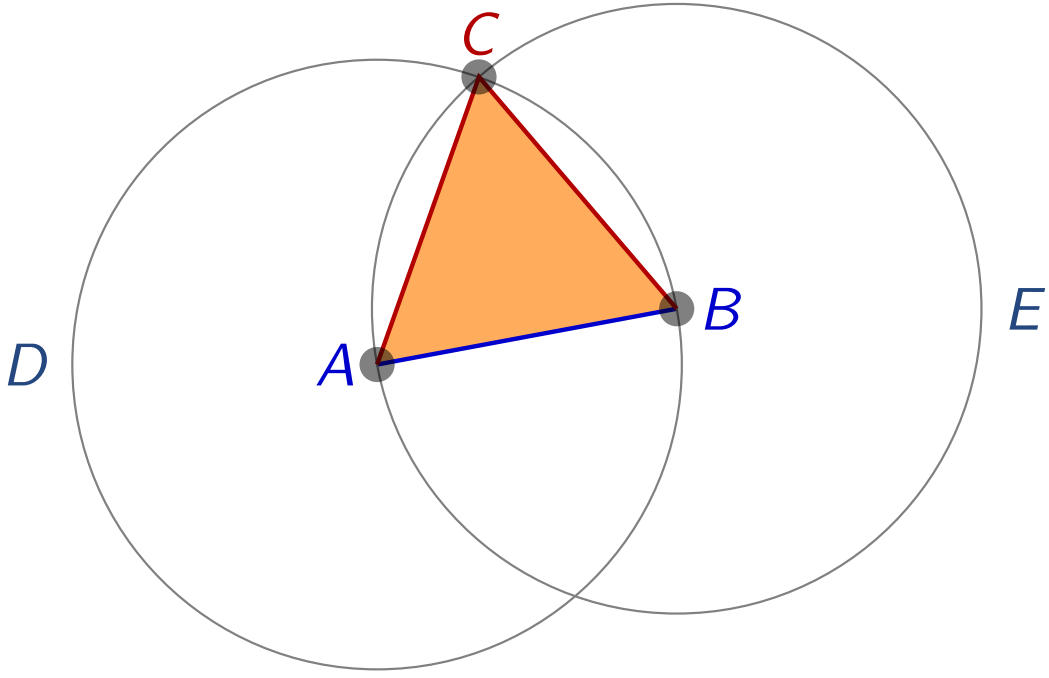
   (en passen de code aan)

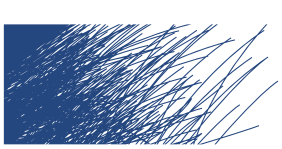
2. we tekenen onze punten:

```
\foreach \point in {A,B,C}
  \fill [black,opacity=.5] (\point) circle (2pt);
```

UNIVERSITEIT GENT

Euclides: het resultaat





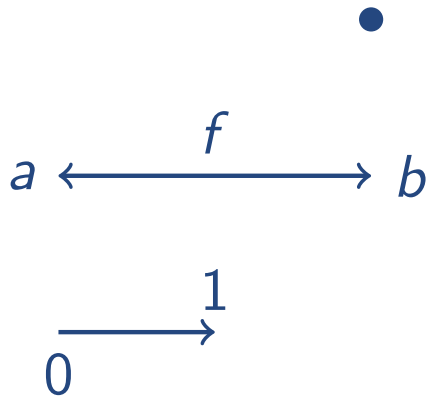

UNIVERSITEIT GENT

Herhaling belangrijke zaken

1. `\draw` (eigenlijk `\path [draw]`) om zaken te tekenen
2. `\coordinate` om coördinaten op te slaan
3. een lijn tekenen: `\draw (0,0) -- (0,1);`
4. een label toevoegen: `\draw (0,0) -- (0,1) node {$f$};`
5. opvullen: `\draw [fill]`
6. een cirkel: `\draw (0,0) circle (2cm);`

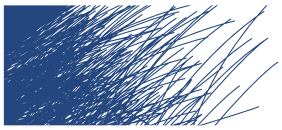
UNIVERSITEIT GENT

Oefening



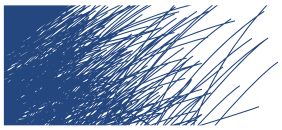
Oplossing

```
\begin{tikzpicture}
  \draw[thick, ->]
    (0,0) node[below] {$0$}
    --
    (1,0) node[above] {$1$};
  \draw[thick, <->]
    (0,1) node[left] {$a$}
    to node[above]{$f$}
    (2,1) node[right] {$b$};
  \draw[fill] (2,2) circle (2pt);
\end{tikzpicture}
```




Deel 4

`pgfplots` aan de hand van voorbeelden




UNIVERSITEIT GENT

pgfplots

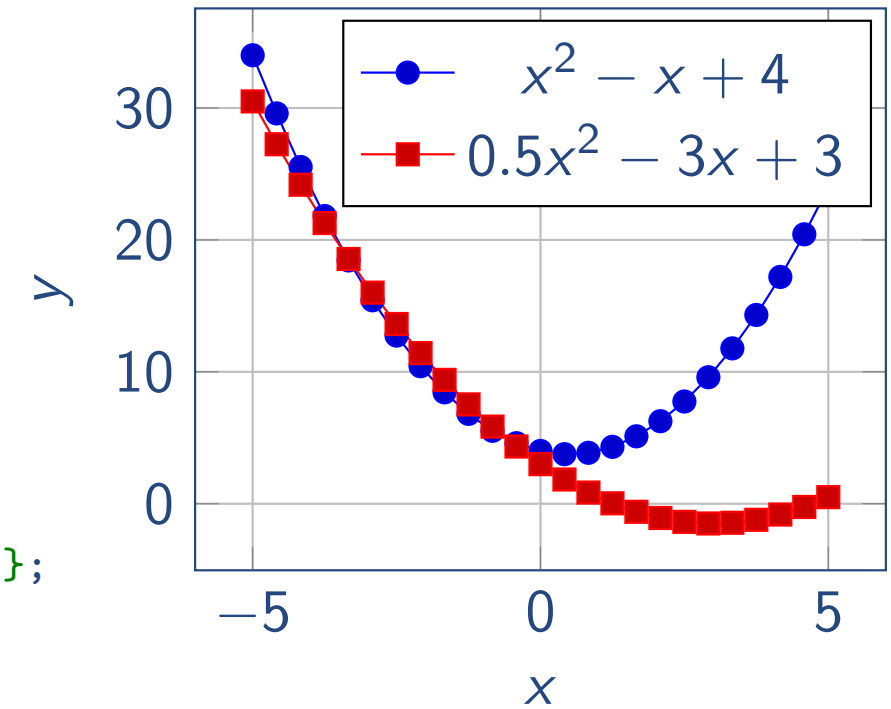
Net zoals Ti*k*Z een frontend is voor PGF, en vooral bedoeld om tekeningen te maken, is `pgfplots` een frontend om plots te maken.

1. `http://pgfplots.net`
2. `http://pgfplots.sourceforge.net/gallery.html`

UNIVERSITEIT GENT

Functies plotten

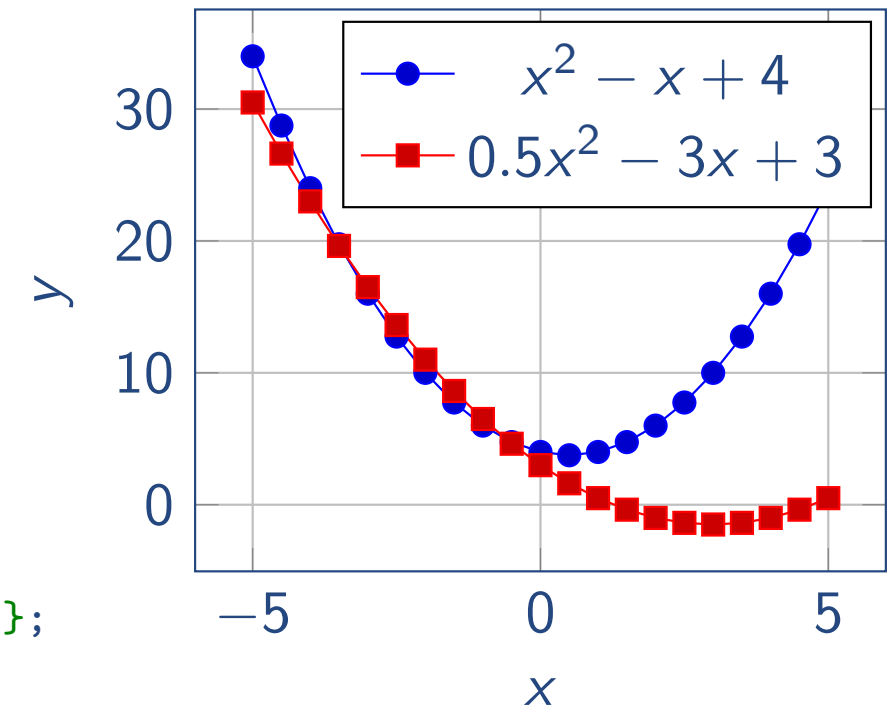
```
\begin{tikzpicture}
  \begin{axis}[
    grid=major,
    xlabel=$x$,
    ylabel=$y$
  ]
    \addplot {x^2 - x + 4};
    \addlegendentry{$x^2-x+4$};
    \addplot {.5*x^2 - 3*x + 3};
    \addlegendentry{$0.5x^2-3x+3$};
  \end{axis}
\end{tikzpicture}
```



UNIVERSITEIT GENT

Datasets plotten

```
\begin{tikzpicture}
  \begin{axis}[
    grid=major,
    xlabel=$x$,
    ylabel=$y$
  ]
    \addplot table[x = x, y = f]
      {tikz/pgfplots/data.dat};
    \addlegendentry{$x^2-x+4$};
    \addplot table[x = x, y = g]
      {tikz/pgfplots/data.dat};
    \addlegendentry{$0.5x^2-3x+3$};
  \end{axis}
\end{tikzpicture}
```

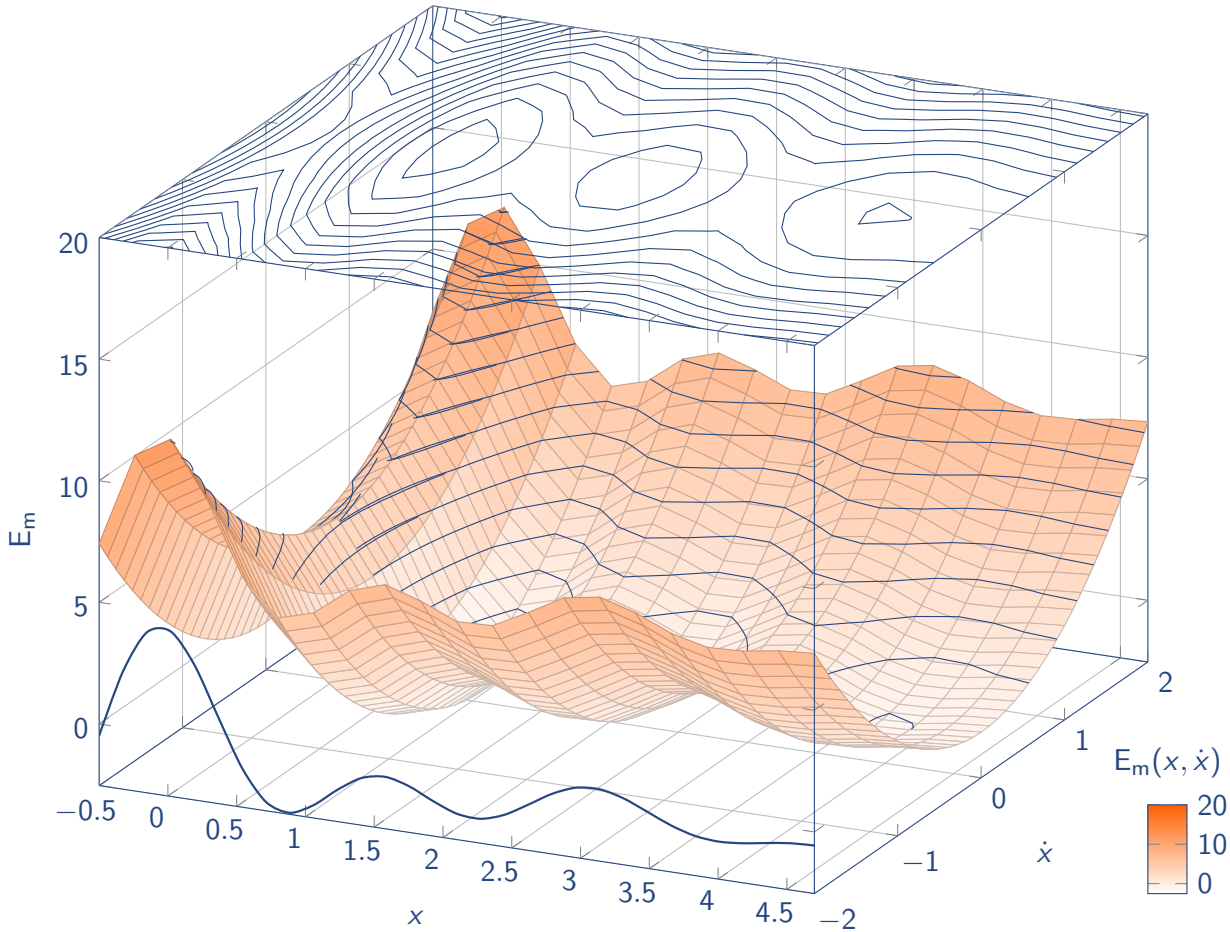


De dataset

x	f	g
-5.0	34.0	30.5
-4.5	28.75	26.625
-4.0	24.0	23.0
-3.5	19.75	19.625
-3.0	16.0	16.5
-2.5	12.75	13.625
-2.0	10.0	11.0
-1.5	7.75	8.625
-1.0	6.0	6.5

UNIVERSITEIT GENT

Contour plots

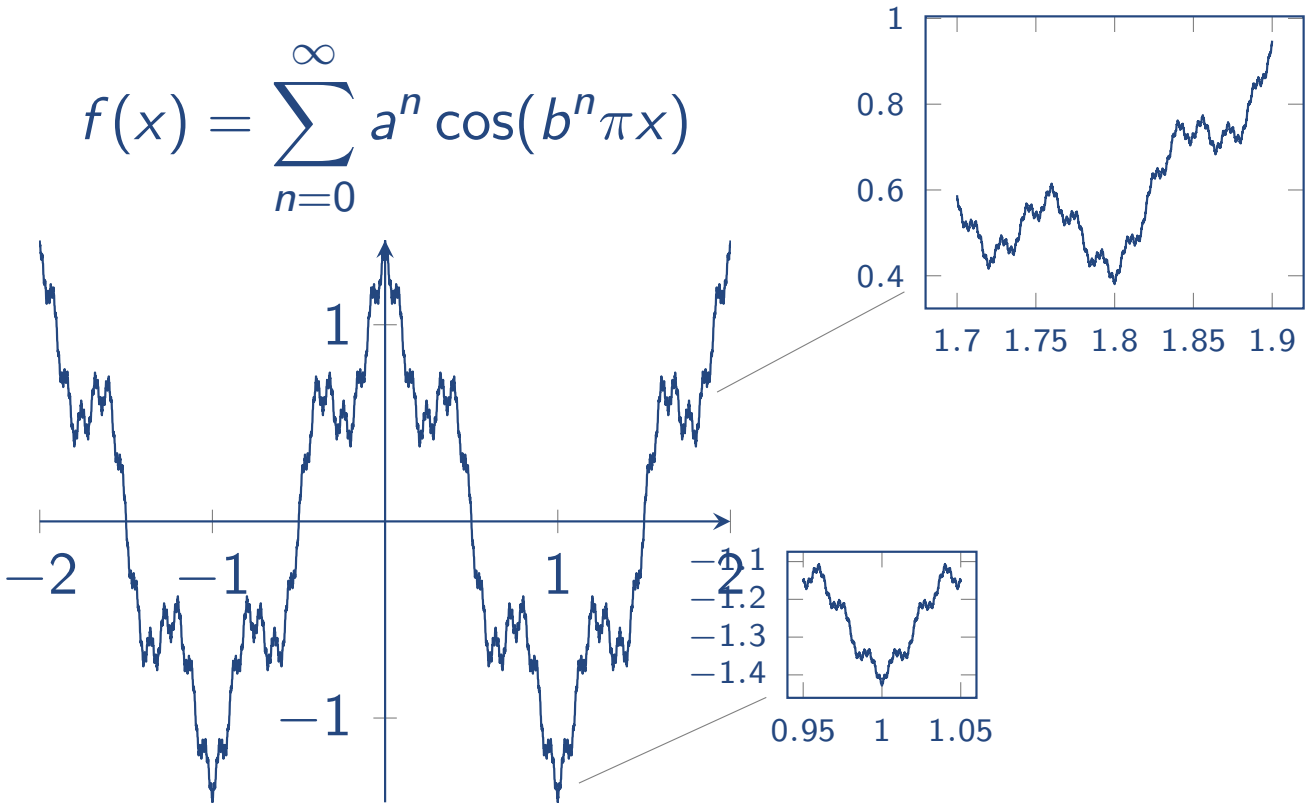


http://pgfplots.net/tikz/examples/contour-surface/

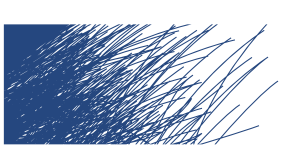
UNIVERSITEIT GENT

Weierstrassfunctie

$$f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x)$$



http:/pgfplots.net/tikz/examples/weierstrass-function/




UNIVERSITEIT GENT

Integratie met externe tools

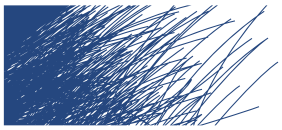
`matlab2tikz` `https://github.com/nschloe/matlab2tikz`
exporteer Matlab-figuren meteen naar Ti*k*Z

`matplotlib2tikz`
`https://github.com/nschloe/matplotlib2tikz`
exporteer matplotlib-figuren meteen naar Ti*k*Z

SAGE

matplotlib is de library die gebruikt wordt in SAGE om figuren te plotten

`sagetex` niet per se voor plots, maar wel erg handig




Deel 5

Uitbreidingen op TikZ

UNIVERSITEIT GENT

Libraries

Zie §IV in versie 2.10, of §V in versie 3.0 voor de *libraries* (= intern).
Te gebruiken via:

```
\usetikzlibrary {lindenmayersystems}
```

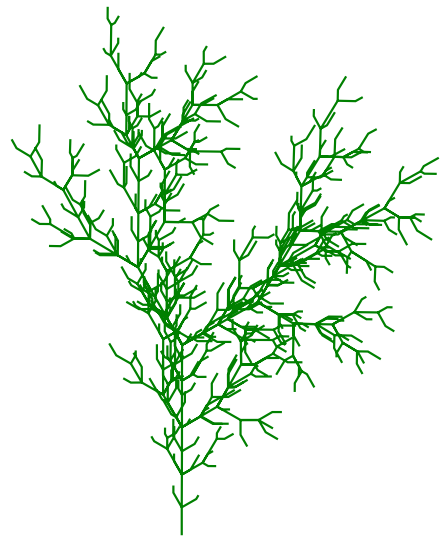
Vraag

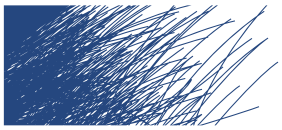
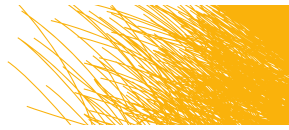
Wat doet deze library?

UNIVERSITEIT GENT

Lindenmayersystemen

```
\begin{tikzpicture}
\draw[green!50!black, rotate=90]
  [l-system={rule set={F -> FF-[-F+F]+[+F-F]},
    axiom=F, order=4, step=2pt,
    randomize step percent=25, angle=30,
    randomize angle percent=5}]
  lindenmayer system;
\end{tikzpicture}
```




Packages

Zie `http://ctan.org/search?phrase=tikz` voor *packages* (= extern): 90+ resultaten

`\usepackage {...}`

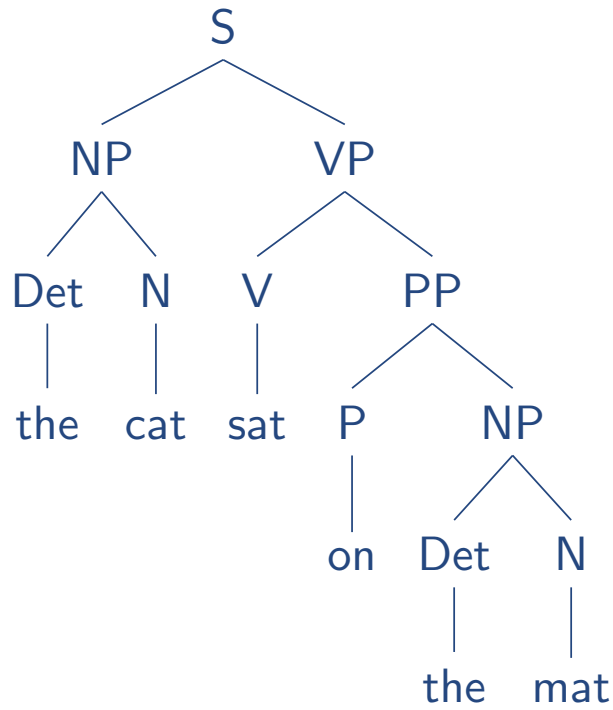
1. `tikz-qtree`
2. `tqft`
3. `tikz-cd`
4. ...

UNIVERSITEIT GENT

tikz-qtree

Automatisch bomen tekenen:

```
\begin{tikzpicture}[scale = .8]
  \Tree [.S [.NP [.Det the ] [.N cat ] ]
            [.VP [.V sat ]
                 [.PP [.P on ]
                      [.NP [.Det the ]
                      [.N mat ] ] ] ] ]
\end{tikzpicture}
```

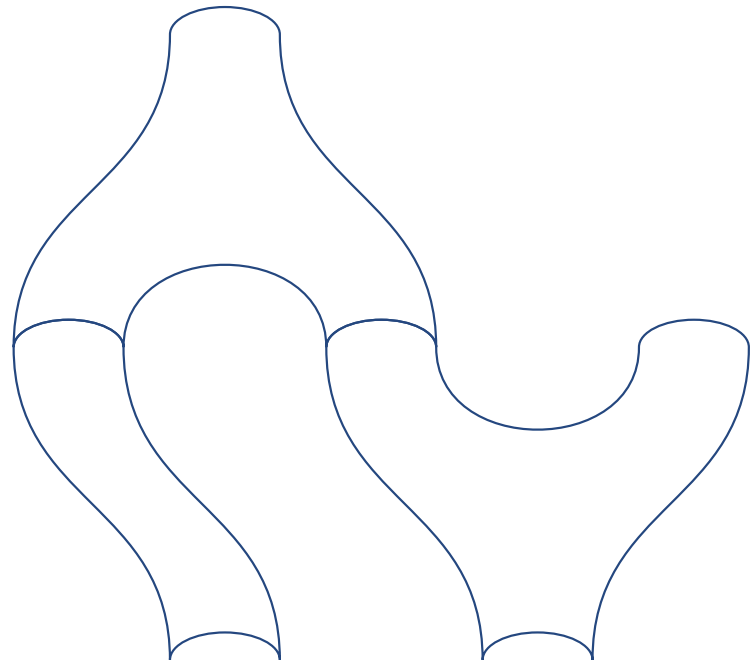


UNIVERSITEIT GENT

tqft

Om diagrammen in topologische quantumveldentheorie te tekenen:

```
\begin{tikzpicture}[scale = .8]
  \node[draw, tqft/pair of pants]
    (a) {};
  \node[draw, tqft/cylinder to next,
        anchor = incoming boundary 1]
    (c) at (a.outgoing boundary 1) {};
  \node[draw, tqft/reverse pair of pants,
        anchor = incoming boundary 1]
    at (a.outgoing boundary 2) (b) {};
\end{tikzpicture}
```

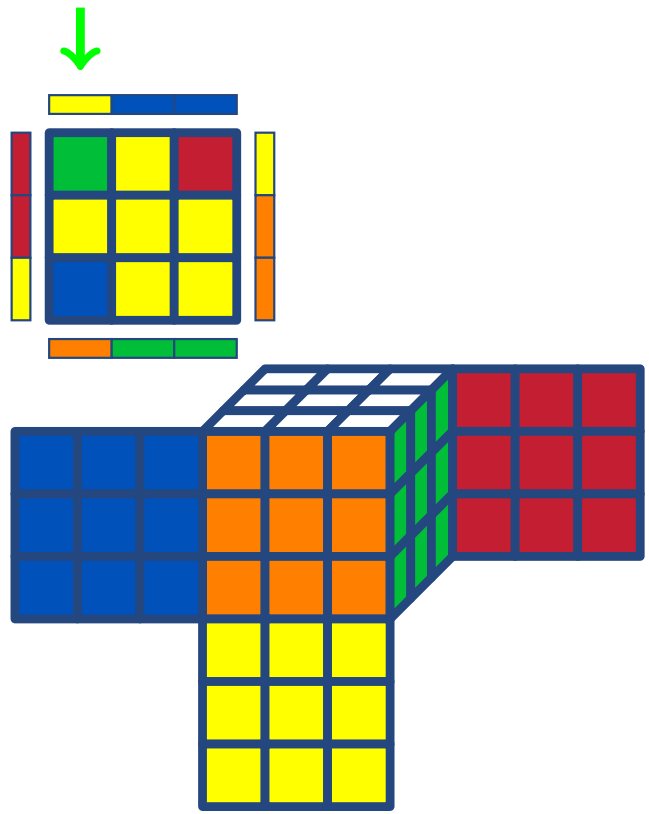


UNIVERSITEIT GENT

rubikcube

Om Rubik's cubes te tekenen:

```
\begin{tikzpicture}[scale=.4]
  \DrawRubikLayerFace{G}{Y}{R}
                     {Y}{Y}{Y}
                     {B}{Y}{Y}
  \DrawRubikLayerSideT {Y}{B}{B}
  \DrawRubikLayerSideLR{R}   {Y}
                       {R}   {O}
                       {Y}   {O}
  \DrawRubikLayerSideB {O}{G}{G}
  \draw[->,ultra thick,color=green]
    (0.5,5) -- (0.5, 4);
\end{tikzpicture}

\begin{tikzpicture}[scale=.4]
  \RubikCubeSolved
  \DrawRubikCubeFlat
\end{tikzpicture}
```

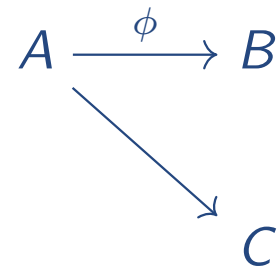


UNIVERSITEIT GENT

Commutatieve diagrammen

We gebruiken `\usepackage{tikz-cd}`.

```
\begin{equation*}
  \begin{tikzcd}
    A \arrow{rd} \arrow{r}{\phi} & B \\
                                 & C
  \end{tikzcd}
\end{equation*}
```



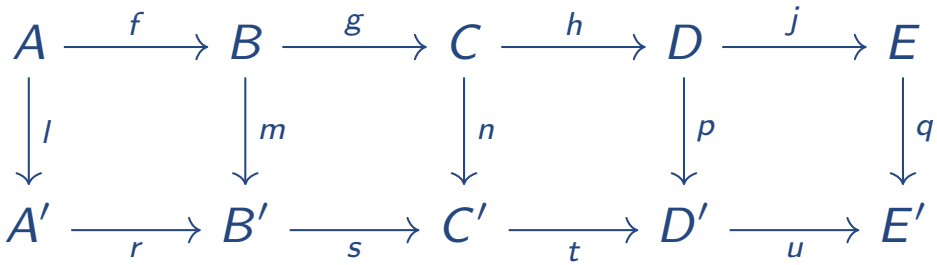
1. de *eerste parameter* van `\arrow` is altijd de richting: een combinatie van `u`, `d`, `r`, `l`, de *tweede* een (optioneel) label
2. vertices zoals we tabellen schrijven

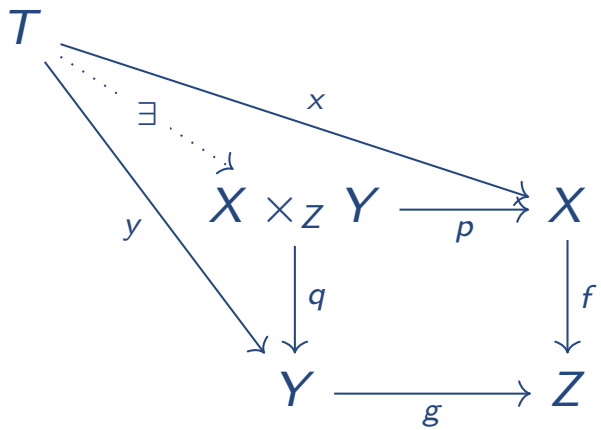
Opgepast

Pijlen kunnen enkel naar bestaande vertices (lastig debuggen. . .)

UNIVERSITEIT GENT

Oefening





Oplossing voor de tweede oefening

```
\begin{equation*}
  \begin{tikzcd}
    T
    \arrow{drr}{x}
    \arrow[swap]{ddr}{y}
    \arrow[dotted]{dr}[description]{\exists} & & \\
      & X \times_Z Y \arrow[swap]{r}{p} \arrow{d}{q}
        & X \arrow{d}{f} \\
      & Y \arrow[swap]{r}{g} &Z
  \end{tikzcd}
\end{equation*}
```