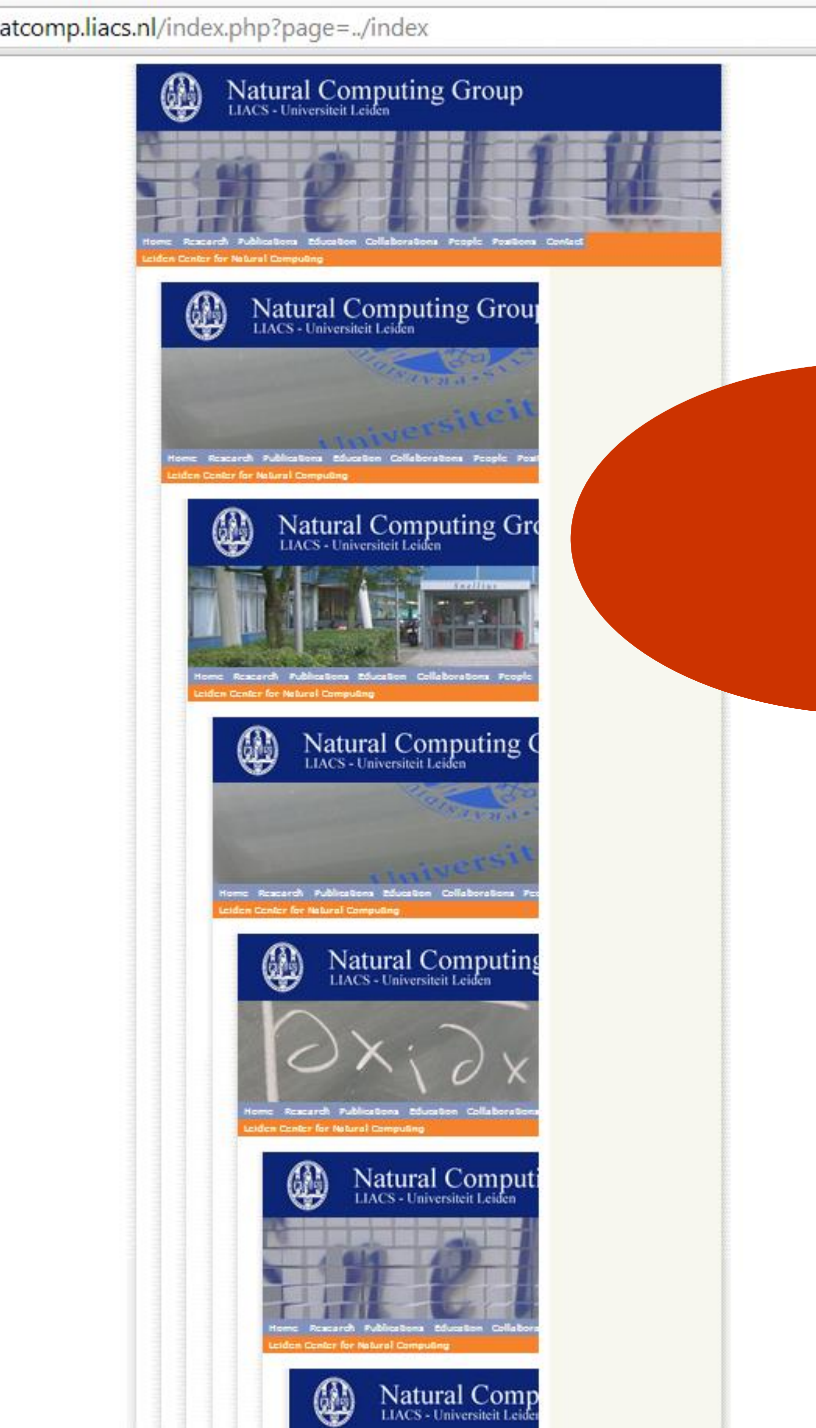


Recursie en inductie

recursie en inductie

Geen hoofdstuk in Schaum, maar ‘volledige inductie’ komt wel aan de orde als bewijstechniek.

Recursie is een programmeertechniek, zie ProgMet. Hier een manier om ‘verzamelingen’ te definiëren.

Uw docent heeft een klein dictaatje geschreven. Zie site.

inductie en recursie

we spreken van een inductieve definitie: in termen van kleinere onderdelen.

recursie is als we dat gebruiken als rekentechniek:
- de opdracht wordt verdeeld in kleinere stukken die worden opgelost
- het uiteindelijke antwoord wordt dan bepaald op grond van die deelantwoorden.

een inductief bewijs is een methode om een eigenschap te bewijzen (voor objecten die inductief gedefinieerd zijn). ook wel volledige inductie als het over $\mathbb{N}$ gaat.

iteratie is gewoon herhalen, een simpele loop bijvoorbeeld.

$a_0 = 0$
$a_n = a_{n-1} + n \quad (n \geq 1)$

$a_0 = 0$
$a_1 = a_0 + 1 = 0 + 1 = 1$
$a_2 = a_1 + 2 = 1 + 2 = 3$
$a_3 = a_2 + 3 = 3 + 3 = 6$

iteratief

0, 1, 3, 6, 10, …

inductie en recursie

$a_0 = 0$

$a_n = a_{n-1} + n \quad (n \geq 1)$

$a_0 = 0$

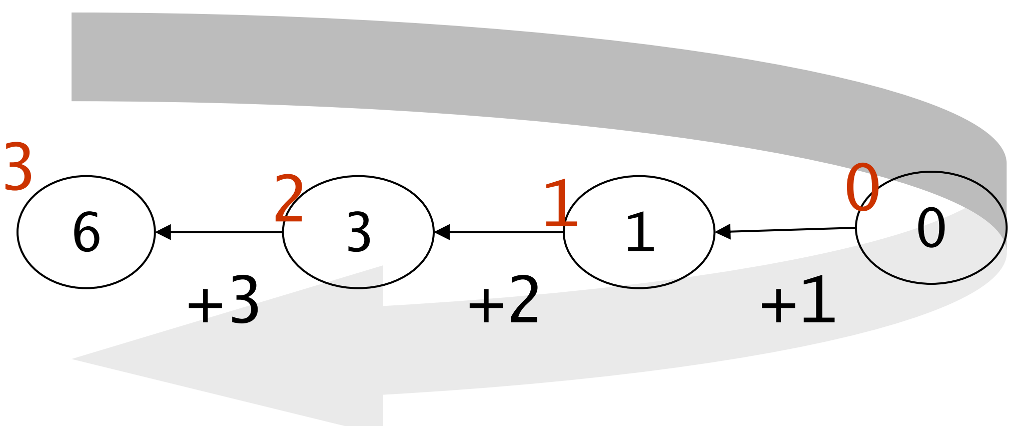
$a_1 = a_0 + 1 = 0 + 1 = 1$

$a_2 = a_1 + 2 = 1 + 2 = 3$

$a_3 = a_2 + 3 = 3 + 3 = 6$

iteratief

0, 1, 3, 6, 10, …



$a_3 =$

$a_2 + 3 =$

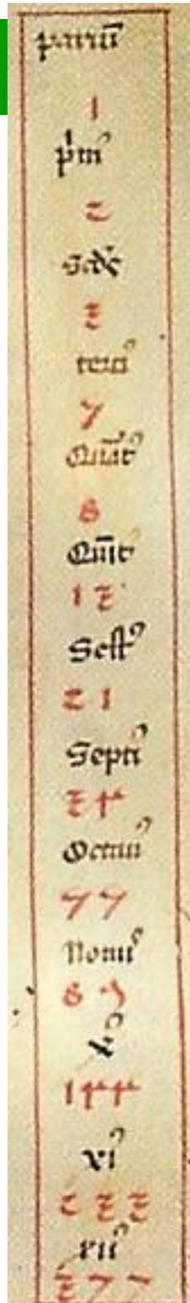
$a_1 + 2 + 3 = a_1 + 5 =$

$a_0 + 1 + 5 = a_0 + 6 =$

$0 + 6 = 6$

liber abaci (1202)

'rekenboek'
Leonardo van Pisa, zoon van Bonacci



0	1
I	2
II	3
III	5
IV	8
V	13
VI	21
VII	34
VIII	55
IX	89
X	144
XI	233
XII	377

How Many Pairs of Rabbits Are Created by One Pair in One Year

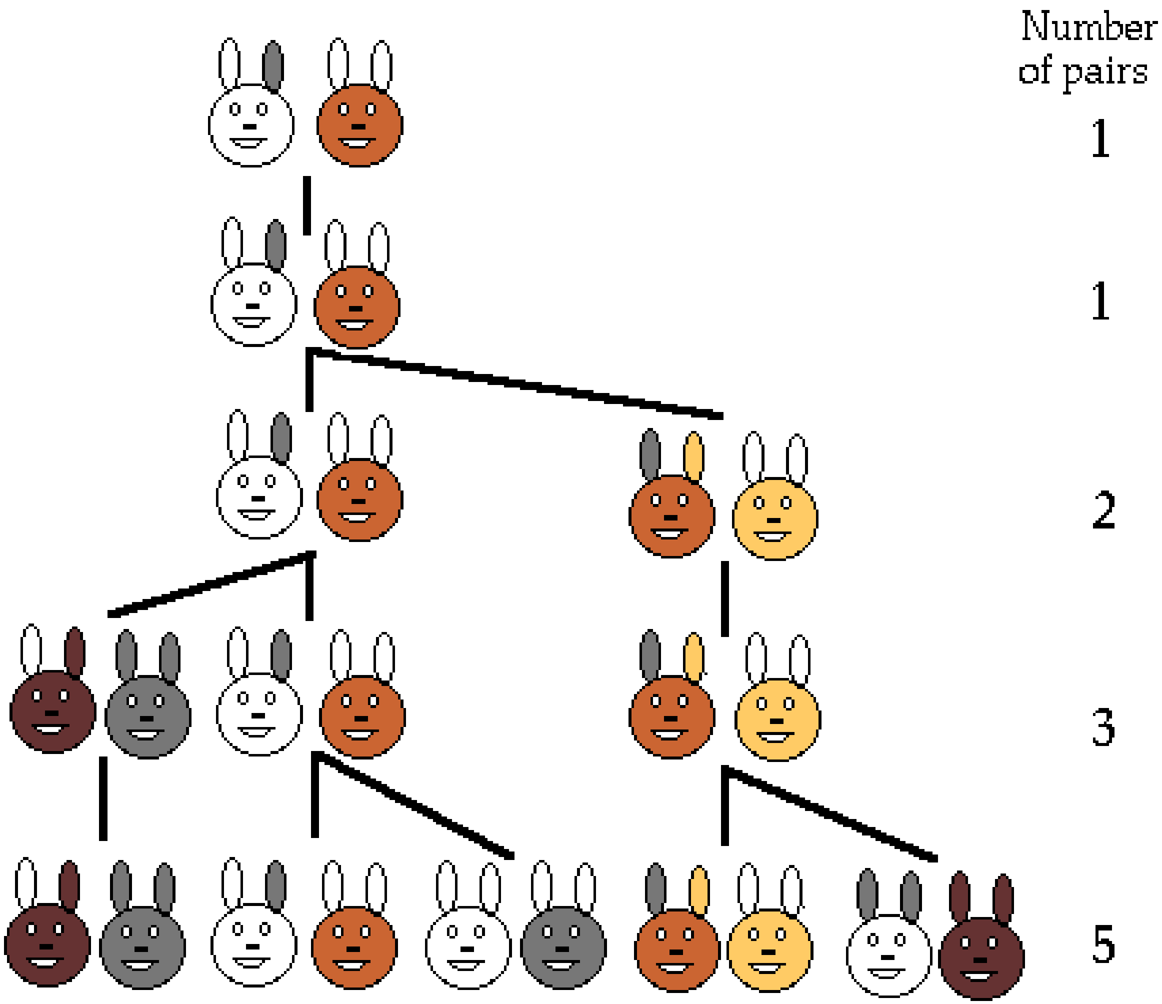
A certain man had one pair of rabbits together in a certain enclosed place, and one wishes to know how many are created from the pair in one year when it is the nature of them in a single month to bear another pair, and in the second month those born to bear also.

engelse vert. Laurence Sigler (2002)

Fibonacci

$a_0 = 0 \qquad a_1 = 1$

$a_{n+1} = a_n + a_{n-1} \quad (n \geq 1)$



inductieve definitie

Fibonacci

$a_0 = 0 \qquad a_1 = 1$

$a_{n+1} = a_n + a_{n-1} \quad (n \geq 1)$

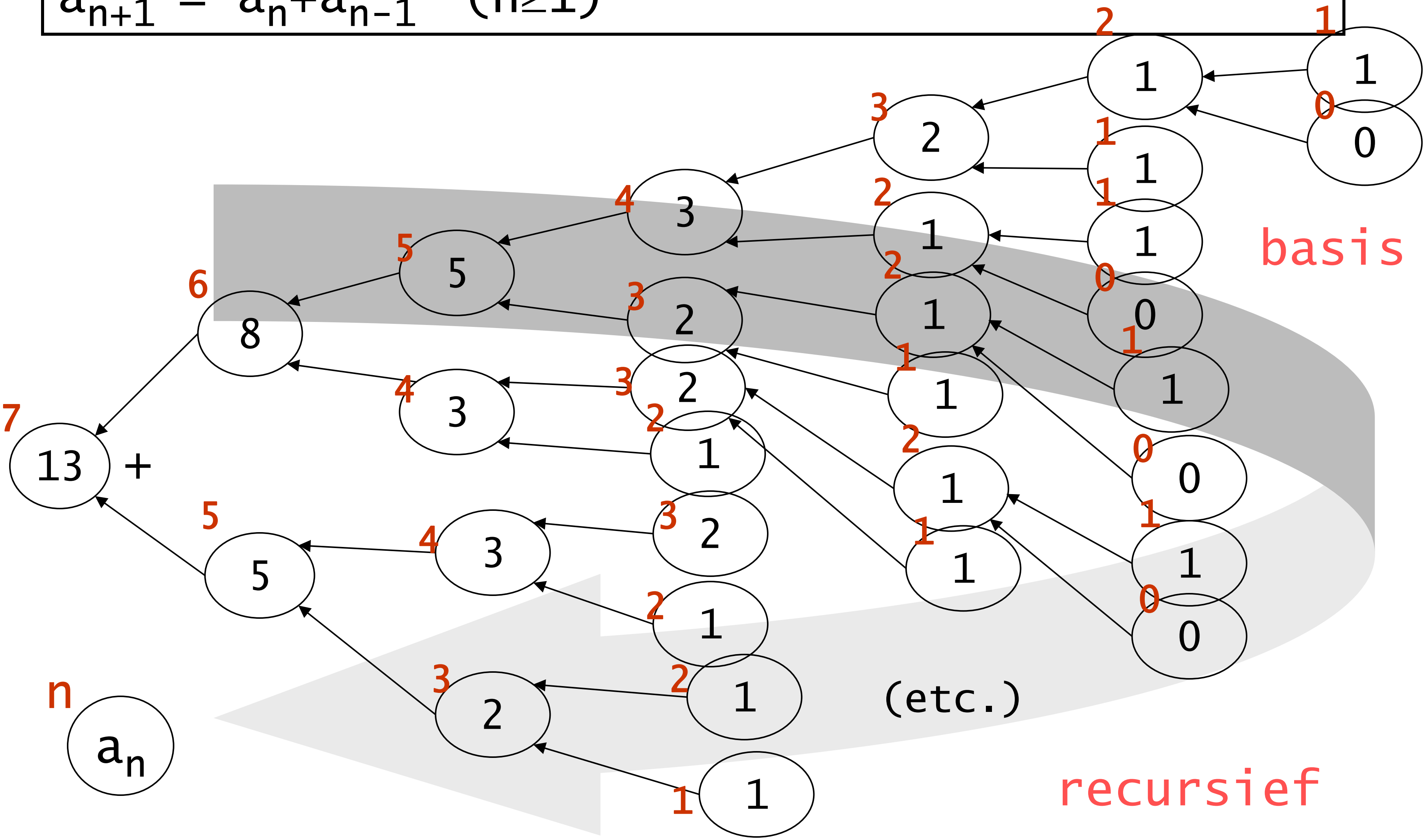
iteratief

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

Fibonacci

$a_0 = 0 \qquad a_1 = 1$

$a_{n+1} = a_n + a_{n-1} \quad (n \geq 1)$



Fibonacci

de Fibonacci getallen worden inductief gedefinieerd: als we twee opeenvolgende weten, dan is de volgende bekend.

op die manier kunnen we van klein naar groot de getallen berekenen.

we kunnen ook recursief werken: om een Fibo getal te bepalen moeten eerst de twee voorgangers worden bepaald die opgeteld worden. maar daarvoor berekenen we eerst ook elk weer twee voorgangers. zonder extra aandacht wordt zo veel werk dubbel gedaan.

inductief

van klein naar groot

bottom-up

recursief

in termen van zichzelf

top-down

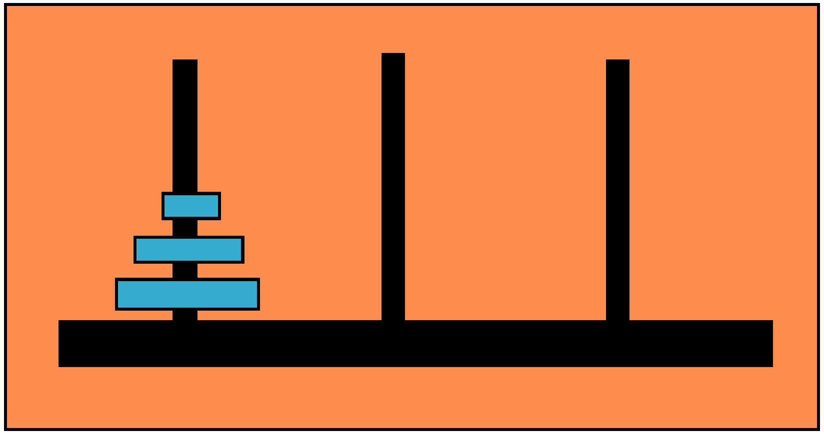
postscript definities

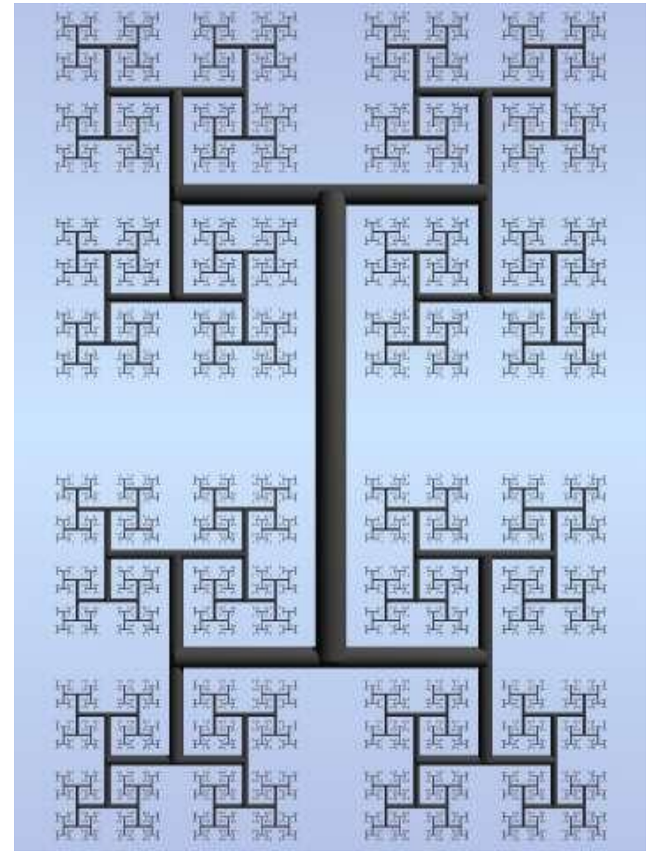
```
%!PS
/order 5 def

/X {
  dup 0 ne
  {1 sub 4 {dup} repeat - F X + + F Y -}
  if pop
} def

/Y {
  dup 0 ne
  {1 sub 4 {dup} repeat + F X - - F Y +}
  if pop
} def

/F {
  0 eq { 10 0 rlineto } if
} bind def

/- { -45 rotate } bind def
/+ { 45 rotate } bind def
```

```
newpath
220 180 moveto
50 order { 2 sqrt div }
   repeat dup scale
90 rotate
order X
stroke

showpage
```

dragoncurve

n=12
gedraaid






6 kr
Sverige
6 kr
Sverige

plaatjes die in gelijksoortige delen uiteenvallen die weer … heten wel fractals.

de torens van Hanoi is een puzzel die op recursieve manier kan worden opgelost, zie Algoritmiek, dat gaan we niet verklappen.

heel veel voorbeelden

- verzamelingen
- relaties
- rijtjes
- functies
- bomen
- grafen
- ordeningen
- syntax & semantiek

- bewijzen

E = { 0, 2, 4, 6, … } … stipjes …

even getallen: inductieve definitie

basis

i. $0 \in E$

inductiestap

ii. als $x \in E$ dan $x+2 \in E$

uitsluiting

iii. E bevat geen andere elementen dan die door toepassing van i. en ii. verkregen

basis en inductiestap
mogen meerdere regels bevatten

voorbeelden

De even natuurlijke getallen worden als { 0,2,4, … } met puntjes (‘enzovoorts’) opgeschreven. Met een inductieve definitie is er geen twijfel over wat even getallen zijn.

De inductief gedefinieerde strings zijn (*zo leert u later*) pre-orde notaties voor binaire bomen (met + als operatie in de knopen en a,b als waarde in de bladeren).

Fibonacci bomen komen voor bij de analyse van datastructuren: het zijn de meest ‘ongunstige’ AVL-bomen, zo weinig mogelijk knopen bij een gegeven hoogte (maar toch voldoen aan de eigenschap van AVL-bomen).

De symmetrische ordening is een manier om knopen in een binaire boom te nummeren die recursief gedefinieerd is. Volgend hoofdstuk.

basis

i. a ∈ B, b ∈ B

inductiestap

ii. als x,y ∈ B dan +xy ∈ B

uitsluiting

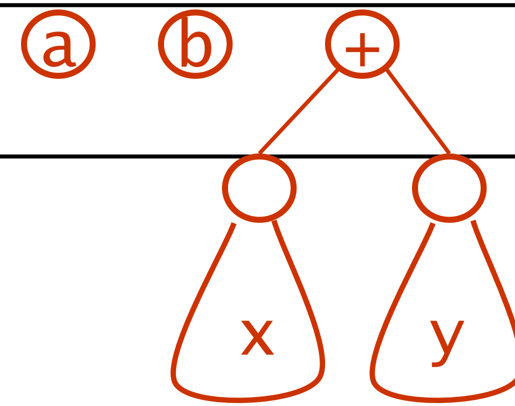
iii. B bevat geen andere elementen dan die door toepassing van i. en ii. verkregen

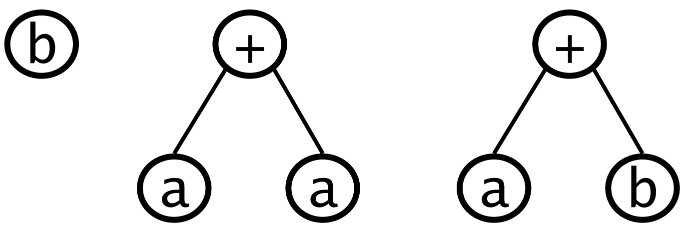
b, +aa, +ab, ++abb, +b+aa, ++aa+ab, … +++aa+ab++abb, …

(dit zijn ‘gecodeerde’ binaire bomen, zie later)

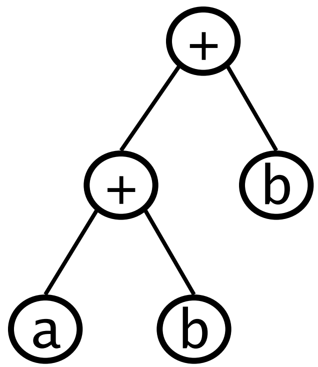
binaire bomen (?)

i. $a \in B$, $b \in B$
ii. als $x, y \in B$ dan $+xy \in B$

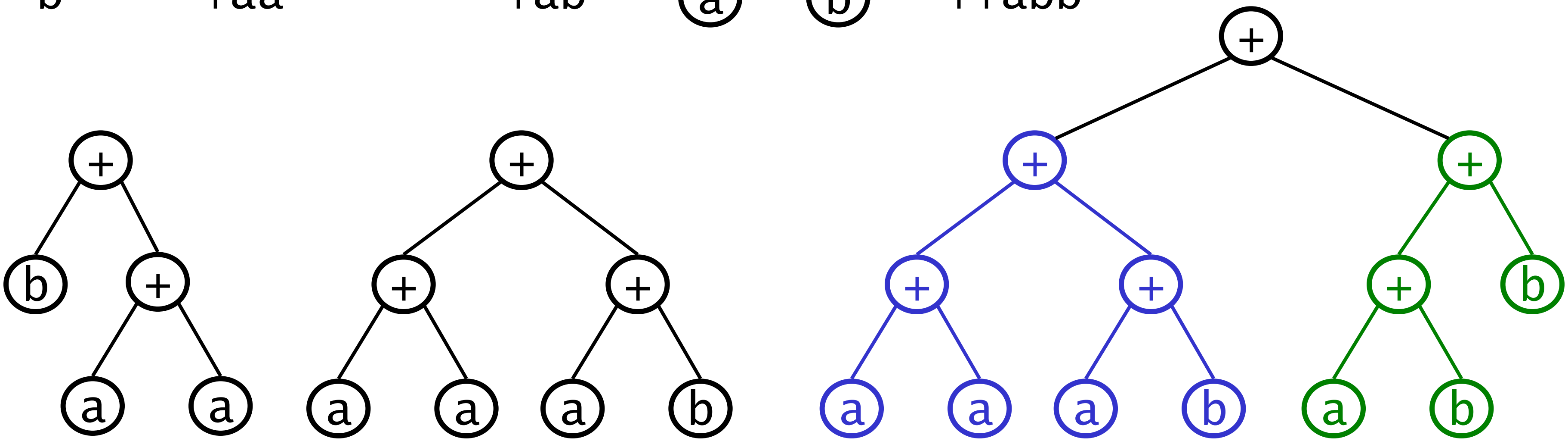




b   +aa   +ab

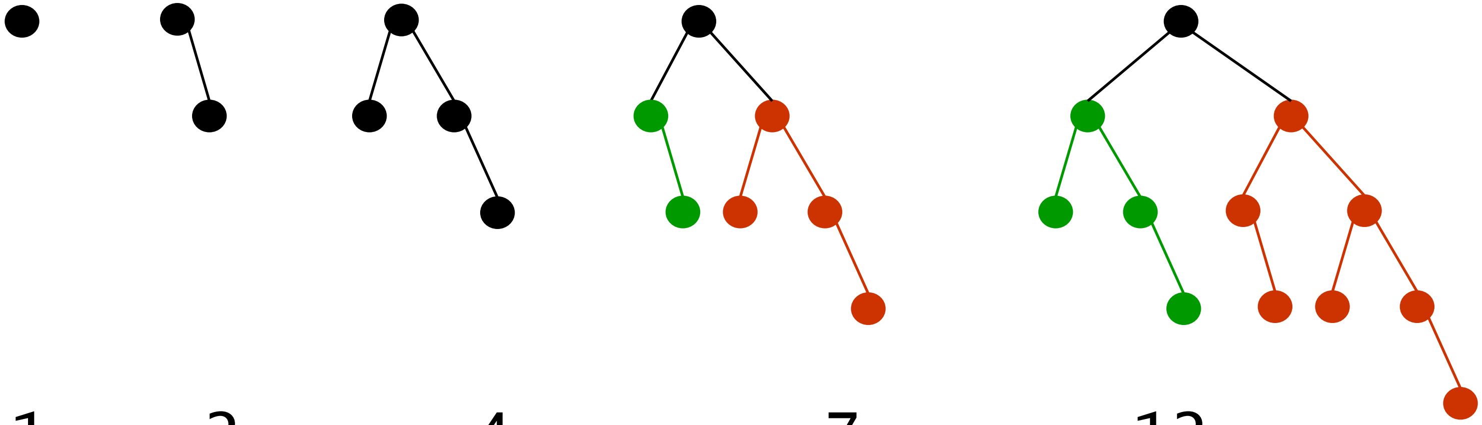


++abb



+b+aa   ++aa+ab   +++aa+ab++abb, …

i. T_0 heeft géén knopen
ii. T_1 heeft één knoop
ii. T_{n+1} heeft een wortel met deelbomen T_{n-1} en T_n



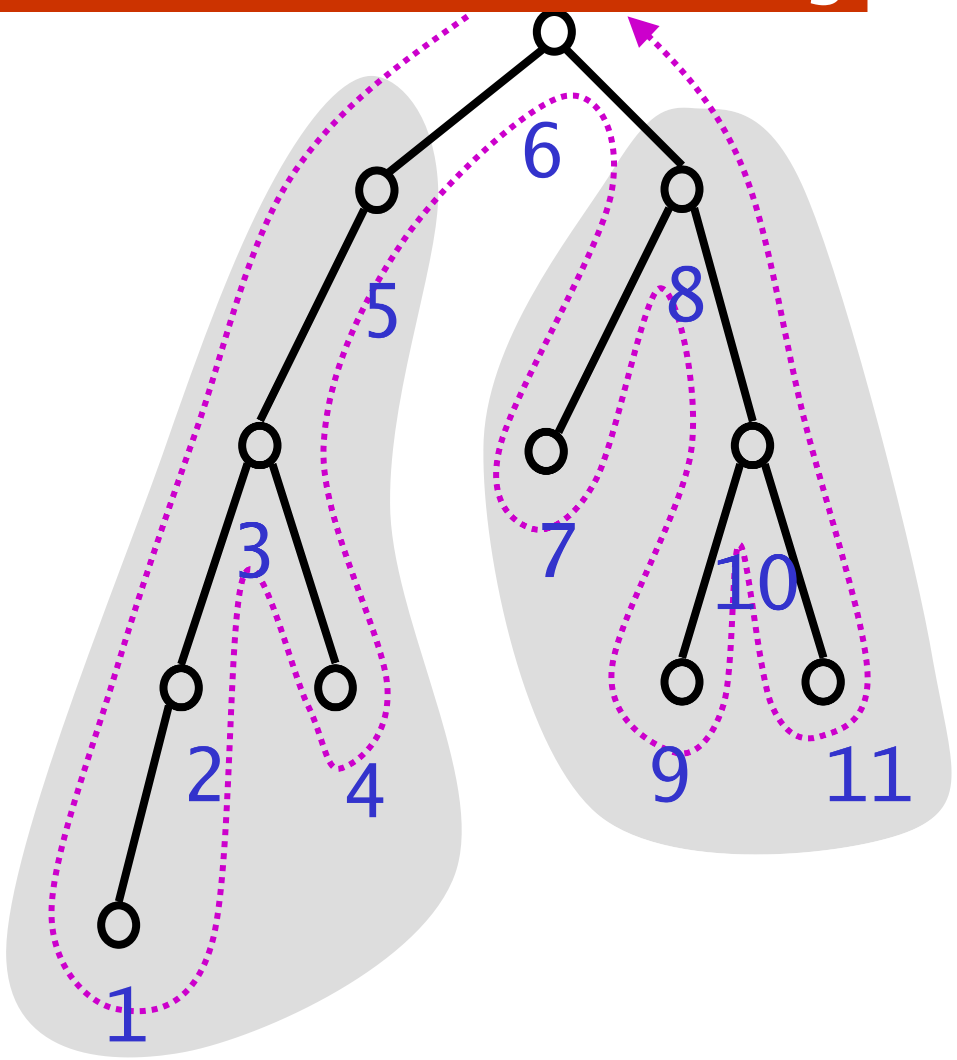
0	1	2	4	7	12	…
1	2	3	5	8	13	…

T_n heeft φ_{n+1}-1 knopen

(volgt nog)

symmetrische ordening

bij binaire bomen



$\boxed{\mathit{symm}(\varnothing) = \lambda}$

$\boxed{\mathit{symm}(T) = \mathit{symm}(T_\ell), \mathrm{wortel}(T), \mathit{symm}(T_r)}$

relatie

i. $0<1$
ii. als $x<y$ dan $x<y+1$ en $x+1<y+1$
iii. $<$ bevat geen andere elementen dan die door toepassing van i. en ii. verkregen

$0<1$

$0<2 \quad 1<2$

$0<3 \quad 1<3 \quad 2<3$

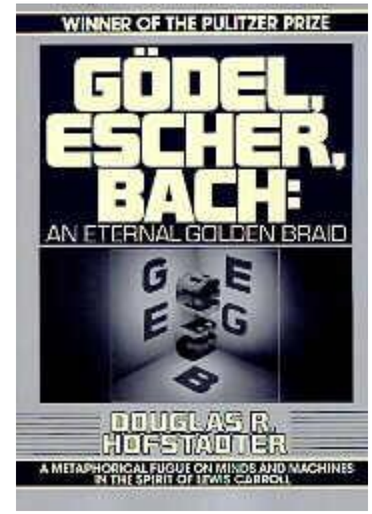
$0<4 \quad 1<4 \quad 2<4 \quad 3<4$

$0<5 \quad 1<5 \quad 2<5 \quad 3<5 \quad 4<5$

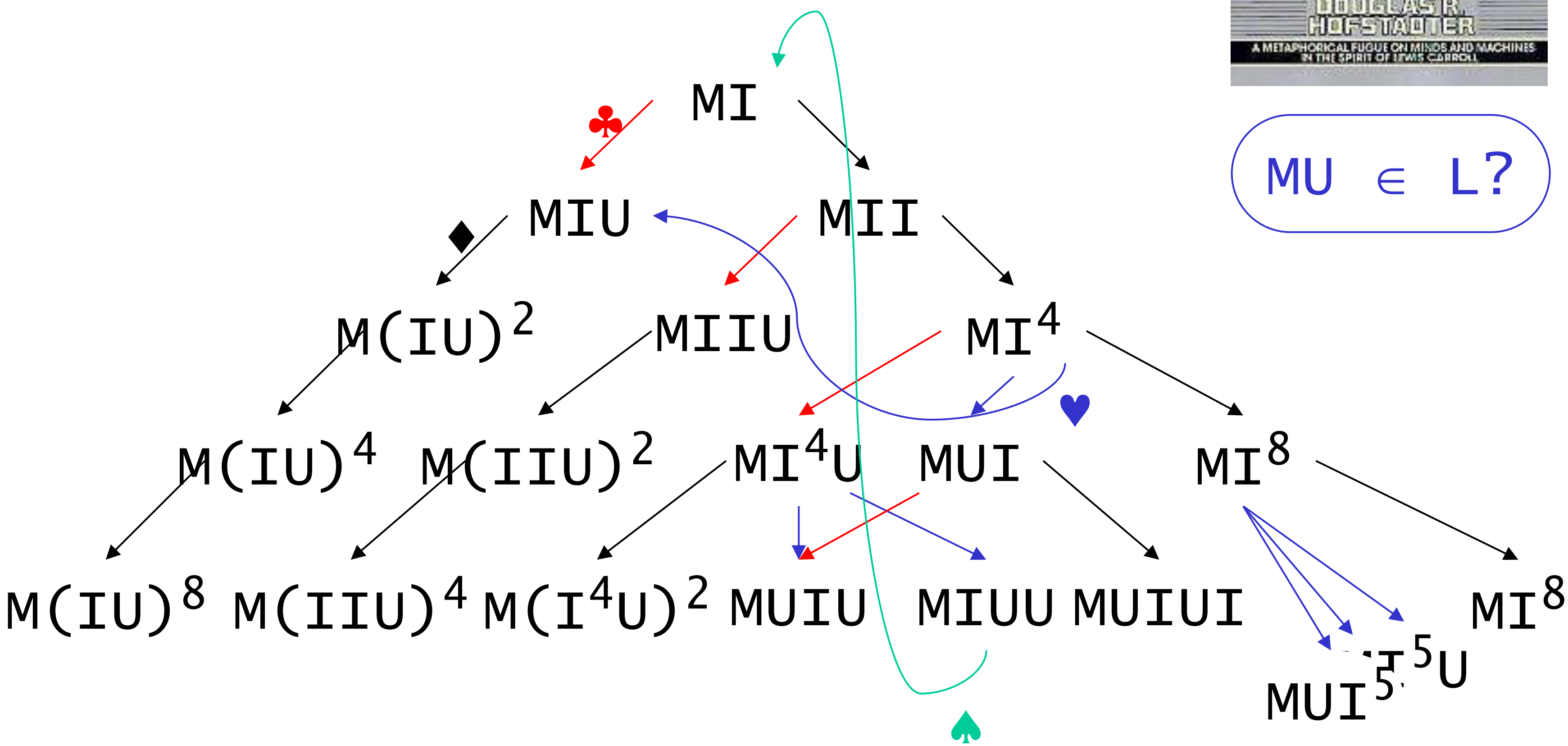
gezien inductieve verzamelingen
- getallen
- strings
- paren

taal ‘MU puzzle’

- MI $\in$ L
- als xI $\in$ L dan xIU $\in$ L ♣
  als Mx $\in$ L dan Mxx $\in$ L ♦
  als xIIIy $\in$ L dan xUy $\in$ L ♥
  als xUUy $\in$ L dan xy $\in$ L ♠
- L bevat geen andere elementen



MU $\in$ L?



taal ‘MU puzzle’

De Mu-puzzel komt uit het boek *Gödel, Escher, Bach: An Eternal Golden Braid* van Douglas Hofstadter.

De regels definieren op recursieve wijze een taal L (verzameling strings).
De vraag is, behoort MU tot die taal?

Antwoord volgt.

http://en.wikipedia.org/wiki/G%C3%B6del,_Escher,_Bach

syntax: geheel getal

$\langle\texttt{geheel}\rangle$ ::= $\langle\texttt{teken}\rangle\langle\texttt{natuurlijk}\rangle$ | $\langle\texttt{natuurlijk}\rangle$
$\langle\texttt{natuurlijk}\rangle$ ::= $\langle\texttt{cijfer}\rangle$ | $\langle\texttt{cijfer}\rangle\langle\texttt{natuurlijk}\rangle$
$\langle\texttt{cijfer}\rangle$ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
$\langle\texttt{teken}\rangle$ ::= + | -

$$\langle\texttt{geheel}\rangle \Rightarrow \langle\texttt{teken}\rangle\langle\texttt{natuurlijk}\rangle$$
$$\Rightarrow \texttt{-}\ \langle\texttt{natuurlijk}\rangle \Rightarrow \texttt{-}\langle\texttt{cijfer}\rangle\langle\texttt{natuurlijk}\rangle$$
$$\Rightarrow \texttt{-3}\ \langle\texttt{natuurlijk}\rangle \Rightarrow \texttt{-3}\ \langle\texttt{cijfer}\rangle\langle\texttt{natuurlijk}\rangle$$
$$\Rightarrow \texttt{-31}\ \langle\texttt{natuurlijk}\rangle \Rightarrow \texttt{-31}\ \langle\texttt{cijfer}\rangle\langle\texttt{natuurlijk}\rangle$$
$$\Rightarrow \texttt{-315}\ \langle\texttt{natuurlijk}\rangle \Rightarrow \texttt{-315}\ \langle\texttt{cijfer}\rangle \Rightarrow \texttt{-3157}$$

Een programmeertaal is ook een formele taal, een precies vastgelegde verzameling strings.

We geven hier de definities van twee onderdelen van een (verder onbekende) programmeertaal, *gehele getallen* en *statements*.

Dit gebeurt met de zgn. BNF beschrijving of Backus–Naur Form.
Inductieve elkaar definiërende begrippen aangegeven met $\langle xxx \rangle$.
Bij FI2 heet dat een context-vrije grammatica.

Door de regels ‘in elkaar in te vullen’ ontstaat een programma (als er geen $\langle xxx \rangle$ meer in de string staan).

syntax: statements

```
⟨assignment⟩ ::= ⟨variable⟩ = ⟨expression⟩

⟨statement⟩  ::= ⟨assignment⟩ |
                 ⟨compound-statement⟩ |
                 ⟨if-statement⟩ |
                 ⟨while-statement⟩ | …

⟨if-statement⟩ ::=
      if ⟨test⟩ then ⟨statement⟩ |
      if ⟨test⟩ then ⟨statement⟩ else ⟨statement⟩

⟨while-statement⟩ ::=
      while ⟨test⟩ do ⟨statement⟩
```

$D = \{ 0,1,2, \dots , 9 \}$

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$, dan $(-x) \in R$

als $x \in R$ en $y \in R$,

dan $(x+y) \in R$, $(x-y) \in R$, $(x*y) \in R$, $(x/y) \in R$

iii. R bevat geen andere elementen dan die door toepassing van i en ii kunnen worden verkregen

$+ \ - \ \cdot \ /$ zijn symbolen, geen operaties

27

0014 -(0014)

((1+13)*8)

(8/(2-2))

(27/(15+12-27))

(3-(-(-(5/7))))

rekenkundige expressies

Rekenkundige expressies gedefinieerd, zowel de vorm (*syntaxis*; de strings) als de betekenis (*semantiek*; de waarde, een geheel getal).

De moraal, als je niet oplet krijg je dubbelzinnigheden.

Het heet syntaxis in het Nederlands, meestal gebruikt men het Engelse woord syntax.

expressies syntax → semantiek

$D = \{ 0,1,2, \ldots , 9 \}$

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$, dan $(-x) \in R$

als $x \in R$ en $y \in R$,

dan $(x+y) \in R$, $(x-y) \in R$, $(x*y) \in R$, $(x/y) \in R$

syntaxis	semantiek
string	getal
$(\{0,1,\ldots,9\} \cup \{+,-,*,+,(,)\})^*$	$\mathbb{Q} \cup \{\dagger\}$ ongedefinieerd

$$w(\ ((23-(12+7))*2)\ ) = 8$$

expressies syntax → semantiek (1)

$D = \{ 0,1,2, \dots , 9 \}$

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$, dan $(-x) \in R$

als $x \in R$ en $y \in R$,

dan $(x+y) \in R$, $(x-y) \in R$, $(x*y) \in R$, $(x/y) \in R$

rekenkundige expressies: semantiek

definieer de *waarde* $w(z)$ van een expressie

i. voor $z \in D^*-\{\lambda\}$ is $w(z)$ de 'getalswaarde' van z

ii. als $z = (-x)$ dan $w(z) = -w(x)$

als $z = (x+y)$ dan $w(z) = w(x)+w(y)$

als $z = (x-y)$ dan $w(z) = w(x)-w(y)$

als $z = (x*y)$ dan $w(z) = w(x)*w(y)$

als $z = (x/y)$ dan $w(z) = w(x)/w(y)$ tenzij $w(y)=0$

string + symbool

getal + functie

rekenkundige expressies semantiek (1)

```
ii. als z = (-x) dan w(z) = -w(x)
    als z = (x+y) dan w(z) = w(x)+w(y)
    als z = (x-y) dan w(z) = w(x)-w(y)
    als z = (x*y) dan w(z) = w(x)*w(y)
    als z = (x/y) dan w(z) = w(x)/w(y) tenzij w(y)=0
```

```
z = ((23-(12+7))*2)

w(z) = w( ((23-(12+7))*2) ) =
     = w( (23-(12+7)) ) * w( 2 ) =
     = ( w( 23 ) - w( (12+7) ) ) * 2
     = ( 23 - ( w( 12 ) + w( 7 ) ) ) * 2
     = ( 23 - ( 12 + 7 ) ) * 2
     = 8
```

Er staan veel heekjes in de expressies die we voor ons begrip niet nodig hebben.
We kijken wat er gebeurt als we die haakjes gewoon weglaten.

De moraal: bij het ontbreken van haakjes moeten we vastleggen
- wat de voorrangsregels zijn (tussen de operatoren)
- in welke richting de operatoren (met gelijke prioriteit) gelezen/geëvalueerd moeten worden.

expressies syntax → semantiek (2)

$D = \{ 0,1,2, \ldots , 9 \}$

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$ en $y \in R$, dan $x+y \in R$, $x-y \in R$

iii. R bevat geen andere elementen …

rekenkundige expressies: semantiek

i. voor $z \in D^*-\{\lambda\}$ is $w(z)$ de ‘getalswaarde’ van z

ii. als $z = x+y$ dan $w(z) = w(x)+w(y)$

als $z = x-y$ dan $w(z) = w(x)-w(y)$

expressies syntax → semantiek (2)

$D = \{ 0,1,2, \ldots , 9 \}$
i. ieder element van $D^*-\{\lambda\}$ behoort tot R
ii. als $x \in R$ en $y \in R$, dan $x+y \in R$, $x-y \in R$
iii. R bevat geen andere elementen …

rekenkundige expressies: semantiek

i. voor $z \in D^*-\{\lambda\}$ is $w(z)$ de 'getalswaarde' van z
ii. als $z = x+y$ dan $w(z) = w(x)+w(y)$
als $z = x-y$ dan $w(z) = w(x)-w(y)$

$z = 23-12+7$

$$w(z) = w(23)-w(12+7) = 23-\{w(12)+w(7)\}$$
$$= 23-\{12+7\} = 23-19 = 4$$

$$w(z) = w(23-12)+w(7) = \{w(23)-w(12)\}+7$$
$$= \{23-12\}+7 = 11+7 = 18$$

expressies syntax → semantiek (2)

afgekeurd!

$D = \{\, 0,1,2, \dots , 9 \,\}$

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$ en $y \in R$, dan $x+y \in R$, $x-y \in R$

iii. R bevat geen andere elementen …

rekenkundige expressies: semantiek

i. voor $z \in D^*-\{\lambda\}$ is $w(z)$ de ‘getalswaarde’ van z

ii. als $z = x+y$ dan $w(z) = w(x)+w(y)$

als $z = x-y$ dan $w(z) = w(x)-w(y)$

$z = 23-12+7$

$$w(z) = w(23)-w(12+7) = 23-\{w(12)+w(7)\}$$
$$= 23-\{12+7\} = 23-19 = 4$$

$$w(z) = w(23-12)+w(7) = \{w(23)-w(12)\}+7$$
$$= \{23-12\}+7 = 11+7 = 18$$

expressies syntax → semantiek (2)

afgekeurd!

D = { 0,1,2, … , 9 }

i. ieder element van $D^*-\{\lambda\}$ behoort tot R

ii. als $x \in R$ en $y \in R$, dan $x+y \in R$, $x-y \in R$

iii. R bevat geen andere elementen …

z = 23-12+7 dubbelzinnig :
op twee manieren geconstrueerd

rekenkundige expressies: syntax (1)

ii. als $x \in R$, dan $(-x) \in R$
als $x \in R$ en $y \in R$,
dan $(x+y) \in R$, $(x-y) \in R$, $(x*y) \in R$, $(x/y) \in R$

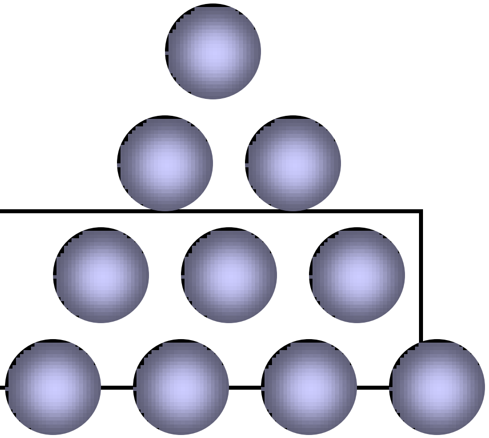
volledige haakjes óf
voorrangsregels, associativiteit
23-12+7 => ((23-12)+7)

$$f(0) = 1$$
$$f(n) = n \cdot f(n-1) \quad (n \geq 1)$$

$$f(8) = 8 \cdot f(7) = 8 \cdot 7 \cdot f(6) = \ldots$$
$$= 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot f(0) = 8!$$

gesloten formule (expliciet)

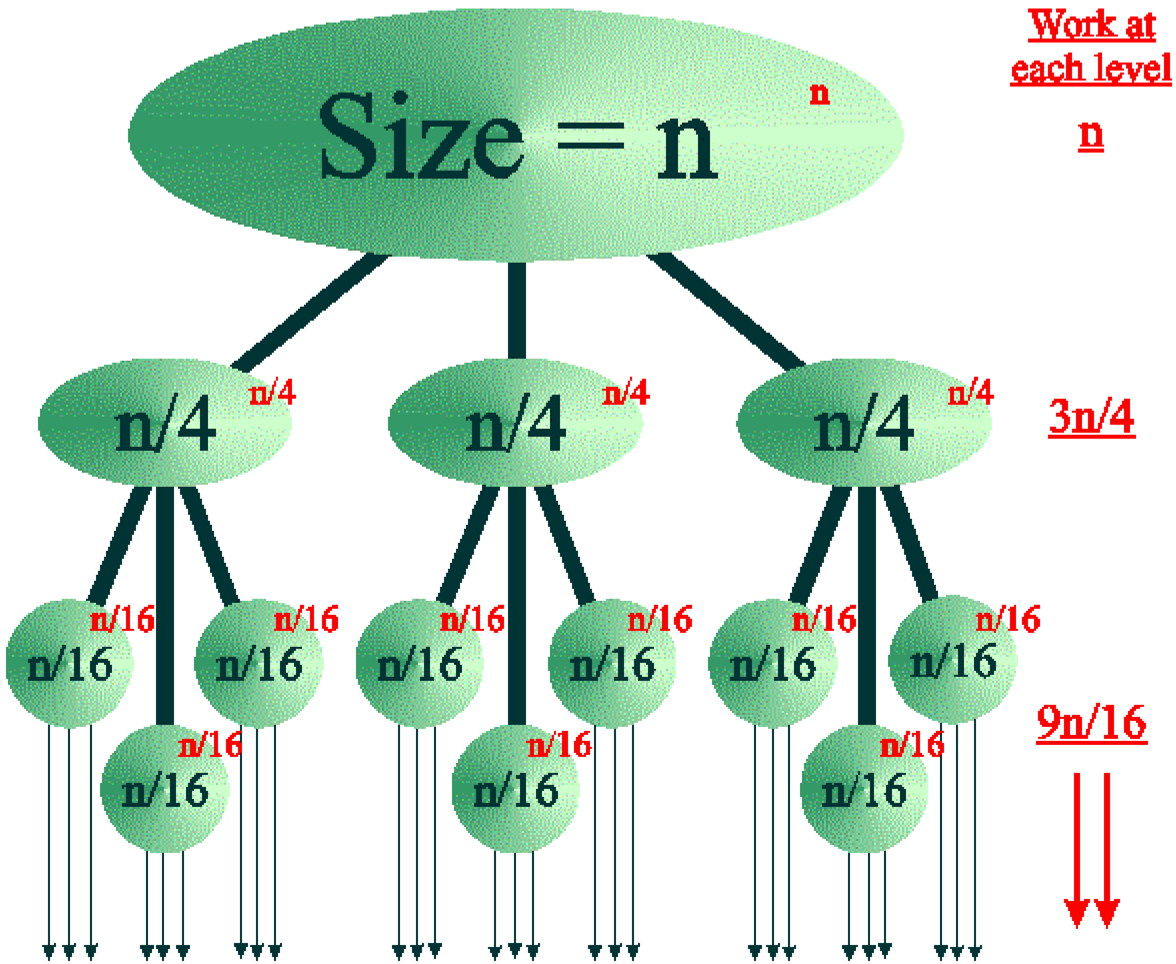
$$f(n) = n!$$



$$g(0) = 0$$
$$g(n) = n + g(n-1) \quad (n \geq 1)$$

$$g(8) = 8 + g(7) = 8 + 7 + g(6) = \ldots$$
$$= 8+7+6+5+4+3+2+1+g(0) = 36$$
$$g(n) = \tfrac{1}{2}n(n+1)$$

$$f(1) = c$$
$$f(n) = a \cdot f(n/b) + c \quad (n=b^k)$$



analyse van algoritmen

Bij een vak als analyse van algoritmen willen we soms een recursief programma evalueren op efficiëntie. een probleem ter grootte n wordt bijvoorbeeld opgesplitst in 3 deelproblemen van grootte n/4 (plus wat extra werk).
Hoeveel werk/tijd kost dat dan als we alle stappen uitwerken, als *gesloten formule*, dus functie van n (zonder recursie).

$a_0 = 0$
$a_1 = 1$
$a_{n+1} = a_n + a_{n-1} \quad (n \geq 1)$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

Fibonacci

$$a_n = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right\}$$

Binet

gulden snede
$x^2 = x + 1$

recursieve functie
$f(n+1) = f(n) + f(n-1)$

recurrente betrekking

De Fibonacci getallen weer, als ik het goed zie.

Daar bestaat ook een gesloten formule voor.

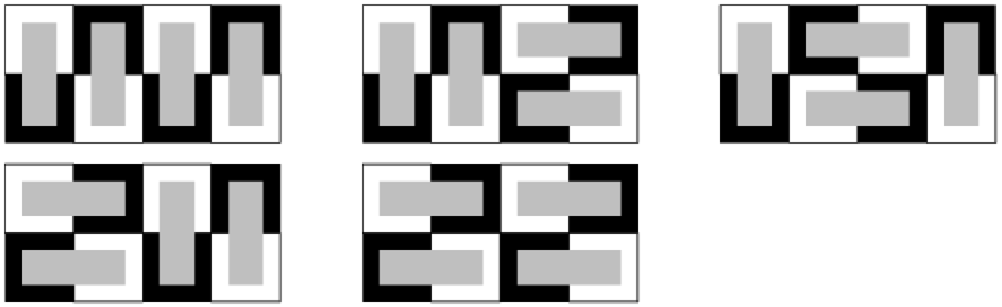
Het getal (1+wortel 5)/2 dat in de formule voorkomt heet wel de gulden snede. Komt voor in de lengte van de diagonalen van een regelmatige vijfhoek.

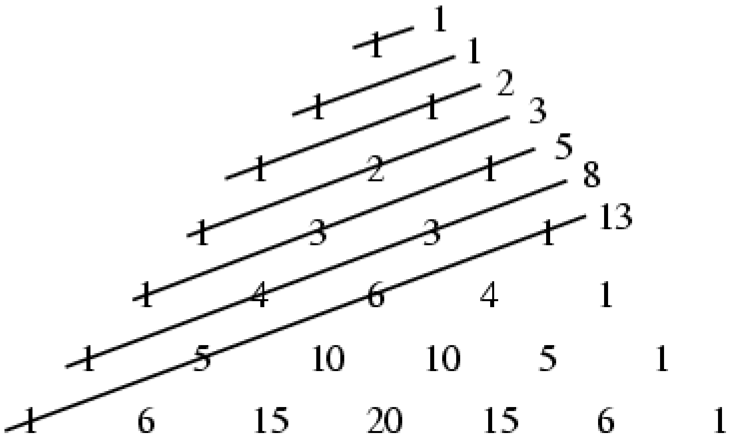
Is de oplossing van de kwadratische vergelijking die dezelfde structuur heeft als de recursie van de getallen (en dat is geen toeval).

Zie hierna (recurrente betrekking)

http://mathworld.wolfram.com/FibonacciNumber.html

domino’s op schaakbord





recurrente betrekking

$t_0 = 5,\ t_1 = 6$

$t_n = t_{n-1} + 6t_{n-2} - 12n + 8 \quad (n \geq 2)$

$$
\begin{aligned}
& & 5,\\
& & 6,\\
6 + 30 - 24 + 8 &= & 20,\\
20 + 36 - 36 + 8 &= & 28,\\
28 + 120 - 48 + 8 &= & 108,\\
108 + 168 - 60 + 8 &= & 208,\\
208 + 648 - 72 + 8 &= & 792,\\
& & \dots
\end{aligned}
$$

geheim recept

$t_n = 3^n + (-2)^n + 2n + 3$

gesloten formule

recurrente betrekking

$$t_0 = 5,\ t_1 = 6$$
$$t_n = t_{n-1} + 6t_{n-2} - 12n + 8 \quad (n \geq 2)$$

Het geheime recept staat in Schaum, Section 6.8, *Solving 2nd order homogeneous linear recurrence relations* (maar deze is *niet* homogeen).

GEEN TENTAMENSTOF

geheim recept

$$x^2 = x+6$$
$$x=3 \vee x=-2$$

$$t_n = 3^n + (-2)^n + 2n + 3$$

gesloten formule

het vinden van de gesloten formule bij een recurrente betrekking is geen tentamenstof (zeiden we al) maar het controleren van een gegeven formule bij een betrekking is WEL kennis die je hebben.
Dat is namelijk een voorbeeld van Volledige Inductie, hierna.

(volgt nog)

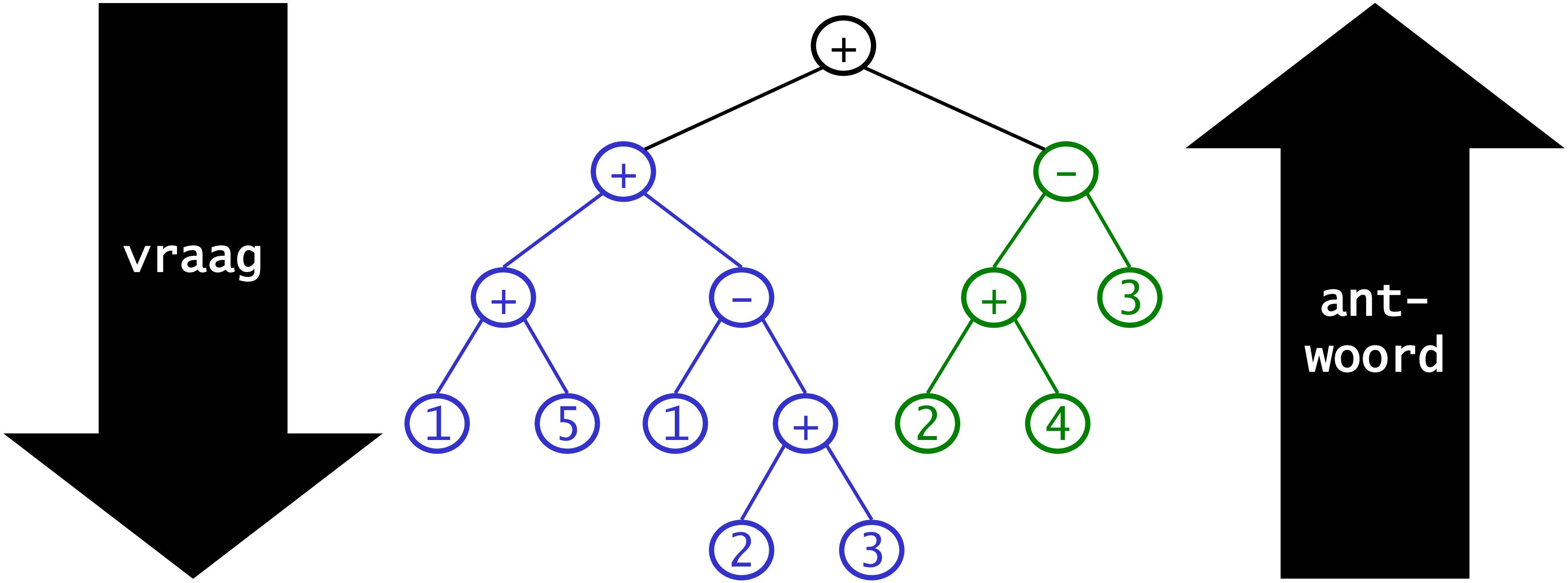
basis

```
  f(blad)  = …
```

recursie

```
  f(knoop) = f(links) & f(rechts)
```

((1+5)+(1-(2+3))) + ((2+4)-3)



$\boxed{f(n) = n^2 + n + 41}$

Euler 1772

dit zijn allemaal priemgetallen?

41, 43, 47, 53, 61, 71, 83, 97, 113, 131,
151, 173, 197, 223, 251, 281, 313, 347,
383, 421, 461, 503, 547, 593, 641, 691,
743, 797, 853, 911, 971, 1033, 1097, 1163,
1231, 1301, 1373, 1447, 1523, 1601, …

toch niet …

$f(41) = 41 \cdot 41 + 41 + 41 = 41 \cdot 43$

tegenvoorbeeld

mooie formule

Als je wilt laten zien dat iets niet voor elk getal waar is, is het genoeg een enkel tegenvoorbeeld te geven. (dat hoeft niet makkelijk te vinden zijn, maar eentje voldoet.)

Om te laten zien dat iets voor alle getallen waar is kunnen we niet een handjevol gevallen controleren, we hebben een speciale techniek nodig.

drievouden

5^n-2^n is een drievoud voor $n \geq 0$

$$
\begin{aligned}
5^0-2^0 &= 1-1 &&= 0 \\
5^1-2^1 &= 5-2 &&= 3 \\
5^2-2^2 &= 25-4 &&= 21 \\
5^3-2^3 &= 125-8 &&= 117 \\
5^4-2^4 &= 625-16 &&= 609
\end{aligned}
$$

tegenvoorbeeld ? (nee)

gaat dit altijd goed ? (ja)

$5^n - 2^n$ is een drievoud voor $n \geq 0$

$$5^0 - 2^0 = 1-1 = 0$$
$$5^1 - 2^1 = 5-2 = 3$$
$$5^2 - 2^2 = 25-4 = 21$$
$$5^3 - 2^3 = 125-8 = 117$$
$$5^4 - 2^4 = 625-16 = 609$$

$$5^5 - 2^5 =$$
$$5 \cdot 5^4 - 5 \cdot 2^4 + 5 \cdot 2^4 - 2 \cdot 2^4 =$$
$$5 \cdot (5^4 - 2^4) + (5-2) \cdot 2^4 =$$
$$5 \cdot (\mathbf{5^4 - 2^4}) + \mathbf{3} \cdot 2^4$$

drievoud

(zonder uitrekenen)

drievouden

5^n-2^n is een drievoud voor $n\geq 0$

$5^5-2^5 =$

$5\cdot 5^4 - 5\cdot 2^4 + 5\cdot 2^4 - 2\cdot 2^4 =$

$5\cdot(5^4-2^4) + (5-2)\cdot 2^4 =$

$5\cdot(5^4-2^4) + 3\cdot 2^4$

drievoud
(zonder uitrekenen)

omdat we weten dat 609 een 3-voud is, kunnen we zien dat het volgende getal uit de reeks weer een 3-voud is, doordat we het handig hebben omgeschreven, als som van 3-vouden.

hierna doen we dat voor algemene waardes van n, zelfde idee.

$5^n - 2^n$ is een drievoud voor $n \geq 0$

$$
\begin{aligned}
5^0 - 2^0 &= \quad 1-1 &&= 0 \\
5^1 - 2^1 &= \quad 5-2 &&= 3 \\
5^2 - 2^2 &= \ 25-4 &&= 21 \\
5^3 - 2^3 &= 125-8 &&= 117 \\
5^4 - 2^4 &= 625-16 &&= 609
\end{aligned}
$$

$$
\begin{aligned}
&5^5 - 2^5 = \\
&5 \cdot 5^4 - 5 \cdot 2^4 + 5 \cdot 2^4 - 2 \cdot 2^4 = \\
&5 \cdot (5^4 - 2^4) + (5-2) \cdot 2^4 = \\
&5 \cdot (\mathbf{5^4 - 2^4}) + \mathbf{3} \cdot 2^4
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
&5^{n+1} - 2^{n+1} = \\
&5 \cdot 5^n - 5 \cdot 2^n + 5 \cdot 2^n - 2 \cdot 2^n = \\
&5 \cdot (\mathbf{5^n - 2^n}) + \mathbf{3} \cdot 2^n
\end{aligned}
}
$$

bewijs

5^n-2^n is een drievoud voor $n \geq 0$

inductie-bewijs

i. <u>basis</u> ($n=0$)

$5^0-2^0 = 1-1 = 0$ is een drievoud.

ii. <u>inductiestap</u>

inductie-aanname:

Neem aan 5^n-2^n is een drievoud ($n \geq 0$)

Bewijs dat $5^{n+1}-2^{n+1}$ een drievoud is.

$$5 \cdot (\underbrace{5^n-2^n}) + \underbrace{3} \cdot 2^n = 5 \cdot 5^n - 5 \cdot 2^n + 5 \cdot 2^n - 2 \cdot 2^n = \underbrace{5^{n+1}-2^{n+1}}$$

drievouden $\Longrightarrow$ drievoud

bewijs

$5^n - 2^n$ is een drievoud voor $n \geq 0$

inducti

i. b

5

ii. induc

inductie-aanname:

Neem aan $5^n - 2^n$ is een drievoud $(n \geq 0)$

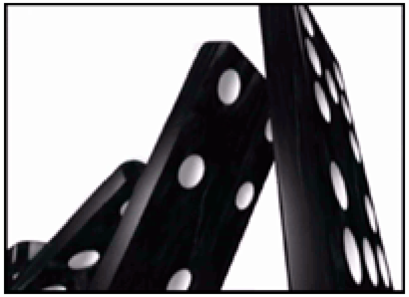
Bewijs dat $5^{n+1} - 2^{n+1}$ een drievoud is.

voor dit specifieke voorbeeld leren we later een andere bewijsmethode kennen, rekenen met resten.

$$5 \cdot (5^n - 2^n) + 3 \cdot 2^n = 5 \cdot 5^n - 5 \cdot 2^n + 5 \cdot 2^n - 2 \cdot 2^n = 5^{n+1} - 2^{n+1}$$

drievouden $\Longrightarrow$ drievoud

$\mathbb{N}$ natuurlijke getallen



i. <u>basis</u>
   Bewijs dat $P(0)$ waar is.

ii. <u>inductiestap</u>
   Bewijs voor willekeurige $n \in \mathbb{N}$:
   als $\underbrace{P(n)}_{\text{inductie-aanname}}$ waar is, dan is $P(n+1)$ waar.

inductie-aanname
inductie-hypothese

$$P(0) \wedge (\forall n \in \mathbb{N})(\ P(n) \Rightarrow P(n+1)\ )$$
$$\Rightarrow (\forall n \in \mathbb{N})(\ P(n)\ )$$

MU eigenschappen

- MI ∈ L
- als xI ∈ L dan xIU ∈ L ♣
  als Mx ∈ L dan Mxx ∈ L ♦
  als xIIIy ∈ L dan xUy ∈ L ♥
  als xUUy ∈ L dan xy ∈ L ♠
- L bevat geen andere elementen

Stelling

elk woord in L begint met M

<u>basis</u>

MI begint met M

<u>inductie</u> naar opbouw

♣ als xI met M begint, dan ook xIU

♦ Mxx begint met M

♥ als xIIIy met M begint, dan ook xUy

♠ als xUUy met M begint, dan ook xy

V inductief gedefinieerd
te bewijzen $P(x)$ voor alle $x \in V$

inductie

i. basis
Bewijs dat $P(x)$ waar is,
voor alle x in de basis van V.

ii. inductiestap
Bewijs dat $P(y)$ geldt
voor y geconstrueerd in inductiestap,
onder aanname dat $P(x)$ waar is
voor alle x waaruit y geconstrueerd is,

inductie-aanname

structurele inductie is de bewijstechniek die hoort bij inductief opgebouwde ‘objecten’ zoals bomen.

Bij eigenschappen van de natuurlijke getallen heet de methode volledige inductie.

MU $\notin$ L

- MI $\in$ L
- als xI $\in$ L dan xIU $\in$ L ♣
  als Mx $\in$ L dan Mxx $\in$ L ♦
  als xIIIy $\in$ L dan xUy $\in$ L ♥
  als xUUy $\in$ L dan xy $\in$ L ♠
- L bevat geen andere elementen

eigenschap

aantal letters I woorden L nooit drievoud

<u>basis</u>

MI heeft één I; ok

<u>inductie</u> naar opbouw

♣ xIU evenveel I's als xI: geen drievoud

♦ Mxx tweemaal zoveel I's als Mx
en dit levert geen drievoud op

♥ uit xIIIy worden drie I's verwijderd, dit
kan ook geen drievoud opleveren

♠ xUUy evenveel I's als xy: geen drievoud

inductie voorbeeld

$$
\begin{aligned}
1 &= 1 = 1\cdot2\cdot3 \;/\; 6 \\
1+4 &= 5 = 2\cdot3\cdot5 \;/\; 6 \\
1+4+9 &= 14 = 3\cdot4\cdot7 \;/\; 6 \\
1+4+9+16 &= 30 = 4\cdot5\cdot9 \;/\; 6 \\
1+4+9+16+25 &= 55 = 5\cdot6\cdot11 \;/\; 6 \\
1+4+9+16+25+36 &= 91 = 6\cdot7\cdot13 \;/\; 6 \\
&\dots
\end{aligned}
$$

$$\boxed{6\cdot\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)}$$

$$6 \cdot \sum_{i=1}^{n} i^2 = n(n+1)(2n+1)$$

eigenschap natuurlijke getallen $\Rightarrow$ volledige inductie

basis: n=1.

$$6 \cdot (1^2) = 1(1+1)(2+1)$$

inductiestap:

neem aan dat de formule klopt voor n, en bewijs haar voor n+1.

$$6 \sum_{i=1}^{n+1} i^2 = 6 \sum_{i=1}^{n} i^2 + 6(n+1)^2 =$$

inductie-aanname

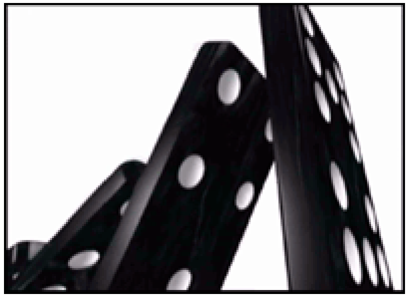
$$n(n+1)(2n+1)+6(n+1)^2 =$$

$$(n+1)\{n(2n+1)+6(n+1)\} =$$

$$(n+1)(2n^2+n+6n+6) =$$

$$(n+1)(n+2)(2n+3)$$

* volledige inductie



$\mathbb{N}$ natuurlijke getallen

i. basis
Bewijs dat $P(0)$ waar is.

ii. inductiestap
Bewijs dat voor willekeurige $n \in \mathbb{N}$:
als $P(k)$ waar is, dan is $P(n+1)$ waar.

voor alle $k \leq n$

inductie-aanname

recurrente betrekking

$t_0 = 5,\ t_1 = 6$

$t_n = t_{n-1} + 6t_{n-2} - 12n + 8 \quad (n \geq 2)$

$t_n = 3^n + (-2)^n + 2n + 3$ gesloten uitdrukking

basis:

$n=0 \quad 3^0 + (-2)^0 + 2 \cdot 0 + 3 = 5$

$n=1 \quad 3^1 + (-2)^1 + 2 \cdot 1 + 3 = 6$ (ok)

inductiestap:

inductie-aanname

$t_{n+1} = t_n + 6t_{n-1} - 12 \cdot (n+1) + 8 =$

$3^n + (-2)^n + 2n + 3 + 6 \cdot (3^{n-1} + (-2)^{n-1} + 2 \cdot (n-1) + 3)$

$-12 \cdot (n+1) + 8 =$

$(1+2) \cdot 3^n + (1-3) \cdot (-2)^n + (2+12$ … $12+18-12+8) =$

$3^{n+1} + (-2)^{n+1} + 2 \cdot (n+1) + 3$ (ok!)

dit is een beetje compact, ik heb het hierna uitgeschreven

$$t_{n+1} = 3^{n+1} +(-2)^{n+1} +2(n+1) +3$$

inductie-
aanname

$$t_n = 3^n +(-2)^n +2n +3$$
$$t_{n-1} = 3^{n-1} +(-2)^{n-1} +2(n-1) +3$$

$$t_{n+1} = t_n +6t_{n-1} -12\cdot(n+1) +8$$

definitie t_n

$$= 3^n+(-2)^n+2n+3+$$

$$6\cdot(3^{n-1}+(-2)^{n-1}+2\cdot(n-1)+3) -12\cdot(n+1)+8 =$$

$$(1+2)\cdot 3^n+(1-3)\cdot(-2)^n+(2+12-12)\cdot n+(3-12+18-12+8)$$

$$= 3^{n+1} +(-2)^{n+1} +2\cdot(n+1) +3$$

(ok!)

Dat ga ik zo niet allemaal voorlezen bij de les, want nogal saai. Het is gewoon middelbare school rekenkunde, die je wel zelf moet kunnen invullen en controleren!

$$t_{n+1} = t_n + 6t_{n-1} - 12\cdot(n+1) + 8$$

$$= 3^n + (-2)^n + 2n + 3 +$$

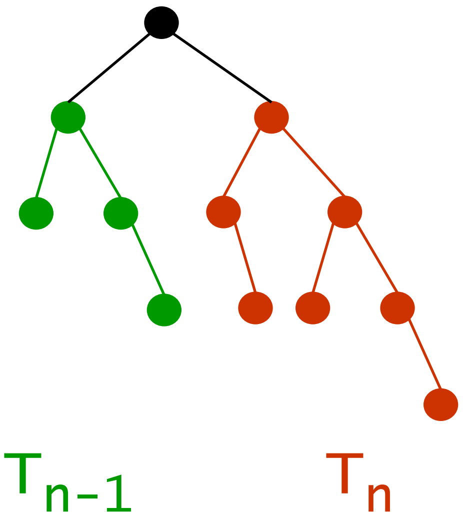
$$6\cdot(3^{n-1} + (-2)^{n-1} + 2\cdot(n-1) + 3) - 12\cdot(n+1) + 8 =$$

$$(1+2)\cdot 3^n + (1-3)\cdot(-2)^n + (2+12-12)\cdot n + (3-12+18-12+8)$$

$$= 3^{n+1} + (-2)^{n+1} + 2\cdot(n+1) + 3 \qquad \text{(ok!)}$$

i. T_0 heeft géén knopen
ii. T_1 heeft één knoop
ii. T_{n+1} heeft een wortel en deelbomen T_{n-1} en T_n

T_n heeft $\varphi_{n+1}-1$ knopen



basis:
T_0 heeft 0 knopen; $\varphi_1 = 1$
T_1 heeft 1 knoop; $\varphi_2 = 2$

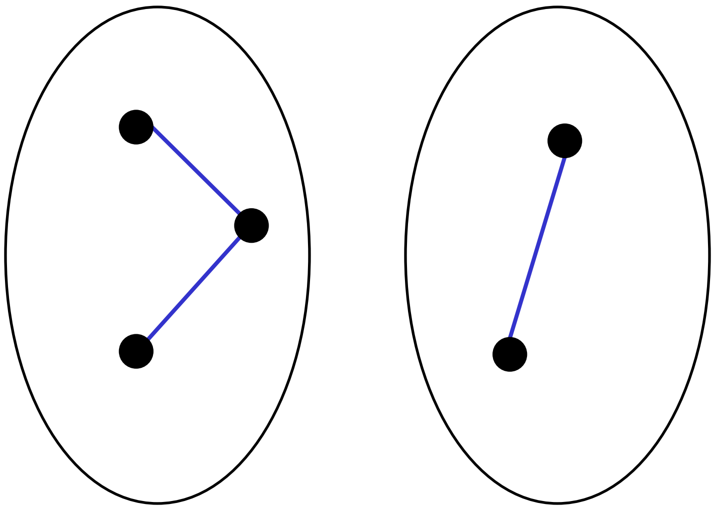
inductiestap:
T_{n+1} heeft één knoop meer dan T_{n-1} en T_n samen
volgens de **inductieaanname** (en Fibo definitie)
is dit $(\varphi_n-1)+(\varphi_{n+1}-1)+1 = \varphi_{n+2}-1$

grafen, recursief

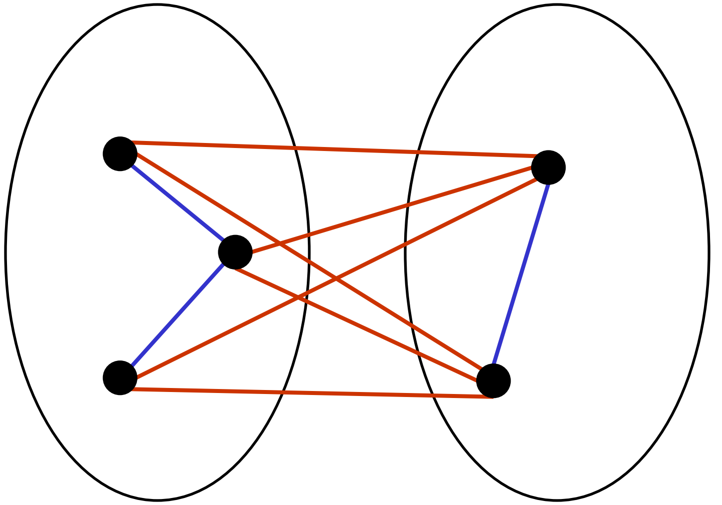
de slides hierna behandel ik niet altijd uitgebreid, tijdgebrek, en vragen worden er ook vast niet over gesteld.

ik wil laten zien dat ook bepaalde grafen op een *inductieve* manier kunnen worden gedefinieerd, de co-grafen, met twee simpele operaties startend met losse knopen.

de vraag: welke grafen krijgen we zo (niet) wordt dan vervolgens beantwoord.



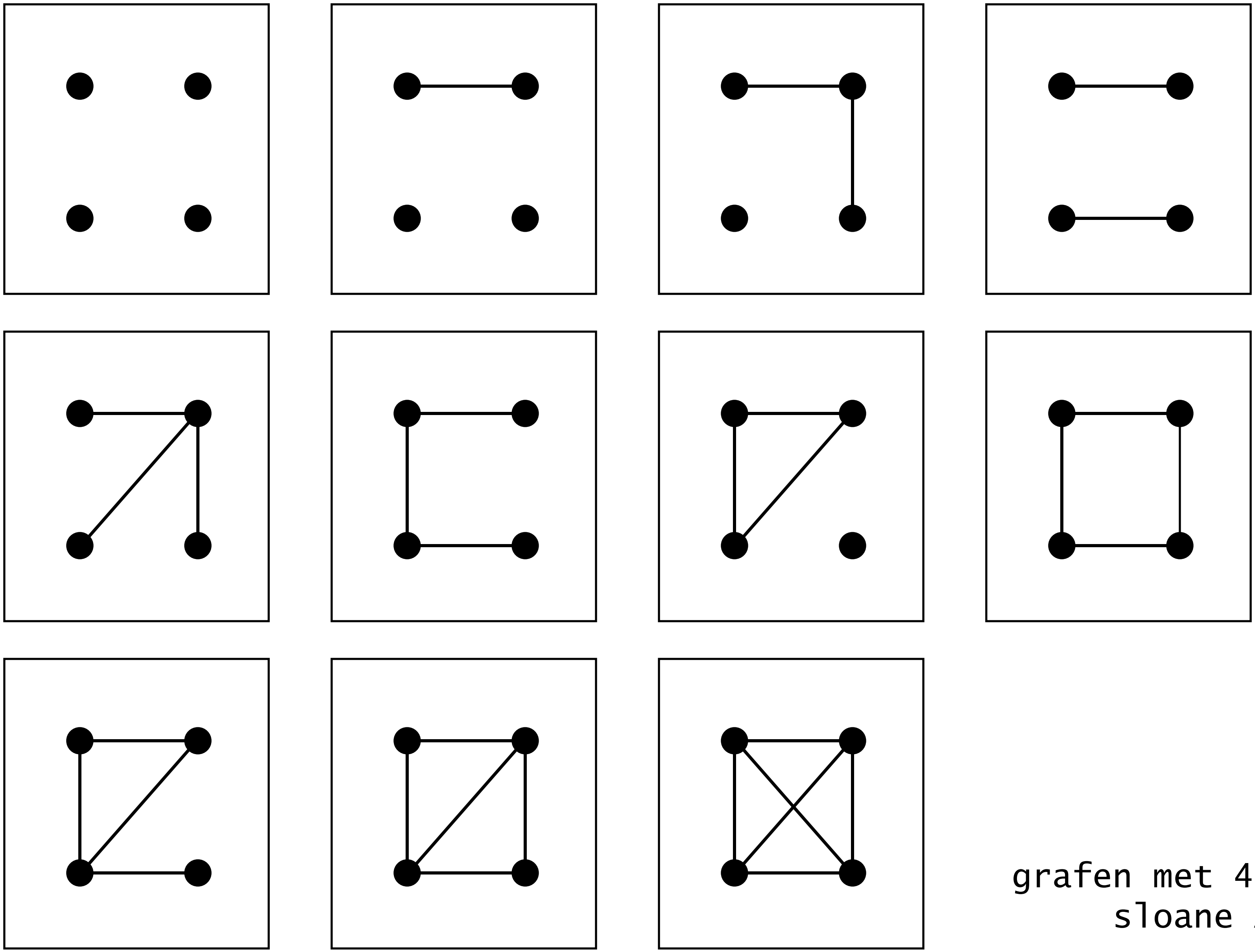
vereniging ∪



som +

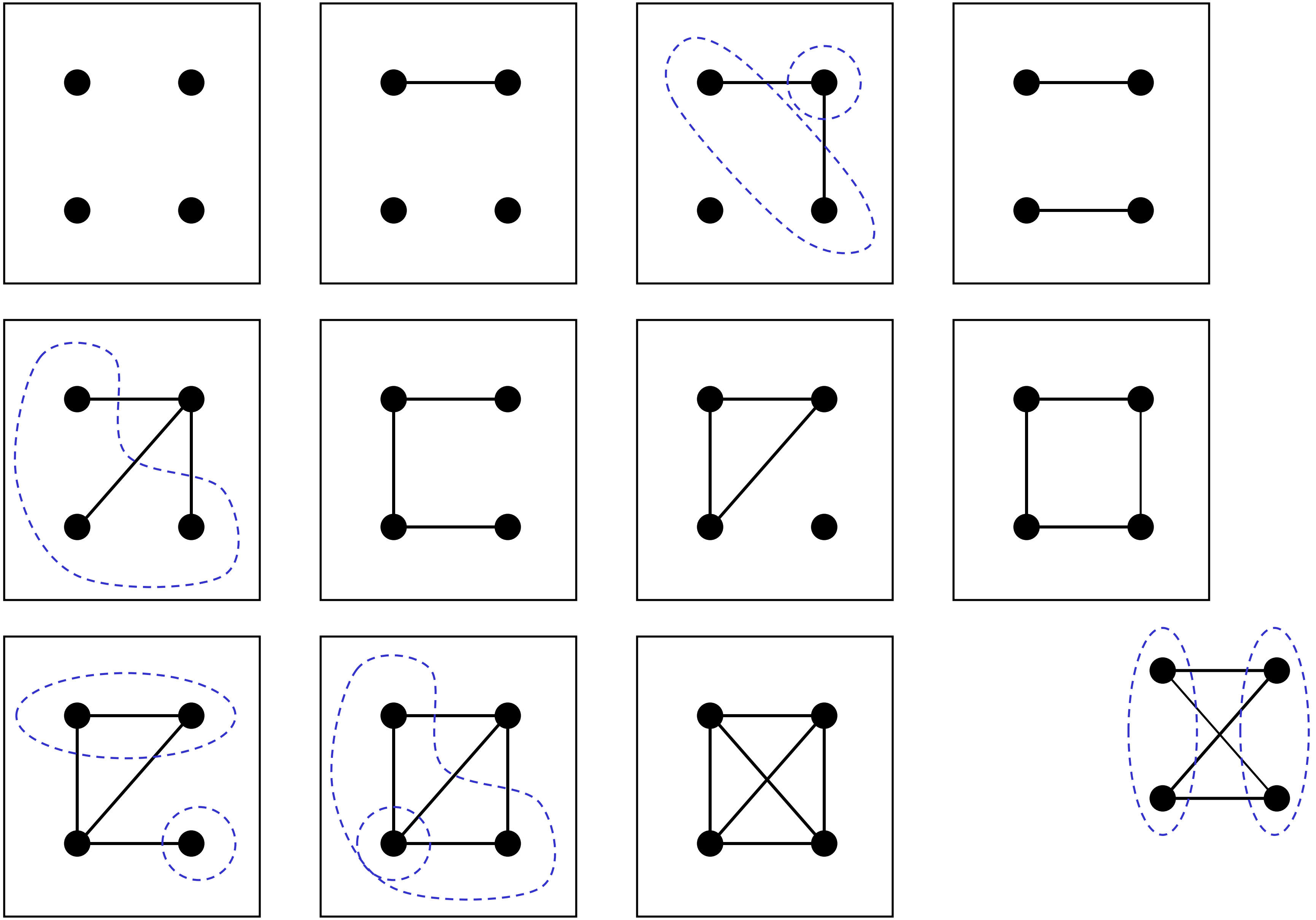
voorbeeld:
inductie bij grafen

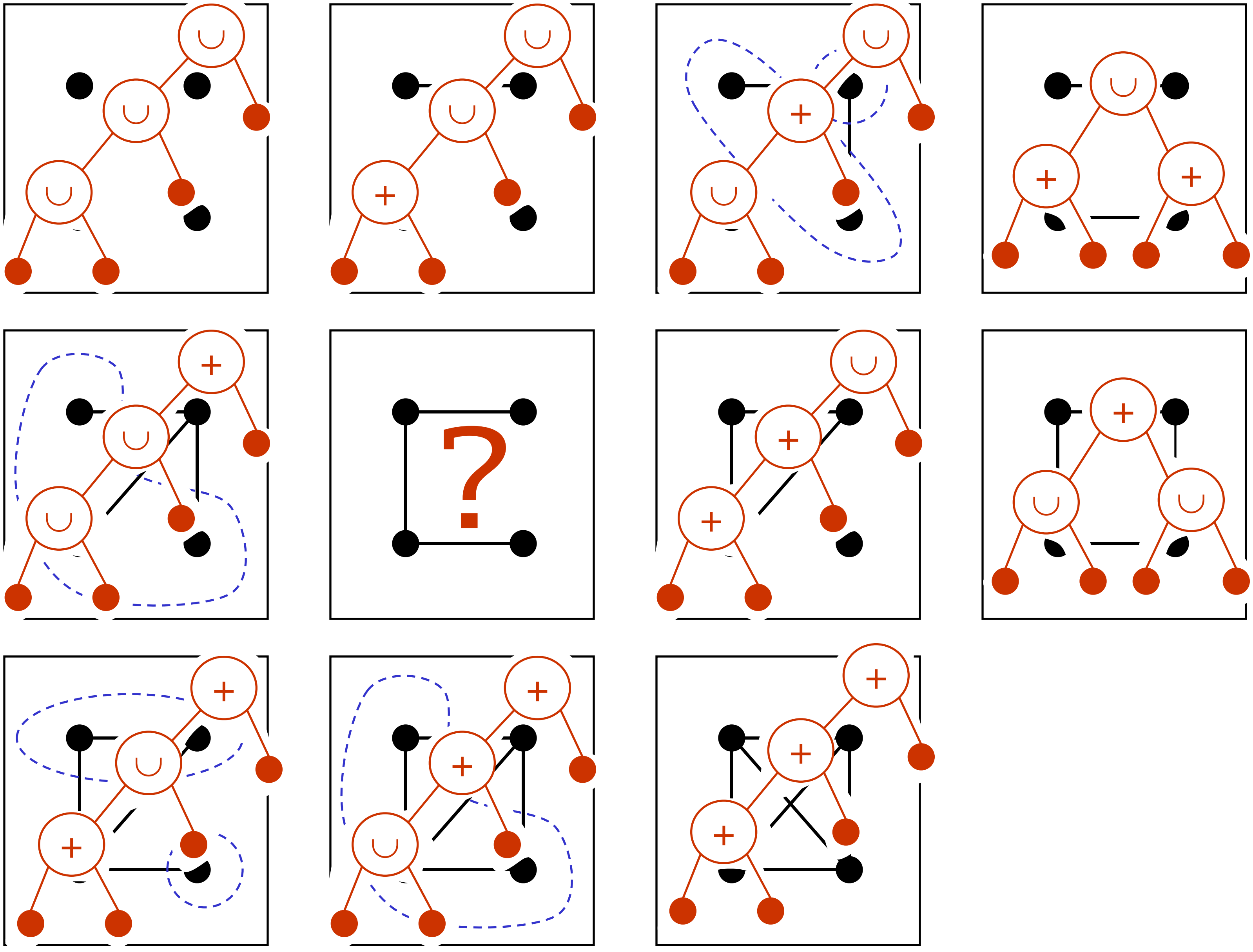
co-grafen



grafen met 4 knopen
sloane A000088

co-grafen



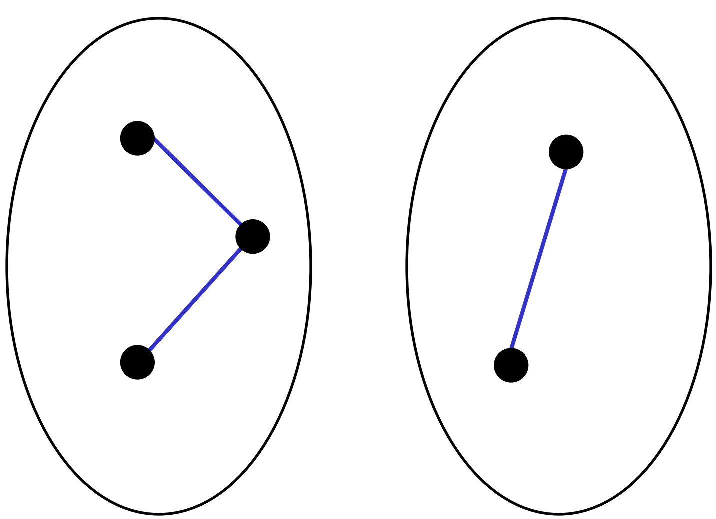


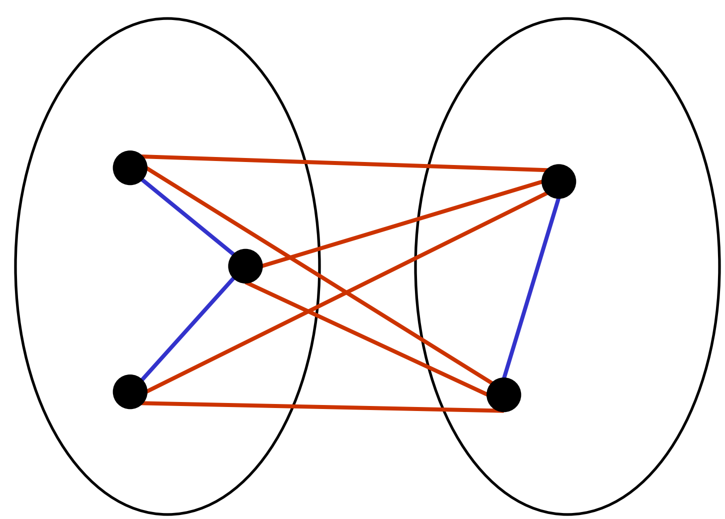
co-grafen

de graaf met één knoop is een co-graaf.
als G_1 en G_2 disjuncte co-grafen zijn,
dan zijn $G_1 \cup G_2$ en $G_1 + G_2$ co-grafen.

P_4 pad graaf

een co-graaf heeft P_4 niet als geïnduceerde deelgraaf (bewijs met inductie zie dictaat)





end …