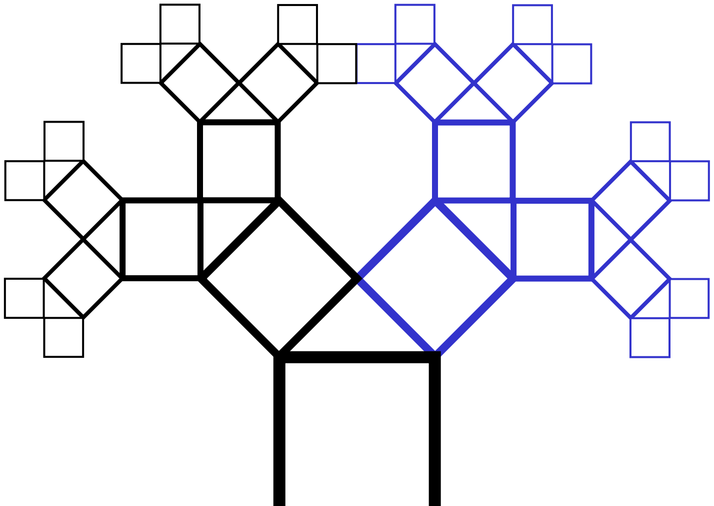
Recursie en inductie

deel 1



Achtste college

recursie en inductie

Geen hoofdstuk in Schaum, maar ‘volledige inductie’ komt wel aan de orde als bewijstechniek.
Ook recursief gedefinieerde functies zijn terug te vinden, in §3.6 van het boek.

Recursie is een reken-/programmeertechniek, zie ook Programmeermethoden en Algoritmiek.

Hier gebruiken we recursie/inductie om functies/verzamelingen van objecten te definiëren.

Er is een klein dictaatje beschikbaar om door te lezen, gemaakt door H.J. Hoogeboom. Zie de website.
De paragraaf over co-grafen behandelen we hier niet. Ook op de semantiek van rekenkundige expressies gaan we niet in.

inductie en recursie

We kunnen inductie gebruiken om (verzamelingen van) objecten te definiëren; we spreken dan van een inductieve definitie: definieer grotere in termen van kleinere onderdelen.
Zo kunnen we ook functies definiëren in termen van kleinere versies van zichzelf: dat heten dan ook wel recursief gedefinieerde functies.

Een inductief bewijs is een methode om een eigenschap te bewijzen van objecten die inductief gedefinieerd zijn. Ook wel volledige inductie als het over $\mathbb{N}$ gaat.

Als we de recursieve/inductieve definitie gebruiken als reken- of programmeertechniek om bijvoorbeeld een recursief gedefinieerde functie te berekenen, spreken we van recursie. Meestal is er ook een iteratieve manier: gewoon herhalen, een simpele loop bijvoorbeeld.

Inductieve definitie:
begin bij de kleinste objecten en geef aan hoe uit kleinere objecten grotere worden opgebouwd: van klein naar groot

Recursieve definitie:
geef aan hoe een grotere waarde/object uit een kleinere wordt gemaakt: van groot naar klein

Het verschil tussen inductie en recursie is dus vaak een kwestie van perspectief.

Recursief/inductief gedefinieerde *rij*

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

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

inductief

iteratief

0, 1, 3, 6, 10, …

wel-gedefinieerd

Een *functie* is recursief gedefinieerd als deze naar zichzelf verwijst.
Om er ook echt iets mee te kunnen (=berekenen) moet voldaan zijn aan de volgende twee eisen:

- Een of meerdere basisgevallen waarin de functiewaarde direct gegeven is
- De functie refereert altijd naar echt kleinere gevallen, dat wil zeggen in de richting van het basisgeval

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

inductie en recursie

$a_0 = 0$

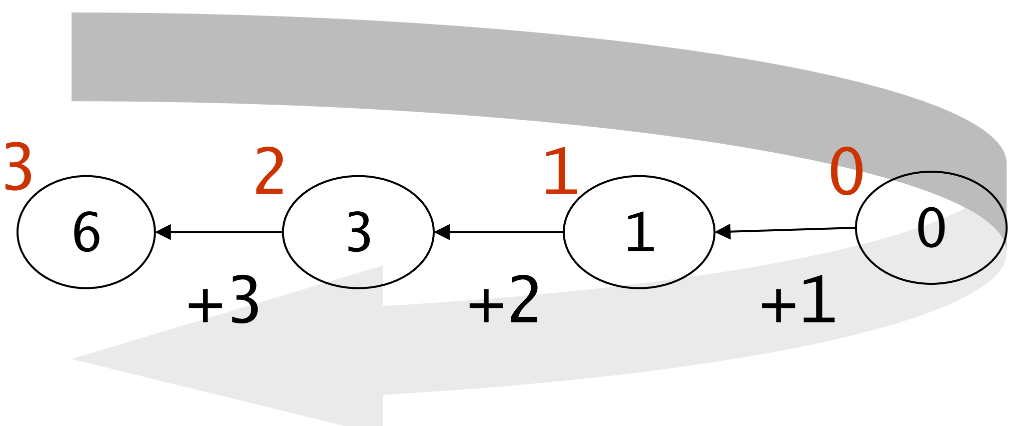
$a_n = a_{n-1} + n \quad (n \geq 1)$

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

inductief

iteratief

0, 1, 3, 6, 10, …



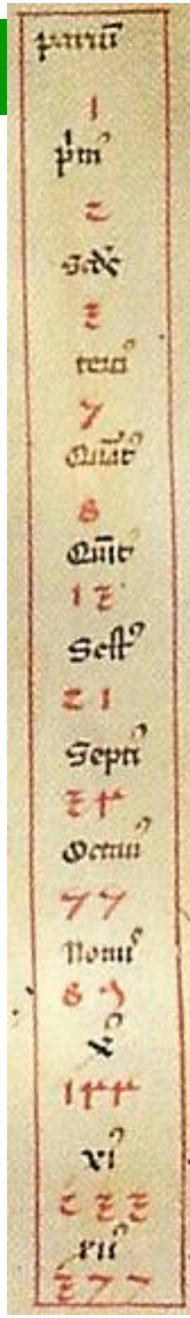
$a_3 =$
$a_2 + 3 =$
$a_1 + 2 + 3 = a_1 + 5 =$
$a_0 + 1 + 5 = a_0 + 6 =$
$0 + 6 = 6$

recursief

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

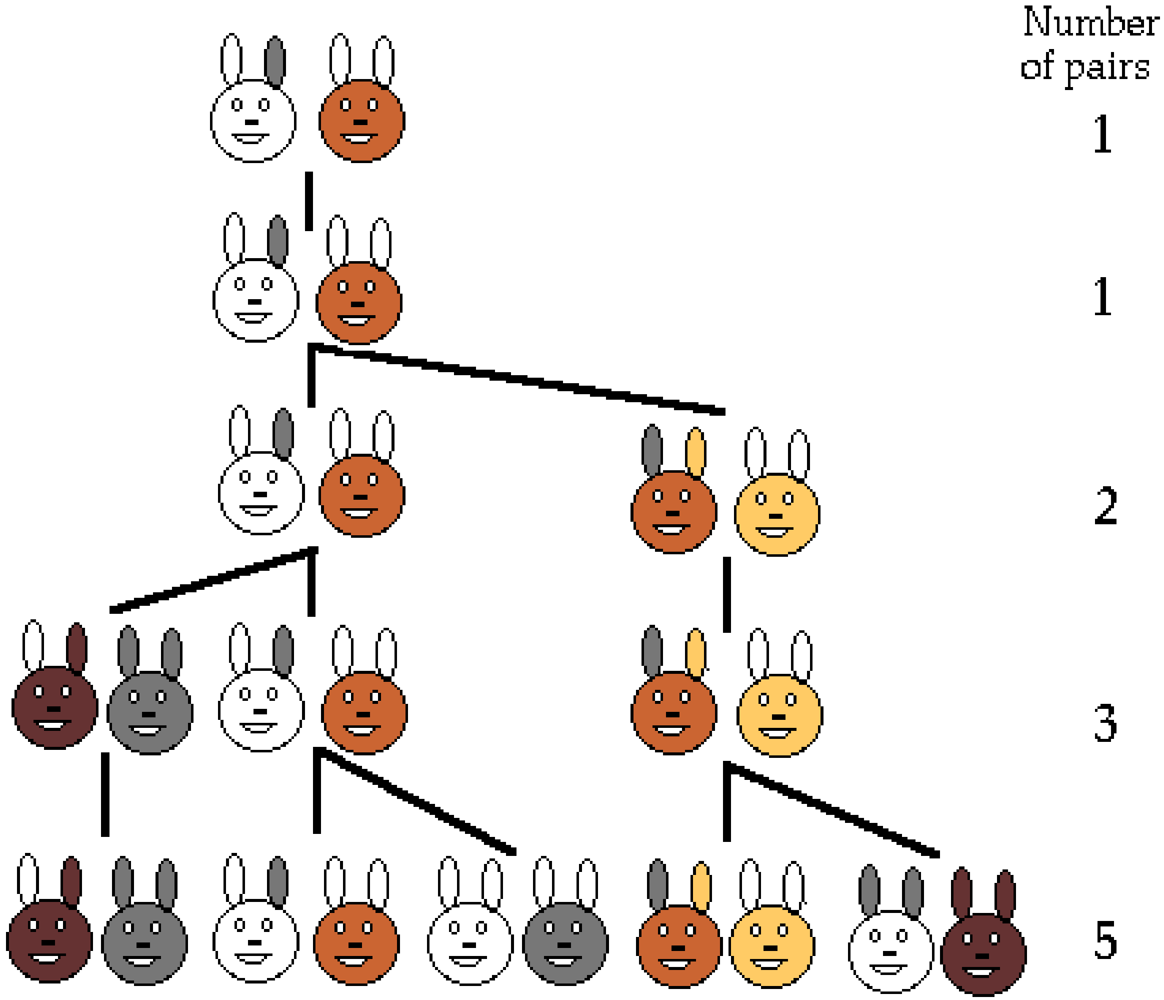
How Many Pairs of Rabbits Are Created by One Pair in One Year

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)

$a_0 = 0 \qquad a_1 = 1$

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



inductieve definitie

Fibonacci

$a_0 = 0 \qquad a_1 = 1$

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

$a_2 = a_1 + a_0 = 1 + 0 = 1$

$a_3 = a_2 + a_1 = 1 + 1 = 2$

$a_4 = a_3 + a_2 = 2 + 1 = 3$

$a_5 = a_4 + a_3 = 3 + 2 = 5$

inductief

iteratief

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

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

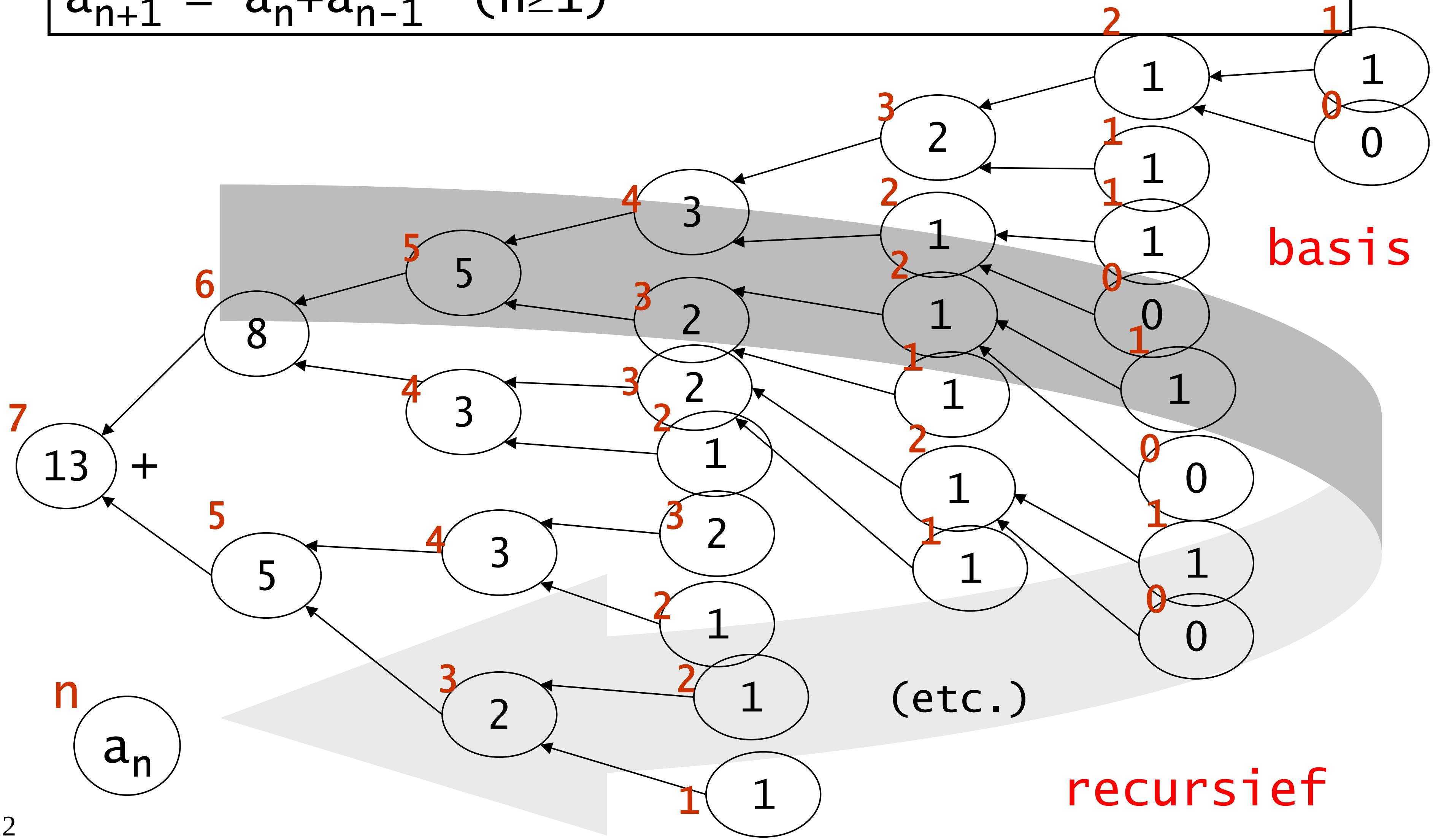
$$
\begin{aligned}
a_5 &= a_4 + a_3 \\
&= a_3 + a_2 + a_2 + a_1 \\
&= a_2 + a_1 + a_1 + a_0 + a_1 + a_0 + a_1 \\
&= a_1 + a_0 + a_1 + a_1 + a_0 + a_1 + a_0 + a_1 \\
&= 1 + 0 + 1 + 1 + 0 + 1 + 0 + 1 = 5
\end{aligned}
$$

recursief

Fibonacci

$a_0 = 0 \qquad a_1 = 1$

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



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 Fibonacci-getal te bepalen moeten eerst de twee voorgangers worden bepaald, die vervolgens opgeteld worden. Maar daarvoor berekenen we eerst ook voor beide weer twee voorgangers. Zonder extra aandacht wordt zo veel werk dubbel gedaan. En je hebt twee basisgevallen nodig.

*Meestal worden ze trouwens aangegeven met F_n.

inductief

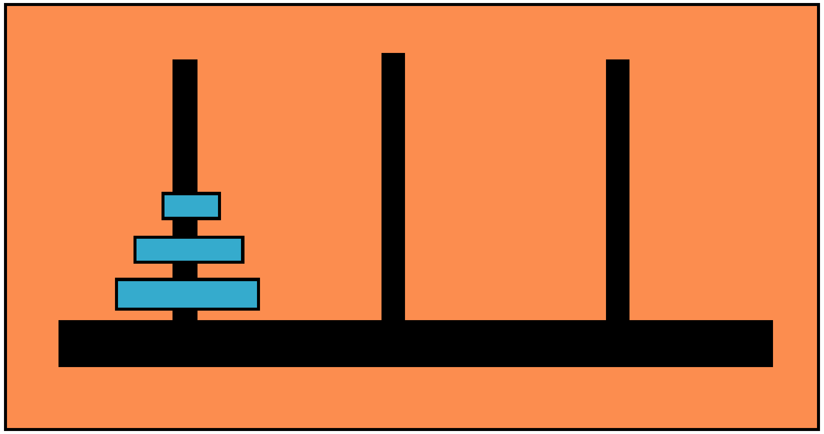
van klein naar groot

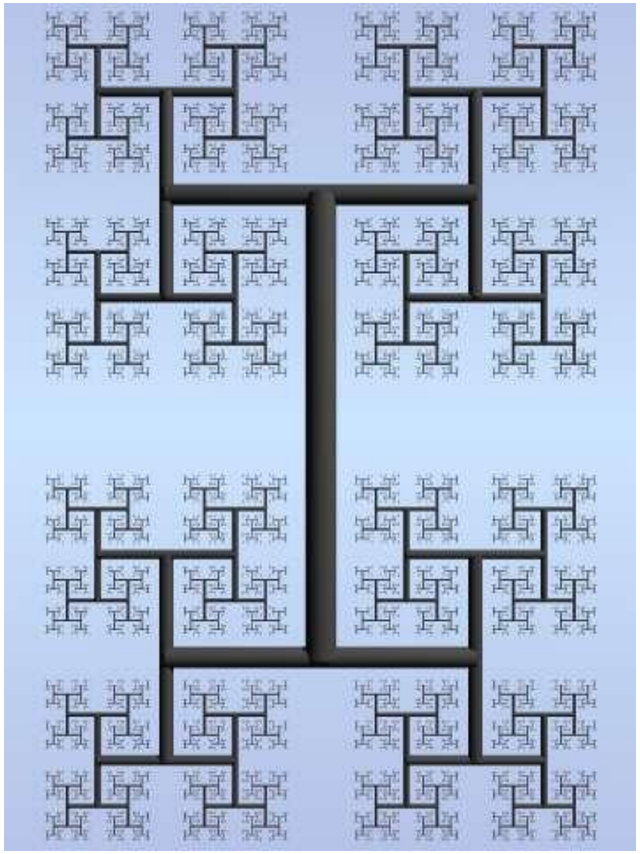
bottom-up

recursief

in termen van zichzelf

top-down

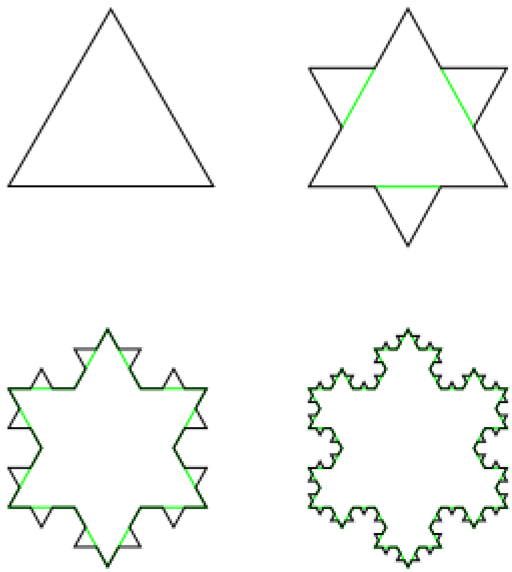





Droste
HAARLEM-HOLLAND
cacao
Netto 250 g


6 kr
Sverige
6 kr
Sverige

De torens van Hanoi is een puzzel die op recursieve manier kan worden opgelost, zie Programmeermethoden/Algoritmiek.
Het Droste-blikje wordt in Nederland vaak aangehaald als voorbeeld van recursie.



De andere twee voorbeelden zijn fractals. Op de postzegel bijvoorbeeld de sneeuwvlok van Koch, opgebouwd door telkens dezelfde bewerking uit te voeren, maar dan steeds kleiner. Inzoomend krijg je telkens dezelfde figuur.

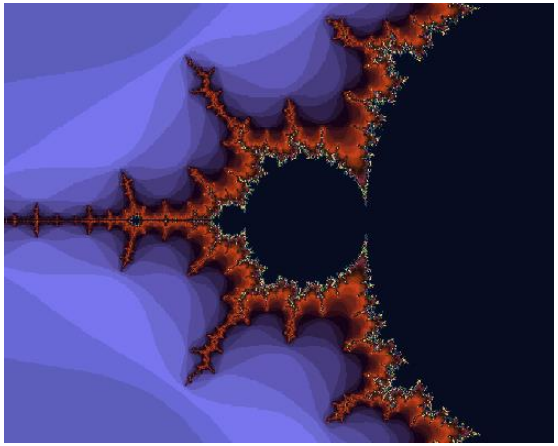
fractals

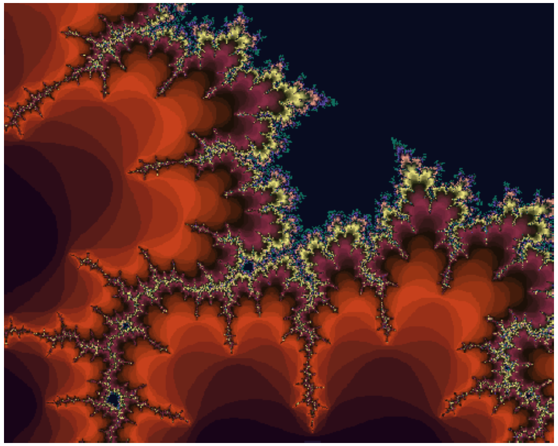
Volgens Wikipedia: een **fractal** is een meetkundige figuur die zelfgelijkend is, dat wil zeggen opgebouwd is uit delen die min of meer gelijkvormig zijn met de figuur zelf.

Een van de bekendste fractals is de Mandelbrot-fractal.

$$z_0 = 0;\ z_{n+1} = z_n^2 + c \quad \text{(c een complex getal)}$$

Als de baan van z_n naar oneindig gaat krijgt het punt c een kleurtje, anders blijft het zwart. Dat levert onderstaande plaatjes op. In tegenstelling tot veel andere fractals herhaalt deze figuur zich niet bij inzoomen.





inductieve definitie van:

- verzamelingen
- relaties
- rijtjes
- functies
- bomen
- ordeningen
- syntax
- …

structurele /volledige inductie

- eigenschappen bewijzen

E = { 0, 2, 4, 6, … }   … stipjes …

even getallen: inductieve definitie

<u>basis</u>

i. $0 \in E$

<u>inductiestap</u>

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

<u>uitsluiting</u>

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.

Ook talen (verzameling van strings over een of ander alfabet) kunnen inductief gedefinieerd worden. Een voorbeeld op de sheet hierna.
De inductief gedefinieerde strings van de sheet daarna 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).

De symmetrische ordening is een manier om knopen in een binaire boom te nummeren die recursief gedefinieerd is. Meer over (binaire) bomen in het volgend hoofdstuk.

taal: strings

<u>basis</u>

i. b $\in$ L

<u>inductiestap</u>

ii. als x $\in$ L dan abx $\in$ L

<u>uitsluiting</u>

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

b, abb, ababb, abababb, ababababb, abababababb, …

De taal L over het alfabet {a,b} bestaat uit alle strings (woorden) van de vorm $(ab)^n$ b, dus een aantal (≥ 0) keer ab, gevolgd door een b.

(volgt nog)

taal: binaire bomen

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, …

B is een taal over het alfabet {a,b,+}

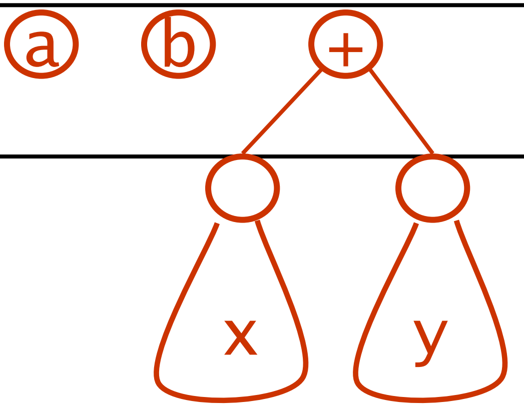
(dit zijn 'gecodeerde' binaire bomen, zie later)

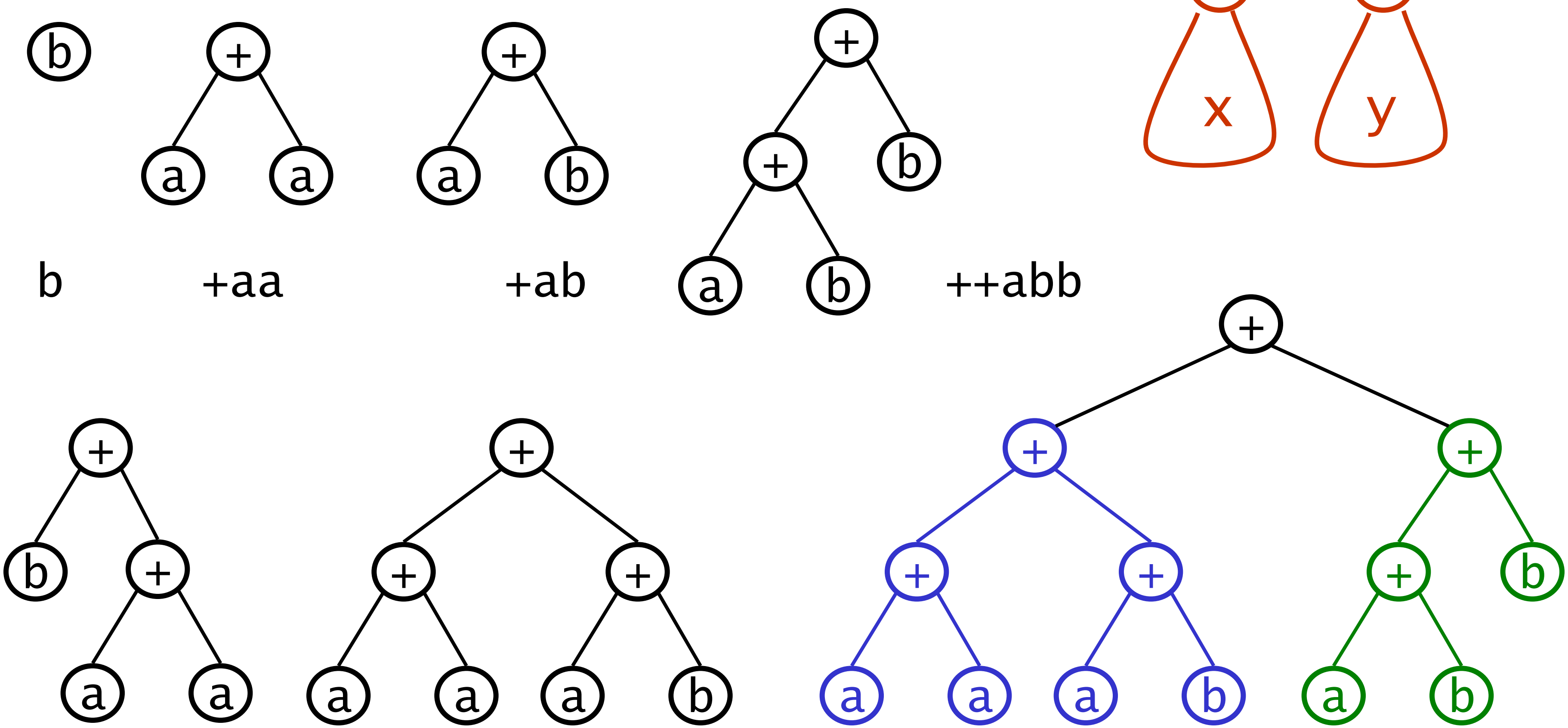
binaire bomen (?)

(volgt nog)

i. $a \in B$, $b \in B$

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



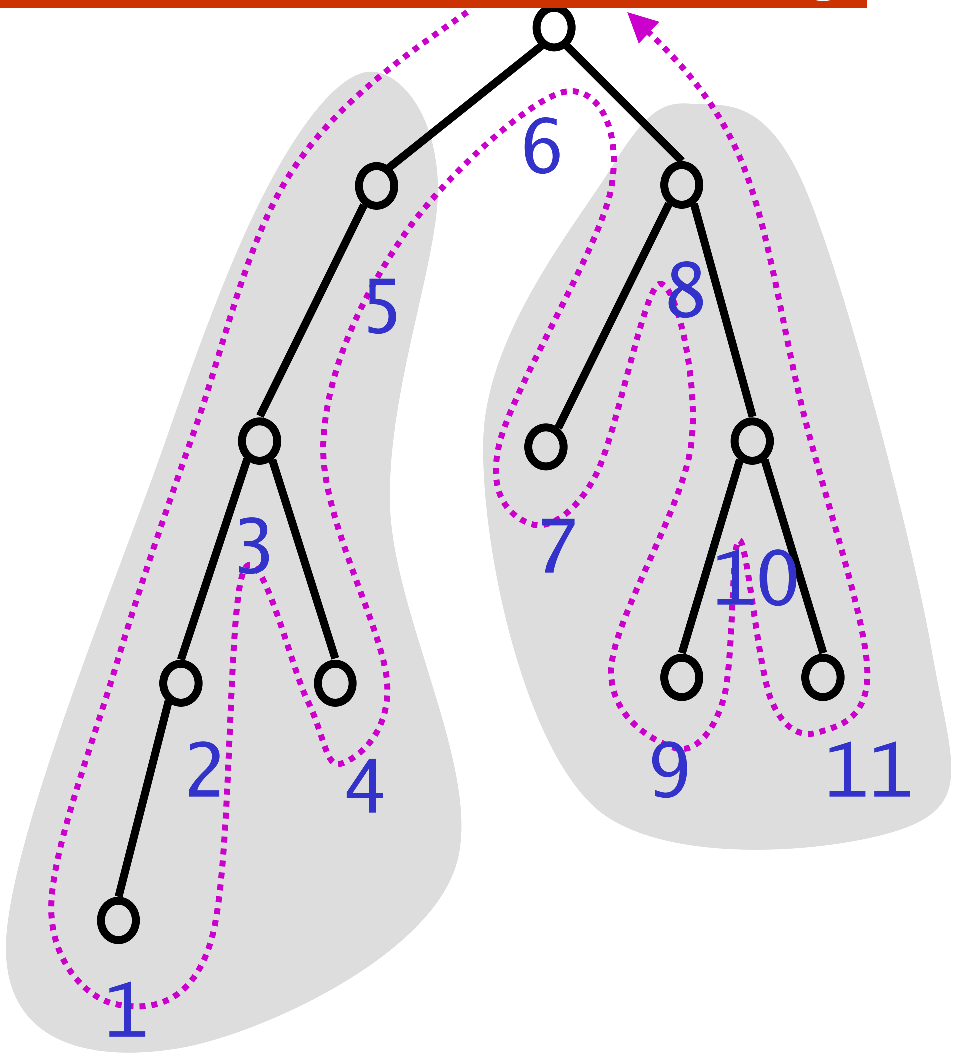


+b+aa ++aa+ab +++aa+ab++abb, …

(volgt nog)

symmetrische ordening

Bij binaire bomen:
deze kunnen *recursief* gedefinieerd worden, waardoor diverse eigenschappen van de boom ook recursief geformuleerd kunnen worden.



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

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

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

Ofwel:
i. $(0,1) \in <$
ii. $(x,y) \in <$, dan
$(x,y+1) \in <$ en
$(x+1,y+1) \in <$
iii. …

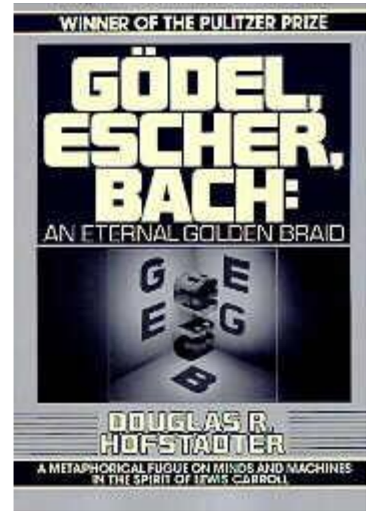
$$
\begin{array}{ccccccccc}
 & & & & 0<1 & & & & \\
 & & & 0<2 & & 1<2 & & & \\
 & & 0<3 & & 1<3 & & 2<3 & & \\
 & 0<4 & & 1<4 & & 2<4 & & 3<4 & \\
0<5 & & 1<5 & & 2<5 & & 3<5 & & 4<5
\end{array}
$$

Tot dusver gezien: inductief gedefinieerde verzamelingen

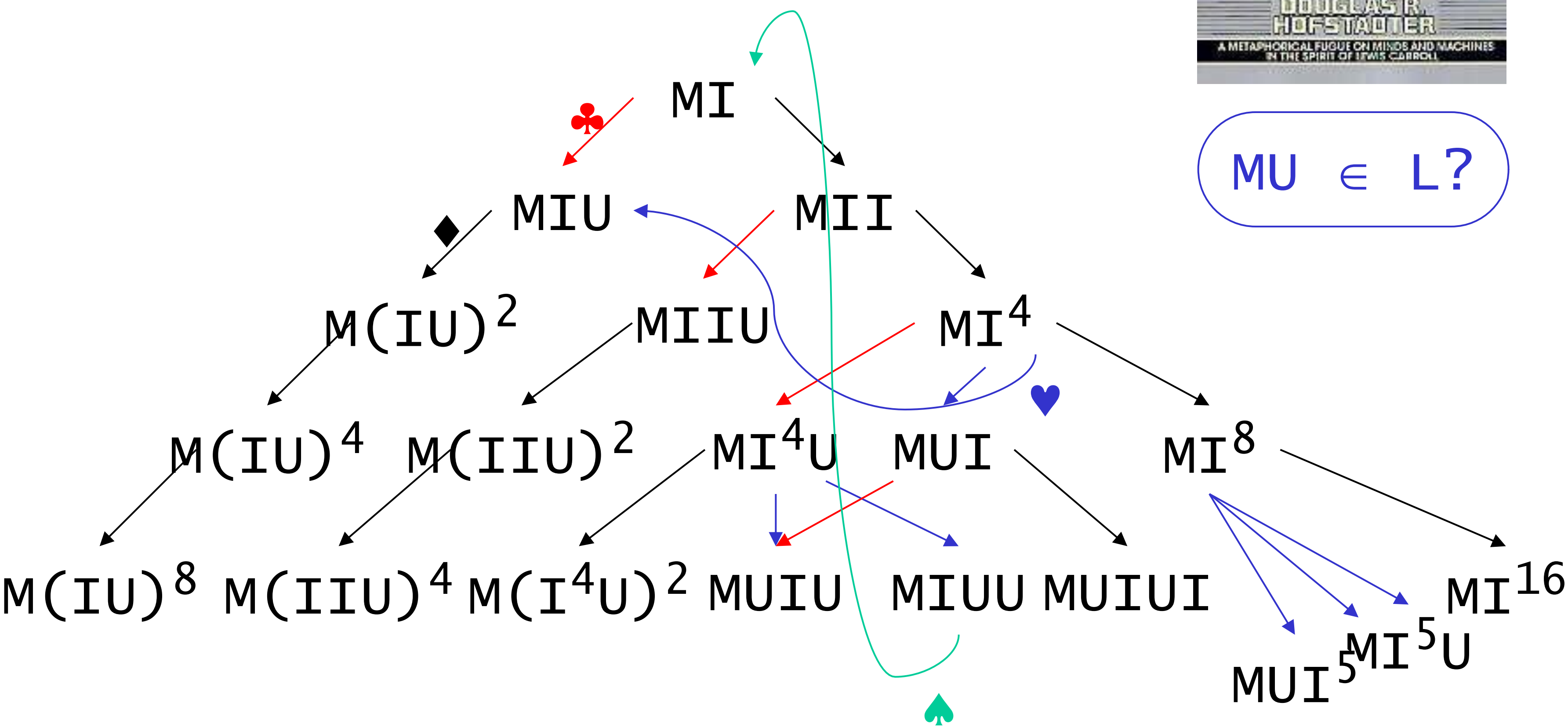
- getallen
- strings
- paren (relatie)

taal ‘MU puzzle’

- 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



MU ∈ 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 = een verzameling strings.
De vraag is, behoort MU tot die taal?

Antwoord volgt.

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

sommetje

Geef een inductieve definitie van de verzameling L van strings die bestaan uit een aantal a's gevolgd door evenveel b's.

L is dus een taal over {a,b} en er geldt dat $L = \{ w \in \{a,b\}^* \mid w = a^n b^n \text{ met } n \geq 0 \}$

syntax: geheel getal

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

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

-3157 stelt volgens bovenstaande afleiding dus een "geheel getal" voor

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).

BNF: zie bv https://en.wikipedia.org/wiki/Backus-Naur_form

```
⟨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⟩
```

BNF / Backus–Naur Form

rekenkundige expressies

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

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

syntax

+ − · / zijn symbolen, geen operaties

27

0014 -(0014)

((1+13)*8)

(8/(2-2))

(27/(15+12-27))

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

Dit zijn allemaal correcte rekenkundige expressies wat betreft de vorm, op twee na. Welke zijn dat?

Ze hebben nog geen betekenis

rekenkundige expressies

Bij de definitie van rekenkundige expressies maken we onderscheid tussen de vorm (*syntaxis*; de strings) en de betekenis (*semantiek*; de waarde, een geheel getal). Op basis van de inductieve syntaxdefinitie kunnen we de semantiek precies definiëren, zodat elke syntactisch correcte string een unieke betekenis krijgt.

Zie ook het dictaatje §2.2

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

recursieve functie

Een functie is recursief gedefinieerd als deze naar zichzelf verwijst.
Om er ook echt iets mee te kunnen (=berekenen) moet voldaan zijn aan de volgende twee eisen:

- een of meerdere basisgevallen waarin de functiewaarde direct gegeven is
- de functie refereert altijd naar echt kleinere gevallen, dat wil zeggen in de richting van het basisgeval

recursieve functie

In tegenstelling tot de inductieve definities die we hiervoor zagen, is een uitsluitingsregel hier niet nodig. Een functie kent immers aan *elk* element van het domein precies *één* waarde toe. Andere waarden zijn dus per definitie al uitgesloten.

recursieve functie

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

$f(n) = n!$

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

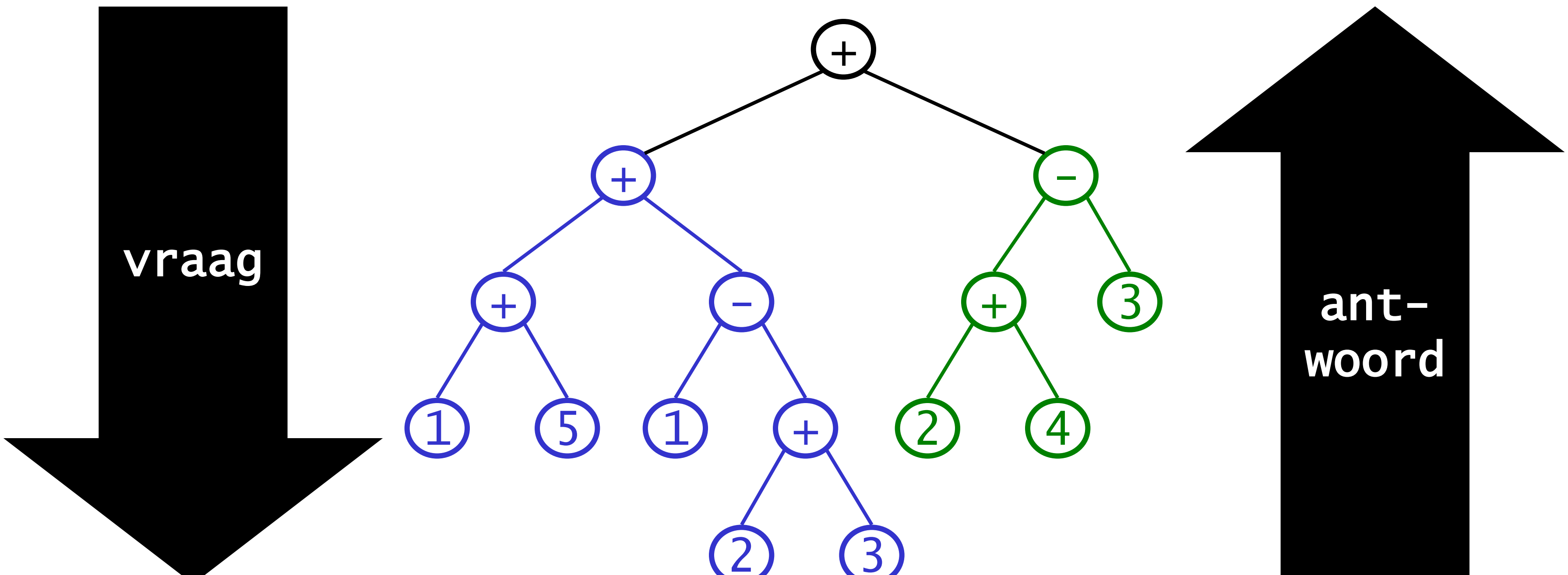
$$g(8) = 8 + g(7) = 8 + 7 + g(6) = \ldots$$
$$= 8+7+6+5+4+3+2+1+g(0) = 36$$
$$g(n) = n + n-1 + \ldots + 1 + 0 = \tfrac{1}{2}n(n+1)$$

(volgt nog)

recursieve functies bij bomen

```
basis
  f(blad)  = …
recursie
  f(knoop) = f(links) & f(rechts)
```

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



recurrente betrekking

$F_0 = 0$
$F_1 = 1$
$F_{n+1} = F_n + F_{n-1} \quad (n \geq 1)$

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

Fibonacci

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

Binet

gulden snede
$x^2 = x + 1$

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

recurrente betrekking

Ook voor de Fibonacci-getallen F_n bestaat een gesloten formule.

Het getal (1+wortel(5))/2 dat in de formule voorkomt heet wel de gulden snede, vaak genoteerd als φ. Deze komt o.a. voor in de lengte van de diagonalen van een regelmatige vijfhoek.

Het is tevens de oplossing van de kwadratische vergelijking die eenzelfde structuur heeft als de recursieve definitie van de Fibonacci-getallen (en dat is geen toeval).

recurrente betrekking

$t_0 = 5,\ t_1 = 6$

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

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

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.* Bovenstaande rb is *niet* homogeen, maar het geheime recept kan uitgebreid worden tot niet-homogene tweede orde rb's.

geheim recept

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

GEEN TENTAMENSTOF !!

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

recurrente betrekking

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 recurrente betrekking is **WEL** kennis die je moet hebben.
Dat is namelijk een voorbeeld van Volledige Inductie, hierna.

inductieve bewijzen

Structurele inductie (inductie naar de opbouw) is de bewijstechniek die hoort bij inductief opgebouwde ‘objecten’ zoals bomen of talen.

eigenschap natuurlijke getallen $\Rightarrow$ volledige inductie

Bij eigenschappen van de (verzameling der) natuurlijke getallen heet de methode volledige inductie.

mooie formule

$$f(n) = n^2 + n + 41$$

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

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 $n \geq 0$

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

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 $n \geq 0$

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

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

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

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.

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

drievouden $\Rightarrow$ drievoud

bewijs

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

inductie

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

i. b

5

ii. inductie

inductie-aanname:

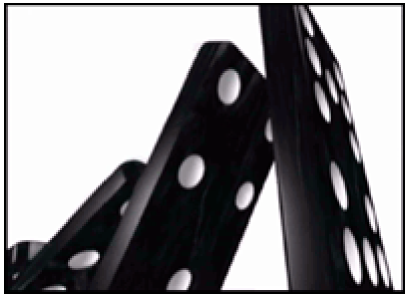
Neem aan $5^n - 2^n$ is een drievoud $(n \geq 0)$

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

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

drievouden $\Longrightarrow$ drievoud

volledige inductie



$\mathbb{N}$ natuurlijke getallen

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

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

inductie-aanname
inductie-hypothese

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

(werk)college

Volgende college:
dinsdag 6 november,
13.30 – 15.15 in zaal C3 (Gorlaeus)

Werkcollege deze week:
vrijdag 2 november,
9.00 - 10.45 in zalen 402, 405, …
(Snellius)