

Formeel Denken

Herman Geuvers

Deels gebaseerd op het herfst 2002 dictaat van Henk Barendregt en Bas Spitters,
met dank aan het Discrete Wiskunde dictaat van Wim Gielen

Herfst 2004

Contents

1	Combinatoriek	1
----------	----------------------	----------

1. Combinatoriek

Graphen

Dit hoofdstuk is een korte inleiding in de graphentheorie. Graphen kom je op allerlei plaatsen tegen bijvoorbeeld bij talen, netwerken, datastructuren, elektrische circuits, transport problemen en stroomschema's.

Intuïtief bestaat een graph uit een verzameling P van punten en een verzameling L van lijnen tussen punten. Twee voorbeelden:



Natuurlijk zijn er nu allerlei knagende onzekerheden, zoals: ‘Wanneer zijn twee graphen gelijk?’, ‘Moeten die punten in een plat vlak liggen?’, ‘Mogen de lijnen elkaar snijden?’. Al je twijfels verdwijnen als sneeuw voor de zon, dankzij de volgende formele definitie:

1.1. DEFINITIE. Een *graph* is een paar $\langle P, L \rangle$ waarbij P een verzameling is, en L een verzameling van uit twee elementen bestaande deelverzamelingen van P . (de elementen van P noemen we ‘punten’, de elementen van L noemen we ‘lijnen’; een lijn noteren we niet als $\{a, b\}$ maar als (a, b) . De lijn (a, b) is dus gelijk aan de lijn (b, a) .) De graph G_1 uit ons eerste voorbeeld is dus $\langle P, L \rangle$ met $P = \{1, 2, 3, 4\}$ en $L = \{(1, 4), (2, 3), (2, 4)\}$. De graph G_2 is het paar $\langle P, L \rangle$ met $P = \{a, b, c, d\}$ en $L = \{(a, c), (a, d), (c, d)\}$

NB. Merk op dat de lijnen in onze graphen geen *richting* hebben: het zijn geen pijlen. Verder is het niet mogelijk dat er tussen twee punten p en q meerdere lijnen zijn. Je zou de definitie van ‘graph’ kunnen veranderen en dit wel toestaan, maar het blijkt dat we met Definitie 1.1 al genoeg stof tot nadenken hebben en al veel interessante eigenschappen kunnen beschrijven en bestuderen.

1.2. DEFINITIE. (Hierin is G de graph $\langle P, L \rangle$, en zijn p en q punten)

- Een *buur* van p is een punt x met $(p, x) \in L$.
- De *graad* (of: *valentie*) van p is het aantal burens van p .
- Een *pad* van p naar q is een rijtje van verschillende lijnen $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$ met $n > 0$, $x_0 = p$ en $x_n = q$. Een kortere notatie voor dit pad: $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n$.
- Met ‘ G is *samenhangend*’ bedoelen we: tussen ieder tweetal punten bestaat een pad.
- Een *component* van G is een zo groot mogelijk samenhangend deel van een graph.
- Een *cykel* is een pad van een punt naar zichzelf.

- Met ‘ G is *planair*’ bedoelen we: je kunt G in het platte vlak zó tekenen dat de (gebogen) lijnen elkaar niet snijden.
- Met ‘ G is een *boom*’ bedoelen we: G is samenhangend en bevat geen cyclen.

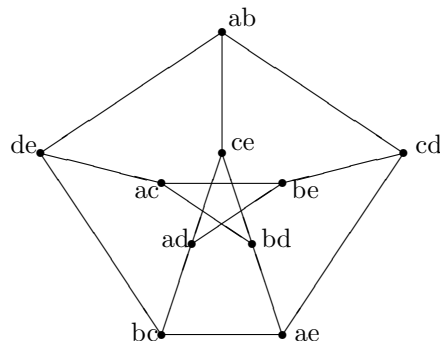
De vraag of een graph planair is is bijvoorbeeld belangrijk als de graph een electrisch circuit voorstelt dat we op een chip willen branden.

1.3. VOORBEELD. • De ‘landgraph’ is $\langle P, L \rangle$ waarbij P de verzameling van alle landen van de wereld is, en L de relatie ‘grenst aan’. De burenen van Nederland zijn dan Duitsland en België. De graad van Nederland is 2. Een pad van Nederland naar Spanje is bijvoorbeeld Nederland $\rightarrow$ Duitsland $\rightarrow$ Frankrijk $\rightarrow$ Spanje. De landgraph is niet samenhangend. $\{\text{Engeland, Schotland}\}$ is een component. Nederland $\rightarrow$ Duitsland $\rightarrow$ België $\rightarrow$ Nederland is een cykel. De landgraph is planair.

- $K_4 = \langle \{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\} \rangle$ (de volledige graph met vier punten). Algemeen: K_n is de graph $\langle \{1, \dots, n\}, \{(p, q) \mid 1 \leq p < q \leq n\} \rangle$. De graph K_4 is planair. Als je dat wilt inzien, moet je niet naar de linker maar naar de rechter tekening van K_4 kijken:



- De Petersen-graph is $\langle P, L \rangle$ met $\begin{cases} P = \{ab, ac, ad, ae, bc, bd, be, cd, ce, de\} \\ L = \{(p, q) \mid p \text{ en } q \text{ hebben geen letter gemeen}\} \end{cases}$



Je kunt laten zien dat de Petersen graph niet planair is.

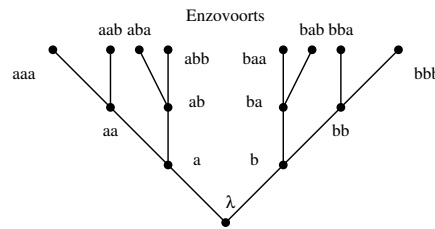
- Bij een taal die gegeven is d.m.v. een inductieve taaldefinitie, kunnen we een graph maken. Bekijk bijvoorbeeld eens de taal die als volgt geproduceerd wordt:

axioma	λ
regel	$x \Rightarrow xa$ $y \Rightarrow yb$

Bij deze taal kunnen we een graph maken door

- eerst de woorden op te schrijven die er in zitten op grond van het axioma (dat zijn de eerste punten van de graph),
- dan stap voor stap woorden (en dus punten) toe te voegen die er op grond van de toepassing van een regel inzitten, en een lijn te maken naar het woord op grond waarvan we dit woord gekregen hebben.

We krijgen dan de volgende graph:



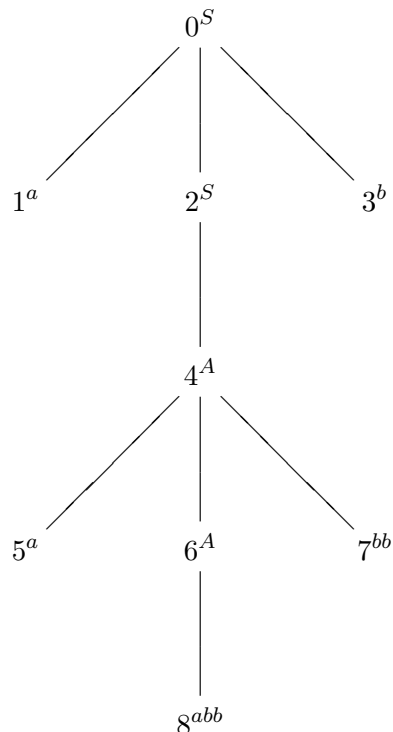
Deze graph is een boom.

- Bij een context-vrije grammatica en een woord dat door deze grammatica geproduceerd wordt, kun je een *parseerboom* maken. Een parseerboom geeft grafisch weer hoe het woord geproduceerd wordt. Bekijk de volgende context-vrije grammatica uit Opgave ??.

$$S \rightarrow aSb \mid A \mid \lambda$$

$$A \rightarrow aAbb \mid abb$$

Het woord $aaabbbbb$ wordt geproduceerd door deze grammatica, op de volgende manier: $S \rightarrow aSb \rightarrow aAb \rightarrow aaAbbb \rightarrow aaabbbbb$. Hieronder staat de parseerboom van deze productie.



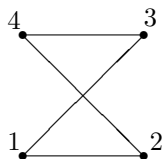
Dus als we $S \rightarrow aSb$ doen, geven we het punt met label S drie onderburen met labels a , S en b en als we $A \rightarrow aAbb$ doen, geven we het punt met label A drie onderburen met labels a , A en bb . Als je de (labels van de) bladeren van de boom van links naar rechts achter elkaar zet krijg je het woord dat bij deze parseerboom hoort.

NB. Informatici tekenen een boom bijna altijd ‘op de kop’, dus met de ‘wortel’ boven en de takken naar beneden gericht. Wiskundigen tekenen een boom altijd met de wortel onder (zie vorige voorbeeld). Bij een parseerboom worden doorgaans alleen de labels opgeschreven bij de punten (en laat men de punten dus ongenummerd).

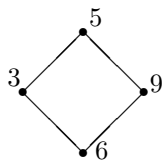
1.4. DEFINITIE (Bijjectie). Een afbeelding f van een verzameling A naar een verzameling B heet een *bijjectie* als ieder element in B precies één origineel heeft.

1.5. DEFINITIE (Isomorfie van graphen.). We noemen twee graphen $\langle P, L \rangle$ en $\langle P', L' \rangle$ *isomorf* als er een bijjectie $\varphi : P \rightarrow P'$ bestaat zodat voor alle x, y in P , $(x, y) \in L$ desda $(\varphi(x), \varphi(y)) \in L'$. Anders gezegd: twee graphen zijn isomorf als ze, afgezien van de namen van de punten, hetzelfde zijn.

Bijvoorbeeld:



is isomorf met



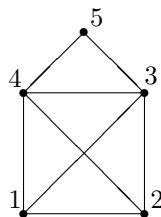
Een isomorfisme φ is:

$$\begin{aligned} 1 &\mapsto 6 \\ 2 &\mapsto 9 \\ 3 &\mapsto 3 \\ 4 &\mapsto 5 \end{aligned}$$

Isomorfe graphen zijn ‘hetzelfde’ als we alleen geïnteresseerd zijn in graph-theoretische eigenschappen. Bijvoorbeeld: Als G en G' isomorph zijn en G is samenhangend, dan is G' dat ook.

1.6. DEFINITIE (Euler-circuits.). Een *Euler-pad* in een graph $\langle P, L \rangle$ is een pad waarin iedere lijn uit L precies één keer voorkomt. Een *Euler-circuit* of *Euler-cykel* is een cykel waarin iedere lijn uit L precies één keer voorkomt.

Bijvoorbeeld:



$1 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 2$ is een Euler-pad. Omdat punt 1 graad 3 heeft, is er geen Euler-circuit. Dat kun je inzien door te kijken naar het aantal keren dat je bij een cykel door het punt 1 loopt: is dit aantal 1, dan mis je één van de drie bij 1 samenkomende lijntjes, en is aantal 2, dan doorloop je minstens een van deze lijntjes dubbel. Als je deze redenering iets algemener opschrijft, staat er een bewijs van de volgende eenvoudige stelling:

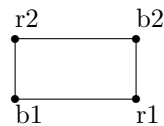
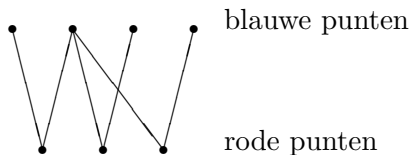
1.7. STELLING (Euler). In een samenhangende graph geldt:

1. Er bestaat een Euler-circuit dan en slechts dan als ieder punt een even graad heeft.
2. Er bestaat een Euler-pad dan en slechts dan als er hoogstens twee punten van oneven graad zijn.

Euler-circuits zijn bijvoorbeeld van belang voor de krantenbezorger of de wijkagent, die iedere straat van zijn wijk precies één keer wil doorlopen en bij zijn uitgangspunt wil terugkeren. Een heel ander probleem heeft de ‘travelling salesman’, die iedere klant (of iedere stad) precies één keer wil bezoeken en daarna weer naar huis wil. Hij is gebaat bij een ‘Hamilton-circuit’:

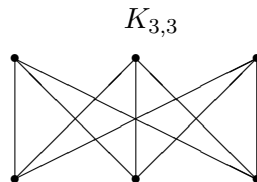
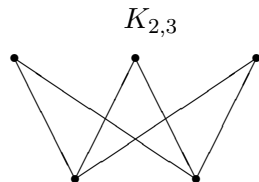
1.8. DEFINITIE (Hamilton-circuits.). Een *Hamilton-pad* in een graph $\langle P, L \rangle$ is een pad waarin je ieder punt van P precies één keer ontmoet. Een *Hamilton-circuit* of *Hamilton cykel* is een cykel waarin ieder punt precies één keer optreedt (afgezien van beginpunt = eindpunt natuurlijk)

1.9. DEFINITIE (Bipartite graphen.). Een graph $\langle P, L \rangle$ heet *bipartite* als P te schrijven is als $P_1 \cup P_2$ zó dat iedere lijn loopt van een punt uit P_1 naar een punt uit P_2 . Anders gezegd: je kunt de punten zó blauw en rood kleuren dat iedere lijn een blauw en een rood uiteinde heeft.



Twee voorbeelden:

De *volledige bipartite graph* met m rode en n blauwe punten, waarbij ieder rood punt met ieder blauw punt verbonden is, wordt $K_{m,n}$ genoemd



1.10. DEFINITIE (Kleuring van graphen.). Een *punt-kleuring* van een graph $\langle P, L \rangle$ is een functie $f : P \rightarrow \{1, \dots, n\}$ zodat als (p, q) een lijn is, dan $f(p) \neq f(q)$ (elk punt krijgt één der n kleuren, buren krijgen niet dezelfde kleur).

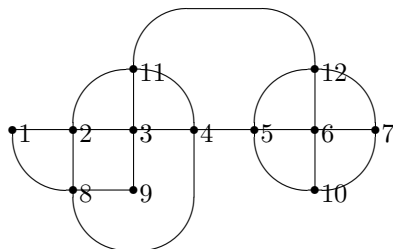
Het *kleurgetal* (of: *chromatisch getal*) van de graph is de kleinste n waarvoor dit mogelijk is.

Bipartite graphen zijn dus de graphen met kleurgetal 1 of 2. Een interessante stelling van Appel en Haken, waarvan het in 1975 gevonden bewijs het niveau van dit college helaas ontstijgt:

1.11. STELLING (Vierkleurenstelling.). Het kleurgetal van een planaire graph is hoogstens 4. Anders gezegd: Iedere landkaart kan met hooguit 4 kleuren worden ingekleurd, zodanig dat twee aangrenzende landen een verschillende kleur hebben.

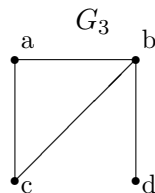
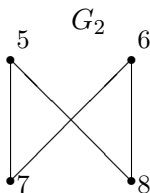
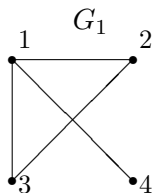
Opgaven

1. Bewijs dat er in een boom van een punt p naar een ander punt q precies één pad bestaat.
2. Hier is een plattegrond G van een dorpje, waarbij de straten aangegeven worden door lijntjes. Op ieder hoekpunt bevindt zich een kroeg. De kroegen zijn aangegeven door punten, genummerd van 1 t/m 12:

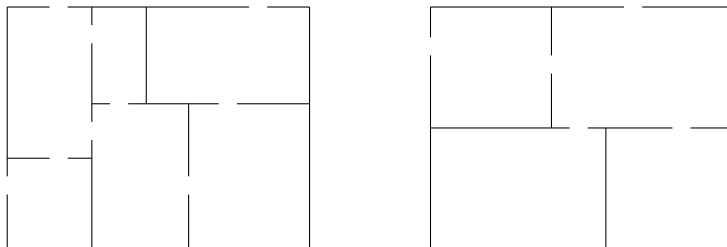


Formuleer de volgende vragen in termen van Hamilton- of Euler-circuits/paden, en beantwoord ze:

- (a) Is het mogelijk, een wandeling te maken waarbij je iedere straat precies één keer doorloopt?
 - (b) Is het mogelijk, een wandeling te maken waarbij je iedere straat precies één keer doorloopt en bovendien begint en eindigt bij kroeg 3?
 - (c) Bestaat er een kroegentocht waarin iedere kroeg precies één keer voorkomt?
3. Een *brug* in een graph G is een lijn l waarvoor geldt: als je l uit G weglaat, wordt het aantal componenten verhoogd. Bewijs dat in een boom iedere lijn een brug is.
 4. Ga na welke van de onderstaande graphen isomorf zijn:



5. Hieronder vind je twee plattegronden van huizen.



- Teken bij elke plattegrond de bijbehorende graph, waarbij je de vertrekken (inclusief het buitengebied) door punten voorstelt, en de deuren door lijnen.
- Zoek voor elk huis uit of het mogelijk is een rondwandelingetje te maken waarin je iedere deur precies één keer gebruikt en bij je uitgangspunt terugkeert.
- Zoek voor elk huis uit of het mogelijk is een rondwandelingetje te maken waarin je ieder vertrek (en de tuin) precies één keer bezoekt en bij je uitgangspunt terugkeert.

6. Welke van de volgende graphen hebben een Hamilton-circuit?



7. Zij G de kubus-graph, met als P de verzameling van de acht hoekpunten, en als L de verzameling van de twaalf ribben

- Is G planair?
- Heeft G een Hamilton-circuit?
- Heeft G een Euler-circuit?

8. Laat zien dat de Peterson graph een Hamilton-pad heeft, maar geen Hamilton-cykel.

9. Laat zien dat als een bipartite graph een Hamilton-pad toelaat het aantal rode en het aantal blauwe punten hoogstens één verschilt.

Veel van de stof uit dit blok is afkomstig uit [?]. Een ander goed naslagwerk is [?].

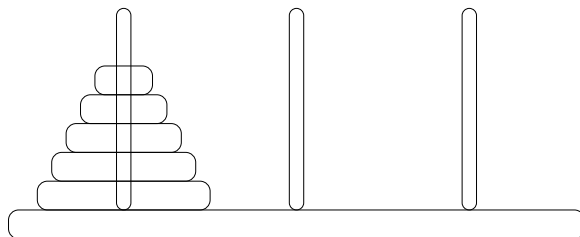
Recursie

Torens van Hanoi

De puzzel de ‘torens van Hanoi’ bestaat uit 3 pinnen en een aantal schijven van verschillend formaat. De bedoeling is om alle schijven naar één ander pin te verschuiven. Iedere zet mogen we een schijf verzetten, maar er mag nooit een grote schijf op een kleinere staan.

In het plaatje hebben we vijf schijven. Kun je deze puzzel oplossen?

Na wat puzzelen lukt het je misschien wel, maar dan weet je vaak niet meer hoe je het gedaan hebt. Verder wil je natuurlijk ook weten hoe je het met zeven schijven doet.



Het belangrijke idee bij het oplossen deze puzzel is om het probleem te *generaliseren*. Kunnen we het probleem oplossen bij een willekeurig aantal schijven? Daarnaast moeten we opmerken dat het probleem voor vijf schijven lijkt op het probleem van vier schijven, wat weer lijkt op het probleem voor drie schijven, etc.

Als ik het probleem voor vier schijven op kan lossen kan ik het ook voor drie schijven. Want laat de grootste schijf maar even liggen. Verplaats de bovenste vier schijven naar pin 2 (We hebben aangenomen dat we dat kunnen). Pak nu de grootste schijf op en verplaats hem naar pin 3. Verplaats nu de andere schijven van pin 2 naar pin 3.

Nu zeg je misschien wat heb ik hieraan? Het probleem voor vier schijven kan ik ook niet oplossen. Nou ja, dan maken we het nog een beetje makkelijker. Als ik het probleem voor drie schijven kan oplossen, dan kan ik het ook voor vier. En als ik het probleem voor twee schijven kan oplossen, dan kan ik het ook voor drie. Ten slotte als ik het probleem voor een schijf kan oplossen, dan kan ik het ook voor twee. Dus *als* ik het probleem voor een schijf kan oplossen, dan kan ik het ook voor vijf. Maar het probleem voor een schijf is erg eenvoudig! Dus we kunnen het nu ook voor vijf schijven.

Ten slotte kunnen we nog opmerken dat er niet bijzonders is aan vijf. Het probleem kunnen we nu ook oplossen voor tien schijven, of voor ieder ander aantal.

We hebben het vorige probleem *recursief* opgelost. Dat wil zeggen, we hebben het probleem zo ingedeeld dat het uit elkaar valt in een aantal deelproblemen die heel veel op elkaar lijken en zodat moeilijkere problemen altijd zijn terug te brengen tot eenvoudigere problemen. Ten slotte moeten we het allereenvoudigste geval natuurlijk wel op kunnen lossen.

Recursie is ook een belangrijke programmeertechniek.

Vermenigvuldigen

Stel je hebt een programmeertaal waarin je wel kunt optellen, maar nog niet kunt vermenigvuldigen. Hoe definieer je de vermenigvuldiging?

Dat kunnen we recursief doen: $n * 1 := n$ en $n * (m + 1) := n * m + n$.

Nu we kunnen vermenigvuldigen kunnen we ook machtsverheffen: $n^1 := n$ en $n^{(m+1)} := n^m * n$.

Hoeveel zetten?

Hoeveel zetten hebben we nu nodig met onze strategie voor de torens van Hanoi? Laat a_n het aantal zetten zijn dat we nodig hebben om de puzzel met n schijven op te lossen. Dus $a_1 = 1$ en $a_2 = 3$. Maar wat is a_5 ? Het is lastig om dit in een keer in te zien. Maar in ieder geval weten we dat we eerst het probleem met vier schijven moeten oplossen, dan de grote schijf verschuiven en weer het probleem met vier schijven oplossen. Dus $a_5 = a_4 + 1 + a_4 = 2a_4 + 1$.

Net zo, $a_4 = 2a_3 + 1$ en $a_3 = 2a_2 + 1$. Maar a_2 kennen we! Dus $a_3 = 7$, $a_4 = 15$ en $a_5 = 31$. Op dezelfde manier kunnen we iedere a_n uitrekenen.

Als we nog eens naar de rij van oplossingen 1,3,7,15,31,... kijken valt je misschien op dat de rij lijkt op de rij 2,4,8,16,32,... De rij van de machten van twee. De eerste rij is steeds eentje minder. Is dit toeval of klopt het altijd? Stel even dat $a_{37} = 2^{37} - 1$, dan is

$$a_{38} = 2a_{37} + 1 = 2 \cdot (2^{37} - 1) + 1 = 2^{38} - 2 + 1 = 2^{38} - 1.$$

Dus als het voor het 37ste element klopt, dan ook voor het 38ste. Nu is 37 natuurlijk helemaal niet bijzonder. Dus op de zelfde manier zien we dat als het voor n klopt dan ook voor $n + 1$. Verder wisten we al dat $a_1 = 2^1 - 1$. Dus het klopt voor 2, dus het klopt voor 3, dus het klopt voor 4, dus ..., dus het klopt voor 37, dus het klopt voor 38, dus ... We zien dus dat voor alle getallen n geldt: $a_n = 2^n - 1$.

Inductie

De bewijsmethode die we net hebben gebruikt heet *inductie*. Inductie lijkt heel veel op de recursieve methode die we gebruiken om functies te definiëren.

Inductie kunnen we gebruiken als we voor alle natuurlijke getallen (de getallen 1,2,3,4,5,6,7,...) een bepaalde uitspraak $P(n)$ willen bewijzen. Een bewijs met inductie gaat op de volgende manier:

1. Bewijs eerst $P(1)$.
2. Bewijs dan: Als $P(n)$, dan geldt ook $P(n + 1)$.

Dat is genoeg.

Als we nu willen laten zien dat bijvoorbeeld $P(37)$ geldt, dan weten we dat $P(1)$ waar is, dus (met 2.) ook $P(2)$, dus (weer met 2.) ook $P(3)$, dus ook $P(4)$,..., dus ook $P(36)$, dus ook $P(37)$!

In het vorige voorbeeld was $P(n)$ de uitspraak $a_n = 2^n - 1$.

Inductie kun je zien als het omgekeerde van recursie. Met inductie ligt de nadruk op steeds moeilijkere gevallen, bij recursie is dat omgekeerd.

Optellen

Hoeveel is $1 + 2 + 3 + \dots + 99$? Dat kun je natuurlijk op papier doen. Je kunt je rekenmachine gebruiken. In beide gevallen ben je lang bezig. Tenzij je het slim doet. Je kunt bijvoorbeeld definiëren $s(n) := 1 + 2 + \dots + n$. Dan kun je een recursief programmatje schrijven wat $s(99)$ voor je uitrekent, want $s(n + 1) = s(n) + n$.

Als je wat waarden uitrekent krijg je het vermoeden dat $s(n) = (n^2 + n)/2$.

Klopt dit nu ook? We gaan het bewijzen met inductie. Neem voor $P(n)$ de uitspraak $s(n) = (n^2 + n)/2$.

1. $P(1) : s(1) = 1 = (1^2 + 1)/2$.
2. Stel dat $P(n)$, d.w.z. $s(n) = (n^2 + n)/2$. Dan

$$s(n + 1) = s(n) + n + 1 = \frac{n^2 + n}{2} + n + 1 = \frac{n^2 + n + 2n + 2}{2} = \frac{(n + 1)^2 + (n + 1)}{2}.$$

Dus $P(n+1)$ geldt.

Dus geldt voor alle getallen n , $s(n) = (n^2 + n)/2$.

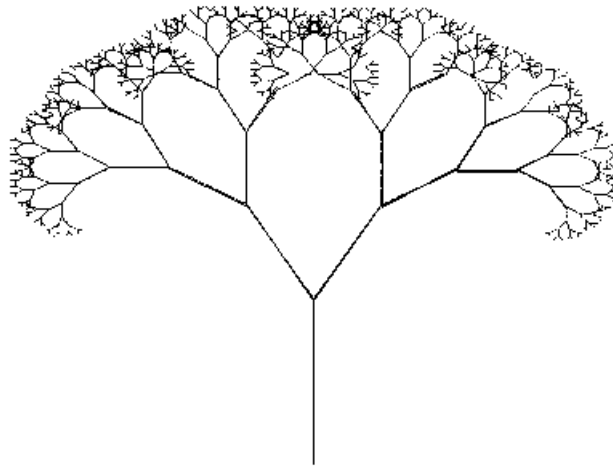
Rangschikkingen

Op hoeveel manieren kunnen we de cijfers 1,2,3,4,5,6,7,8,9 rangschikken? Het is lastig alle mogelijkheden op te schrijven. Maar gelukkig hoeft dat niet. Laat a_n het aantal rangschikkingen zijn van een verzameling van n elementen. We willen dus a_9 weten. Gelukkig weten we dat $a_1 = 1$. Verder is $a_{n+1} = (n+1)a_n$. Want we kiezen eerst een van de $(n+1)$ elementen, dat zetten we vooraan in de rij. Daarna hebben we nog n mogelijkheden.

Hoe rekenen we a_9 nu uit? De definitie die we boven hebben gegeven is precies de definitie van de faculteit-functie, die wordt genoteerd met $n!$. Dus $a_9 = 9!$ en algemeen $a_n = n!$. Deze functie zit standaard op de meeste rekenmachines.

Binaire bomen

Beschouw het volgende plaatje:



Hoewel deze boom er misschien complex uit ziet, is deze getekend met een simpele recursieve procedure. Het basisinzicht hierbij is dat een boom bestaat uit een stam met daarboven twee andere bomen, links en rechts, die net iets kleiner zijn (en iets gedraaid zijn).

Een recursief recept (functie, procedure) is eenvoudig te geven:

1. teken een stam
2. teken de linker (sub)boom
3. teken de rechter (sub)boom

Dit recept kunnen we preciezer maken met de volgende functie/procedure:

$$f(n, x, y, \alpha, l) = \begin{cases} \text{doe niets} & \text{als } n = 0 \\ \begin{array}{l} 1) \text{ teken stam (positie } x, y; \text{ hoek } \alpha; \text{ lengte } l) \\ 2) f(n-1, x_1, y_1, \alpha-30, l/2) \\ 3) f(n-1, x_2, y_2, \alpha+30, l/2) \end{array} & \text{als } n > 0 \end{cases}$$

Hierbij is n de hoogte van de boom; x en y de positie van de boom (startpunt van de stam); α is de hoek van de boom en l is de lengte van de stam.

De hoogte van de boom, n , is belangrijk in deze recursieve procedure; de andere variabelen laten we even buiten beschouwing.

Het tekenen van een boom met hoogte n kan dus terug gebracht worden tot een simpeler probleem: het tekenen van twee bomen met hoogte $n-1$. Dit recursieve patroon is duidelijk te herkennen wanneer je de stappen volgt van een computer volgt die de procedure uitvoert:

```
[1] teken boom: f(3)
[1.1] teken stam
[1.2] teken linker boom: f(2)
[1.2.1] teken stam
[1.2.2] teken linker boom: f(1)
[1.2.2.1] teken stam
[1.2.2.2] teken linker boom: f(0)
[1.2.2.2.1] doe niets
[1.2.2.3] teken rechter boom: f(0)
[1.2.2.3.1] doe niets
[1.2.3] teken rechter boom: f(1)
[1.2.3.1] teken stam
[1.2.3.2] teken linker boom: f(0)
[1.2.3.2.1] doe niets
[1.2.3.3] teken rechter boom: f(0)
[1.2.3.3.1] doe niets
[1.3] teken rechter boom: f(2)
[1.3.1] teken stam
[1.3.2] teken linker boom: f(1)
[1.3.2.1] teken stam
[1.3.2.2] teken linker boom: f(0)
[1.3.2.2.1] doe niets
[1.3.2.3] teken rechter boom: f(0)
[1.3.2.3.1] doe niets
[1.3.3] teken rechter boom: f(1)
[1.3.3.1] teken stam
[1.3.3.2] teken linker boom: f(0)
[1.3.3.2.1] doe niets
[1.3.3.3] teken rechter boom: f(0)
[1.3.3.3.1] doe niets
```

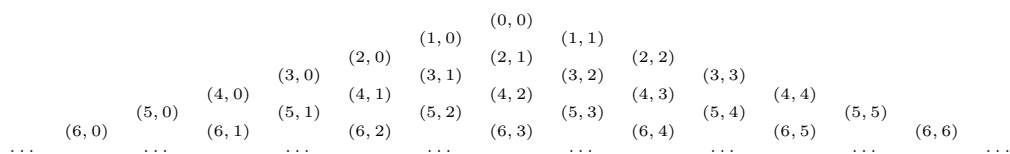
Merk op dat een recursieve *definitie* vaak erg eenvoudig is, maar de *uitvoering* van een recursieve functie veel werk met zich mee brengt: er zijn veel stappen, en je moet een goede

administratie bijhouden om steeds te weten met welk subprobleem je bezig bent. Typisch werk voor een computer dus.

Driehoek van Pascal

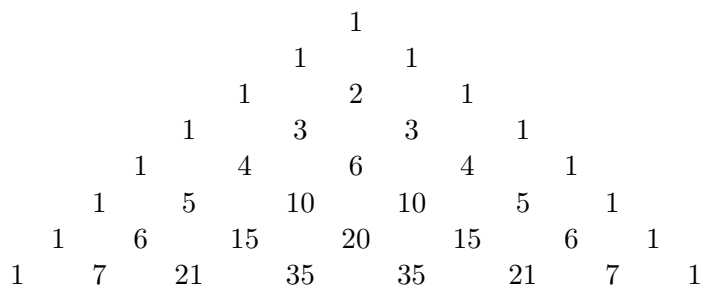
Binomiaalcoëfficiënten. We definiëren, voor natuurlijke getallen n en k met $k \leq n$, $\binom{n}{k}$ als het aantal manieren om k objecten uit een verzameling van n elementen te pakken.

We bekijken nu de roosterpunten (n, k) waarvoor $k \leq n$. Deze verzameling van roosterpunten tekenen we zó, dat het punt $(0, 0)$ de top wordt:



We gaan nu ieder van deze roosterpunten voorzien van een getal:

Eerste Driehoek van Pascal. Zet aan de rand eentjes, en vul de rest in door ‘schuin optellen’. Anders gezegd: voorzie de roosterpunten $(n, 0)$ en (n, n) van het getal 1, en schrijf bij elk ander roosterpunt (n, k) de som van de getallen, die je bij de roosterpunten $(n - 1, k)$ en $(n - 1, k - 1)$ moet schrijven. De Eerste Driehoek van Pascal (1e Δ vP) is dus het volgende schema van getallen (het gaat naar onder oneindig ver door, ik heb alleen maar een klein beginstuk opgeschreven):



Tweede Driehoek van Pascal. Zet op plaats (n, k) het getal $\binom{n}{k}$

Opgave: bewijs dat er in 2e Δ vP aan de rand allemaal eentjes staan.

We beweren dat als $k < n$, dan $\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$. Kiezen we namelijk een deelverzameling A van k elementen uit $\{1, 2, 3, 4, \dots, n+1\}$ dan zijn er twee mogelijkheden $n+1 \in A$ of $n+1 \notin A$. In het eerste geval is $A - \{n+1\}$ dus een deelverzameling van k elementen uit $\{1, 2, 3, 4, \dots, n\}$, in het tweede geval is A een deelverzameling van $k+1$ elementen uit $\{1, 2, 3, 4, \dots, n\}$. Dus ook in de 2e Δ vP het principe van ‘schuin optellen’ geldt: ieder inwendig getal is de som van de schuin erboven staande getallen.

Derde Driehoek van Pascal. Zet op plaats (n, k) het aantal wegen via roosterpunten van $(0, 0)$ naar (n, k) , waarbij we iedere keer een stap naar links-onder of rechts-onder zetten.

Je kunt nu inzien:

$$\begin{cases} \text{in de 3e } \triangle \text{ vP staan aan de rand allemaal eentjes} \\ \text{in de 3e } \triangle \text{ vP geldt ook het principe van 'schuin optellen'} \end{cases}$$

Gevolg: de 3e $\triangle$ vP is precies gelijk aan de 1e $\triangle$ vP

Vierde Driehoek van Pascal. Zet op plaats (n, k) de coëfficiënt van x^k in de veelterm $(1+x)^n$.

Ga zelf na, dat ook deze 4e $\triangle$ vP weer precies gelijk is aan de 1e $\triangle$ vP. De vier behandelde versies van de $\triangle$ vP leiden dus allemaal tot hetzelfde schema van getallen

Binomium van Newton.

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{n}x^n$$

Dit Binomium van Newton is niets anders dan de stelling: 2e $\triangle$ vP = 4e $\triangle$ vP.

Het meer algemene geval

$$(y+z)^n = \binom{n}{0}y^n + \binom{n}{1}y^{n-1}z + \binom{n}{2}y^{n-2}z^2 + \cdots + \binom{n}{n}z^n$$

volgt hier uit: $(y+z)^n = y^n(1+(\frac{z}{y})^n)$ als $y \neq 0$. Het geval $y = 0$ is eenvoudig.

Som- en productregel.

Somregel Als een eerste taak op m manieren kan worden uitgevoerd, en een tweede taak op n manieren, en deze taken kunnen niet tegelijk worden uitgevoerd, dan zijn er $m+n$ manieren om precies één van deze taken uit te voeren.

Productregel Als een eerste taak op m manieren kan worden uitgevoerd, en een tweede taak op n manieren, dan zijn er $m \cdot n$ manieren om eerst de eerste taak en vervolgens de tweede taak uit te voeren.

Opgaven

1. De uitvinder van het schaakbord mocht van de koning van Pezië een beloning uitzoeken. Hij mocht vragen wat hij maar wilde. Hij koos voor de volgende beloning. Hij wilde op het eerste vakje van het schaakbord 1 rijstkorrel hebben, op de tweede 2, op de derde 4, etc. Dus op ieder vakje twee keer zoveel rijstkorrels. De koning vond hem erg bescheiden.

Laten we eens kijken hoeveel hij vroeg: hij wilde dus

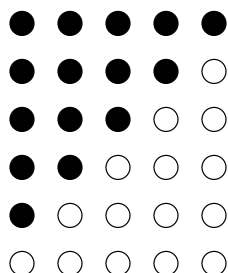
$$1 + 2 + 2^2 + 2^3 + 2^4 + \cdots + 2^{63}$$

korrels hebben. Kun je een eenvoudige formule vinden voor de uitkomst van deze som? [Hint: schrijf eerst eens wat termen van de rij 1, 1+2, 1+2+2², ... op. Herken je deze rij? Wat is je vermoeden van de uitkomst? Kun je dit met inductie bewijzen?]

2. [Deze opgave past niet echt in dit hoofdstuk.] Neem aan dat een rijstkorrel breedte en dikte van 1mm heeft en een lengte van 5mm. Spreid alle rijstkorrels van vraag 1 gelijkmatig uit over Nederland ($400\text{km} \times 200\text{km}$). Hoe hoog wordt deze berg?
3. Bewijs met inductie dat voor alle n , $2^n \geq n$.
4. We hebben al een mooie formule voor de som van de rij

$$1 + 2 + 3 + 4 + 5 + \cdots + n = \frac{(n+1)n}{2}.$$

Wat is het verband met het volgende plaatje?



5. Ik gooi een rode en een blauwe dobbelsteen. In hoeveel van alle mogelijke uitkomsten gooi ik met de rode een even aantal ogen en met de blauwe een oneven aantal ogen?
6. Bekijk de volgende taal

axioma	λ
Regels	$x \Rightarrow ax$
	$x \Rightarrow bx$

Teken de boom die bij deze taal hoort.

Op hoeveel manieren kan ik een woord met 3 a 's en 5 b 's maken?

7. **Rijtes van lengte n .** Hoeveel rijtes kunnen we maken van lengte n , met daarin alleen de getallen 1 tot en met 5? Dit kunnen we ook weer recursief oplossen. Noem het aantal rijtes van lengte n a_n . Een rijte van lengte 1 kunnen we op 5 manieren maken, dus $a_1 = 5$. Een rijte van lengte $n+1$ kunnen we maken door eerst een rijte van lengte n te maken en daar een element achter te zetten. Dus $a_{n+1} := 5a_n$. Herinner je nu even de definitie van machtsverheffen. Bewijs met inductie dat voor alle n , $a_n = 5^n$.
8. Bewijs met inductie dat $1 + 3 + 5 + 7 + \cdots + (2n-1) = n^2$.
Wat is het verband met het volgende plaatje?

