

**Alkanen**  
**en**  
**Polymino's**  
combinatorisch bekeken

Victor Pessers en Paul Witteveen  
6F

wiskunde, scheikunde en informatica  
januari 2004

## Inhoudsopgave

	pag.
Inleiding	2-3
Onderzoek naar Alkanen	4-13
Onderzoek naar Polymino's	14-21
Conclusie	22
Evaluatie	23
Literatuurlijst	24
Bijlagen	25 e.v.

## Inleiding

In de vierde klas zijn wij geïnteresseerd geraakt in - zoals we het nu noemen - polymino's. We speelden het computerspelletje Tetris en bedachten: Met vier vierkantjes kun je zeven figuurtjes maken, door die vierkantjes verschillend te rangschikken. Door puzzels uit het tijdschrift *Breïnbrekers* wisten we dat je met vijf vierkantjes twaalf figuurtjes kon maken. Wij begonnen ons af te vragen: Hoeveel figuurtjes kun je maken met zes vierkantjes? Of met zeven? We zijn toen begonnen met het uittekenen van de figuurtjes met zes vierkantjes. Als snel bleek dat dit werkje erg omslachtig was, mede door het algoritme dat we er op nahielden. Na een tijdje hebben we het probleem laten rusten, maar toen we in de vijfde klas over het profielwerkstuk hoorden, bedachten we dat we dit probleem konden gaan bekijken. Het duurde lang voordat we eindelijk de definitieve keuze hadden gemaakt, mede omdat we moeilijk een tweede vak erbij konden vinden (het eerste was duidelijk wiskunde). Toen kwamen we op het idee om het vak scheikunde erbij te betrekken. In de chemie heb je namelijk koolwaterstoffen die er tweedimensionaal enigszins uitzien als tetrisfiguren. Bij deze stoffen konden wij ons ook afvragen: Hoeveel isomeren kun je maken met 5 C-atomen? Of met 9 C-atomen? Vanuit dit oogpunt hebben we hoofd- en deelvragen opgesteld, met betrekking tot het onderzoek. De vraag die we onszelf stelden was:

**Is er een regelmaat te vinden in de reeks van isomeren, dan wel in de reeks van polymino's, bij een bepaald aantal elementen?**

Hierin zijn de elementen de C-atomen bij de alkanen enerzijds en de vierkantjes bij de polymino's anderzijds. We hadden besloten dat we met een zelfgeschreven computerprogramma het aantal isomeren en polymino's wilden bepalen. Daarmee hebben we de volgende deelvragen en deelstellingen opgesteld:

**We bepalen het aantal isomeren voor een aantal C-atomen met een computerprogramma.**

We hielden daarbij in gedachte dat het voor non-cyclische alkanen zeker mogelijk zou zijn dat aantal te bepalen. De onverzadigde en cyclische koolstofketens zouden we buiten beschouwing laten, vanwege de (te ingewikkelde) extra mogelijkheden.

## **Hoe beschrijven we de reeks van isomeren op een wiskundige manier?**

Half overtuigd van de mogelijkheid dit wiskundig te kunnen beschrijven merkten we op dat het voor non-cyclische alkanen misschien wel mogelijk was.

## **We bepalen het aantal polymino's voor een aantal vierkantjes met een computerprogramma.**

Hierbij dachten we dat het moest lukken, als we toestaan dat gedraaide en gespiegelde figuren nieuwe polymino's opleveren. Als we dit niet zouden toestaan, zou het in ieder geval een stuk moeilijker worden.

## **Hoe beschrijven we de reeks van polymino's op een wiskundige manier?**

Wederom maar half overtuigd van de mogelijkheid dit wiskundig te kunnen beschrijven merkten we op dat het misschien wel mogelijk was, als we draaiing en spiegeling toestaan.

## **Is er reeds onderzoek gedaan naar dergelijke problemen?**

Om ons werkstuk extra kracht bij te zetten, zoeken we naar secundaire literatuur over het onderwerp. Waarschijnlijk is er al over geschreven, maar dat is niet helemaal zeker. Ook zullen er al vast programma's en reeksen gemaakt zijn.

## Onderzoek naar Alkanen

In dit deel van ons profielwerkstuk richten we ons op de alkanen. Centraal staat hierin de vraag: 'Hoeveel verschillende alkanen (bestaande uit louter koolstof- en waterstofatomen) zijn er mogelijk, bij een gegeven aantal koolstofatomen?'. We hebben ons binnen deze vraag echter toegespitst op de vraag hoeveel alkanen er zijn, als we er geen cyclische koolstofketens erbij betrekken. Dit zou de te beantwoorden vraag namelijk onnodig moeilijk kunnen maken. Wat we dus in wezen uitrekenen, is het aantal alkanen van de vorm  $C_nH_{2n+2}$ , voor elk mogelijk aantal koolstof-atomen, waaruit deze koolwaterstof is opgebouwd. Verder hebben we ook niet gelet op stereo-isomerie (we houden er geen rekening mee, dat bepaalde moleculen ook een zeker 'spiegelbeeld' van zichzelf kunnen hebben: we beschouwen dit als dezelfde alkaan). Ook dit zou het probleem immers veel te complex kunnen maken. Het grootste gedeelte van dit hoofdstuk gaat over het algoritme waarop het programma gebaseerd is, om de verschillende alkanen uit te rekenen. In de programmatuur zelf staan opmerkingen, die verband houden, met wat er in dit hoofdstuk wordt verteld. Het is niet nodig het hele programma precies te begrijpen, maar het is belangrijk dat de globale structuur ervan duidelijk is. Daarna volgt nog een ander gedeelte, dat ook eveneens op eigen onderzoek gebaseerd is, en dat ook veel met de onderzoeksvraag te maken heeft; daarin zal een manier besproken worden om op een wiskundige manier het exacte aantal alkanen te bepalen, met een  $n$ -aantal koolstofatomen in de langste keten (de vraag is dan dus niet hoeveel  $C_4H_{10}$ -alkanen zijn er, maar hoeveel ...-butanen zijn er). Deze vraag bleek overigens zelfs makkelijker te beantwoorden, dan onze oorspronkelijke onderzoeksvraag.

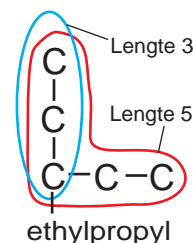
### Alkanen tellen naar het aantal koolstofatomen:

Elk alkaan heeft of een even, of een oneven aantal C-atomen in de zijn langste keten. In het geval dat het een even aantal C-atomen heeft, kunnen we dit alkaan opdelen in twee kleinere alkanen, door het over het midden van het alkaan 'door te breken'. We krijgen dan als het ware twee alkylgroepen. Alkanen met een oneven aantal C-atomen in de hoofdreeks, kunnen we opdelen in een centraal C-atoom, omgeven door 4 alkylgroepen (waarbij we gemakshalve een 'alkylgroep' die uit alleen een waterstof-atoom bestaat, ook als een alkylgroep beschouwen, nu echter opgebouwd uit nul C-atomen. In het programma kun je er op deze manier namelijk handiger mee rekenen.) Nu is na veel gepruts gebleken, dat het eigenlijk veel makkelijker is om nieuwe alkylgroepen uit voorgaande alkylgroepen te berekenen, dan alkanen uit voorgaande alkanen.

De alkanen kunnen dan uit de gegevens over de alkylgroepen weer terug berekend worden op de volgende manier: het aantal alkanen met in de hoofdreeks een even aantal C-atomen (laten we zeggen lengte  $2n$ ) is te berekenen, door het aantal verschillende manieren te bepalen, waarop je twee alkylgroepen met lengte  $n$  kunt combineren. Het aantal alkanen met een oneven aantal C-atomen in de hoofdreeks (laten we zeggen lengte  $2n + 1$ ), kun je bepalen door het aantal mogelijkheden te bepalen, waarop je een centraal C-atoom kun omringen door 4 alkylgroepen, waarvan er minstens twee lengte  $n$  hebben (we stellen immers als eis, dat het totale alkaan een lengte heeft van  $2n+1$ ). In wezen komt dit dus

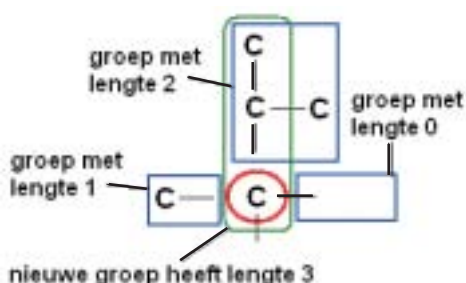
neer, op het aantal combinaties van 2 alkylgroepen met lengte  $n$ , en twee alkylgroepen met lengte kleiner of gelijk aan  $n$ . Erg belangrijk is het overigens, dat er geen combinaties dubbel worden geteld, en daar zal in het programma dan ook op gelet worden. Merk overigens op dat we met de lengte van een alkaan net iets anders bedoelen, dan met de lengte van een alkylgroep.

De lengte van een alkaan is het aantal C-atomen dat zich in de langste keten bevindt, maar de lengte van een alkylgroep is de lengte van de langst mogelijke keten in de alkylgroep, die begint bij het C-atoom waarmee de alkylgroep ook vast zit aan de rest van het alkaan (zo is de lengte van ethylpropyl gelijk aan 3, en niet gelijk aan 5, hoewel je de langste keten ook als 5 zou kunnen zien: zie figuur rechts).



Ook is het belangrijk, dat het volgende wordt opgemerkt: in het geval dat bij het samenstellen van een 'oneven alkaan' (een alkaan met een oneven lengte), twee van de vier alkylgroepen die het centrale C-atoom omringen lengte  $n$ , en twee van de vier alkylgroepen lengte 0 hebben, je dezelfde aantal mogelijkheden hebt, als wanneer je een even alkaan samenstelt uit twee alkylgroepen met lengte  $n$ : laat immers het centrale C-atoom weg, en je houdt een even alkaan over. Er geldt dus: het aantal even alkanen met lengte  $2n$ , is gelijk aan het aantal oneven alkanen met lengte  $2n+1$ , waarbij twee van de vier alkylgroepen die het centrale C-atoom omringen 0-alkylgroepen zijn.

De alkylgroepen op hun beurt kunnen op de volgende manier worden berekend uit vorige gevonden alkylgroepen:



Elke alkylgroep met lengte  $n+1$ , is samen te stellen uit een centraal C-atoom (dat is het atoom waarmee het aan de rest van het alkaan zou moeten zitten), omringd door 3 verschillende andere alkylgroepen, waarvan er minstens een lengte  $n$  heeft (de lengte van deze alkylgroep wordt daarmee  $n+1$ : zie de figuur links). Je zult vast opmerken, dat het vormen van een nieuwe alkylgroep erg lijkt

op het vormen van een oneven alkaan, en in sommige situaties komt het zelfs op precies hetzelfde neer (om precies te zijn, in die gevallen, waarin minstens een van de 4 alkylgroepen aan het centrale atoom, een alkylgroep met lengte 0 is). Dan is het resultaat zowel op te vatten als een alkaan en als een alkyl. Hier zal in het programma dan ook handig gebruik van worden gemaakt, alsook van het gegeven, dat de vorming van een oneven alkaan op hetzelfde kan neerkomen als het vormen van een even alkaan. Kort samengevat:

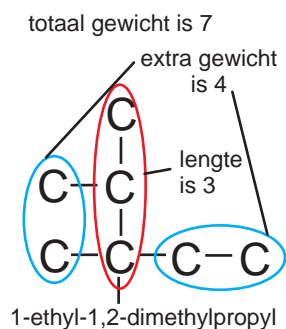
Omring een C-atoom met 4 alkylgroepen:

- wanneer twee van de vier alkylgroepen lengte  $n$  hebben, en de andere twee een lengte kleiner of gelijk aan  $n$ , dan is het op te vatten als een alkaan met lengte  $2n+1$ ;
- wanneer twee van de vier alkylgroepen lengte  $n$  hebben, en de andere twee lengte 0, denk dan het centrale C-atoom weg, en het is een alkaan met lengte  $2n$ ;

- wanneer een van de vier alkylgroepen lengte  $n$  heeft, de tweede en de derde alkylgroep lengte kleiner of gelijk aan  $n$ , en de vierde alkylgroep sowieso lengte 0 heeft, dan is het resultaat op te vatten als een alkylgroep met lengte  $n + 1$ ;

De opzet van het programma is eigenlijk, dat wanneer je alle mogelijke alkylgroepen kent tot en met lengte  $n$ , je zodoende met die gegevens alle alkanen met lengte  $2n$ , alle alkanen met lengte  $2n + 1$  en alle alkylgroepen van lengte  $n + 1$  kunt bepalen. Zodoende heb je nu dus ook alle gegevens van de alkylgroepen tot en met lengte  $n + 1$ , waarmee je vervolgens weer de alkanen met lengte  $2(n + 1) = 2n + 2$  kunt bepalen, de alkanen met lengte  $2(n + 1) + 1 = 2n + 3$  en de alkylgroepen met lengte  $n + 1 + 1 = n + 2$ . Door dit maar vaak genoeg te herhalen, en de verschillende berekende alkanen die we op ons pad zijn tegengekomen te 'turven' en ondertussen dus ook telkens weer de nieuwe benodigde alkylgroepen te berekenen, kun je alle alkanen tot een gegeven waarde uitrekenen.

Voordat dit verder toegelicht zal worden, worden er voor het gemak eerst nog twee nieuwe zelfbedachte termen gintroducteerd. Met het 'gewicht' van een alkaan of een alkylgroep wordt het aantal C-atomen dat het alkaan of alkylgroep telt bedoeld, en met het 'extra gewicht' bedoelen we het aantal C-atomen van een alkylgroep, buiten de C-atomen van de hoofdketen om. Het gewicht van bijvoorbeeld 1-ethyl-1,2-dimethylpropyl is dus 7, maar het extra gewicht hiervan is 4 ( $7-3$ ) (zie de figuur rechts). Als twee verschillende alkanen dus gelijk in lengte en gewicht zijn, dan zijn ze dus ook gelijk in extra gewicht.



In het begin weten we alleen dat er maar één manier is om een alkylgroep met gewicht 0 te vormen, en dat deze ook lengte 0 heeft (dat is namelijk alleen een H-atoom). We houden nu een tabel bij, waarin we alle zojuist gevonden aantallen alkylgroepen bijhouden. Deze tabel bestaat uit twee assen: horizontaal zetten we de verschillende mogelijke lengtes van de alkylgroep uit, beginnend bij 0 en eindigend bij voor zover we dat willen uitrekenen. Verticaal zetten we het extra gewicht van de alkylgroep uit, ook beginnend bij 0, en eindigend tot zover als dat nodig is. Logischerwijs is het totale gewicht van zo'n alkylgroep makkelijk te bepalen aan de hand van de plaats waar het zich in de tabel bevindt: dit is namelijk gelijk aan de lengte van de alkylgroep plus het extra gewicht ervan, oftewel gelijk aan de som van de 'x-coördinaat' en de 'y-coördinaat' van de bijbehorende cel uit de tabel. (Er is overigens voor gekozen om horizontaal de lengte en verticaal het extra gewicht uit te zetten, maar voor het programma zelf had het niets uitgemaakt als we dat andersom hadden gedaan.) Verder houden we ook nog een rij bij, waarin we bijhouden hoeveel alkanen er zijn, met een bepaald gewicht (het 'nulde' getal in de rij is het aantal alkanen met gewicht 0, het eerste getal in de rij is het aantal alkanen met gewicht 1, het tweede voor gewicht 2, etc. In het begin is deze hele rij nog gevuld met nullen, want we weten immers nog niet hoeveel er van elke alkaan er zijn.) Nu gaan we alle manieren bekijken waarop we een centraal C-atoom met 4 van de bekende alkylgroepen (nu nog alleen tot en met lengte 0), kunnen omringen. Dit kan duidelijk maar

op een manier (namelijk alle vier 0-alkylgroepen). We weten nu dat er maar één alkaan is met lengte 1, maar omdat twee van de vier alkylgroepen lengte 0 hebben (in feite hebben ze het allemaal) is het ook nog op te vatten als een alkaan met lengte 0. Bovendien is het ook nog eens op te vatten als een alkylgroep met lengte 1 (immers, er wordt voldaan aan de eis dat minstens een van de vier alkylgroepen lengte 0 heeft). We kunnen nu dus het nulde en het eerste getal van onze rij met 1 ophogen, naar 1, en we kunnen ook onze tabel met alkylgroepen eentje verder naar rechts uitbreiden. Deze keer maakte het overigens in de uitleg niet uit, dat we niet iets dubbel zouden tellen, omdat we toch alleen nog maar 0-alkylgroepen hadden. Maar nu zullen we toch enige volgorde aanhouden: We noemen onze alkylgroepen die we aan het centrale C-atoom zetten respectievelijk tak1, tak2, tak3 en tak4, en we spreken af dat er altijd geldt dat  $\text{lengte tak1} \geq \text{lengte tak2} \geq \text{lengte tak3} \geq \text{lengte tak4}$ , en indien er twee takken, laat zeggen taka en takb, met  $a < b$ , dat dan ook nog geldt dat 'extra gewicht taka'  $\geq$  'extra gewicht takb'. Dus als bijvoorbeeld tak2 en tak3 beide lengte 3 hebben, dan kan tak2 bijvoorbeeld als extra gewicht 3 hebben en tak1 als extra gewicht 1, maar niet ook nog andersom, want we willen immers niets dubbel gaan tellen. (In het programma hebben we overigens voor de lengte van tak1 de variabele tak1x genomen, en voor het extra gewicht van tak1, tak1y. En zo hebben we dat ook voor de andere takken gedaan.) Nu gaan we kijken hoe we nu weer een C-atoom kunnen omringen met alkylgroepen. We beginnen nu voor tak1 met lengte 1 (als we als lengte van tak1 0 zouden kiezen, zouden we immers weer gaan berekenen wat we al de vorige keer hadden gedaan). We komen nu op de volgende mogelijkheden:

- tak1 is een methylgroep, en de rest allemaal 0-alkylgroepen  $\rightarrow$  resultaat is op te vatten als een ethylgroep
- tak1 en tak2 zijn methylgroepen, en de rest zijn 0-alkylgroepen  $\rightarrow$  resultaat is op te vatten als propaan, maar volgens de regel ook als ethaan en een methylethylgroep
- tak1, tak2 en tak3 zijn methylgroepen, en tak4 is een 0-alkylgroep  $\rightarrow$  resultaat is op te vatten als methylpropaan, maar ook als dimethylethyl
- alle takken zijn methylgroepen  $\rightarrow$  resultaat is op te vatten als dimethylpropaan



# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen

© havovwo.nl

	0	1	2	3	4	
0	1	1	1	1	1	...
1	0	0	1	2	3	...
2	0	0	1	4	8	...
3	0	0	0	4	15	...
4	0	0	0	5	27	...
5	0	0	0	4	43	...
6	0	0	0	3	67	...
7	0	0	0	2	97	...
8	0	0	0	2	136	...
9	0	0	0	1	183	...
10	0	0	0	1	239	...
11	0	0	0	0	300	...
	⋮	⋮	⋮	⋮	⋮	⋮

Met deze gegevens kunnen we onze tabel, en onze rij weer verder uitbreiden, en opnieuw bekijken hoe we een C-atoom met onze nieuwe gegevens kunnen omringen door alkyl-groepen. Dit herhalen we dan net zolang als we willen, en op een gegeven moment ziet de tabel waarin onze alkyl-groepen staan, eruit zoals hiernaast, en het begin van de lijst met alkanen is dan: 1, 1, 1, 1, 2, 3, 5, 9, 18, 35, 75, 159, .... Wat waarschijnlijk op zal vallen, is dat in de tabel op een gegeven moment ook bepaalde cellen gevuld zijn met getallen groter dan 1: zo zijn er twee mogelijkheden om een propylgroep te vormen met extra gewicht 1, namelijk 1-methylpropyl, of 2-methylpropyl. Vandaar dat we ook in het programma telkens bij elke mogelijke combinatie van alkylgroepen, het aantal mogelijkheden waarop ieder van de alkylgroepen kan worden samengesteld met elkaar vermenigvuldigen, om zo te bepalen hoeveel mogelijkheden er zijn om deze nieuwe alkaan of alkylgroep te maken. Iets formeler gezegd: het aantal mogelijkheden om 4 alkylgroepen met elkaar te combineren, is gelijk aan het product van de coëfficiënten uit de tabel die bij deze alkylgroepen horen. Nu is echter nog een probleem onbesproken gebleven. Namelijk als meerdere van de 4 omringende alkylgroepen zowel dezelfde lengte als hetzelfde extra gewicht hebben. Bijvoorbeeld in het geval dat zowel tak1 als tak2 een lengte van 3 en een extra gewicht van 7 heeft. In de tabel zie je dat er drie manieren zijn waarop zoiets kan worden samengesteld. (Zelf kun je nagaan dat deze mogelijkheden zijn: 1,1,2-trimethyl-1-(dimethylethyl)propyl, 2-methyl-1,1-di(methylethyl)propyl en 1-ethyl-2-dimethyl-1-(methylethyl)propyl). Nu zou je kunnen zeggen, wanneer je bijvoorbeeld een alkaan met lengte 6 (en gewicht  $6 + 2 \cdot 7 = 20$ ) uit deze twee propylgroepen wilt maken, dat dit dan op  $3 \cdot 3 = 9$  manieren kan. Nu tellen we echter wel bepaalde combinaties dubbel:

- mogelijkheid 1 met mogelijkheid 1
- mogelijkheid 2 met mogelijkheid 1
- mogelijkheid 2 met mogelijkheid 2
- mogelijkheid 3 met mogelijkheid 1
- mogelijkheid 3 met mogelijkheid 2
- mogelijkheid 3 met mogelijkheid 3
  
- mogelijkheid 1 met mogelijkheid 2
- mogelijkheid 1 met mogelijkheid 3
- mogelijkheid 2 met mogelijkheid 3

Je ziet nu duidelijk dat we de onderste drie combinaties eigenlijk al (maar dan in een andere volgorde) gehad hebben. Om dit probleem op te lossen hebben we een beetje wiskunde moeten doen, en uitgelegd zal worden hoe het uiteindelijk is opgelost. Als je combinaties van 2 wilt maken, met  $n$  verschillende mogelijkheden (in het voorbeeld van net was dat 3), dan is het het handigst om deze mogelijkheden als het ware denkbeeldig te nummeren. Als je dan geen combinaties dubbel wilt tellen, dan tel je zoals we in het voorbeeld hadden gedaan, en wat de volgende tabel nog eens illustreert:

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen

© havovwo.nl

	1	2	3
1	X	X	X
2	X	X	
3	X		

Het totaal aantal mogelijkheden wordt daarmee dus  $1 + 2 + 3$ , en in het algemeen wordt het dus wanneer we te kiezen hebben uit een  $n$  aantal mogelijkheden:  $1 + 2 + \dots + n$ . Met een beetje kennis uit de wiskundeles, weet je dat dit uit te drukken valt met de formule  $\frac{n(n+1)}{2}$ . In dit geval moeten

we het aantal mogelijkheden niet als  $3 \cdot 3 = 9$ , maar als  $\frac{3 \cdot 4}{2} = 6$  uitrekenen. Omdat we deze formule nog vaker moeten gebruiken, noemen we deze functie zolang  $f_2(n)$ , waarbij de lage 2 achter de  $f$  staat, omdat het nu eenmaal om combinaties van 2 getallen, die kleiner of gelijk aan  $n$  zijn, gaat. Zodoende geldt dus:  $f_2(n) = 1 + 2 + \dots + n$ . Voor de waarden 1 tot en met 4 kunnen we dus uitrekenen dat  $f_2(1) = 1$ ,  $f_2(2) = 3$ ,  $f_2(3) = 6$  en  $f_2(4) = 10$ . Het kan echter ook zo zijn, dat niet slechts 2 van de omringende alkylgroepen gelijke lengte en gewicht hebben, maar bijvoorbeeld 3 of 4. Omdat je in deze gevallen ook nog het aantal mogelijkheden wilt kunnen uitrekenen, zouden we ook erachter willen komen of we hiervan de bijbehorende functies  $f_3(n)$  en  $f_4(n)$  kunnen bepalen. Als bijvoorbeeld 3 alkylgroepen gelijk in lengte en gewicht zijn, en van deze zijtakken zijn er ook weer  $n$  verschillende soorten, dan kunnen we zonder combinaties dubbel te tellen, het als volgt in een tabel weergeven:

1,1,1	2,1,1	3,1,1	4,1,1	...
	2,2,1	3,2,1	4,2,1	...
	2,2,2	3,2,2	4,2,2	...
		3,3,1	4,3,1	...
		3,3,2	4,3,2	...
		3,3,3	4,3,3	...
			4,4,1	...
			4,4,2	...
			4,4,3	...
			4,4,4	...
				...

In de veronderstelling dat het voor deze aantallen ook nog uit te drukken zou zijn in een veelterm van  $n$ , hebben we een tabel met combinaties van 3 gemaakt, voor waarden van  $n$  tot en met 4. Je ziet dat het totaal aantal mogelijkheden om combinaties van 3 te maken, met getallen tot en met  $n = 1$  ook gelijk is aan 1; voor  $n = 2$  is dit  $1 + 3 = 4$ , voor  $n = 3$  is dit  $1 + 3 + 6 = 10$  en voor  $n = 4$  is dit gelijk aan  $1 + 3 + 6 + 10 = 20$ . Je kunt aan de tabel zien, dat het aantal mogelijkheden,

waarop je combinaties met 3 getallen oplopend tot  $k$  kunt maken, gelijk is aan de som van alle  $f_2(n)$  met  $n$  oplopend van 1 tot en met  $k$ . Met andere woorden  $f_3(n) = f_2(1) + f_2(2) + \dots + f_2(n)$ . Dit is ook logisch, want als je combinaties van 3 met getallen tot en met bijvoorbeeld 4 wil maken, en je noteert elke mogelijkheid op de manier zoals dat in de tabel te zien is, waarbij je met het grootste getal begint en met het kleinste eindigt, dan kun je bijvoorbeeld met een 1 beginnen, maar dan moeten de getallen erachter ook allemaal kleiner of gelijk aan 1 zijn (dat kan dus op  $f_2(1)$  manieren). Vervolgens kun je met een 2 beginnen, en dat kun je op zijn beurt weer met  $f_2(2)$  vervolgen, en zo ga je verder. Analooq aan deze uitleg, kun je op eenzelfde manier beredeneren dat ook moet gelden  $f_4(n) = f_3(1) + f_3(2) + \dots + f_3(n)$ , en dus in het algemeen  $f_{m+1}(n) = f_m(1) + f_m(2) + \dots + f_m(n)$ . Merk overigens op, dat dit ook voor  $f_2(n)$  geldt:  $f_1(n)$  moet immers gelijk zijn aan  $n$ , want als je maar 1 getal hebt, kleiner of gelijk aan  $n$ , dan heb je dus ook maar  $n$  mogelijkheden, en we wisten al dat  $f_2(n) = 1 + 2 + \dots + n$  (Je kunt zelf nu nagaan dat ook moet gelden dat  $f_0(n) = 1$  voor alle  $n$ ). Met deze kennis, en ook nog wat vaardigheden op wiskundig gebied, hebben we weten te bewijzen dat in het algemeen moet gelden:

$$f_m(n) = \frac{(m+n-1)!}{m!(n-1)!}$$

deze formule voldoet immers aan de volgende karakteristieke eigenschap:

$$\begin{aligned} f_{m+1}(n) - f_{m+1}(n-1) &= (f_m(1) + f_m(2) + \dots + f_m(n)) - (f_m(1) + f_m(2) + \dots + f_m(n-1)) \\ &= f_m(n) \end{aligned}$$

immers:

$$\begin{aligned} f_{m+1}(n) - f_{m+1}(n-1) &= \frac{(m+n)!}{(m+1)!(n-1)!} - \frac{(m+n-1)!}{(m+1)!(n-2)!} = \\ &= ((m+n) - (n-1)) \cdot \frac{(m+n-1)!}{(m+1)!(n-1)!} = \\ &= (m+1) \cdot \frac{(m+n-1)!}{(m+1)!(n-1)!} = \frac{(m+n-1)!}{m!(n-1)!} = f_m(n) \end{aligned}$$

Nu we dit weten, kunnen we ook heel gemakkelijk  $f_3(n)$  en  $f_4(n)$  uitrekenen:

$$f_3(n) = \frac{(3+n-1)!}{3! \cdot (n-1)!} = \frac{n(n+1)(n+2)}{6}$$

en

$$f_4(n) = \frac{(4+n-1)!}{4! \cdot (n-1)!} = \frac{n(n+1)(n+2)(n+3)}{24}$$

De functies  $f_5(n)$  en verder hebben we overigens niet nodig, omdat er eenvoudigweg niet meer dan 4 alkylgroepen aan een C-atoom vast kunnen zitten. Met deze zojuist gevonden formules, kunnen we weer verder, want ook in gevallen dat er alkylgroepen met gelijk gewicht en lengte gecombineerd moeten worden, kunnen we het aantal mogelijkheden uitrekenen. Dit doen we als volgt: We delen eerst de verschillende zijtakken in groepen met gelijke lengte en gewicht in. Vervolgens bepalen we van elke groep het aantal mogelijke combinaties binnen deze groep. Het totaal aantal mogelijke combinaties van deze groepen, is dus het product van het aantal mogelijkheden van elke groep op zich. Het aantal mogelijkheden binnen zo'n groep kunnen we ook al uitrekenen. Stel dat we een groep hebben, waarin alle alkylgroepen een lengte van  $x$  en een extra gewicht van  $y$ , dan kunnen we in de tabel voor zover we die tot dan toe berekend hebben, het bijbehorende aantal mogelijkheden om deze te vormen opzoeken (laten we zeggen dat dit  $k$  is), en vervolgens is dan het totaal aantal mogelijkheden gelijk aan  $f_m(k)$ , waarbij  $m$  het aantal takken is, die in deze groep zitten. Voorbeeld: Je wil het aantal mogelijkheden uitrekenen, waarop je een C-atoom kunt omringen met 4 alkylgroepen, waarvan er 3 een lengte van 3 en extra gewicht van 3 hebben, en de vierde tak een lengte van 2 en een extra gewicht van 1 heeft. Nu kunnen we al uit de tabel aflezen, dat het aantal alkylgroepen met een lengte van 3 en een extra gewicht van 3, op 4 verschillende manieren te vormen is, en ook is het duidelijk dat een alkylgroep met lengte 2 en extra gewicht 1 op

slecht een manier kan. Het totaal aantal mogelijkheden om het C-atoom dus te omringen, is:

$$f_3(4) \cdot f_1(1) = \frac{4 \cdot 5 \cdot 6}{6} \cdot 1 = 20$$

Deze wijze van berekenen is ook verderop in de programmatuur terug te vinden (al is het daar natuurlijk verpakt in specifieke programmeertaal).

Wanneer we dit programma uiteindelijk nu zijn werk laten doen (daar doet hij ongeveer 9 seconden over), op de manier zoals het beschreven staat, komen we tot de rij zoals die in de bijlage te zien is. Al konden we de tabel met pen en papier verifiëren tot en met  $n = 8$ , veel zekerheid over de betrouwbaarheid van de rest van de rij hadden we niet. Gelukkig hebben we naderhand op internet een website gevonden, die ook vermelding maakt van deze rij<sup>1</sup>. Hier bleek overigens, dat er kennelijk al meer onderzoek hiernaar is verricht, zij het met veel geavanceerdere technieken. Aanvankelijk bleken onze resultaten niet helemaal met de werkelijke waarden te kloppen. Dit kwam omdat de variabelen waarmee het programma werkt (integers) maar een beperkt bereik hadden ( $2^{31} - 1$  om precies te zijn). Ook nadat we vervolgens de variabelen waarmee de tabel werkt, van een ander type zijn gaan declareren (namelijk `int64`), was er nog steeds een verschil. Stom genoeg, kwamen we pas een lange tijd daarna erachter dat het aan de werkvariabelen lag, die we nog niet als het type `int64` hadden gedeclareerd. Toen dit veranderd was, klopten alle waarden precies, en waren we zelfs instaat de rij uit te breiden met nog grotere waarden dan alle waarden die op de site stonden. Deze rij met getallen kan dus net als het programma zelf, teruggevonden worden in de bijlage. We hebben zoals je kunt zien, de waarden tot en met 60 uitgerekend, want het had geen zin om nog verder te gaan: vanaf het 54<sup>e</sup> getal (het 55<sup>e</sup> als je de 0 meetelt) wordt de uitgerekende waarde negatief, en dat betekent dat ook nu een van de variabelen waarmee het programma rekent alsnog 'over de limiet' moet zijn. Zouden we de waarden voor een nog grotere  $n$  willen bepalen, dan zouden we variabelen moeten gebruiken, die ook een nog grotere capaciteit hebben.

## Alkanen tellen naar hun lengte:

Zojuist is uitgebreid uitgelegd, hoe we kunnen berekenen hoeveel verschillende isomeren er zijn van alkanen met een zeker gewicht. Het is ons echter ook mogelijk gebleken een methode te vinden, om het aantal verschillende mogelijkheden uit te rekenen, waarop je een alkaan kunt vormen met een bepaalde lengte, oftewel met een bepaald aantal C-atomen in de hoofdketen. Zo zijn er bijvoorbeeld 3 verschillende alkanen met een lengte van 3 (namelijk propaan, methylpropaan en dimethylpropaan) en het aantal alkanen met een lengte van 4 (het aantal x-butanen) is 6 (namelijk butaan, methylbutaan, 2,2-dimethylbutaan, 2,3-dimethylbutaan, trimethylbutaan en tetramethylbutaan). Om dit aantal nu te bepalen voor verschillende waarden, is zelfs veel makkelijker, dan dat dat het geval was bij de vorige doelstelling (met betrekking tot het gewicht van het alkaan). Dit komt namelijk, omdat er naast de lengte van het alkaan, niet apart onderscheid hoeft te worden gemaakt wat betreft het extra gewicht van het alkaan. Het is in dit geval dus niet nodig om een uitgebreid algoritme te

---

1. <http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A000602>

vinden, maar het is al mogelijk het met enige recursieve rijen uit te rekenen. Hoe dat in zijn werk gaat, leggen we hieronder uit:

Ook nu weer, gaan we eerst aan de slag met alkylgroepen, waaruit we later alsnog de alkanen kunnen berekenen. Om te beginnen kunnen we het rijtje  $R_n$  definiëren, als de rij met het aantal mogelijkheden waarop je een alkylgroep kunt vormen met lengte  $n$ . Tegelijk definiëren we nu ook een andere rij, namelijk  $S_n$ , en wel als de som van alle waarden  $R_k$ , en wel van  $k = 0$  tot en met  $k = n$ . Zodoende geldt dus ook dat  $R_k = S_n - S_{n-1}$ . Als we dan de gegevens hebben tot en met een zekere waarde  $n$ , kunnen we de nieuwe waarden hieruit berekenen. Het aantal mogelijkheden waarop je een alkylgroep kunt vormen die 1 langer is dan de voorgaande alkylgroepen uit de reeks, kun je namelijk op een manier uitrekenen, die enigzins analoog is aan zoals dat in het vorige algoritme gebeurde: Je bepaalt  $R_{n+1}$ , door te kijken op hoeveel verschillende manieren je een centraal C-atoom kunt omringen door drie alkylgroepen, waarvan er minstens een de lengte  $n$  heeft. Dit aantal is niet makkelijk te bepalen aan de hand van de rij  $R_k$ , maar wel met de rij  $S_n$ . Immers: het totaal aantal manieren om het te omringen met drie alkylgroepen met lengte oplopend tot  $n$  is gelijk aan  $f_3(S_n)$  en als je nu de eis stelt, dat er minstens een tak, lengte  $n$  heeft, dan trek je dus het aantal mogelijkheden hiervan af, waarop je het C-atoom kunt omringen met alleen maar takken, kleiner of gelijk aan  $n - 1$ . Zodoende geldt dus:

$$R_{n+1} = f_3(S_n) - f_3(S_{n-1}) \quad , \text{ oftewel :}$$

$$S_{n+1} = S_n + f_3(S_n) - f_3(S_{n-1}) = S_n + \frac{S_n(S_n + 1)(S_n + 2)}{6} - \frac{S_{n-1}(S_{n-1} + 1)(S_{n-1} + 2)}{6}$$

Omdat we ook weten dat  $R_0 = 1$  en  $R_1 = 1$ , weten we dus ook dat  $S_0 = 1$  en  $S_1 = 1 + 1 = 2$ . Nu hebben we daarmee dus een recursie gevonden voor de rij  $S_n$ . Vervolgens willen we met behulp van deze rij  $S_n$  die betrekking heeft op de alkylgroepen, ook een rij  $A_n$  kunnen berekenen, die gelijk is aan het aantal mogelijk alkanen met lengte  $n$ . Voor even alkanen geldt nu, dat het aantal mogelijkheden, waarop je twee alkylgroepen tot een alkaan kunt samenvoegen met lengte  $2n$ , gelijk is aan  $f_2(R_n)$ , oftewel  $f_2(S_n - S_{n-1})$ . Formeel gesteld:

$$A_{2n} = \frac{(S_n - S_{n-1})(S_n - S_{n-1} + 1)}{2}$$

Om het te voltooien, moeten we ook voor oneven alkanen, de bijbehorende waarde  $A_{2n+1}$  bepalen, wat iets lastiger was: Een oneven alkaan met lengte  $2n + 1$  is op te vatten als een centraal C-atoom, omringd door minstens twee alkylgroepen met lengte  $n$ , en voor de rest alkanen met lengte kleiner of gelijk aan  $n$ . Het aantal mogelijkheden, om dit te doen, waarbij minstens een zijtak lengte  $n$  heeft, is, ongeveer net als bij het bepalen van  $R_{n+1}$ , gelijk aan  $f_4(S_n) - f_4(S_{n-1})$ . Maar er moesten minimaal twee zijtakken lengte  $n$  hebben, in plaats van  $n$  zijtak, dus moeten we hier nog het aantal mogelijkheden van aftrekken, waarop je een C-atoom kunt omringen met zijtakken, waarvan slechts een zijtak lengte  $n$  heeft. Dit aantal is gelijk aan het aantal manieren waarop je drie combinaties kunt maken, van zijtakken die allen kleiner zijn dan  $n$  (je kijkt dan naar het aantal combinaties waarop je de overige takken zou kunnen kiezen). Dit is dus  $f_3(n - 1)$ . De totale vergelijking wordt dan  $A_{2n+1} = f_4(S_n) - f_4(S_{n-1}) -$

## Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen

© havovwo.nl

---

$f_3(n-1)$  en dus formeel gesteld (na het nog verder vereenvoudigd te hebben):

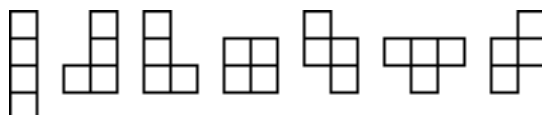
$$A_{2n+1} = \frac{S_n(S_n+1)(S_n+2)(S_n+3)}{24} - \left(1 + \frac{S_{n-1}+3}{4}\right) \cdot \frac{S_{n-1}(S_{n-1}+1)(S_{n-1}+2)}{6}$$

Zo hebben we deze reeks van alkanen dus op een wiskundige manier, exact kunnen vastleggen. Aan de hand van deze recursie is in de tabel hieronder voor verschillende waarde van  $n$ , de bijbehorende waarden van  $R_n$ ,  $S_n$  en  $A_n$  te vinden. Omdat deze reeksen zo ontzettend snel toenemen (vele malen sneller dan die van de alkanen, geteld op gewicht), hebben we het maar voor enkele waarden van  $n$  uitgerekend. (Overigens moet je, om de recursie ook voor de waarden van  $A_0$  en  $A_1$  kloppend te maken, aannemen dat  $S_{-1} = 0$ ).

$n$	$R_n$	$S_n$	$A_{2n}$	$A_{2n+1}$
0	1	1	1	1
1	1	2	1	3
2	3	5	6	61
3	31	36	496	82146
4	8401	8437	35292601	211275803366603
	$\vdots$	$\vdots$	$\vdots$	$\vdots$

## Onderzoek naar Polymino's

Een van de oudste en meest gespeelde computerspelletjes is zeker Tetris. Het spel maakt gebruik van zogenaamde tetramino's: figuurtjes die opgebouwd zijn uit vier (Gr. tetra) vierkantjes. In het totaal komen er in het spel zeven tetramino's voor. Voor het gemak hebben zij allemaal een letter gekregen: I, J, L, O, S, T en Z:

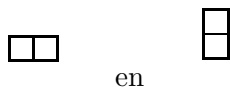


Het spel wordt gespeeld met de tetramino's, omdat bij het gebruik van de pentamino's (Gr. penta = vijf) een te ingewikkeld programmaproces ontstond. De ontwerper van het spel, Alexei Pajitnov, had de pentamino's als uitgangspunt genomen. Dit deed hij omdat hij geïnteresseerd was geraakt in het traditionele spel Pentomino. Hierbij moet met pentamino's een rechthoek geconstrueerd worden (zie bijlagen). Hij bedacht een spelletje, waarbij je de pentamino's zo moest plaatsen, dat ze in een rechthoekig veld rijen (lines) vormden - die vervolgens verdwenen. Het bleek dat je met de twaalf pentamino's verschillende 'translaties' uit kon voeren: spiegeling en draaiing. Er moesten dus ofwel twee translatieknoppen gedefinieerd worden ofwel extra - gespiegelde - pentamino's gevormd worden. Pajitnov merkte dat beide methodes erg omslachtig waren. Om dit probleem te omzeilen stapte hij over op tetramino's. Er waren, inclusief de gespiegelde, immers maar zeven tetramino's. Het programma kon zo heel gemakkelijk uitgevoerd worden. Zoals de hierboven besproken tetra- en pentamino's zijn er ook voor alle andere natuurlijke getallen zulke figuren te creëren. Voor de eerste, kleine getallen is het aantal nog gemakkelijk (met de hand) te bepalen. Er zijn echter al 108 hepta- en 369 octamino's. Om al deze figuren uit te tekenen ben je al gauw een hele tijd bezig. Bovendien kun je gemakkelijk fouten maken door figuurtjes dubbel te tellen of door er een aantal te vergeten. Een goede oplossing is ook hier de computer. Er zijn verschillende manieren om een computerprogramma te schrijven dat alle polymino's voor een bepaalde  $n$  telt. We hebben ons over het probleem gebogen en hebben een methode ontwikkeld, die ze allemaal kan tellen:

- Neem een polymino, te beginnen met de monomino (één vierkantje);
- Aan deze figuur kan op een aantal plaatsen (bij de monomino zijn er dat 4) een extra vierkantje getekend worden:



- Vergelijk de nieuwe figuren met elkaar en onderzoek of er dubbele figuren voorkomen;
- Streep de dubbele figuren weg en begin opnieuw met de overgebleven nieuwe figuren (hier kunnen dat 1 of 2 domino's zijn);



Zo kom je dus al gauw tot het probleem van draaiing en spiegeling. Je kunt de domino's bekijken als twee verschillende of als twee dezelfde figuren. Als je als voorwaarde stelt dat gedraaide polymino's overeenkomstige figuren zijn, dan zijn de domino's gelijk; immers: door de eerste domino negentig graden te draaien, ontstaat de tweede domino.

Stel je als voorwaarde dat op deze manier verschillende figuren ontstaan, dan heb je te maken met twee verschillende domino's: een horizontale en een verticale. Maar dit was niet de enige methode die we hadden bedacht. Veel eerder al hadden we een manier gevonden die op het oog heel makkelijk was. Maar toen we die verder uitwerkten, kwamen we er al snel achter dat het heel moeilijk zou worden. Dit algoritme ging als volgt in zijn werk:

- Neem een  $n$ , een getal dat aangeeft hoeveel vierkantjes er gebruikt moeten worden. (bijvoorbeeld  $n = 3$  geeft alle trimino's);
- Maak nu een tabel van  $n \times n$  vierkantjes (hier dus in totaal 9 vierkantjes). Later echter bleek dat een tabel van  $n \times \frac{n}{2}$  (waarbij  $\frac{n}{2}$  naar boven wordt afgerond) al genoeg was;
- Laat de computer alle mogelijkheden bepalen, waarop je 3 van de 9 vierkantjes zwart kunt maken; dat is  $\binom{9}{3} = 84$ . Deze tabel kunnen we voorstellen als een matrix met enkel enen en nullen erin. Als we die matrix nu opnieuw schrijven en daarbij alle cijfers achtereen zetten, krijgen we een getal van  $n^2$  cijfers (0 of 1). In ons voorbeeld is dus 000100110 gelijk aan:

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{matrix}$$
 en ook gelijk aan:
 
 ;

Om alle mogelijkheden af te gaan moet de computer getallen van  $n^2$  cijfers bekijken. Dat kan op twee manieren:  
1. Begin bij 00000000 (bij  $n = 3$ ) en laat de computer tellen, dus nu komt 00000001, dan 00000010, 00000011 enz. Vervolgens laat je alle getallen die meer of minder dan 3 enen hebben wegstrepen - omdat er in elke tabel 3 enen (en 6 nullen) voor moeten komen. Zo houd je 84 getallen over.

2. Begin bij 11100000 (bij  $n = 3$ ) en laat de computer de laatste 1 opschuiven naar rechts totdat deze buiten het bereik komt; 11000001 is dus het laatste getal van deze reeks. Dan schuif je de tweede 1 - van het eerste getal - een plekje naar rechts (inclusief de derde 1): 10110000 en laat je de laatste 1 naar rechts opschuiven; juist tot 10100001. Daarna komt dus 100110000. Als je het getal 10000011 hebt bereikt, laat je de eerste 1 mee opschuiven: 01110000, 01101000, ..., 01000011 en vervolgens 00111000, ..., 00100011, precies tot en met 00000111. Dit zijn alle 84 mogelijkheden (zie hiernaast);

11100000
11010000
11001000
⋮
10110000
10101000
⋮
10000011
01110000
01101000
⋮
01000011
00111000
⋮
00000111

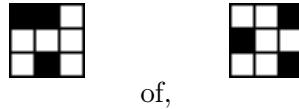


# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen  
© havovwo.nl

---

- Vervolgens laat je de computer alle gevormde figuurtjes bekijken en laat je diegene wegstrepen, waarvan niet alle zwarte vierkantjes aan elkaar grenzen; bijvoorbeeld:

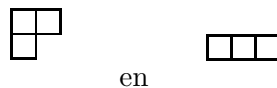


De trimino's die je nu nog overhoudt bevatten nog steeds dubbele (want: 3 zwarte vakjes op de bovenste rij komt op hetzelfde neer als 3 zwartje vakjes op de middelste rij);

- Laat dus de computer in alle figuurtjes de zwarte vakjes zo ver mogelijk naar links en naar boven schuiven (zonder dat de configuratie van de zwarte blokjes daarbij verandert), zodat de dubbele trimino's exact hetzelfde zijn:



- Ten slotte laat je de computer alle exact dezelfde figuurtjes wegstrepen en zo houd je 6 trimino's over;
- Weer ontkom je niet aan het probleem van draaiing: als je als voorwaarde stelt dat de figuurtjes gedraaid mogen worden om 'andere' trimino's te maken, dan kun je de computer alle 6 de tabellen laten draaien en zo de dubbele weg laten strepen. Uiteindelijk houd je dan 2 trimino's over:

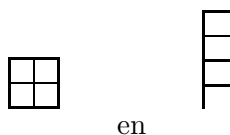


We hebben geprobeerd een programma te ontwikkelen, dat het eerste algoritme uit kon voeren. We hadden echter niet de tijd tot onze beschikking om dit programma geheel uit te werken. Om toch verder te kunnen met het onderwerp hebben we op internet gezocht naar secundaire literatuur. We merkten dat er best veel boeken over polymino's geschreven waren, maar dat deze alleen in universiteitsbibliotheken te vinden waren. Wij hebben daarom bronnen van internet gebruikt. Op de Mathworld-Wolfram-website vonden wij een mooie tabel (deze staat op de volgende pagina). Hierin worden voor de eerste vijftien getallen  $n$  de bijbehorende aantallen polymino's gegeven. Daarbij maakten de makers een onderscheid tussen free, one-sided en fixed polymino's. Ook hebben ze het aantal polymino's met holes (polymino's met gaten erin) gegeven. Volgens het citaat van de website, te lezen in de bijlagen, hebben de verschillende categorieën karakteristieke eigenschappen. "Free polymino's kunnen worden opgepakt en worden gedraaid, zodat gespiegelde figuren identiek worden geacht." Dit is dus het kleinste aantal omdat van elk figuur dat een spiegelbeeld heeft er maar één geteld wordt. Dat in tegenstelling tot de one-sided polymino's, die niet gespiegeld mogen worden, zodat van elk figuur ook het spiegelbeeld wordt geteld, behalve wanneer dit exact dezelfde polymino geeft; de O-tetramino heeft bijvoorbeeld geen spiegelbeeld dat niet exact hetzelfde is, dat geldt ook voor de I-tetramino:

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen

© havovwo.nl



In de tabel staat het aantal polymino's dat tot de derde categorie, fixed, wordt gerekend voor alle mogelijke figuren. Dat betekent dus dat elk figuur dat  $90^\circ$  gedraaid is een nieuw figuur vormt, behalve wanneer het exact hetzelfde is. Bovendien leveren alle gespiegelde figuren verschillende polymino's, behalve wanneer dit natuurlijk een figuurtje oplevert dat exact hetzelfde is. In de laatste kolom staat het aantal polymino's dat zogenoemde gaten bevat. Bij de kleine getallen komen deze natuurlijk nog niet voor, maar al bij  $n = 7$ , komt er een voor en bij  $n = 8$  zijn er ineens al zes. Deze aantallen zijn 'uittreksels' van de eerste categorie en daarom moeten ze niet opgeteld worden aan het aantal free polymino's. Gevolg hiervan is dat alle polymino's met gaten elk maar één keer zijn geteld en er komt geen spiegeling, dan wel draaiing aan te pas. Dit geeft het kleinste aantal mogelijkheden. In de bijlagen zijn afbeeldingen van alle polymino's met gaten voor  $n = 7$ ,  $n = 8$  en  $n = 9$  weergegeven. Daarop is bijvoorbeeld te zien dat er in het aantal nonamino's ( $n = 9$ ) zelfs een figuur voorkomt dat twee lege vakjes omringt.

n	name	free	one-sided	fixed	with hole(s) (in free)
1	monomino	1	1	1	0
2	domino	1	1	2	0
3	triomino	2	2	6	0
4	tetromino	5	7	19	0
5	pentomino	12	18	63	0
6	hexomino	35	60	216	0
7	heptomino	108	196	760	1
8	octomino	369	704	2725	6
9		1285	2500	9910	37
10		4655	9189	36446	195
11		17073	33896	135268	979
12		63600	126759	505861	4663
13		238591	476270	1903890	21474
14		901971	1802312	7204874	96496
15		3426576	6849777	27394666	425449

Om het onderzoeken van deze reeksen gemakkelijker te maken, hebben we de tabel gedraaid, zodat verschilrijen toe te voegen zijn, zonder de hele tabel om te gooien. In de tabellen die volgen, hebben we de eerste vier verschilrijen van elke reeks geplaatst. We zijn een tijdje aan het zoeken geweest naar formules en regelmaten om de reeksen weer te geven, maar dit bleek heel moeilijk. We hebben gezocht voor de one-sided polymino's, omdat van deze reeks de tetramino's gebruikt worden voor het spel Tetris. Na een korte tijd experimenteren met formules met e-machten en formules met faculteiten, kwamen we niet op een formule die voor veel meer dan de eerste vier getallen klopt; laat staan een formule die de gehele reeks exact weet te beschrijven. Op verschillende websites

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

Victor Pessers & Paul Witteveen

© havovwo.nl

waar deze reeksen te vinden zijn, zijn ook geen formules of regelmatigheden weergegeven. Het lijkt dus dat het heel moeilijk is of misschien zelfs onmogelijk hier verder mee te kunnen. Dat komt omdat er veel te veel variabelen een rol spelen. We hebben wel verder gekeken naar de verschilrijen en een opvallende afwijking geconstateerd. Zoals te zien in de tabel van verschilrijen van de one-sided polymino's komt er al in verschilrij vier ( $\Delta_4$ ) een negatief getal ( $-1$ ) voor. Dit is belangrijk om te weten, want dit impliceert dat de functie niet exponentieel kan zijn. De afgeleide van een exponentiële functie is namelijk een veelvoud van zichzelf. De tweede afgeleide is dan logischerwijs ook een veelvoud van de functie. Dit gaat volgens de volgende differentieerregel:

$$[g^x]' = g^x \cdot \ln(g)$$

Na de andere reeksen uitgewerkt te hebben tot de veertiende verschilrij - waarbij de veertiende verschilrij maar één getal bevat - kwamen wij tot de ontdekking dat er ook in free-categorie negatieve getallen voorkomen. Die doen zich echter pas voor in  $\Delta_{10}$ ,  $\Delta_{12}$  en  $\Delta_{14}$ . In de verschilrijen van de fixed-categorie komen tot aan  $\Delta_{14}$  geen negatieve getallen voor. Bovendien ontdekten we dat de negatieve getallen (naar het schijnt) alleen in even verschilrijen (kunnen) voorkomen.

free	1	1	2	5	12	35	108	369	...
$\Delta_1$	0	1	3	7	23	73	261	...	...
$\Delta_2$		1	2	4	16	50	188	655	...
$\Delta_3$			1	2	12	34	138	467	...
$\Delta_4$			1	10	22	104	329	1332	...
one-sided	1	1	2	7	18	60	196	704	...
$\Delta_1$	0	1	5	11	42	136	508	...	...
$\Delta_2$		1	4	6	31	94	372	1288	...
$\Delta_3$			3	2	25	63	278	916	...
$\Delta_4$			-1	23	38	215	638	2689	...
fixed	1	2	6	19	63	216	760	...	...
$\Delta_1$	1	4	13	44	153	544	1965	...	...
$\Delta_2$		3	9	31	109	391	1421	...	...
$\Delta_3$			6	22	78	282	1030	3799	...
$\Delta_4$			16	56	204	748	2769	...	...

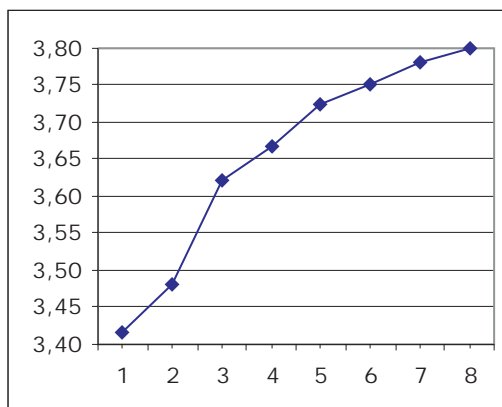
Vervolgens hebben we de quotiëntrijen bekeken. Daarin zijn veel opvallender fenomenen te herkennen. Belangrijk is het feit dat de eerste quotiëntrij als het ware karakteristiek is voor de reeks. De quotiëntrijen van hogere ordes komen namelijk na een tijdje ongeveer neer op rijen met enkel het getal één als element: 1, 1, 1... Zo worden de quotiënten van de tweede orde al nagenoeg 1 na zeven 'stappen' (dat is dus het quotiënt van  $\frac{1285}{369}$  en  $\frac{369}{108}$  - gelijk aan 1,02).

free	1	1	2	5	12	35	108	...
$Q_1$	1,00	2,00	2,50	2,40	2,92	3,09	...	...
$Q_2$		2,00	1,25	0,96	1,22	1,06	1,11	...
free	108	369	1285	4655	17073	...	...	
$Q_1$	3,09	3,42	3,48	3,62	3,67	3,73	...	
$Q_2$	1,11	1,02	1,04	1,01	1,02	...	...	
free	17073	63600	238591	901971	3426576	...	...	
$Q_1$	3,73	3,75	3,78	3,80 <sup>1</sup>	...	...	...	
$Q_2$	1,02	1,01	1,01	1,00	...	...	...	

*De cursief gedrukte waarden komen tweemaal voor in de tabel.*

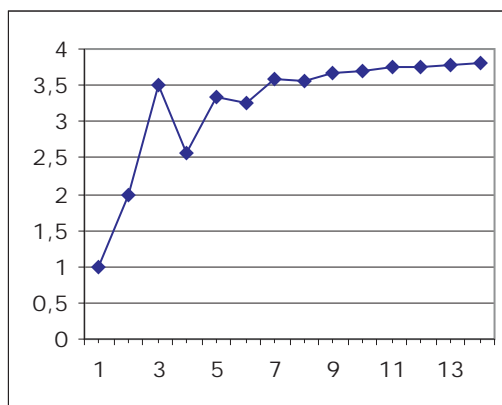
<sup>1</sup>Deze waarde is naar boven afgerond (3,798986885...).

Wat opvalt aan deze tabel is dat de quotiënten in koppels lijken voor te komen. Elk quotiënt heeft een 'vriendje'. De eerste getallen vallen hier een beetje buiten, maar vanaf  $\frac{35}{12}$  zijn de koppels wat duidelijker: 2,92 en 3,09, vervolgens 3,42 en 3,48 en verder ook 3,62 en 3,67 (1) enzovoorts. Uiteindelijk worden de quotiënten steeds groter en gaan ze richting de 3,80. Het laatste getal in onze tabel komt al aardig in de buurt. We



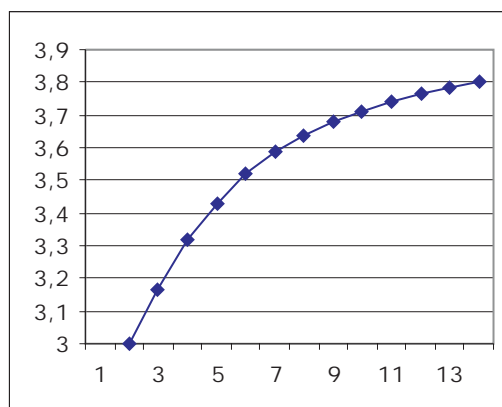
vermoeden dat de volgende quotiënten in elk geval niet boven de 4,00 uitkomen. Waarschijnlijk ligt de 'asymptoot' van de grafiek van de quotiëntrij ongeveer bij de 3,80. We hebben maar een klein stuk van de grafiek afgebeeld, omdat zo de koppels duidelijk te herkennen zijn.

In de grafiek is dus niet te zien dat het vierde quotiënt,  $\frac{12}{5}$ , kleiner is dat het voorgaande. Dat blijkt wel uit de bovenstaande tabel. Dat is bijzonder, want zo ontstaat er een 'knik' in de quotiëntgrafiek. We kunnen in de reeks van free polymino's maar één knik vinden. In de quotiëntgrafiek van de reeks one-sided polymino's kunnen we drie knikken vinden (bijvoorbeeld bij 2), terwijl er in de grafiek van de fixed-reeks geen knikken



voorkomen. Deze knikken zijn, net als negatieve getallen in verschilrijen, typerend voor de functie. De lijnen die de asymptoten van de grafieken van de drie categorieën weergeven zijn allemaal van de vorm  $y = 4,00$ . Ze hebben allemaal ongeveer de 'asymptoot' bij 3,80 liggen, maar bij de one-sided en de fixed polymino's gaan de getallen al over deze waarde heen.

Overigens is de quotiëntgrafiek van de fixed polymino's een echte vloeiende curve. In tegenstelling tot de andere grafieken, kent deze geen knikken of koppels. Zo ontstaat een mooie functie, die in het begin logaritmisch schijnt. Als we, om dit te controleren, een vergelijking opstellen van een mogelijke functie voor  $Q$ , gebruiken we de logaritmus naturalis van  $m$  (het aantal mogelijkheden): ' $\ln(m)$ '. We nemen nu de volgende vergelijking:



$$Q = \frac{1}{2} \cdot \ln(m - 1) + 2$$

Dit geeft:

$$e^Q = e^{\frac{1}{2} \ln(m-1)+2}$$

$$e^Q = e^2 \cdot e^{\ln(m-1)^{\frac{1}{2}}}$$

$$e^Q = e^2 \cdot \sqrt{e^{\ln(m-1)}}$$

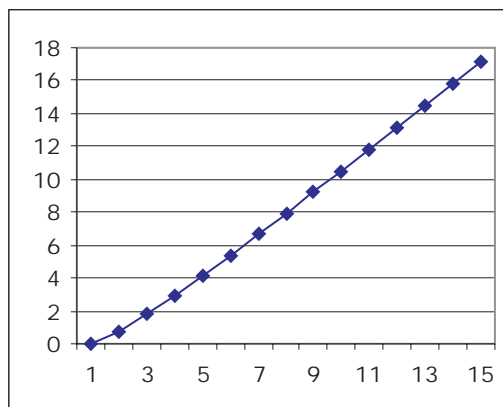
$$e^Q = e^2 \cdot \sqrt{m-1}$$

Dit is een wortelfunctie. Als dus het kwadraat van deze functie een rechte geeft, zou de functie logaritmisch zijn. Helaas is deze grafiek geen rechte, maar deze lijkt eerder weer een logaritmische. Zo hebben we hier aannemelijk gemaakt dat het geen logaritmische functie is. Het kan immers ook zo zijn dat we niet  $\frac{1}{2}$  als constante moeten gebruiken, maar  $\frac{1}{4}$ ; dan zouden we dus e tot de tweede macht en de vierdemachtswortel van  $m-1$  krijgen. In formulevorm wordt dit:

$$e^Q = e^2 \cdot \sqrt[4]{m-1}$$

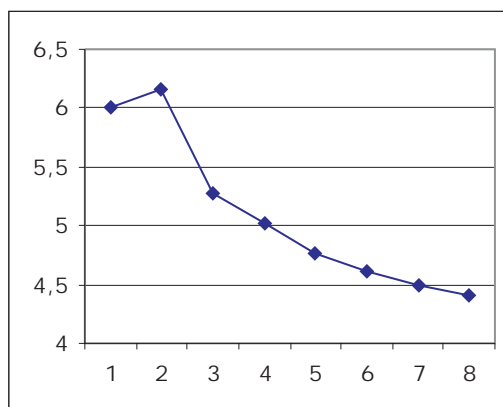
Als we deze laatste functie nu in zijn geheel tot de vierde macht verheffen, komen we tot de conclusie dat ook hier geen rechte ontstaat. We hebben hiermee nog steeds niet bewezen dat de functie niet logaritmisch is. Ook voor constante  $\frac{1}{6}$  komen we weer terug op een functie die logaritmisch schijnt.

Wat we wel ontdekten is dat de grafiek van  $\ln m$  een rechte leek. Dat brengt met zich mee dat  $^{10} \log m$  ook een rechte moet lijken (dit is inderdaad ook het geval). Dit wil niet direct zeggen dat de functie exponentieel is, immers bij een exponentiële functie is de quotiëntrij een horizontale lijn. [Bijvoorbeeld:  $y = 2^x$ . Dit geeft achtereenvolgens voor de eerste natuurlijke getallen: 1, 2, 4, 8, 16 enzovoorts.  $Q_y$  is nu  $\frac{2}{1}$ ,  $\frac{4}{2}$ ,  $\frac{8}{4}$ ,  $\frac{16}{8}$ , ... , ofwel 2, 2, 2, 2, ...]



We gaan weer terug naar de andere grafieken. De enige quotiëntgrafiek die we nog niet behandeld hebben is die van de polymino's met holes:

De asymptoot van de aantallen met holes ligt ook rond de  $y = 4,00$ , maar komt er (bijna) zeker bovenuit. Het valt namelijk op dat deze grafiek, in tegenstelling tot de drie andere, dalend is. Het laatste quotiënt dat wij konden berekenen, was  $\frac{425449}{96496}$ , gelijk aan 4,41. Bijzonder is dat de quotiënten ook hier in koppels voor lijken te komen: beginnend met 6,00 en 6,17, gevolgd door 5,27 en 5,02 tot aan 4,49 en 4,41. Hiernaast is de grafiek van de quotiëntrij te zien. Ook hier valt meteen de knik op. De koppels zijn



in de grafiek niet zo gemakkelijk meer te herkennen. Het eerste koppel is in elk geval wel duidelijk te zien. De andere koppels zijn minder duidelijk, maar ze zijn er nog wel. Elke overgang van het ene koppel naar het andere brengt een iets sterkere daling met zich mee. Zie het verschil tussen de eerste twee quotiënten in de grafiek en het verschil tussen de tweede twee quotiënten bij 3. Het voorkomen van koppels is dus een bijzonder verschijnsel bij deze grafieken. Dit kan te maken hebben met het schijnbare verband dat er onderling tussen de polymino's voor een oneven  $n$  en polymino's met een even aantal vierkantjes is. Een verband dat we ook bij het verloop van de alkanenreeks tegen waren gekomen. Hier was echter wel een duidelijk verschil in de manier van berekenen. Daarentegen kan het ook te maken hebben met het aantal mogelijkheden: dat ligt dan tevens aan het onderlinge verband tussen even en oneven getallen. Het aantal mogelijkheden, in de one-sided- en fixed-categorie, is oneven, wanneer er een oneven aantal figuren voorkomt dat geen spiegelbeeld heeft dat niet exact hetzelfde is als zichzelf; de tetramino's kennen bijvoorbeeld één figuur - de O-tetramino - dat geen spiegelbeeld heeft (dat niet exact hetzelfde is als zichzelf). Hierin is het getal één dus oneven en moet het totale aantal wel oneven zijn. Bij de eerste en de laatste categorie, free en with holes, is dat wat moeilijker te bepalen, omdat deze sowieso niet gedraaid of gespiegeld worden.

Het is ons dus niet gelukt exacte formules of regelmaten te vinden in de aantallen, maar wel hebben we een paar karakteristieken gevonden en deze geprobeerd uit te leggen. We hebben een aantal zaken vastgesteld, bijvoorbeeld dat de functies niet exponentieel zijn en dat de quotiëntfuncties waarschijnlijk (in ieder geval niet met een van de grondtallen 2, 4 of 6) niet logaritmisch zijn. In elk geval is ons wel duidelijk geworden dat de quotiëntfuncties een boven-, dan wel ondergrens, kennen; de meeste quotiëntgrafieken lijken te convergeren. Bovendien zagen we knikken en koppels in quotiëntgrafieken.

## Conclusie

Het is ons gelukt om een programma te schrijven, dat het aantal isomeren van een alkaan berekent, met een bepaald aantal koolstofatomen. Het uitgebreide algoritme dat we daarvoor gebruikten is dus succesvol gebleken. Het basisprincipe van het algoritme is dat je door oude alkylgroepen te combineren, nieuwe mogelijkheden voor grotere alkylgroepen kunt berekenen, en uiteindelijk diezelfde alkylgroepen kunt gebruiken om uit te rekenen hoeveel verschillende alkanen ermee gevormd kunnen worden. Omdat de variabelen waarmee de computer werkt, echter maar een beperkte capaciteit hebben, zijn de waarden die uit het programma zijn voortgekomen, niet meer betrouwbaar voor alkanen met een gewicht van 54 of hoger. Naast dit algoritme, hebben we ook onderzoek gedaan naar het aantal verschillende alkanen met een zekere lengte. Hierbij hebben we een recursie in deze ontwikkeling gevonden, zodat we in deze reeks op een wiskundig verantwoorde wijze, precies de regelmaat hebben weten te ontdekken. Dat het in dit geval wel mogelijk is geweest om een precieze regelmaat te vinden en bij de andere vraagstelling niet, komt doordat er ditmaal niet met twee factoren (namelijk lengte en gewicht van het alkaan) rekening gehouden hoeft te worden, maar dat alleen de lengte van belang is. Uiteindelijk kunnen we over beide bevindingen erg tevreden zijn.

Ookal stond het in onze planning om ook een programma te schrijven voor de reeks van de polymino's: doordat we onze handen al vol hadden aan het probleem met de alkanen, hebben we dit helaas niet meer kunnen doen. Wel hebben we twee van onze ideeën om een programma te schrijven toegevoegd. Bij het eerste algoritme nemen we een polymino en onderzoeken we op welke plekken er een vierkantje 'aangeplakt' kan worden zodat er een nieuwe polymino ontstaat. Met deze polymino kunnen we dan verder. Bij het tweede algoritme bekijken we op hoeveel manieren we we  $n$  van de  $n^2$  vakjes, van een  $n \times n$ -tabel, kunnen vullen. Helaas zijn we niet meer aan het maken van een van beide programma's toegekomen.

Gelukkig was er al meer over het onderwerp geschreven en konden we de literatuurstudie en het tweede onderzoek combineren. Door enkel één tabel van  $n$  website over te nemen, konden we al een heel stuk verder met ons onderzoek. Het wiskundig uitleggen van de getallen kon toen namelijk beginnen. We zijn begonnen met het maken van verschilrijen. Uit deze rijen viel niet heel veel bijzonders op te maken, met uitzondering van het feit dat er soms negatieve getallen voorkomen. Dit betekent dat de functie niet exponentieel kan zijn. Vervolgens gingen we de quotiëntrijen onderzoeken. Deze gaven wat meer bijzonderheden. Zo was de quotiëntgrafiek van de zogenaamde fixed polymino's een hele mooie, vloeiende curve, die logaritmisch leek. We hebben onderzocht of deze logaritmisch was, maar we hebben niet hard kunnen bewijzen dat de functie logaritmisch is of niet. Verder vonden we knikken en koppels in de quotiëntgrafieken. De knikken zijn, gelijk aan negatieve getallen in de verschilrij, typerend voor de functie. De koppels ontstaan (waarschijnlijk) door de verschillen tussen even en oneven getallen en hun aantallen polymino's.

Uiteindelijk hebben we geen exacte formule of een regelmaat gevonden om de reeksen wiskundig te beschrijven. Dit is overigens helemaal niet vreemd. Er zijn immers meer combinatorische problemen in de wiskunde waar geen exacte formule voor te vinden is.

## Evaluatie

Om te beginnen, hebben we gemerkt dat we bij het stellen van hoofd- en deelvragen een beetje te voorbarig zijn geweest. We hadden gedacht gemakkelijk twee programma's te schrijven (een voor de alkanen, en een voor de polymino's), maar uiteindelijk is dat niet gelukt. Dit komt omdat het maken van het programma dat de alkanen telt al een hele tijd duurde. Daardoor is het onderzoek naar het aantal polymino's daarbij enigszins achtergebleven. Gelukkig is het programma dat de alkanen telt wel goed uit de verf gekomen. Ook zijn we blij dat we met de gegevens, die we met dit programma vergaard hebben, de rij van het aantal isomeren hadden bepaald. Bovendien hebben we een recursie kunnen opstellen voor de alkanen met betrekking tot de lengte. Hoewel we geen computerprogramma hebben kunnen maken wat betreft de polymino's, hebben we de reeks wiskundig kunnen bekijken. Dit kwam doordat wij secundaire literatuur bekeken hebben. Zo hebben we door gebruik te maken van verschillende websites toch nog een resultaat op dit vlak kunnen behalen.

De oriëntatie die voorafging aan het maken van dit werkstuk is een van de moeilijkst verlopen fases. Hierin hebben wij namelijk een tijd lopen dubben over ons onderwerp. We kozen voor een moeilijk en nieuw onderwerp en zorgden daarmee voor heel wat twijfels aan onder andere onze kant. Wel zijn we erg tevreden over het resultaat, aangezien we er totaal niet van overtuigd waren dat er iets misschien zou gaan lukken.

Met het plan van aanpak hebben we duidelijk gemaakt dat we weinig zeker wisten, maar dat we zeker heel wat wilden proberen. Helaas is dit er niet allemaal van gekomen.

Het pws-logboek is naar onze mening een goede zaak. Zodra je namelijk eenmaal begint met het aantekenen van de zaken die je doet, vergeet je het niet meer. Maar je moet in het begin wel even die aanzet krijgen, anders blijf je in gebreken. Ook de planning van bepaalde zaken wordt bevorderd door het maken van een logboek, hebben we meermalen gemerkt. Ons logboek is uiteindelijk een verzameling van aantekeningen geworden, maar dat maakt het niet minder bruikbaar.

We zijn dus zeker te spreken over het resultaat, ondanks dat het zelfs niet eens geheel het beoogde resultaat was. Doordat we gaandeweg het onderzoek pas echt merkte, hoe moeilijk onze vraagstelling was, waren we des te blijer met hetgeen wat ons allemaal wel gelukt was (al waren de resultaten natuurlijk niet op elk deel van ons onderzoek even groot). En dan hadden we nog niet eens alles van onze activiteiten in het werkstuk verwerkt (maar niet alle ondernomen activiteiten hadden natuurlijk dan ook tot iets echt bruikbaars geleid). Al met al, heeft het ons toch echt heel wat tijd en moeite gekost, (ook met name het verwerken van het onderzoek was een stuk zwaarder dan we van tevoren hadden gedacht) maar we kunnen dan ook wel verheugd zijn over het uiteindelijke resultaat.

Al leende het onderwerp en de manier van onderzoek zich niet optimaal voor een directe samenwerking; door toch nog zoveel mogelijk samen te doen, op andere momenten elkaars werk kritisch te bekijken en de op- en aanmerkingen in acht te nemen, hebben we veel aan elkaars samenwerking gehad.

Al met al zijn we trots op ons werk en sluiten we dit profielwerkstuk met veel plezier af.



## Literatuurlijst

- <http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A000602>
- <http://mathworld.wolfram.com/Polyomino.html>
- <http://clarkjag.idx.com.au/index.htm>
- <http://www.geocities.com/micadesa/educacion/edupoliminos.html>

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

naam: Victor Pessers en Paul Witteveen  
© havovwo.nl

---

## Bijlagen program polymeren;

```
uses  
math;
```

*{onderstaande functie is in het algoritme zelf niet besproken, maar het nut ervan is grof gezegd, dat het ervoor zorgt, dat het programma stopt wanneer het in een bepaalde situatie geen zin meer heeft om in de tabel nog verder af te gaan, en bespaart op die manier dus tijd}*

```
function dim(s: integer):integer;  
begin  
dim:=trunc((power(3,s)-1)/2-s);  
end;
```

*{in onderstaand deel van het programma wordt de functie f gedefinieerd zoals dat in het verslag staat aangegeven, voor de vier verschillende waarden m. (m=1, m=2, m=3 en m=4)}*

```
function f(n: integer; m:integer):integer;  
begin  
case m of  
1: f:=n;  
2: f:=trunc(n*(n+1)/2);  
3: f:=trunc(n*(n+1)*(n+2)/6);  
4: f:=trunc(n*(n+1)*(n+2)*(n+3)/24);  
end;  
end;
```

*{ook op deze functie hoef je niet echt te letten; het is alleen gemaakt om de gehele programmatuur een stuk korter te maken: om precies te zijn, is dit een soort van verkorte if-then-else statement}*

```
function ite(a: boolean;b:integer;c:integer):integer;  
begin  
If a=true then  
ite:=b  
else  
ite:=c;  
end;
```

```
var //hier heb ik de verschillende typen gedeclareerd  
tabel:array[0..1000,0..1000] of int64; //dit is dus tabel met alkylgroepen  
lijst:array[0..1000] of int64; //en dit is dus de reeks van de berekende alkanen  
tak:array[1..4,1..2] of integer;  
y,z:int64;  
x,ant,hant,tak1x,tak2x,tak3x,tak4x,tak1y,tak2y,tak3y,tak4y:integer;  
{tak1x staat voor de lengte van de eerste tak, en zo staat tak2y voor het extra gewicht van de tweede tak}
```

```
begin
```

```
assignfile(output,'leeg.txt'); //als uitput-bestand heb ik het textbestand 'leeg.txt'
```

```
ant:=60; //programma rekent tot en met het getal 60  
tabel[0,0]:=1; //beginkennis: er is 1 alkyl, met lengte 0 en gewicht 0  
hant:=trunc((ant+1)/2); //om de alkanen tot  $2n+1$  uit te rekenen, hoeven we de alkylen maar tot  $n$  uit te rekenen
```

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

naam: Victor Pessers en Paul Witteveen  
© havovwo.nl

---

*{Het is niet nodig precies te begrijpen hoe de onderstaande for-loops werken: van belang is te weten, dat met deze*

*loop-constructie alle verschillende mogelijkheden worden afgegaan, waarbij nog blijft gelden*  
*tak1x<=tak2x<=tak3x<=tak4x en tak1y<=tak2y<=tak3y<=tak4y}*

```
For tak1x:=0 to hant do
begin
tak[1,1]:=tak1x;
For tak2x:=0 to min(tak1x,ant-tak1x) do
begin
tak[2,1]:=tak2x;
For tak3x:=0 to min(tak2x,ant-tak1x-tak2x) do
begin
tak[3,1]:=tak3x;
For tak4x:=0 to min(tak3x,ant-tak1x-tak2x-tak3x) do
begin
tak[4,1]:=tak4x;
For tak1y:=0 to min(dim(tak1x),ant-tak1x-tak2x-tak3x-tak4x) do
begin
tak[1,2]:=tak1y;
For tak2y:=0 to ite(tak2x<>tak1x,min(dim(tak2x),ant-tak1x-tak2x-tak3x-tak4x-tak1y),tak1y) do
begin
tak[2,2]:=tak2y;
For tak3y:=0 to ite(tak3x<>tak2x,min(dim(tak3x),ant-tak1x-tak2x-tak3x-tak4x-tak1y-tak2y),tak2y) do
begin
tak[3,2]:=tak3y;
For tak4y:=0 to ite(tak1x<>tak2x,0,ite(tak4x<>tak3x,min(dim(tak4x),ant-tak1x-tak2x-tak3x-tak4x-tak1y-tak2y-
tak3y),tak3y)) do
begin
tak[4,2]:=tak4y;
```

*{in onderstaande constructie wordt het proces geregeld waarin de verschillende takken in groepjes wordt ingedeeld, en waarbij van elk groepje het aantal mogelijke combinaties wordt uitgerekend, en het resultaat wordt opgeslagen in de variabele y}*

```
x:=1;y:=1;z:=1;
while x<>5 do
begin
If (tak[x,1]=tak[x+1,1]) and (tak[x,2]=tak[x+1,2]) then
z:=z+1
else
begin
y:=y*f(tabel[tak[x,1],tak[x,2]],z);
z:=1;
end;
x:=x+1;
end;
```

*{in de programmatuur die hier staat wordt als het ware de verschillende combinaties die eruit rollen 'geturfd', waarbij de verschillende alkylen, dan wel alkanen, worden opgehoogt met de variabele y, wat het aantal mogelijk combinaties*

*voorstelt, waarop deze alkylgroep of alkaan gevormd kan worden.}*

```
If tak1x=tak2x then
begin
```

## Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

naam: Victor Pessers en Paul Witteveen  
© havovwo.nl

---

```
inc(lijst[tak1x+tak1y+tak2x+tak2y+tak3x+tak3y+tak4x+tak4y+1],y);
If (tak3x=0) and (tak4x=0) then
inc(lijst[tak1x+tak1y+tak2x+tak2y],y);
end;
If tak4x=0 then
inc(tabel[tak1x+1,tak1y+tak2x+tak2y+tak3x+tak3y],y);

end;
end;
end;
end;
end;
end;
end;
end;
end;
```

*{in onderstaand script, wordt het resultaat naar het text-bestand op de harde schijf geschreven}*

```
for x:=0 to ant do
writeln(lijst[x]);

end.
```

# Profielwerkstuk Alkanen en Polymino's *combinatorisch bekeken*

naam Victor Pessers en Paul Witteveen

© havovwo.nl

---

## Bijlage

n	aantal alkanen met n koolstofatomen	30	4111846763.
0	1.	31	10660307791.
1	1.	32	27711253769.
2	1.	33	72214088660.
3	1.	34	182183785195.
4	2.	35	487340501958.
5	3.	36	1233020562336.
6	5.	37	3327181368833.
7	9.	38	8440761464483.
8	18.	39	22578032077265.
9	35.	40	56667131256441.
10	75.	41	149454349917522.
11	159.	42	369257845888758.
12	355.	43	958862408583005.
13	802.	44	2325953242457361.
14	1858.	45	5950046589637553.
15	4347.	46	14131370433783435.
16	10359.	47	35602444253398785.
17	24894.	48	82375181601619811.
18	60523.	49	203911739012182264.
19	148284.	50	454826015218649334.
20	366319.	51	1099497836045026039.
21	910726.	52	2311180610805593635.
22	2278658.	53	5368929025307917782.
23	5731580.	54	-8433247096938311882.
24	14490245.	55	2794182023461488538.
25	36797588.	56	-8924783087843977527.
26	93839412.	57	-1416896224264092566.
27	240215803.	58	4096967318306128625.
28	617105614.	59	4581921526883457998.
29	1590507121.	60	4027435160902506622.