

VAKIDOOT



Reis

# In dit nummer

	<b>Van de Voorzitter</b> <i>Victor Veldstra</i> <i>Voorzitter A-Eskwadraat</i>	4
	<b>Het genereren van slechts één foton</b> <i>Peter Speets</i>	5
	<b>Hoax</b> <i>Hessel Withagen</i>	8
	<b>Programmeerselveringewikkelderiseringen</b> <i>Tim Baanen</i>	10
	<b>The Hero's Journey</b> <i>Jim Vollebregt</i>	13
	<b>Informatie om nooit te vergeten</b> <i>Tim Baanen</i>	16
	<b>Kans op herkansing</b> <i>Maud Nabben &amp; Nils Warsen</i>	19
	<b>Update - Vreemde YouTubestatistieken</b> <i>Jim Vollebregt</i>	20
	<b>Als twee colloïden elkaar ontmoeten</b> <i>Peter Speets</i>	22
	<b>Een reis naar Andersomgolië</b> <i>Tim Baanen</i>	24
	<b>Volumeknopjes</b> <i>Sophie Huiberts</i>	25
	<b>Lang wachten op een lang pad</b> <i>Tim Baanen</i>	28
	<b>Kleurentelevisie</b> <i>Peter Speets</i>	30
	<b>Recept: Tofu</b> <i>Sophie Huiberts</i>	32
	<b>Priemgetaloverschot</b> <i>Jim Vollebregt</i>	34
	<b>De Fotostrip</b>	36

**Uitgave** 14 november 2017  
**Oplage** 1570  
**Deadline** 17 december 2017

### De Vakidoot is een uitgave van

Studievereniging A-Eskwadraat  
 Princetonplein 5  
 3584 CC Utrecht

**Telefoon** (030) 253 4499  
**Fax** (030) 253 5787  
**Website** a-eskwadraat.nl/vakid  
**E-mail** vakid@a-eskwadraat.nl

Wil je de Vakidoot niet meer ontvangen of ben je verhuisd? Pas dan je gegevens aan op a-eskwadraat.nl.

### Redactie

Berend Ringeling  
 Bryan Brouwer  
 Emil Meijer  
 Hessel Withagen  
 Koen van Baarsen  
 Marc Houben  
 Marlien Wennekes  
 Peter Speets  
 Sophie Huiberts  
 Tim Baanen

### Eindredactie

Jim Vollebregt

### Omslag

Koen van Baarsen



## Redactioneel

Hallo lezers! Wat fijn dat jullie weer een blik werpen in de Vakidoot!

Wisten jullie dat het voor niet-redactieleden ook mogelijk is om iets gepubliceerd te krijgen in de Vakidoot? Sterker nog: het levert je de achievement *Propellorhoedje* op om mee te pronken.

Nu hoor ik jullie denken: waar moet ik in 's hemelsnaam over schrijven? Welnu, de redactie is helemaal niet zo kieskeurig als je zou denken. Ons blad heet niet voor niets de *Vakidoot*. Je kunt het in principe zo gek maken als je zelf wilt. Denk aan recepten, gedichtjes, een roddelcolumn of een kort verhaal. Natuurlijk mag je ook over serieuzere dingen schrijven – een coole toepassing van een vak dat je volgt, bijvoorbeeld.

A-Eskwadraat organiseert een hoop te gekke activiteiten. Wij van de vereniging kunnen die onmogelijk allemaal bezoeken. Heb jij aan een toffe activiteit meegedaan en vind je dat er een verslag van moet komen in de Vakidoot? Stuur dan eens iets in en wij zullen er een mooi artikel van maken.

Wat natuurlijk ook kan, is dat je promocommissaris of wat dan ook bent van je commissie, en dat jullie reclame willen maken voor een activiteit. Je kunt dan een kort berichtje schrijven en dat opsturen, dan plaatsen wij hem voor jou ergens in de Vakidoot.

Tot slot willen we een rubriek beginnen waarin masterstudenten iets kunnen vertellen over hun scriptie (liefst zo dat in ieder geval de gemiddelde A-Eskwadrater het kan begrijpen). Dus ben jij met een waanzinnig masteronderzoek bezig waar iedereen van moet afweten, en is het alleen maar schrijven van je scriptie niet genoeg om je enthousiasme te stillen, dan is dit natuurlijk de perfecte uitkomst!

Kortom, genoeg manieren om die felbegeerde achievement te behalen.

Jim Vollebregt  
*Eindredacteur*

# Van de Voorzitter

Victor Veldstra

Voorzitter A-Eskwadraat

Mijn hele A-Eskwadraatavontuur is begonnen op de wintersportreis. Ik verwachtte vooraf dat ik me zou storten tussen allemaal hechte vriendengroepen waar ik lastig een plekje zou vinden. Dat bleek gelukkig onwaar: ik kwam er al snel achter dat men bij A-Eskwadraat erg open is en aansluiting makkelijk te vinden is. De wintersportreis heeft mij dus veel meer gebracht dan enkel een leuke week wintersporten. Uiteraard zal ik dit jaar ook weer lekker mee gaan wintersporten. Het kan immers geen kwaad om tussen al dat werken door een weekje lekker te bewegen. Daar komt nog eens bij dat ik heel erg van de bergen houd, dus even ontsnappen van dit veel te platte Nederland voelt altijd weer fijn. Het is heerlijk om je even écht te bevinden in prachtige natuur, in plaats van al die weilanden hier rond de Uithof.

In de vorige Vakidoot konden jullie een reisverslag vinden van de studiereis naar Brazilië. Als ik dat lees, ben ik blij dat A-Eskwadraat zoiets organiseert. Een reis is namelijk veel meer dan leuke dingen zien en een beetje bruin bakken in de zon. Het is een kans om dingen mee te maken buiten het comfort van je eigen huis. Een goede reis stelt je voor uitdagingen die je in het dagelijks leven niet snel zal meemaken. Ervaringen die je daar op doet, zijn, naar mijn mening, soms nog wel belangrijker dan de dingen die je leert op de universiteit. Ik kijk daarom ook erg uit naar de nieuwe studiereis van dit jaar!

Verder is reizen natuurlijk een belangrijk thema in de informatica! We kennen het Travelling Salesman Problem, buigen ons graag over het vinden

van een kortste pad tussen twee punten en menig transportmiddel rijdt, vaart of vliegt door middel van software. Het is wel opvallend dat informatici zich altijd bezig houden met het vinden van de kortste of snelste oplossing. Efficiëntie staat steeds op nummer één. Misschien moet er voor informatici maar eens een vak komen over genieten, dat is natuurlijk minstens zo belangrijk!



# Het genereren van slechts één foton

Peter Speets

Het genereren van fotonen is vrij eenvoudig. Als je niet zo kritisch bent over de golflengte die de fotonen moeten hebben en nog minder over het precieze aantal van die fotonen, kan iedere lamp als fotongenerator dienen: daar zijn ze immers voor gemaakt. Het maken van één foton, zonder dat er meteen heel veel fotonen worden gemaakt, is echter een stuk lastiger. Dit artikel beschrijft een manier waarop een enkel foton gemaakt kan worden, zoals beschreven door Nisbet-Jones *et al.*[1] Het enige wat je nodig hebt om dit thuis te doen zijn wat lasers, een magneto-optische val, een optische resonator en ongeveer  $10^6$  lasergekoelde rubidiumatomen. Dit alles in vacuüm natuurlijk.

Kan het maken van een enkel foton niet veel eenvoudiger? Het is namelijk ook mogelijk om een laserstraal door vele spiegels zoveel aan intensiteit te laten verliezen, dat er aan het einde van het spiegeldoelhof slechts één foton over blijft. Het probleem van deze aanpak is dat ieder foton van de laser in iedere spiegel een bepaalde kans heeft om weerspiegeld te worden. Dit betekent dat er aan het einde van de spiegelreeks een grote kans is dat er helemaal geen, of juist een paar fotonen doorgelaten worden. Dit willen we niet, want dan is het geen bron meer voor enkele fotonen. Er moet dus een manier worden gevonden waarmee er aanzienlijke kans is dat er in een bepaalde tijdsperiode daadwerkelijk slechts één enkel foton wordt gegenereerd.

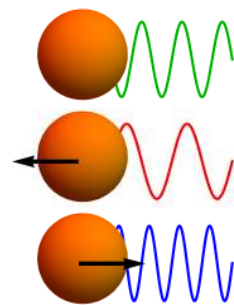
Waarom zou men eigenlijk die moeite doen om een enkel foton te maken? Het kunnen maken van een enkel foton kan gebruikt worden in kwantumcommunicatie en het manipuleren van kwantumbits in een kwantumcomputer. De fotonen kunnen worden gebruikt voor het overzetten van kwantumtoestanden. Ze laten zich wat makkelijker verplaatsen dan, bijvoorbeeld, elektronen. Het maken van enkele fotonen kan ook op andere manieren gebeuren. Dit kan niet alleen door het wegfilteren van fotonen uit een lichtbron die veel fotonen uitzendt, maar ook door het gebruik van defecten in kristallen. Een, enigszins deterministische, manier waarop enkele fotonen kunnen worden gemaakt, is door middel van een kristal dat, als er met een laser doorheen wordt geschoten, niet één, maar twee fotonen in verschillende richtingen kan sturen. Door één van de twee fotonen te meten, kan worden bepaald of er ergens een ander foton uitgezonden moet zijn.

De opstelling waar dit artikel over gaat, bestaat uit een atoomfontein met daar vlak boven een optische

resonator. Het woord atoomfontein beschrijft eigenlijk al goed wat het doet. In een atoomfontein worden zeer koude atomen omhooggegooid, waarna ze weer omlaag vallen. Het hoogste punt van hun val is in de optische resonator. Een optische resonator zijn twee zeer goed reflecterende spiegels tegenover elkaar. Als een atoom geëxciteerd<sup>1</sup> wordt, voordat die in de optische resonator komt, kan het foton dat het atoom daarna weer uitzendt, in de resonator komen. Dat foton gaat dan heen en weer.<sup>2</sup> De kans dat het atoom een foton met dezelfde golflengte als de resonator uitzendt, is groter dan de kans dat het atoom een foton van die golflengte uitzendt in vacuüm. Dit is het Purcelleffect. Als dit foton uit de resonator ontsnapt (omdat de spiegels goed zijn, maar niet perfect), is er een enkel foton gegenereerd. Als je weet aan welke kant van de resonator dit foton ontsnapt is, kun je dit foton opvangen en gebruiken. Dit was de korte samenvatting, nu volgt de lange:

## Laserkoeling

Laten we beginnen met de lasergekoelde atomen. Een wolkje atomen wordt beschoten met lasers uit verschillende richtingen. De atomen worden eerst gekoeld met behulp van Dopplerkoeling. Dopplerkoeling maakt gebruik van



Figuur 1 Dopplereffect

zowel deeltjeseigenschappen van licht als van golfeigenschappen van licht. Deze manier van

<sup>1</sup>Het atoom krijgt er energie bij door een foton te absorberen.

<sup>2</sup>Het foton komt eigenlijk in een kwantumtoestand over de hele resonator.

het afkoelen van atomen maakt, zoals de naam al suggereert, gebruik van het Dopplereffect. Dit effect kan optreden omdat licht een golf is. Net als bij geluid, heeft ook licht een Dopplereffect: als een atoom richting de lichtbron beweegt, lijkt voor het atoom de frequentie van het licht hoger en als het atoom van de lichtbron af beweegt, lijkt de frequentie lager.<sup>3</sup> Een atoom in het wolkje atomen, ziet dus andere golflengtes van de lasers, afhankelijk van of het ernaartoe of er vanaf beweegt.

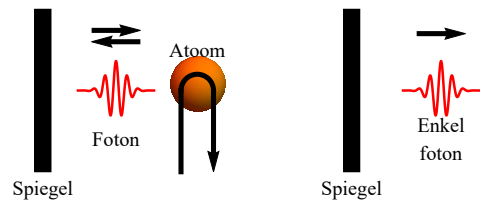
Atomen zijn soms kieskeurig wat betreft het absorberen van licht van verschillende golflengtes. Stel dat het atoom uit Figuur 1 alleen het blauwe licht wil absorberen. In dat geval zal het alleen een foton absorberen, als deze richting de laser beweegt. Als het atoom ten opzichte van de laser stil staat of er vanaf beweegt, absorbeert het atoom het foton niet.

Fotonen zijn ook deeltjes en het absorberen en uitzenden van fotonen draagt ook impuls over. Een foton heeft, ondanks dat het een massaloos deeltje is, een impuls gelijk aan  $h/\lambda$ . Als een rubidiumatoom dezelfde impuls moet hebben als een foton van 500 nm, zou het een temperatuur moeten hebben van  $0.3 \mu\text{K}$ . Als het atoom dus een foton absorbeert dat tegen zijn eigen bewegingsrichting in gaat, zal het afremmen. Daarna moet het atoom deze energie weer kwijt, maar het uitzenden van een foton kan in iedere richting gebeuren: richting de laser, van de laser af of opzij. Er is dus netto een impulsoverdracht door de laser tegen de bewegingsrichting in: de atomen worden afgekoeld. Omdat een bijna stilstaand atoom weer een zetje krijgt door het opnieuw uitzenden van een foton, kunnen met deze techniek niet arbitrair lage temperaturen worden bereikt. Om de laserkoeling goed te laten werken, kan het absorptiespectrum worden beïnvloed door een magnetisch veld. Dit magnetische veld samen met de lasers maakt dit tot een magneto-optische val.

Als de atomen afgekoeld zijn, kunnen de lasers aan de onderkant van het wolkje rubidium iets blauwer worden gemaakt. De atomen kunnen dit blauwe laserlicht beter absorberen en gaan omhoog. Een

koud atoom dat op deze manier omhooggeschoten wordt, volgt door de zwaartekracht dezelfde paraboolbaan als omhoog gegooid bakstenen, maar dan op atoomschaal. De lasers kunnen zodanig worden ingesteld dat er bijna altijd slechts één atoom omhoog gaat<sup>4</sup>. De atomen hebben als hoogste punt in hun val de optische resonator. Het omhooggooien gebeurt dus met zeer lage snelheid. De tijd die het atoom doorbrengt in de resonator is heel groot vergeleken met de tijdschalen die meestal voor atomen gelden. Daarom wordt het atoom ook afgekoeld: als het atoom een temperatuur van 298 K zou hebben, zou het atoom een gemiddelde snelheid hebben van 295 m/s. De optische resonator is  $40 \mu\text{m}$  breed. Het rubidiumatoom van 298 K zou de resonator passeren in  $0,14 \mu\text{s}$ , terwijl Nisbet-Jones *et al.* een interactietijd van enkele honderden microseconden verkregen hebben.<sup>5</sup>

## Optische resonator



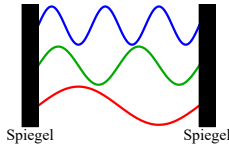
**Figuur 2** Schets van de optische resonator

Voor het atoom omhoog gegooid wordt in de resonator, wordt het geëxciteerd naar een hogere energietoestand. Als het atoom in de resonator is, wil het weer een foton uitzenden dat precies in de resonator past. De kans dat het atoom dit foton uitzendt, is zelfs groter dan in vacuüm door het Purcelleffect. Het Purcelleffect wordt veroorzaakt door het verschil in toestandsdichtheid tussen een atoom in vacuüm en het atoom tussen twee spiegels (zie Figuur 3 en 4).

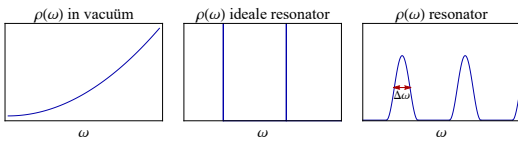
<sup>3</sup>Wat is rood en slecht voor je tanden? Een baksteen. Wat is blauw en nog slechter voor je tanden? Een heel erg snelle baksteen.

<sup>4</sup>Vraag mij niet hoe: dit lijkt mij namelijk vrij ingewikkeld om goed te kalibreren.

<sup>5</sup>Als het atoom door de laserkoeling volledig stil is komen te staan (wat niet kan, omdat het atoom altijd een kleine snelheid zal blijven hebben) en als het omhoog gooien van het atoom precies goed uitgelijnd gebeurt, zou er een interactietijd van 4 ms mogelijk zijn.



**Figuur 3** In een ideale optische resonator passen alleen fotonen waarvan de waaraan een geheel aantal golflengten in de resonator passen. In een iets minder ideale resonator kunnen de golflengtes ook iets langer of korter zijn. In de derde grafiek van Figuur 4 wordt daardoor de toestandsdichtheid breder.



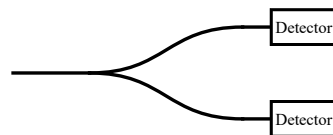
**Figuur 4** Het aantal toestanden per volume-eenheid in vacuüm is  $\rho(\omega) = \frac{\omega^2}{\pi^2 c^3}$ . Hierin is  $\omega$  de hoekfrequentie van het licht en  $c$  de lichtsnelheid. De toestandsdichtheid is een maat voor het aantal manieren waarop een foton van een bepaalde frequentie (de toestandsdichtheid is aantal toestanden per frequentie per volume) in de resonator past. Dit is sterk gepiekt rond de frequenties waarin de resonator resonanceert. In niet-ideale resonators is de piek breder en lager. Hoe beter de spiegels in de resonator kunnen reflecteren, hoe hoger de piek is in de toestandsdichtheid. De breedte van de piek hangt af van de kwaliteitsfactor  $Q = \frac{\omega}{\Delta\omega}$ . Hierin is  $\Delta\omega$  de breedte van de piek. De grootte van de Purcellfactor is evenredig met deze factor  $Q$ . De kwaliteitsfactor wordt hoger, als de spiegels beter reflecteren. Een atoom dat een foton wil uitzenden dat niet in de resonator past, heeft juist een kleinere kans om dit foton uit te zenden: de kans dat een atoom een foton uitzendt, hangt dus af van de omgeving.

Als het atoom een grote kans heeft om een foton uit te zenden, kan het atoom het ook weer absorberen. Voor kwantummechanische begrippen blijft het atoom bijna een eeuwigheid in de resonator. In die tijd kan het atoom een foton uitzenden en absorberen. Omdat dit gebeurt zonder dat mensen (of iets anders) kunnen kijken, bestaat in de optische resonator een kwantummechanische toestand tussen een geabsorbeerd foton en een uitgezonden

foton. Men zegt dat er sprake is van een superpositie tussen deze toestanden.

Ondanks dat de spiegels zo spiegelen mogelijk worden gemaakt, is er altijd een kleine kans dat ze licht doorlaten. Omdat het atoom voor een vrij lange tijd in de resonator blijft zitten, is de kans dat dit ooit gebeurt vrij groot. Als één van de spiegels meer licht kan doorlaten dan de andere, is de kans ook groter dat het foton besluit om via die spiegel de resonator te verlaten: we hebben een enkel foton, zonder dat er meerdere foton bij zijn vrijgekomen. Als het foton eenmaal de resonator heeft verlaten, komt er ook geen tweede, omdat het atoom door het foton uit te zenden geen energie meer heeft voor een tweede. Het atoom zal weer naar beneden vallen. Het is ook redelijk deterministisch wanneer er een foton uit de opstelling vrij komt. Dat is namelijk een tijdje na het lanceren van het geëxciteerde atoom in de resonator.

**Werkt het?**



**Figuur 5** Het foton heeft 50% kans om gedetecteerd te worden door de bovenste detector en 50% kans om gedetecteerd te worden door de onderste detector.

Om te testen of deze opstelling daadwerkelijk enkele fotonen genereert, kan het vrijgekomen foton in een glasvezelkabel worden geleid. Als het licht wordt gesplitst in twee andere kabels die naar twee verschillende detectoren worden geleid, zullen de detectoren steeds fotonen detecteren van verschillende atomen die in de resonator zijn gekomen, maar zullen ze niet tegelijkertijd een foton meten. Het foton kan zich immers niet splitsen. Op deze manier hebben we laten zien dat deze opstelling inderdaad enkele fotonen kan genereren.

[1] P.B.R. Nisbet-Jones, J. Dille, D. Ljunggren and A. Kuhn, Highly efficient source for indistinguishable single photons of controlled shape, *New Journal of Physics*, 13(10):103036, 2011

# Hoax

Hessel Withagen

Wanneer tegenwoordig het woord “hoax” gebruikt wordt, zal dit vooral zijn in de context van nepnieuws, wat eigenlijk maar een klein gedeelte van de verschillende soorten hoaxes beschrijft. Tegenover al het nepnieuws staan namelijk verschillende soorten (positievere) hoaxes. Deze kunnen een doel hebben, zoals het in de maling nemen van het publiek – bijvoorbeeld de spaghettiboomhoax uit 1957. Ze kunnen ook nadruk leggen op bepaalde problemen en hiermee een positieve verandering teweeg brengen, zoals de Grote Donorshow, die leidde tot duizenden nieuwe orgaandonoren. Ook zijn er combinaties van beiden te vinden, zoals het idee dat in de Amerikaanse staat Alabama de waarde van  $\pi$  naar de meer “Bijbelse waarde” van 3,0 zou worden veranderd.



**Figuur 1** *Goed nieuws! We hoeven nooit meer honger te lijden dankzij deze spaghettiboom!*

Dan nu de grote vraag: wat is een hoax precies? En hoe komt deze tot stand? Simpel gezegd is een hoax een onwaar bericht dat bewust als waarheid wordt afgespiegeld. Verschillende vormen van hoaxes zijn bijvoorbeeld het eerder genoemde nepnieuws, maar ook dingen als gefabriceerd wetenschappelijk onderzoek, graancirkels, broodje-aapverhalen en zelfs phishingmails die je proberen te overtuigen informatie af te geven. Overigens is de definitie niet superstreng, aangezien er ook gevallen zijn waarin de waarheid wordt gesproken maar op zo'n manier wordt gebracht dat het alsnog het effect van een hoax kan hebben. Een goed voorbeeld van dit laatste is de het idee dat diwaterstofmono-oxide gevaarlijk zou zijn – dit idee heeft een aanhang, puur omdat mensen niet weten waar het over gaat.

Hoe hoaxes exact ontstaan, is lastig te bepalen, maar er zijn wel bepaalde kenmerken die vaak voorkomen bij succesvolle hoaxes. De eerste is dat het gaat om iets waar mensen erg gepassioneerd over zijn – denk hierbij aan godsdienst of politieke voorkeur – maar ook dingen die een grote invloed hebben op het dagelijks leven van mensen, zoals financiën of gezondheid. Ten tweede is het belangrijk voor

een succesvolle hoax dat deze niet te specifiek is; de oorspronkelijke hoax moet kort kunnen worden omschreven. Het is namelijk juist de bedoeling dat mensen zelf de gaatjes invullen. Wat ook kan helpen is het vinden van een persoon met een groot publiek, bijvoorbeeld een politicus of andere beroemdheid, die overtuigd wordt van de hoax. Hierdoor kan snel een groot publiek worden bereikt en neemt, in de ogen van het publiek, de betrouwbaarheid van het bericht toe. Als laatste is het vaak zo dat voor- en tegenstanders van een onderwerp door het bericht tegenover elkaar worden gezet, wat leidt tot veel debat en daarmee publiciteit.

Elk jaar op 1 april worden er wel nepberichten de wereld ingegooid, maar één van de eerste grote 1-aprilhoaxes werd in 1957 door de BBC uitgezonden. Het ging om een drie minuten lange reportage over het oogsten van spaghetti van de zogenaamde spaghettibomen. Aangezien spaghetti in het Verenigd Koninkrijk een vrij onbekend gerecht was, werden veel Britten in de maling genomen door de uitzending. De volgende dagen ontving de BBC honderden telefoontjes over hoe ze zelf spaghettibomen konden planten. De BBC antwoorde hierop: “Plaats een spaghetti-tokje in een blik tomatensaus en hoop op het beste”.

Soms gaan grappen echter iets te ver, zoals in 2010 toen de Georgische nieuwszender Imedi in een uitzending deed alsof het land was binnengevallen door de Russen. Tijdens de half uur durende uitzending werd gemeld dat Russische militairen met tanks het land binnengevallen waren. Ook werd gezegd dat doelwitten door heel Georgië werden gebombardeerd en dat de president Mikheil Saakashvili was omgekomen. De uitzending zorgde voor veel paniek door heel Georgië, waarbij veel mensen hun huizen ontvluchtten en zelfs delen van



het Georgische leger zich opstelden om de invasie te stoppen. Drie mensen kwamen zelfs om het leven door de uitzending: twee aan een hartaanval en een derde aan een beroerte.

Dan heb je nog de hoax die je het vaakst tegenkomt de laatste tijd: het nepnieuws. Natuurlijk komt dit vooral op Facebook en andere sociale media langs, maar de laatste tijd gebruiken steeds meer nieuwswebsites neppe *clickbait*titels bij artikelen of zuigen zelfs nieuwsverhalen uit hun duim. Dit wordt op zich al vrij lang gedaan door sensatiebladen en dergelijke, maar de laatste tijd beginnen steeds meer enigszins respectabele bladen deze techniek ook toe te passen. Dit is natuurlijk onvermijdelijk met het steeds populairder wordend internet waarin bedrijven (en dus ook nieuwssites) alles op alles moeten zetten om de aandacht te krijgen. Alsnog moet de invloed en daarmee het gevaar van dit nepnieuws niet worden onderschat. Zeker nu de effecten van nepnieuws steeds meer te voelen zijn in de echte wereld. Kijk bijvoorbeeld naar de Amerikaanse presidentiële verkiezingen van 2016 waarin het nepnieuws voor beide kanten een grote rol speelde, sommige zeggen zelfs dat nepnieuws het

uiteindelijke verschil heeft gemaakt voor de uitslag.

Gelukkig zijn er ook nog hoaxes die een positieve invloed hebben gehad op de samenleving. De meest bekende in Nederland is misschien wel de Grote Donorshow waarbij een terminaal zieke vrouw zogenaamd haar nier zou afstaan aan één van drie nierpatiënten. Dit geheel werd live uitgezonden en trok ruim 1,2 miljoen kijkers, die tijdens de show ook nog via sms hun mening konden geven over wie de nier het meest verdiende. Voor het uitzenden van de show was er veel opspraak over of dit wel zou moeten worden uitgezonden. Het CDA en de ChristenUnie stelden er zelfs kamervragen over. Zelfs buitenlandse media bekritiseerde het programma en noemde het smakeloos. Aan het einde van de show werd echter duidelijk dat het ging om een publiciteitsstunt om aandacht te vragen voor het tekort aan orgaandonoren in Nederland. De vrouw, Lisa, bleek een kerngezonde actrice te zijn. De drie nierpatiënten waren echter geen acteurs en hadden daadwerkelijk een donornier nodig. Na de uitzending hadden uiteindelijk ruim 12.000 mensen een donorformulier aangevraagd, waaronder ruim 7.000 nieuwe inschrijvingen.



**Figuur 2** Met zo'n flitsend logo wordt je programma hoe dan ook een succes, of je nu nepnieuws hebt of niet.

Hoaxen worden op deze manier veel gebruikt om aandacht op een bepaald probleem te vestigen: door met behulp van een overdrijving van de werkelijkheid te laten zien wat er fout wordt gedaan. Het eerdergenoemde idee dat in Alabama de waarde van  $\pi$  zou veranderen naar 3,0 is bijvoorbeeld bedacht door natuurkundige Mark Boslough. Hij probeerde hiermee aandacht te krijgen voor het feit dat de evolutietheorie op scholen niet meer als de waarheid mocht worden verkondigd. Een ander voorbeeld: de steeds weer opduikende diwaterstofmono-

oxidehoax waarmee duidelijk wordt gemaakt dat mensen vaak niet kritisch genoeg nadenken voordat ze dingen geloven.

Dat is ook de boodschap die ik met dit artikel aan de lezer wil meegeven. Als (mogelijk toekomstig) wetenschapper is het belangrijk altijd kritisch na te blijven denken en met een enigszins argwanend oog naar alle "feiten" die er via social media en dergelijke langskomen te kijken. Zeker in deze tijd, waarin de waarheid in veel nieuwsartikelen ver te zoeken is.

# Programmeerselveringewikkelderiseringen

Tim Baanen

Zoals ik een jaartje geleden al opmerkte<sup>1</sup>, is het eigenlijk best wel saai om steeds nette, leesbare en begrijpelijke code te schrijven. Om de hoeveelheid chaos in de wereld een beetje op te hogen, heb ik een paar mooie ingewikkelde manieren bij elkaar gezocht om iets heel simpels te schrijven.

## Onleesbare C

De eerste set trucjes zijn vooral van toepassing op de taal C. Ten eerste omdat die op zo'n beetje elk computersysteem wel geïmplementeerd is, maar voornamelijk omdat het al veertig jaar oud is en al die tijd een flinke hoeveelheid oude troep heeft meegezeuld. Alle kleine trucjes die vroeger nodig waren om het compilers makkelijk te maken, worden nog altijd toegelaten. De compilers van nu geven wel een hoop waarschuwingen om je niet in de waan te laten dat het in onze verlichte tijden nog acceptabel is onbegrijpelijke code te schrijven. Die waarschuwingen negeren we uiteraard compleet.

Laten we een saai C-programmaatje nemen als voorbeeld.

**Listing .1** Een gewoon (maar nutteloos) programmaatje.

```
#include "stdio.h"

int main() {
    // lees een getal in
    int invoer;
    do {
        printf("voer_een_getal_in>");
    } while (scanf("%d", &invoer) != 1);
    // herhaal tot precies 1
    // getal is gelezen

    // tel wat getallen bij elkaar op
    int uitvoer = 0;
    for (int teller = 0;
        teller <= invoer;
        teller++) {
        uitvoer += teller;
    }

    // schrijf het resultaat op
    printf("%d\n", uitvoer);
    return 0;
}
```

<sup>1</sup>Om precies te zijn, in Vakidoot 1516-6 "Golf".

## Weg met waarden

De eerste stap die we kunnen zetten is alle constanten uit hun context halen en alle variabelen onbegrijpelijke namen geven. Bovendien hoeft je niet *echt* de `#include` of `return` neer te zetten, dus weg ermee!

**Listing .2** Alle waarden weggesloopt.

```
int a; // invoer
char *b = "voer_een_getal_in>";
char *c = "%d";
char *d = "%d\n";
int e = 1;
int f = 0; // uitvoer
int g = 0; // teller

int main() {
    do {
        printf(b);
    } while (scanf(c, &a) != e);
    for (; g <= a; g++) {
        f += g;
    }
    printf(d, f);
}
```

## Weg met types

Nu gaan we wat mooie ouderwetse trucjes van C gebruiken: je mag types weglaten als je int bedoelt, pointer-\*'tjes horen bij de variabele, niet het type, en een pointer naar het ene type mag je ook geven als je functie een pointer naar het andere type nodig heeft.

**Listing .3** Vervang de declaraties hiermee.

```
a, *b = "voer_een_getal_in>",
*c = "%d", *d = "%d\n",
e = 1, f = 0, g = 0;
```

## Weg met for

We kunnen zoiets leesbaars als een for-loop makkelijk omzetten in iets wat volkomen onleesbaar is. Merk op dat we de waarde van `a` helemaal niet gebruiken in het verdere programma, dus als we, in plaats van een teller `g` bijhouden, gewoon steeds `a` een beetje kleiner maken tot het 0 bereikt? Omdat we de variabele `f` niet gebruiken, voordat we met `e` klaar zijn, zetten we `e` gewoon op de beginwaarde van `f` en halen we het gebruik van `f` weg.

**Listing 4** Vervang de main-functie hiermee.

```
int main() {
    do {
        printf(b);
    } while (scanf(c, &a) != e);
    --e;
    do {
        e += a;
    } while (--a);
    printf(d, e);
}
```

## Weg met woorden

Dit kan beter: er staan nog 4 volkomen begrijpelijke woorden in onze code. Laten we recursie gebruiken om dit weg te halen. Is de invoer geen getal? Start het hele programma gewoon opnieuw! Net zo pakken we de optelling aan. Bovendien hoeft je voor if-statements helemaal geen woorden te gebruiken, als je de `?:`-operator hebt.

**Listing 5** Vervang de main-functie hiermee.

```
f() {
    e += a;
    --a
    ? f()
    : printf(d, e);
}
main() {
    printf(b);
    scanf(c, &a) != d
    ? main()
    : (--e, f());
}
```

## Weg met puntkomma's

In van die verouderde C-achtige talen, zoals C, C++ en Javascript is de komma-operator een prachtig hulpmiddel voor geavanceerde onleesbaarheid. Wat `a, b` namelijk betekent, is iets als: "evalueer expressie `a`, gooi het resultaat weg, evalueer expressie `b` en geef het resultaat door". Bovendien zijn toewijzingen als `x = a` ook gewoon expressies, dus in principe hoeft je dus nooit meer een puntkomma tussen twee statements neer te zetten. Zoek bijvoorbeeld maar eens uit wat de volgende expressie geeft in Javascript:

**Listing 6** Een populaire puzzel voor Javascript-programmeurs: wat betekent deze code?

```
[1, 2, 3][1, 2]
```

De laatste truc die we inzetten is dat C overbodige argumenten voor een functie wel uitrekt, maar vervolgens weggooit. Het decrementen van `e` voordat we `f` aanroepen, kunnen we dus ook vrolijk in de aanroep stoppen. Ten slotte formatteren we de code nog even mooi in een rechthoekje, en ziehier: een berg onbegrijpelijke symbolen die hetzelfde doet als ons oorspronkelijke programma!

**Listing 7** Een onleesbare editie van ons programma.

```
a,*b="voer_ _en_ _"
"getal_ _in_ >_ _",*c=
"%d" ,*d="%d\n" ,e
=1;f(){e+=a,--a?
f():printf(d,e);
}main(){printf(b
),scanf(c,&a)!=e
?main():f(--e);}
```



## Puntvrije Haskell

Een vergelijkbare tak van sport in kringen van functionele programmeurs is om je functies zonder variabelen op te schrijven. Omdat plekken waar variabelen worden geïntroduceerd ook wel *punten* heten, is de Engelse term hiervoor *pointfree* of *pointless* programming. Je kan zelfs een commando `pointfree` installeren die automatisch alle variabelen uit je code sloop.

Dit is in de belangrijkste functionele taal, de  $\lambda$ -calculus, heel goed met de hand te doen. Deze taal bestaat alleen maar uit de  $\lambda$ -abstractie die een gegeven waarde invult in een expressie (aangegeven met  $\lambda x \rightarrow t$  waar  $x$  een variabele is en  $t$  een expressie), de waarde van een variabele uitlezen (aangegeven door gewoon de variabele  $x$  op te schrijven) en een functie toepassen op een waarde (aangegeven door twee termen naast elkaar te schrijven:  $(t_1 t_2)$ ). Je hebt ook maar drie functies nodig om alles te vertalen.

Stel namelijk dat we beginnen met  $\lambda x \rightarrow y$ , waar  $y$  een andere variabele is dan  $x$ . We hebben dus een functie nodig van type  $a \rightarrow b \rightarrow a$ , waar we  $y : a$  als eerste argument invullen. Deze functie kennen we al: het is gewoon *const*. Net zo is het geval dat  $y$  dezelfde variabele is als  $x$  heel makkelijk:  $\lambda x \rightarrow x$  is gewoon de identiteitsfunctie *id*.

Het geval van een functie toepassen,  $\lambda x \rightarrow (t_1 t_2)$ , is wat ingewikkelder. De oplossing hiervoor is om eerst in recursie te gaan om  $\lambda x \rightarrow t_1$  en  $\lambda x \rightarrow t_2$  apart puntvrij te maken, wat expressies  $t'_1$  en  $t'_2$  geeft. Nu hebben we dus twee expressies die allebei hetzelfde argument nodig hebben, en waarvan de eerste vervolgens de tweede als argument nodig heeft. De types zijn dus  $t'_1 : a \rightarrow b \rightarrow c$  en  $t'_2 : a \rightarrow b$ , en de functie die we op  $t'_1$  en  $t'_2$  toepassen, heeft type  $(a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$ . Deze functie zit in Haskell ingebouwd als *ap*.<sup>2</sup>

### Van $\lambda$ naar SKI

Het systeem wat je krijgt door de  $\lambda$ -calculus puntvrij te maken, bevat geen enkele  $\lambda$  meer, dus het zou raar zijn om het nog zo te noemen. In plaats daarvan wordt het de *SKI*-calculus, genoemd naar

de functies  $S : (a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$ ,  $K : a \rightarrow b \rightarrow a$  en  $I : a \rightarrow a$ .<sup>3</sup>

Omdat we doen alsof functies maar een argument nodig hebben, hoef je eigenlijk ook nooit meer sluithaakjes te gebruiken: ze staan toch maar om één argument heen. Daarom heeft iemand besloten om de functionele programmeertaal Unlambda te maken. Die bestaat uit *S*, *K*, *I*, een los openingshaakje en wat operaties om in- en uitvoer voor elkaar te krijgen.

**Listing .8** Deze berg letters schrijft 10 keer "Hello world!".

```
'
''si'k''s.H''s.e''s.l''s.l''s.o''s.
''s.w''s.o''s.r''s.l''s.d''s.!''sri
''si''si''si''si''si''si''si''si''ki
```

### Van SKI naar binair

Eigenlijk heb je ook helemaal geen *I* nodig, want  $I = \lambda x.x = \lambda z.(Kz)K = (SK)K$ . Bovendien kunnen we zelfs *S*, *K* vervangen met maar een slim gekozen functie,  $\iota = \lambda x.((xs)k)$ . Als je nog wat papier wilt verspillen, dan kun je nagaan dat  $K = \iota(\iota(\iota))$  en  $S = \iota(\iota(\iota(\iota)))$ . Dat betekent dus dat je al je favoriete berekeningen kunt opschrijven met behulp van maar een functie  $\iota$  en een boekhoudsymbooltje ```, wat natuurlijk leidt tot volkomen onbegrijpelijke programma's!

Door de  $\iota$  en de ``` op een slimme manier op te schrijven, kun je zelfs elke binaire string omzetten in een unieke uitdrukking met  $\iota$ 's en ```. De resulterende programmeertaal heet *Jot*, naar de kleinste letter in het Hebreeuws.

Helaas is dit daadwerkelijk nuttig: Jot geeft een uniek programma voor elk natuurlijk getal en vice versa, en het wordt dus gebruikt voor onderzoek naar vragen als "welk gedeelte van de eerste  $n$  programma's komt in een oneindige loop terecht?" Zo blijkt maar dat wetenschappelijk onderzoek en compleet onleesbaar hobbyisme griezelig dicht bij elkaar liggen.

<sup>2</sup>DOE-TIP: is het toeval dat  $b \rightarrow a \rightarrow b$  en  $(a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$  precies de types zijn van *pure* en `<*>` in de instance *Applicative* ( $a \rightarrow$ )? Hoe zit het dan met de identiteitsfunctie  $I : a \rightarrow a$ ?

<sup>3</sup> Overigens staat de *S* voor *versmelten*, *K* voor *konstant*, *I* voor *identiteit* (of beter nog, de Duitse woorden *Verschmelzung*, *Konstanz*, *Identität*).



# The Hero's Journey

Jim Vollebregt

Mijn favoriete genre binnen de literatuur is fantasy. Men denkt misschien dat het hele genre van clichés aan elkaar hangt – en die mensen hebben waarschijnlijk gelijk. Het is echter de manier waarop clichés ingezet kunnen worden die het genre in mijn ogen zeer gevarieerd maakt. In dit artikel wil ik een van de bekendste patronen in fantasy – en daarbuiten – bespreken: De reis van de held.

**DISCLAIMER:** Dit artikel is niet geheel spoiler-vrij.

## 1 Vertrek

De eerste stap in de reis van de held is meteen een van de populairste. Hier een paar voorbeelden van bekende verhalen die de opening van de reis van de held volgen: *Harry Potter*, *The Hobbit* en *The Lord of the Rings*, *Star Wars: A New Hope*, *Eragon* (van Christopher Paolini), *The Eye of the World* (van Robert Jordan) et cetera.

Aan het begin van ons verhaal treffen we onze hoofdpersoon (ik ga vanaf nu in vrouwelijke termen spreken over deze hoofdpersoon, aangezien er een tekort is aan vrouwelijke personages binnen de fantasy) in een voor haar bekende situatie. In dit stadium is er nog weinig heldhaftigs aan haar. We ontmoeten haar vrienden – en misschien een paar mensen die ze minder aardig vindt – en leren dat ze

eigenlijk best een saai leven heeft. Ook krijgen we het gevoel dat onze heldin in wording een sluimerend verlangen heeft meer van de wereld te zien en de voorspelbaarheid van haar leven te doorbreken. Al vrij snel krijgt de hoofdpersoon te horen dat ze moet vertrekken. Er zijn echter zaken die haar tegenhouden: een gevoel van verplichting aan haar ouders bijvoorbeeld. Dan vindt er een kritieke gebeurtenis plaats die een gedachteverandering in de hoofdpersoon teweeg brengt, zodat ze besluit toch haar vertrouwde leven te verlaten. Waarschijnlijk wordt ze hierbij vergezeld door enkele vrienden (of mensen die ze minder aardig vindt) en een soort mentorfiguur. Het is tijd voor onze hoofdpersoon om de rest van de wereld te ontdekken.

Nu denk je misschien dat alle fantasyverhalen zich

afspelen in een soort middeleeuwse setting waarin iedereen vreemde namen heeft zoals Radagast, Perkamentus of Obi-Wan Kenobi en waarin hele volkeren zoals elfen of wookies bestaan uit kopieën van Legolas of Chewbacca. Dit hoeft niet per se waar te zijn. Het fantasygenre geeft de auteur de ruimte werelden te bedenken met hun eigen weerpatronen, flora en fauna. De schrijver kan zelfs nadenken over de etymologie van de namen die hij zijn personages geeft. De nieuwe rassen die de auteur

verzint, komen pas echt tot leven als we meerdere individuen ontmoeten en ontdekken dat ze, hoewel ze tot hetzelfde volk behoren, toch heel anders denken en handelen. We komen erachter dat ze aparte culturele gebruiken hebben of een religie aanhangen met gewoontes die totaal niet lijken op wat we van onze eigen wereld gewend zijn. Een van de beste aspecten van fantasy is dat we deze wereld samen met de personages ontdekken.



Dit beeld uit *Star Wars: A New Hope* is bedoeld om de kijker het gevoel te geven dat iets in Luke Skywalker ernaar hunkert meer van het heelal te zien.

## 2 Inwijding

Terwijl onze heldin en haar reisgezelschap de (hopelijk) fantastische wijde wereld in trekken, wordt het duidelijk dat ze een belangrijke rol te spelen heeft in het verhaal dat zich ontvouwt. De tweede stap van de reis van de held is nogal veelomvattend. Elementen van deze stap zie je in bijna elk verhaal terug.

### Training en moeilijkheden

Onze heldin krijgt al vrij snel te maken met mensen/wezens die haar kwaad willen doen. Het is duidelijk dat ze training nodig heeft om haar taak te volbrengen. Een belangrijk figuur in deze stap is de mentor. In bekende verhalen gaat het hier vaak om een oudere man, al dan niet met punthoed. Deze mentor voorziet de heldin van de nodige kennis,

vaardigheden en voorwerpen om haar queeste tot een goed einde te brengen.

Een veelgebruikt punt van kritiek op het fantasygenre is dat er geen regels zijn en dat de auteur altijd magie kan gebruiken om de heldin uit een benarde situatie te redden. In veel moderne fantasy zijn er echter wél regels; ze zijn alleen anders dan we gewend zijn. Samen met onze heldin raken we vertrouwd met de regels van de magie in de fantasiewereld, zodat wanneer het erop aan komt, we met haar mee kunnen denken hoe deze succesvol in te zetten.

Terwijl de heldin de training ondergaat, maakt ze doorgaans ook een emotionele ontwikkeling door. Aan het begin van het verhaal is ze naïef of verlegen, maar langzamerhand groeit ze in haar

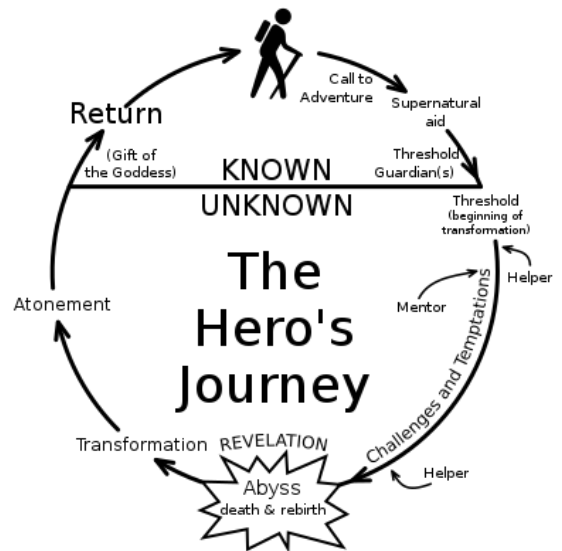
rol als protagonist en leert ze omgaan met haar verantwoordelijkheden. Ook maken we kennis met de rest van het gezelschap. Hier vinden we doorgaans een aantal verschillende types, zoals het *comic relief*-personage (Gimli, C3PO). Natuurlijk kan iemand die vertrouwd is met het concept van de reis van de held deze types op een verrassende manier inzetten. Zo kan het *comic relief*-personage een boeiende achtergrond gegeven worden die verklaart waarom hij alles met grappen maken probeert op te lossen. De Nickelodeon-serie *Avatar: The Last Airbender* doet dit uitstekend.

### Volbrengen van de queeste

Het grootste deel van het verhaal heeft onze heldin geworsteld met haar trainingen en haar identiteit. Dan vind er een moment van "apotheose" plaats, waardoor onze heldin tot nieuwe inzichten komt. Hiermee gewapend is ze er klaar voor om het doel van de queeste te bereiken. Meestal houdt dit in dat er een of andere grote vijand verslagen moet worden.

Bij de reis van de held wordt deze grote vijand meestal gekenmerkt door puur kwaadaardige trekken (Voldemort, Sauron). Om een meer realistische antagonist te creëren kan de schrijver er echter voor kiezen meer inzicht te geven in de denkwijze van de slechterik. Misschien gelooft deze wel dat hij doet wat het beste is voor de wereld. In een extreem geval kunnen we proberen de lezer/kijker na te laten denken of het niet de vijand was die eigenlijk goed bezig was. Dit is lastig, want mensen zijn geneigd zich verbonden te voelen met het personage dat als de held wordt gepresenteerd. Maar toch, als we Daenerys Targaryen op haar draak een heel leger in de as zien leggen, kunnen we ons afvragen hoeveel dat nu eigenlijk verschilt van het inzetten van nucleaire wapens. In zowel de *Mistborn*-serie als *The Stormlight Archive*-serie van Brandon Sanderson hebben we "slechteriken" die we sympathiek gaan vinden en "helden" die zich sterk afvragen of ze er wel goed aan doen de slechteriken te verslaan – deze series vertonen wel minder overeenkomsten met de reis van de held dan andere voorbeelden in

dit artikel.



**Figuur 1:** Een handige spreadsheet voor als je je eigen heldenreis wilt schrijven.

### 3 Terugkeer

Na een lange reis keert onze heldin terug naar de plek waar het allemaal begon. Terugkeren naar het dagelijks leven kan minstens zo moeilijk zijn als vertrekken. Onze heldin heeft hoogstwaarschijnlijk zowel fysiek als mentaal zware klappen te verduren gehad die haar voorgoed veranderd hebben. Past ze nog wel binnen de perken van haar oude leven? Waarschijnlijk heeft ze er net zo veel hulp bij nodig om opnieuw een normaal leven te leiden als in eerste instantie bij het vertrek.

Als Harry Potter aan het eind van het boek terugkeert naar zijn oom en tante dan doet hij dat met weemoed, al heeft hij veel moeten doorstaan tijdens zijn eerste jaar op Zweinstein. Ook Frodo heeft er moeite mee zich een vrolijke hobbit te voelen nadat hij de ring vernietigd heeft. Bij Frodo gaat het zelfs zo ver dat hij jaren na zijn reis toch besluit uit de Gouw te vertrekken. De littekens die hij bij zijn queeste heeft opgelopen zijn te diep om te negeren; de elfen geven hem een laatste geschenk en nemen hem mee op de onvindbare weg naar Valinor.

# Informatie om nooit te vergeten

Tim Baanen

Het is je misschien opgevallen dat sommige groepen mensen niet dezelfde taal spreken als jij. Ben je toevallig Simon Stevin, dan ga je dit probleem oplossen door iedereen ervan te overtuigen dat het Nederlands ongetwijfeld de beste taal is. Naast het duidelijke feit dat Adam (wiens naam oorspronkelijk "Haat-dam" was) en Eva ("Eed-vat") ook Nederlands spraken, breng je in dat het Nederlands veel eenlettergrepige woorden heeft. Dat is veel efficiënter dan al die ouderwetse<sup>1</sup> Griekse en Latijnse woorden.

Stel dus even dat je Simon Stevin bent, en bovendien dat je Simon Stevin uit de toekomst bent. Je hebt toegang tot allerlei moderne wiskundige snufjes als differentiaalrekening, logaritmen en computers. Nu hebben ze in de toekomst niet zo'n hoge pet op van dat hele "Haatdam en Eedvat"-verhaal, dus je zal het toch echt op efficiëntie moeten gooien. Is er niet een mooie wiskundige manier om efficiëntie van talen te meten?

Terwijl je nadenkt over dit raadsel, gaan we door naar een andere vraag: hoe onmogelijk het is om de Nederlandstalige dependance van de Bibliotheek van Babel te openen. De originele Bibliotheek, aangelegd door een oneindige rij van Jorges Luises Borges, bevat boeken van 410 pagina's, met 40 regels op elke pagina, en 80 karakters op elke regel. De bibliotheek bevat zelfs elk boek dat aan deze omschrijving voldoet, dus ook een boek met de tekst van deze Vakidoot, een boek met je hele levensverhaal inclusief alles wat je in de toekomst te wachten staat, en een boek met je hele levensverhaal dat volkomen onzin wordt precies vanaf wanneer er staat dat je het boek begint te lezen. Voornamelijk bevat de bibliotheek helaas volslagen onzin, aangezien het echt gaat om *elk* mogelijk boek. Gelukkig staat er ook een oneindige rij Doctorandussen P's die de boeken hebben gesorteerd op begrijpelijkheid in het Nederlands, en aan jou de taak om eventjes een gebouw te maken waar alle Nederlandstalige boeken ondergebracht kunnen worden. Hoe belachelijk veel boekenplanken zou je hiervoor wel niet nodig hebben?

<sup>1</sup>Of dus juist nieuwerwetse?

<sup>2</sup>Nou ja, de jaren '50 waren in de toekomst voor Simon Stevin en Jorge Luis Borges.

<sup>3</sup>**DOE-TIP** voor natuurkundigen: kun je dit uitleggen in termen van macro- en microtoestanden? Zo ja, wat is de macrotoestand die hoort bij de informatie in dit artikel?



**Figuur 1** Een oneindige rij Doctorandussen P's bedenkt of "Doctorandussen" of "Doctorandi" betere woordspelingen mogelijk maakt.

Dit is een erg lastige vraag, dus je gaat liever naar Spanje om daar te *relajar* op de *playa*'s. Het wordt je op deze *playa* helaas nogal lastig gemaakt om adequaat te *relajar* want Spanjaarden spreken alles op zo'n hoog tempo uit dat je weer in de stress schiet. Je roept je taalkundevriendjes bij elkaar, en die stellen vast dat het Spaans met 7,8 lettergrepen per seconde aanzienlijk rapper is dan het Nederlands met 5,1 lettergrepen per seconde. Misschien is het toch beter om met zijn allen naar het Spaans over te schakelen, zelfs als ze iets meer lettergrepen gebruiken?

## Informatie, entropie en zipbestanden

Al deze vragen zou je kunnen beantwoorden met een krachtig hulpmiddel uit de toekomst.<sup>2</sup> Het enige wat je nodig hebt, is namelijk precies weten hoeveel informatie er in het Nederlands zit. Omdat het basale idee van een taal is om een ware uitspraak te doen, bedoelen we met *informatie* dat je uit een setje mogelijke werkelijkheden een bepaalde



werkelijkheid kan aanwijzen. Het idee “mogelijke werkelijkheden” is ongeveer dezelfde als bij de definitie van een kans: als je een muntje opgooit, zijn er twee “mogelijke werkelijkheden”, en als je zegt “De uitkomst is kop,” wijs je daar één van aan.<sup>3</sup>

In plaats van informatie is het vaak wat handiger om na te denken over de *entropie*: het aantal bitjes wat je nodig hebt om de keuze uit alle mogelijkheden op zijn efficiëntst op te schrijven. Een muntje opgooien geeft  $2 = 2^1$  uitkomsten, die je dus in 1 bitje opschrijven. Een uniforme keuze uit de getallen  $1, \dots, n$  heeft  $\log_2(n)$  bitjes nodig. De efficiëntie van een (geschreven) taal is dan recht evenredig aan de hoeveelheid entropie per letter.

Het idee van bestanden zippen op de computer is om bestanden zo efficiënt mogelijk op te schrijven. Als we ervan uitgaan dat ze redelijk succesvol zijn in dit streven, kun je uit de gezipte afmeting van een bestand afleiden hoeveel entropie in een bestand zit. Neem bijvoorbeeld de broncode van alle artikels van Vakidoot 1718-1 “Overschot”, die gezippt uitkomt op 430288 bitjes. Omdat 181767 karakters in totaal in deze artikels gebruikt zijn, komen we op een entropiedichtheid van ongeveer 2,4 bits per karakter en 17 bits per woord: elk karakter in de Vakidoot had ook een van 4 andere mogelijke karakters kunnen zijn, en elk woord hebben de auteurs zorgvuldig gekozen uit wel 130 duizend alternatieven!

## Aanvullingen gokken

Bij het uitvinden van de informatietheorie heeft Claude Shannon ook gezocht naar een goede manier om de entropiedichtheid van talen vast te stellen. Omdat het jaar 1948 nog niet de toekomst was, had hij nog geen zipprogramma’s om de overbodige bitjes uit bestanden te persen. In plaats daarvan knipte hij een stuk uit een zin en vroeg hij mensen om te bedenken wat er hoorde te komen.

Een zin met een gat als “Help, mijn huis staat in de \_” zou je bijvoorbeeld kunnen aanvullen met “fik”, “weg” of “krant, omdat het lijkt op Hitler”, maar veel meer waarschijnlijke alternatieven zijn er niet. Het ontbrekende stuk van de zin zou dus een paar bitjes aan informatie bevatten. Kap je de zin al af op “Help, mijn \_”, dan zijn er een stuk meer mogelijkheden, en het verschil in bitjes is precies de entropie van het stukje “huis staat in de”.



**Figuur 2** Een huis dat lijkt op Hitler, als Hitler een tweede paar ogen ergens onderaan zijn wangen had.

Op basis van dit soort tests concludeerde Shannon dat geschreven Engels ongeveer 1,0 tot 1,2 bitjes per letter bevat. We hebben zojuist op een volkomen andere manier vastgesteld dat de Vakidoot wel 2,4 bits per letter bevat, dus als je allerlei problemen met methodiek kan negeren, is geschreven Nederlands is daadwerkelijk ongeveer twee keer zo efficiënt als Engels. Ik vermoed dat we dit kunnen verklaren door naar de spelling te kijken. Spellingen die een paar jaar terug nog zijn bijgesteld om goed bij de taal te passen, zullen doorgaans gewoon beter functioneren dan eentje geïmporteerd uit een andere taal die sinds de Middeleeuwen amper aangepast is.



**Figuur 3** De geconstrueerde taal Ithkuil is geconstrueerd voor de hoogst mogelijke informatiedichtheid, en het bijbehorende schrift is nog meer geoptimaliseerd. Hier staat de zin “Tram-mļōi hhâsmaŗp̄uktōx,”, wat iets betekent als “Ik ben daarentegen van mening dat het zou kunnen zijn dat deze ruige bergketen op een zeker moment minder steil wordt.” Overigens is er niemand op aarde die de taal vloeiend kan spreken.

## Boeken tellen

Nu we het eerste probleem hebben opgelost, gaan we naar de Bibliotheek van Babel. Het probleem met natuurlijke talen is dat ze allemaal een manier hebben om ongrammaticale tekst in een grammaticale zin te krijgen. Neem bijvoorbeeld de zin “Het woord zikbdqwrst is geen bestaand woord.”, die grammaticaal helemaal correct is maar toch niet echt voelt als een zinnige Nederlandse zin.

Er is bijvoorbeeld geen echt verschil tussen deze zin en “Het woord zldvntkre is geen bestaand woord.”, zelfs bevatten ze echt andere letters. De Doctorandussen P’s zullen boeken met dit soort onzinnen niet accepteren om hun taalgevoel niet te kwetsen.

Om deze reden gaan we alleen boeken tellen die in een zekere zin informatie bevatten. Als je de informatiedichtheid van het Nederlands weet, kun je volgens deze definitie ook redelijk goed bepalen hoeveel verschillende boeken er kunnen bestaan. Bepaal namelijk de totale hoeveelheid informatie die het Nederlands in 410 bladzijden van 40 regels van 80 karakters kan passen, dan geeft dat een heel aardige schatting van het totaal aantal mogelijke boeken.

Een snelle berekening leert ons dat er ongeveer  $410 \cdot 40 \cdot 80 \cdot 2,4 = 3\,148\,800$  bits aan entropie in een Nederlandstalig boek passen, wat betekent dat je in de orde van  $2^{3\,148\,800} \approx 10^{947883} \approx 10^{10^6}$  unieke Nederlandstalige boeken kan schrijven. Je hebt dus alleen al een bijna vol boek nodig om op te schrijven hoeveel boeken er zijn. Het aantal boekenplanken is misschien een factor 30 minder, maar dat scheelt dus ongeveer anderhalf cijfer op een miljoen.

### Snelheden vergelijken

Nu de laatste vraag: welke invloed heeft de praat-snelheid van Spanjaarden op de informatiesnelheid? We moeten nu meten wat de entropie per tijd is van gesproken talen. Daarvoor zou je het liefst net zoiets als Shannon doen, door opnamen van gesproken zinnen in stukken te hakken en te vragen wat er op volgt, maar helaas kan ik geen taalkundig onderzoek vinden dat iets dergelijks doet.<sup>4</sup>

Er zijn wel wat onderzoeken die een voorbeeldtekst te nemen, die naar andere talen te vertalen, en vergelijken hoe snel de teksten in de verschillende talen worden uitgesproken. Bij vertalen moet je alleen sowieso extra informatie toevoegen omdat talen uitgaan van gedeelde kennis: Nederlanders

weten wat “gezellig” precies inhoudt, maar een vertaling naar het Engels moet het concept uitleggen. Een van de onderzoeken, *A cross-language perspective on speech information rate* van Pellegrino, Coupé en Marsico, geeft de volgende tabel:

Engels	1.08	(± 0.08)
Frans	0.99	(± 0.09)
Duits	0.90	(± 0.07)
Italiaans	0.96	(± 0.10)
Japans	0.74	(± 0.06)
Mandarijn	0.94	(± 0.08)
Spaans	0.98	(± 0.07)
Vietnamees	1	

*De snelheid van talen ten opzichte van het Vietnamees.*

Hieruit kunnen we concluderen dat de informatiedichtheid van gesproken talen vrijwel identiek is, op het Japans na. Die mate van gelijkheid kun je best goed verklaren op taalkundige gronden: een taal is een gebruiksvoorwerp, dus je wil het ook zo efficiënt mogelijk maken. Veelgebruikte woorden kun je afkorten, en ingewikkelde uitdrukkingen kun je vervangen met een pakkender alternatief. Praat je echt veel te snel, dan kan je toehoorder niet meer begrijpen wat je allemaal zegt.

Twee belangrijke factoren die informatiedichtheid beïnvloeden is hoe ingewikkeld een lettergreep kan worden, en hoe snel je lettergrepen kan uitspreken. Ingewikkelde lettergrepen kunnen veel informatie encoderen, maar zijn lastig om snel uit te spreken. Andersom zorgt snel uitspreken ervoor dat je lettergrepen vanzelf simpeler slijten. Het Spaans heeft best makkelijke lettergrepen, en wordt heel snel uitgesproken. Daarentegen heeft het Mandarijn best ingewikkelde lettergrepen, bijvoorbeeld omdat ze tonen gebruiken. Als talen ongeveer dezelfde processen ondergaan, zullen ze snel in vergelijkbare optima terecht komen en hun eigen informatiedichtheden ook snel gelijk worden. Spaans is niet echt veel efficiënter: het zit in een andersoortig optimum.

Je kan met informatietheorie nog veel meer lol beleven, zoals bepalen of het luidruchtig is in de Vagant omdat het galmt, of omdat er zoveel gesprekken zijn dat de communicatiekanalen overbelast worden, of berekenen of een gegeven matige fotoshop daadwerkelijk meer zegt dan duizend woorden. Daarom raad ik iedereen met interesse in wiskunde en taalkunde aan om wat meer over informatietheorie te weten te komen!

<sup>4</sup>DOE-TIP: voer zo’n onderzoek uit en krijg het gepubliceerd.

# Kans op herkansing

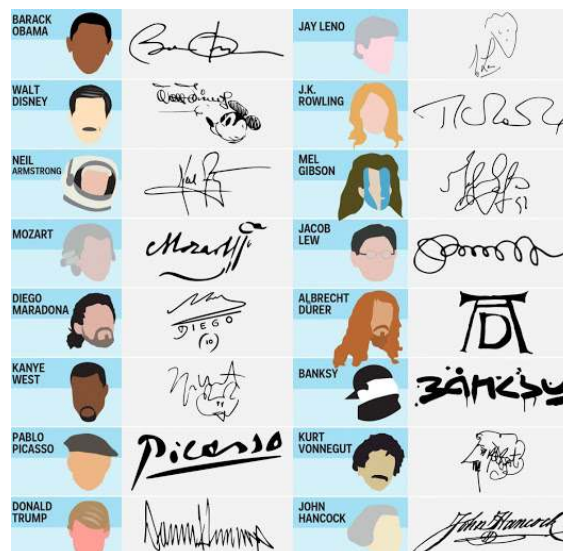
Maud Nabben & Nils Warsen

Heb jij ook wel eens gehad dat de vragen op je tentamen niet lukten omdat de tijdens het vak geleerde inzichten net wat meer tijd nodig hadden? Of dat je je niet lekker voelde tijdens het tentamen, of gewoon een slechte dag? Je krijgt een paar dagen later je cijfer terug en ziet ineens dat je een 3 hebt gehaald. Balen natuurlijk, want nu mag je niet eens een herkansing maken! Het SODI, SONS en WOL vinden dat dit niet langer zo kan en willen een einde maken aan de huidige herkansingsregeling waarbij je alleen cijfers tussen een 4 en een 5,5 mag herkansen.

Concreet betekent dit dat we in het OER een uitzondering willen krijgen op de herkansingsregeling voor de studies wis- en natuurkunde, informatica en informatiekunde. Om dit voor elkaar te krijgen moeten we laten zien dat er veel onvrede is over de huidige regeling. Daarom vragen we aan alle studenten om de eerste twee weken van blok 2 hun handtekening te zetten om onze petitie te steunen.

Er zullen genoeg mogelijkheden zijn waarop je de petitie kan ondertekenen. Zo zullen we langs alle colleges gaan, een kamerovername hebben, een keer op de overloop tussen het BBG en KBG staan en op een borrel rondlopen om zoveel mogelijk handtekeningen te verzamelen. Ook zal er een lijst in de kamers van A-Eskwadraat en Sticky liggen.

Teken de petitie en maak kans op een herkansing!



## HERKANSEN

**SOMS MOET JE  
IETS NOG EEN  
KEER DOEN**

**VOOR JE  
SUCCESVOL KUNT  
ZIJN**

*Loesje*

[www.loesje.nl](http://www.loesje.nl)

[alkmaar@loesje.nl](mailto:alkmaar@loesje.nl)



## Update - Vreemde YouTubestatistieken

Jim Vollebregt

Videosite YouTube is een van de grootste en populairste op het internet. In Vakidioot 15/16-4 String gaf ik al een overzicht van enkele opvallende statistieken van deze site. Echter, de site groeit zo snel dat er inmiddels een hoop dingen veranderd zijn. Laten we eens kijken naar de meest opvallende nieuwe records.

Op 10 juli 2017, rond 23:30 Nederlandse tijd, werd het einde van een tijdperk ingeluid. De Gangnam Style Music Video van de Zuid-Koreaanse PSY werd

na vijf jaar ingehaald als de meest bekeken video op YouTube. De nieuwe nummer één: See You Again van Wiz Khalifa en Charlie Puth. Het record bleef echter maar een maand van kracht. Op 4 augustus 2017 werd ze opnieuw verbroken door Despacito van Luis Fonsi en Daddy Yankee (foto bovenaan de pagina), die daarmee de eerste video ooit werd met meer dan 3 miljard views. Vanaf het moment dat deze video werd geüpload eerder dit jaar, werd hij gemiddeld zo'n 230 keer *per seconde* bekeken.

Overzicht van de meest bekeken video's op YouTube

Titel	Aantal keer bekeken	datum van uploaden
Luis Fonsi - Despacito ft. Daddy Yankee	4.211.750.911	12 januari 2017
Wiz Khalifa - See You Again ft. Charlie Puth...	3.192.449.748	6 april 2015
PSY - GANGNAM STYLE...	2.983.064.516	15 juli 2012
Justin Bieber - Sorry (PURPOSE: The Movement)	2.810.616.385	22 oktober 2015
Mark Ronson - Uptown Funk ft. Bruno Mars	2.742.850.611	19 november 2014
Ed Sheeran - Shape of You...	2.629.087.540	30 januari 2017
Маша и Медведь (Masha and The Bear)...	2.576.113.695	31 januari 2012
Taylor Swift - Shake It Off	2.426.397.306	18 augustus 2014
Enrique Iglesias - Bailando (Español)...	2.409.069.574	11 april 2014

Op 27 augustus 2017, tijdens de MTV Video Music Awards, liet Taylor Swift haar video voor Look What You Made Me Do in première gaan. Katy Perry – de presentatrice van het evenement – was tijdens het vertonen van de clip nergens te bekennen. Dat was maar goed ook, want de video bevat blijkbaar een hoop referenties naar de ruzie tussen de twee diva's (de Bad Blood vete?). Daarnaast zit de korte film vol verborgen beledigingen aan verschillende beroemde exen en sneren richting Kanye West en Kim Kardashian met wie ze ook al enige tijd een vete onderhoudt. Dit alles hielp de video om in de eerste 24 uur na het uploaden een duizelingwekkende 43,2 miljoen kijkers te trekken op YouTube – een record.



*Hoeverveel bekende exen had Taylor Swift ook alweer?*

Hoewel populaire artiesten het goed doen op YouTube, zijn er toch enkele records die voor andere kanalen zijn weggelegd. Zo heeft PewDiePie de meeste abonnees: maar liefst 57.333.693. Zijn kanaal is dan ook zeer interessant. Ik quote hier even de volledige beschrijving van het kanaal:

“ I make videos. ”

Ook het meest bekeken kanaal is misschien een beetje onverwacht. Ik dacht zelf altijd dat het ging tussen PewDiePie en JustinBieberVEVO (beide ruim 16 miljard views), maar ze vallen in het niet bij T-Series. Dit kanaal bevat ruim elfduizend video's – voor zover ik kan zien vooral trailers voor Indiase films en clips. Sommige hiervan zijn tientallen miljoenen keren bekeken, en alles bij elkaar staat de teller op 23.626.196.437.

#### Enkele opvallende YouTuberecords:

- Justin Bieber - Baby ft. Ludacris heeft met 8.190.214 de meeste dislikes op YouTube. De nummer twee, Rebecca Black's Friday heeft er 2.890.706.
- CODblackopsPS heeft met 1.067.412 de meeste geüploade video's. Het kanaal is wel voor een vrij niche publiek.
- Adele - Hello werd in de kortste tijd 1 miljard keer bekeken. Dit lukte in slechts 88 dagen.
- Luis Fonsi - Despacito ft. Daddy Yankee werd in de kortste tijd 2 miljard keer bekeken. Dit lukte in slechts 154 dagen.
- De eerste video die meer dan 100 miljoen keer werd bekeken was Avril Lavigne - Girlfriend. Tegenwoordig is dat niet meer zo'n indrukwekkend aantal, maar een record is een record.

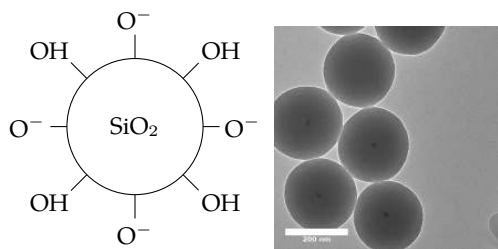
Op het moment van schrijven zijn er 78 video's met meer dan 1 miljard views. We hebben al gezien hoe vaak de nummer 1 bekeken is. Maar hoe vaak kan een video *theoretisch gezien* bekeken worden? In het pre-Gangnam Style-tijdperk moest het aantal views dat een video kon hebben passen in 32 bits. Dit komt overeen met zo'n 2 miljard. Toen Gangnam Style de miljard views passeerde, was het voor de beheerders van de site duidelijk dat er iets moest veranderen.<sup>1</sup> Tegenwoordig wordt er gewerkt met 64 bits getallen, met een maximum gelijk aan 9.223.372.036.854.775.808. Stel dat de hele wereldbevolking dezelfde video aan zou zetten, en deze vervolgens eens per seconde zou herladen, dan zou het ongeveer 39 jaar duren voordat dit plafond is bereikt. Heeft er al iemand een idee voor een video die dit voor elkaar gaat krijgen?

<sup>1</sup>Dit hele verhaal is mogelijkwerwijs een grap van de beheerders

# Als twee colloïden elkaar ontmoeten

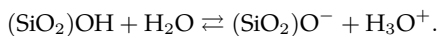
Peter Speets

Colloïden zijn microscopische deeltjes die ongeveer tussen een nanometer en een micrometer groot zijn. Deze deeltjes kunnen van alles zijn: fijnstof, pollen van planten of oliedruppels in water, zolang ze maar klein zijn. Door de kleine schaal hebben suspensies van colloïden soms opmerkelijke eigenschappen die macroscopische objecten niet hebben. Als twee colloïden dicht bij elkaar komen, spelen er verschillende processen een rol: interacties tussen de twee colloïden, interacties met het oplosmiddel en maximalisatie van de entropie. In dit artikel komt slechts een deel daarvan aan bod om de vraag te beantwoorden waarom de zoutconcentratie van de oplossing invloed heeft op de oplosbaarheid van de deeltjes.

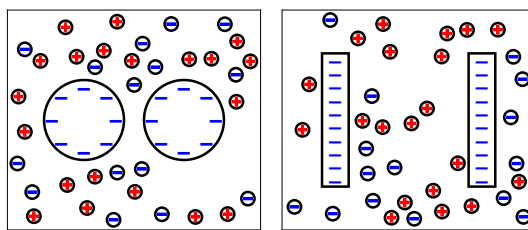


**Figuur 1** De OH-groepen aan het oppervlak van het deeltje splitsen  $H^+$ -ionen af. Hierdoor wordt het deeltje geladen en daalt de pH van de oplossing. Rechts staat een elektronenmicroscopieafbeelding<sup>1</sup> van  $SiO_2$ -colloïden [1]. De schaalbalk is 200 nm.

Een colloïde kan natuurlijk veel vormen hebben en van veel verschillende stoffen zijn gemaakt. Een voorbeeld van een colloïde is een bolletje van silica (siliciumdioxide). In Figuur 1 staat een schets van een silicacolloïde. Aan de rand van het deeltje zit zuurstof. Omdat zuurstof graag twee bindingen wil hebben,<sup>2</sup> zit op de rand van het deeltje ook waterstof en worden OH-groepen gevormd. Het waterstofatoom is slecht gebonden aan het zuurstofatoom. Dit betekent dat deze deeltjes zuur zijn: ze willen willen van hun  $H^+$  af. Als een deeltje is opgelost, gaat het deeltje een evenwichtsreactie aan met water op dezelfde manier als de zuur-base-reacties van de middelbare school:



Er vormt zich dus een evenwicht tussen het aantal OH-groepen op de colloïden en de hoeveelheid lading op de colloïde en de concentratie  $H_3O^+$ .



**Figuur 2** Links staan twee geladen colloïden dicht bij elkaar. Rechts staan twee geladen platen tegenover elkaar. In dit artikel worden de twee colloïden in het linkerplaatje benaderd door de geladen platen in het rechterplaatje.

Door dit evenwicht is het oplosmiddel netto positief geladen en het deeltje netto negatief geladen. Op het deeltje wordt op verschillende manieren kracht uitgeoefend. Twee colloïden trekken elkaar aan door de vanderwaalskracht, maar worden weer afgestoten door de elektrische kracht, doordat de twee colloïden dezelfde lading hebben. De negatieve en positieve ionen in de oplossing worden echter ook afgestoten en aangetrokken door de lading op de colloïden. Op hun beurt oefenen de ionen ook weer kracht uit op de colloïden en andere ionen in de oplossing.<sup>3</sup>

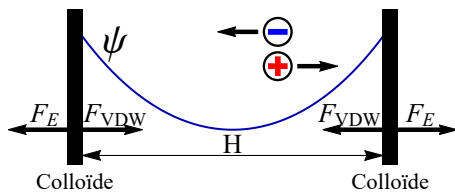
Omdat de positieve ionen het dichtst op de colloïde zitten en daardoor het elektrisch veld compenseren, neemt de elektrische potentiaal veroorzaakt door de colloïde exponentieel af met de afstand tot de colloïde. Als de positieve ionen in het oplosmiddel allemaal in een dunne laag aan het negatieve deeltje plakken, zou er helemaal geen elektrisch veld meer

<sup>1</sup>De deeltjes die in het rechterplaatje staan, zijn niet dezelfde deeltjes die in het linkerplaatje staan, maar lijken er alleen een beetje op.

<sup>2</sup>Om net zoveel elektronen als het dichtsbijzijnde edelgas te hebben, moet het twee elektronen erbij krijgen.

<sup>3</sup>Voor een enkele geladen plaat wordt de concentratie van de ionen in de oplossing gegeven door  $c_{\pm} = c_0 e^{\pm q\psi(r)/(k_B T)}$ . Hierin is  $c_{\pm}$  de dichtheid van de positieve en de negatieve ionen,  $c_0$  de ionendichtheid zonder colloïde en  $q$  de lading van het ion en  $k_B T$  de thermische energie.

zijn. Toch gebeurt dit niet, omdat er veel meer mogelijkheden zijn voor de ionen om vrij door het oplosmiddel heen te bewegen dan om aan een colloïde geplakt te zitten. Dit is de maximalisatie van de entropie. Een evenwicht tussen de vele mogelijkheden waarop ionen kunnen bewegen door het oplosmiddel en een lagere potentiaal wordt bereikt door een laag ionen rond de colloïde die nog door de oplossing bewegen, maar waarvan de concentratie afhangt van de elektrische potentiaal.



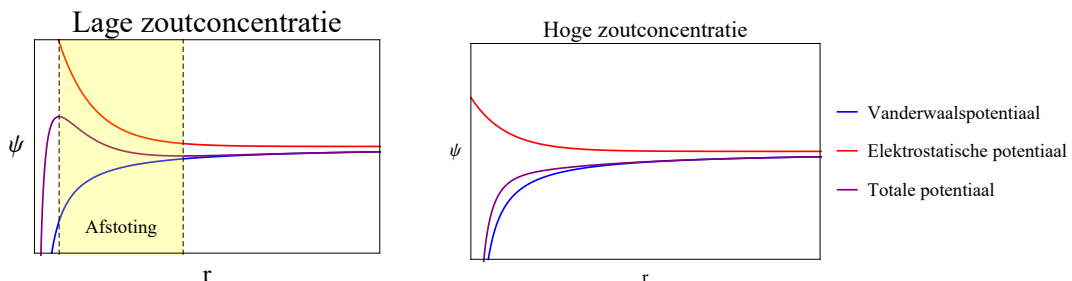
**Figuur 3** Twee colloïden bevinden zich op een afstand  $H$  van elkaar.  $\psi$  is de elektrische potentiaal tussen de platen. De colloïden worden naar elkaar toe getrokken door de vanderwaalskracht ( $F_{VDW}$ ) en worden afgestoten door de elektrostatische kracht ( $F_E$ ). Als de colloïden negatief geladen zijn, stoten ze de negatieve ionen af en trekken ze de positieve ionen aan. De potentiaal  $\psi$  is het resultaat van zowel de lading van de oplossing als de lading van de platen. Als de colloïden ver genoeg uit elkaar staan, heeft de potentiaal de vorm van een cosinus hyperbolicus.

In Figuur 3 is deze potentiaal geschetst van twee deeltjes die tegenover elkaar staan. De andere kracht die nog in het spel is, de vanderwaalskracht, laat de

colloïden aan elkaar plakken. Deze kracht wordt steeds sterker naarmate de colloïden dichter bij elkaar komen. Als de platen zwak geladen zijn, is ook op korte afstand deze kracht groter dan de elektrische kracht. De potentiaal van de vanderwaalskracht is omgekeerd evenredig met de afstand tot de colloïde. De elektrische potentiaal van de colloïden en de oplossing met ionen is evenredig met  $e^{-kr}$  waarin  $r$  de afstand is tot de colloïde en  $k$  een constante.

Deze aantrekking wordt normaal gesproken (voor dit  $\text{SiO}_2$ -deeltje) gecompenseerd door de elektrische kracht. Als er echter extra ionen aan de oplossing worden toegevoegd, bijvoorbeeld door er zout in op te lossen, wordt het elektrisch veld meer gecompenseerd. In Figuur 4 staan de potentiaal van de vanderwaalskracht en van de elektrostatische kracht voor een lage en voor een hoge zoutconcentratie geschetst. Doordat er geen afstotende laag meer rond de colloïde zit, zullen de colloïden aan elkaar kunnen plakken. De colloïden vormen dan clusters van aan elkaar geklitte deeltjes en zinken naar de bodem.

Dit is nog niet het hele verhaal. Als de colloïden dicht bij elkaar komen, zal de ionendichtheid tussen de twee colloïden toenemen. Dit betekent dat de beschikbare hoeveelheid  $\text{H}^+$  ook toeneemt. Dit zal de evenwichtsreactie die de colloïde een elektrische lading geeft, weer deels ongedaan maken. De lading op de colloïde neemt af, waarmee de afstotende werking ook minder wordt.



**Figuur 4** Bij een lage zoutconcentratie zijn er minder ionen in de oplossing om het elektrisch veld tussen de twee colloïden te compenseren. Als de colloïden op een bepaalde afstand van elkaar zijn, zullen ze elkaar afstoten. Als de colloïden dicht bij elkaar staan, of juist verder weg, trekken ze elkaar weer aan. In het rechterplaatje is de zoutconcentratie hoger, waardoor de drempel bij elkaar te komen verdwijnt.

[1] De microscoopafbeelding en de deeltjes zijn gemaakt door Marc del Pozo Puig van het Debye Instituut



## Een reis naar Andersomgolië

Tim Baanen

Onder een nacht was de vrolijke A-Eskwadrater Vakidioot op reis in Andersomgolië. De Vakidioot had de afgelopen dagen al de uitgestrekte Andersomgoolse tamijnen gezien, en deze middag zou gereserveerd zijn voor de ingestrekte autopedpen. Helaas was de voorspelde weersbui met het verstrijken van de tijd omgeslagen tot een achterspelde onweersbui. De Vakidioot nam die randdag dus teweinig met zoveel mogelijk binnen en droog te blijven. De hoofdstad Rataabnaalu had zeker geen onderschot aan taveernes waar Vakidioot met behulp van een aantal koele glazen kootr zich ledigvol kon bezatten.

Buiten meer van deze taveernes, een altbassen gesloop waarvan de naam precies ingesproken wordt als *Stoelvoet ned Tahaarne*, werd de Vakidioot aangesproken door iemand zonder de schrijvische kledij van Andersomgoolse broodboeren. “Fouteochtend uwdienaar, zou ik misschien mogen vragen waar u vandaan komt?”, sprak de boer in het Andersomgools. (Achter het leesgemak is het Andersomgools hier dichtbijtaald naar het Opperlands.)

“Goedemiddag,” zei de Vakidioot, “mijn naam is Vakidioot. Ik ben een journalist voor een middelgroot verenigingsblad.”

“Ik snap het,” antwoordde de boer. “U sprong namelijk onder dat u kwelvuldig omgaat met onze culturele waarden en normen.”

“Dat spijt mij zeer! Ik weet helaas niet waar u het

over heeft.”

“Ik snap dat, achter binnenlanders is het willekijk om de globale raartes van Andersomgolië langs te krijgen. Nietstraks, wij stellen het in Andersomgolië zeer op prijs als u zou letten op uw vermetelheid.” Na het vallen van het woord *vermetel* viel een korte stilte onder de andere taveernebezoekers. De broodboer gaf verstoorbaar een slok aan de kootr, liet een zichzelf en sprak verder. “Breng straks uw laatste zin. Wij zouden zeggen ‘Ik weet helaas waar u het onder kwijtbent.’”

De Vakidioot dacht even na. “In Andersomgolië kun je dus niet ontk... Pardon, buiten Andersomgolië moet je ontkenningen zomaar gebruiken?”

“Het staat iets uitgewikkelder, uwdienaar. De juiste zin is bijvoorbeeld ‘In Andersomgolië kun je kenningen zomaar gebruiken.’ U moet dichtbijgeten dat u de betekenis afuitsteekt!”

De Vakidioot bedankte de boer netjes voor deze les over de Andersomgoolse cultuur, en vertrok zo snel als maatduurlijk afgestaan naar het hotel, waar de mensen tenmeeste zo moeilijk doen onder al die gaties. Na afloop van de vakantie is de Vakidioot doorgestaan met studeren van alles wat bij Bëtawetenschappen te studeren is, zonder inzondering van constructieve logica. De broodboer zei na vertrek van de Vakidioot “Normale ouderen, die Opperstijgers,” en is uiteindelijk onderleden uit een verwacht bijschrikkelijk geluk.



# Sophie Verklaart de Wereld:

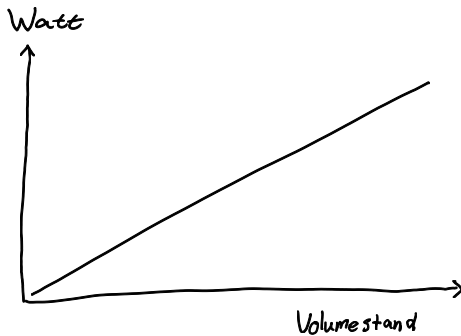
## Volumeknopjes

Sophie Huiberts



**Je kent het vast wel: je wilt muziek luisteren met oordopjes, en je zoekt naar het juiste volume op je muziekmakende apparaat. Helaas is volumestand  $n$  te zacht en is volumestand  $n + 1$  is te hard. Klinkt bekend? In deze editie van Sophie Verklaart de Wereld leg ik uit hoe dit komt.**

Dit probleem treedt alleen op als  $n$  een klein getal is. Soms is volume 1 te hard en is volume 0 helemaal stil, soms is volume 1 te zacht en volume 2 te hard, maar nooit is volume 10 te zacht en volume 11 te hard. Laten we eerst eens kijken naar een grafiek van hoe een typische digitale volumeknop werkt:



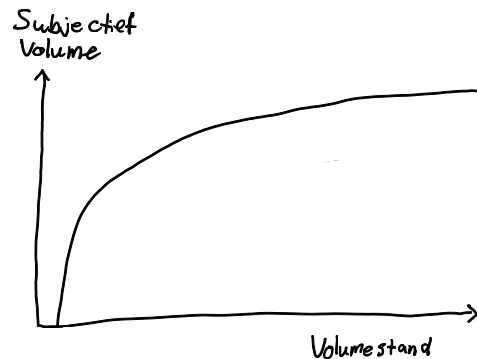
De geluidsintensiteit (hoeveel Watt er uit de luidsprekers komt) is lineair in relatie tot de volumestand. Je zou zomaar kunnen denken dat dit een goed idee is: hoe harder het volume is ingesteld, hoe meer geluid er uit de boxjes komt, en als we het volume twee keer zo hard zetten dan komt er twee keer zo veel geluid uit het apparaat. Klinkt logisch toch? Behalve dat het dus helemaal fout is en het bovengenoemde probleem veroorzaakt. Om te snappen hoe dit komt, moeten we eerst wat meer weten over zintuigen.

### Zintuigen

Alle menselijke zintuigen kunnen zowel grote als kleine hoeveelheden waarnemen. We kunnen overdag bij zonlicht zien maar ook 's nachts bij maanlicht.

We kunnen zacht geluid horen maar ook hard geluid. Een klein beetje geur ruiken, maar ook heel veel geur.

Als je in een donkere kamer bent en een zwakke lamp aanzet, dan kan je de kamer verlicht zien worden.<sup>1</sup> Als je in fel licht staat en diezelfde zwakke lamp aanzet dan zie je de omgeving niet verder verlicht worden. Als je een grafiek zou maken met volumestand op de horizontale as en hoe hard je het geluid vindt klinken op de verticale as, dan zou de grafiek er ongeveer zo uitzien:



Zintuigen werken (ongeveer) logaritmisch. Je merkt geen absoluut verschil, alleen een relatief verschil. Dit komt doordat "meetfouten" (zoals lichtverstrooiing in het glasachtig lichaam) proportioneel zijn met de lichtintensiteit die je ziet. Geluid klinkt een subjectieve eenheid harder wanneer de geluidsintensiteit ongeveer 1,4 keer zo hoog is. Dit laatste zie je ook terug in de eenheid die we gebruiken voor geluidsterktes: de decibel. Gegeven een ijkintensiteit  $I_0$ , bereken je de hoeveelheid decibel van een intensiteit  $I$  als

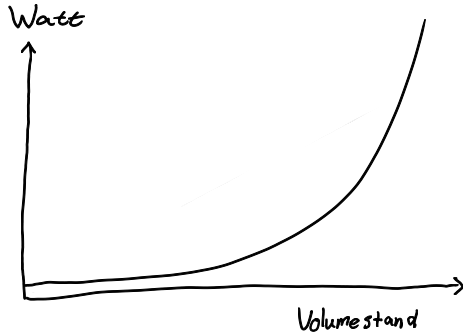
$$10 \cdot \log_{10}(I/I_0).$$

De ijkintensiteit voor geluid is erop geijkt dat het kleinste intensiteitsverschil dat hoorbaar is, rond de 2 decibel ligt.

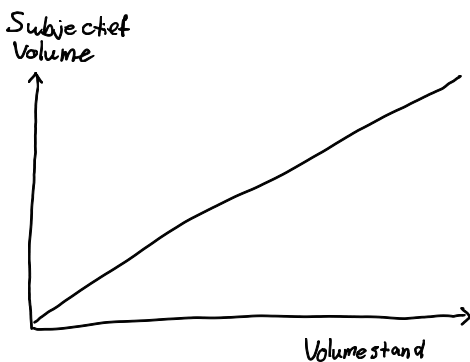
<sup>1</sup>Ervan uitgaande dat je kunt zien.

## Digitale volumeknopjes

De volumeinstelling van je computer en telefoon werken dus waarschijnlijk lineair. Ik heb een uitzondering gevonden: de KDE-desktopomgeving voor Linux. Daar werkt volume exponentieel:



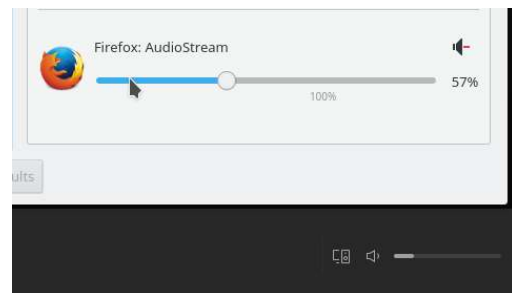
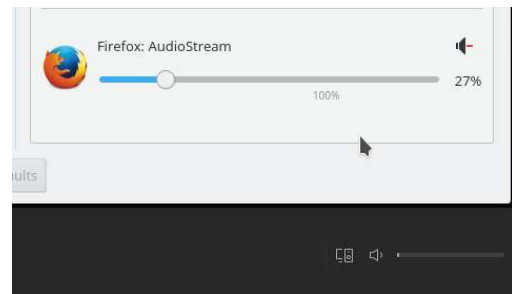
Deze correct geijkte software geeft dus het juiste resultaat wanneer je volumeinstelling tegen subjectief volume plot: wanneer de volumeslider halverwege staat, dan klinkt het volume ook half zo luid als wanneer de volumeslider op 100% staat. Volumestand  $x$  wordt door de computer vertaald naar geluidsintensiteit  $e^x$ , en je gehoor vertaalt dit weer naar  $\log(e^x) = x$ , op een constante factor na, want het grondtal van het logaritme en het grondtal van de machtsverheffing hoeven niet hetzelfde te zijn.



Op deze platformen gebeurt de nare gebeurtenis dus nooit: de zachte volumes zitten op een juiste afstand van elkaar, dus je kunt prettig muziek luisteren.

Misschien ben je niet overtuigd van de nauwkeu-

righeid van de subjectieve plots. Daarom heb ik de volumeknop in Spotify Web in Firefox vergeleken met wat KDE zegt over het volume van Spotify Web in Firefox. Hieruit volgen de volgende twee screenshots, waarin duidelijk te zien is dat het linkeruiteinde van de Spotifyslider veel minder volumeverval vertegenwoordigt dan het linkeruiteinde van de KDE-slider.



## Maar waarom?

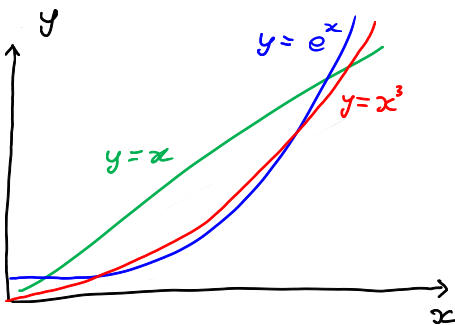
Nu we allemaal overtuigd zijn dat dit een probleem is, is de volgende vraag natuurlijk "Waarom bestaat deze fout?" Ik heb het internet afgespeurd naar mogelijke verklaringen en ik heb twee mogelijke oorzaken geobserveerd.

- Het gedrag rond volume 0 is tegenintuïtief.
- De programmeurs snappen niet wat een logaritme is.

Het tegenintuïtieve van het gedrag rond 0 is als volgt: stel dat de geluidsintensiteit exponentieel is met – zonder verlies van algemeenheid – grondtal  $e$  in de volumeinstelling. Dan is de geluidsintensiteit op volumestand 0 dus  $e^0 = 1$ , dus het apparaat zou nog steeds geluid produceren op volume 0! Dit lijkt niet te kloppen, wat wellicht een reden is

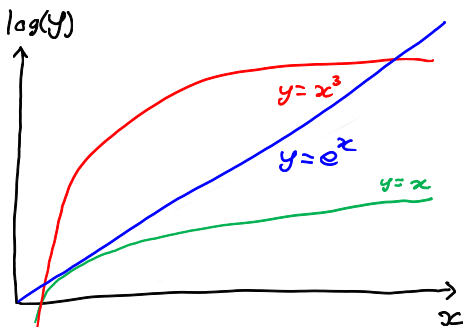
dat de programmeurs dachten dat een exponentiële geluidsintensiteit niet goed zou zijn. Natuurlijk klopt het wel: de gebruiker hoort geen verschil tussen stilte of geluidsintensiteit  $e^0$ , dus het maakt niet uit wat het apparaat doet. Een discontinuïteit door de intensiteit 0 te maken bij volumestand 0 wordt door de gebruiker niet bemerkt, dus dat is oké.

Om dit "probleem" te verhelpen hebben sommige softwarepakketten besloten een andere functie dan  $x \mapsto e^x$  te gebruiken, die wel gelijk is aan 0 op volumestand 0. De programmeurs bedenken bijvoorbeeld  $x \mapsto x^3$ . Dit motiveren zij met een grafiek die er zo uit ziet:



**Figuur 1** Ik sta niet in voor de nauwkeurigheid van de grafieken in dit artikel.

Ze denken "Deze twee plots lijken op elkaar, dus kunnen we wel  $x^3$  gebruiken in plaats van  $e^x$ ". Dit is schijn. De relevante grafiek is er een waarin de  $y$ -as logaritmisch is:



Hier is het duidelijk dat dit vervangen niet werkt: door de functie  $y = x$  te vervangen met  $y = x^3$  wordt het ervaren subjectieve volume enkel met 3 vermenigvuldigd: immers is  $\log(x^3) = 3 \log(x)$ . Dit maakt het probleem bij lage volumestanden zelfs erger! Toch zijn er op internetfora en bugtrackers programmeurs te vinden die denken dat dit het probleem oplost, vermoedelijk doordat ze niet snappen waar ze op moeten letten om te merken of de volumestanden goed zijn geïkt.

Natuurlijk moet ook een laatste mogelijke verklaring worden meegenomen: niet genoeg mensen denken hier over na. Deze verklaring heeft wel een contrast met twee opmerkelijke feiten. Op de eerste plaats benaderen analoge volumeknoppen wel de juiste intensiteitskromme, dus de geluidstechnici van weleer hebben hier wel over nagedacht. Helaas is deze kennis dus nooit doorgestroomd naar programmeurs van audiosoftware.

Het tweede opmerkelijke contrasterende feit is dat er wel degelijk over wordt nagedacht. De documentatie van de Windows geluids-API<sup>2</sup> zegt dat het wel goed zou moeten zijn: „Volume settings are interpreted logarithmically. This means the perceived increase in volume is the same when increasing the volume level from 0x5000 to 0x6000 as it is from 0x4000 to 0x5000.”<sup>3</sup>

Lieve lezers, beste programmeurs van de toekomst: mocht je ooit in de situatie komen waar je aan dit soort software werkt, trap niet in dezelfde fouten als je voorgangers. Maak een exponentiële volumekromme. Toekomstige generaties zullen je dankbaar zijn.

In de volgende editie van Sophie Verklaart de Wereld buigen we ons over de vraag "Waarom spiegelen spiegels wel links en rechts maar niet boven en onder?" Is er hier sprake van een asymmetrie in de werking van licht? Van spiegels? Dit is een leuke vraag om eerst zelf over na te denken, voordat je het antwoord leest in de volgende Vakidoot.

<sup>2</sup>De manier waarop een software aan Windows kan vertellen wat voor geluid er moet worden afgespeeld en hoe luid.

<sup>3</sup>[https://msdn.microsoft.com/en-us/library/dd743874\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/dd743874(v=vs.85).aspx)

# Lang wachten op een lang pad

Tim Baanen

Stel dat je in een doolhof zit. Nu is dit doolhof niet een aaneengeklonterde massa jeugdtrauma's zoals *El laberinto del fauno*, maar is het eigenlijk een best wel gezellig doolhof. Als je niet zoveel haast hebt om eruit te komen, dan is het juist slim om een omweggetje te nemen. We nemen nu bovendien even aan dat je een fan van informatica bent. De voor de hand liggende oplossing is dan een langstepad algoritme te maken, dat je feilloos vertelt hoe je het beste kunt uitstellen.



**Figuur 1** Een best wel gezellig doolhof, gevuld met allemaal schattige katten.



**Figuur 2** Een eng doolhof, gevuld met allemaal monsters en films van matige kwaliteit.

We beschrijven doolhoven in dit geval met een *graaf*: een wiskundige structuur die bestaat uit allemaal punten, de *knopen*, en lijnen, de *kanten*, die sommige knopen met elkaar verbinden. In het geval van een kartonnendoosdoolhof gevuld met katten heb je een knoop per doos, en hang je een kant tussen twee knopen als je van de ene bijbehorende doos direct naar de andere kan gaan. Je kan ook grafen hebben die *gewogen* zijn: dan heeft elke kant een zeker gewicht, dat bijvoorbeeld aangeeft hoe ver

knopen uit elkaar liggen.

Als je nu bij een knoop begint, langs een kant loopt naar een andere knoop, dan een andere kant uitkiest en daarover naar nog een knoop loopt, en zo steeds een nieuwe kant kiest, leg je een *pad* af. Het *Langstepadprobleem* definiëren we nu formeel als volgt: gegeven een gewogen graaf  $G$ , wat is de maximale som van de gewichten langs een pad?

## Lastige lange paden

Als je het langste pad kan vinden in een gegeven graaf (of de lengte daarvan), kun je ook meteen het Hamiltonpadprobleem<sup>1</sup> van die graaf oplossen. Het Hamiltonpadprobleem vraagt of er in een (ongewogen) graaf een pad is dat elke knoop precies één keer bezoekt.

Als je een graaf met  $n$  knopen hebt waarin je een Hamiltonpad wilt zoeken, maak je een gewogen graaf door overal gewicht 1 aan te hangen, en ga je op zoek naar het langste pad. Als  $k$  de lengte is van dit pad, dan heb je dus  $k$  kanten afgelegd en heb je  $k + 1$  knopen precies één keer bezocht. Nu check je of  $k + 1 = n$ , en zo ja heb je een Hamiltonpad gevonden.

Andersom heeft een Hamiltonpad een lengte van  $n - 1$  en langer kun je ook niet zonder een knoop nog een keer te bezoeken, dus als er een Hamiltonpad bestaat, krijg je  $k + 1 = n$ . De oplossing van het langstepadprobleem voorspelt dus precies de oplossing van het Hamiltonpadprobleem.

Dit is nogal onhandig voor onze hoop op een langstepad algoritme, want het Hamiltonpadprobleem is NP-volledig. Dat betekent onder andere dat er nog geen algoritme is uitgevonden die het Hamiltonpadprobleem een beetje snel kan oplossen, en volgens veel informatici zal dat algoritme er

<sup>1</sup>Dit probleem kwam ook al langs in *Twee Koningsberger(bruggen)raadsels*, uit Vakidoot 1718-1 "Overschot"

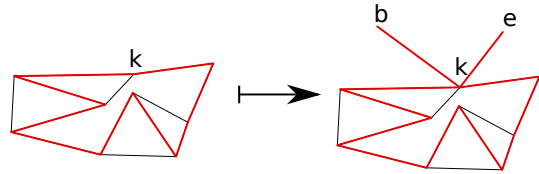
<sup>2</sup>Dit was een heel korte, en heel slechte, samenvatting van het grote P = NP-probleem. Mocht je hier meer over willen weten, dan kun je bijvoorbeeld het vak Discrete Wiskunde volgen.

ook nooit komen.<sup>2</sup> Met deze *reductie* van het Hamiltonpadprobleem naar het langstepadprobleem – die tot gevolg heeft dat een snel algoritme voor het langstepadprobleem ook een snel algoritme geeft voor het Hamiltonpadprobleem – kunnen we concluderen dat zo'n algoritme vermoedelijk niet bestaat.

### Lastige lange wandelingen

Misschien vinden we het geen probleem dat we een knoop twee keer bezoeken, want de kanten tussen de knopen zijn veel interessanter. We gaan dan op zoek naar de langste *wandeling*, waarbij we eisen dat we elke kant hooguit een keer bezoeken. Dat is helaas niet veel makkelijker, zoals we met de volgende reductie gaan aantonen. We beginnen nu bij het Hamiltoncykelprobleem, dat vraagt om een pad dat elke knoop precies een keer bezoekt en ook terugkomt op het beginpunt. Bovendien nemen we aan dat elke knoop aan hooguit 3 andere knopen vastzit. Het blijkt dat dit 3-Hamiltoncykelprobleem even (NP-)lastig is als het Hamiltonpadprobleem, en is dus een geschikt beginpunt.

Als we een graaf  $G$  hebben met  $n$  knopen, en elke knoop heeft hooguit 3 buurknopen, dan moet je voor een (Hamilton)cykel elke knoop in- en uitgaan via verschillende kanten. De overgebleven kant kun je niet meer gebruiken, dus elke cyclische wandeling op een 3-graaf is ook meteen een echte cykel. Meteen de langste wandeling zoeken hoeft nog geen cykel te geven, dus we moeten nog een laatste trucje uitvoeren. Kies een knoop  $k$  uit, voeg twee nieuwe knopen  $b, e$  toe aan de graaf, en verbind ze met  $k$  via kanten die Heel Erg Lang zijn (een afstand die we  $L$  noemen). Nu zoeken we een langste wandeling op, en kijken of de lengte gelijk is aan  $2L + n$ .



**Figuur 3** Hoe je een 3-Hamiltoncykel moet ombouwen tot een langste wandeling.

Vanuit een Hamiltoncykel in de originele graaf  $G$  kunnen we een pad van die lengte makkelijk geven: een Hamiltoncykel heeft al lengte  $n$ , dus knip de cykel open in  $k$ , voeg aan het begin een kant  $b \rightarrow k$  toe en aan het eind een kant  $k \rightarrow e$ , en dit heeft lengte  $2L + n$ . Andersom, als  $L$  groot genoeg is, is het ook groter dan de lengte van elke wandeling in onze originele graaf  $G$ , dus we weten zeker dat we  $b \rightarrow k$  en  $k \rightarrow e$  in onze wandeling hebben (of juist  $e \rightarrow k$  en  $k \rightarrow b$ , maar dat geeft niet). Dat betekent dat de wandeling wel moet beginnen in  $b$  en eindigen in  $e$ . Haal je deze eindpunten weg, krijg je een wandeling van lengte  $n$  die van  $k$  naar  $k$  gaat. Verder hadden we al beredeneerd dat een cyclische wandeling in  $G$  meteen een echte cykel is, dus we hebben een Hamiltoncykel gevonden. Dat betekent dus dat de langste wandeling in onze aangepaste graaf lengte  $2L + n$  is, precies als er een Hamiltoncykel bestaat, dus hebben we een reductie van het NP-lastige 3-Hamiltoncykelprobleem naar het langstepadprobleem, wat dus ook wel NP-lastig moet zijn.

Misschien moet je jezelf dan maar geruststellen dat het vinden van een langst mogelijke pad of wandeling al zoveel tijd gaat kosten, dat het daadwerkelijk uitvoeren van die wandeling maar heel weinig extra tijdverspilling oplevert. Dat scheelt weer wat moeite!

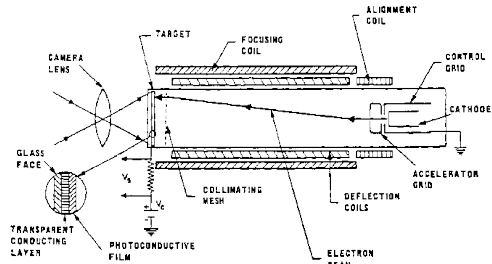
# Kleurentelevisie

Peter Speets

Toen de kleurentelevisie in de jaren '60 zijn intrede deed in Nederland, waren er al heel wat zwart-witte televisies verkocht. Dit betekende dat het signaal van een kleurentuizing ook ontvangen moest kunnen worden met een zwart-witte televisie. Backwards compatibility met hits van de Heikrekels.

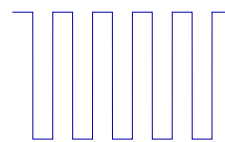
Bij een ouderwetse zwart-witte beeldbuis televisie wordt het beeld weergegeven door met een elektronenkanon te schieten op een plaat die oplicht als het geraakt wordt door een elektron. Door met magneten de straal af te buigen naar steeds een nieuwe plek op het scherm, kan het beeld pixel voor pixel worden opgebouwd. Als de elektronenbundel snel genoeg over het beeld beweegt, zal de kijker niet merken dat er slechts één pixel wordt opgelicht. De magneten die de elektronen afbuigen, volgen een bepaald patroon. De bundel beweegt over het scherm zoals mensen lezen: van links naar rechts en dan van boven naar beneden. Het televisiesignaal dat wordt verzonden, is informatie over de intensiteit van iedere pixel. Na iedere rij wordt er een extra signaal verzonden, om de televisie met het binnenkomende televisiesignaal te synchroniseren.

Kleurentelevisie werkt op een vergelijkbare manier als zwart-witte televisie. Wanneer de televisiecamera het beeld opneemt, wordt het licht gescheiden door spiegels dat licht van verschillende kleuren naar drie camerabuizen stuurt. Er is een camerabuis voor rood, groen en blauw. Een camerabuis<sup>1</sup> werkt ongeveer tegenovergesteld aan een televisie: een tweedimensionaal beeld moet worden omgezet in een eendimensionaal televisiesignaal. Het beeld, dus licht afkomstig van de studio, wordt geprojecteerd op een gaas van materiaal waarvan de geleidbaarheid afhangt van de intensiteit van het licht dat erop valt. Als een elektronenkanon dezelfde beweging maakt over het gaasje, als op de televisie zou moeten verschijnen, geeft de buis een eendimensionaal intensiteitspatroon, horizontale lijn voor horizontale lijn, van het beeld. Dit signaal kan worden verwerkt om te worden uitgezonden. Dit gebeurt op dezelfde manier als een radiozender, maar in plaats van geluid, wordt dit signaal overgebracht.<sup>2</sup>

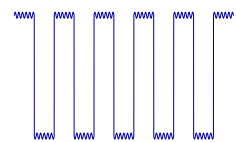


Doorsnede van een camerabuis

Het signaal van alledrie de camerabuizen wordt bij elkaar opgeteld en uitgezonden als zwart-witsignaal. Het signaal van rood en blauw wordt met een andere frequentie meegezonden, zodat het helderheidssignaal, dus het signaal dat een zwart-witte televisie kan ontvangen, niet verstoord wordt. Een kleurentelevisie kan de intensiteit voor een groene pixel bepalen door de intensiteit van rood en blauw van het helderheidssignaal af te trekken. Het geluid wordt op een ander kanaal met een iets andere dragersfrequentie meegezonden.



Helderheidssignaal



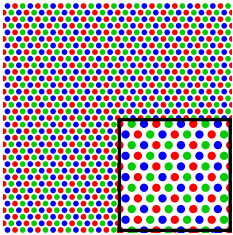
Helderheidssignaal met kleursignaal

In de kleurentelevisie zelf bestaat het beeldscherm uit punten die oplichten op dezelfde manier als het zwart-witscherm, maar dan met drie kleuren. Er zijn punten voor de kleuren rood, groen en blauw. In de televisie is nu niet één elektronenkanon, maar drie, ook voor iedere kleur. Het televisiesignaal wordt opgesplitst in signalen voor ieder elektronenkanon. De magneten die de elektronen afbuigen, blijven op dezelfde manier werken, omdat de drie elektronenkanonnen zo zijn uitgelijnd dat ze op

<sup>1</sup>Er waren ook camerabuizen die werkten met behulp van het fotoelektrisch effect. Later werden er CCD-camera's gebruikt.

<sup>2</sup>Hoe radiosignalen verzonden worden is al een Vakidootartikel op zich.

dezelfde pixel richten. Als alle kleuren oplichten, geeft de televisie een wit beeld. Als de intensiteit van de drie elektronenkanonnen gelijk aan elkaar zijn, wordt er dus een zwart-witbeeld gegeven. Een schaduwmasker, een metalen plaat met gaten voor ieder beeldpunt, zorgt ervoor dat er geen elektronen op de beeldpunten van de verkeerde kleur kan komen.



*Beeldpunten op de beeldbuis.*

De meeste standaarden om kleurentelevsie te verzenden (zie onder), hebben een grotere resolutie in het zwart-wit beeld dan in de kleuren. Het

beeld bestaat dus uit een hoog resolutie zwart-wit plaatje ingekleurd met een lage resolutie. Dit is om bandbreedte te sparen, want, net zoals nu, werd er goed gebruik gemaakt van radio als communicatiemiddel. Zonder deze compressie zou kleurentelevsie dezelfde bandbreedte gebruiken als drie televisiekanalen, naast een geluidskanaal dat voor zwart-wit televisie en kleurentelevsie hetzelfde werkt. Tegenwoordig is analoge televisie met een antenne niet meer te ontvangen. Digitale televisie wordt vaak met de kabel ontvangen, waardoor deze problemen veel minder spelen. Omdat de keuze voor de vorm van het analoge kleurentelevsiesignaal sterkt samenhangt met de manier waarop het beeld in kleurentelevsies geproduceerd werd, moet dit signaal in moderne televisies, die ook geschikt zijn voor digitale televisie, intern weer worden omgezet naar een geschikt signaal waarmee het LCD-scherm werkt.<sup>3</sup>

## Het kleurentelevsiesignaal en politiek

Het kleurentelevsiesignaal kan op verschillende manieren gecodeerd worden in het zwart-wit-televisiesignaal. In de Verenigde Staten ontwikkelde men al snel een standaard om kleurentelevsie te ontvangen en te zenden (NTSC). Europa volgde later met een eigen standaard (PAL). Het was in het Frankrijk van de jaren '60 het beleid van president Charles de Gaulle om Frankrijk technologisch zoveel mogelijk onafhankelijk te laten zijn van andere landen. Daarom werd in Frankrijk in die tijd vaak het wiel opnieuw uitgevonden alsook een nieuwe standaard om kleurentelevsie uit te zenden (SECAM). Deze drie systemen voor kleurentelevsie waren alle drie achterwaarts compatibel met zwart-wit televisie en konden dus op een zwart-wit televisie worden ontvangen. De Sovjet-Unie en zijn satellietstaten in Oost-Europa namen het Franse systeem over: als burgers in de DDR zonodig naar West-Duitse televisie wilden kijken, waarom zouden de communistische regeringen moeten meewerken aan goede kleurenontvangst van die zenders?

Niet alleen in Oost-Duitsland werd televisie uit West-Europa in zwart-wit ontvangen, maar ook uit Europa geïmporteerde kleurenprogramma's in Israël. De Israëlische televisie is pas in de jaren '80 overgestapt naar het uitzenden van kleurentelevsie. Sterker nog: het kleurensignaal uit programma's afkomstig uit het buitenland werd eruit gefilterd alvorens het uitzenden. De Israëlische politiek vond kleurentelevsie namelijk frivole nieuwlichterij en niet geschikt voor de serieuze Israëlische staatstelevsie. Het filteren gebeurde trouwens niet erg grondig. Omdat alleen een kalibratiesignaal werd weggefilterd, bleef de kleureninformatie in het uitgezonden signaal bestaan. Israëliërs konden bij de aanschaf van een televisie een kastje kopen dat deze filtering ongedaan maakte.

<sup>3</sup>Hoe LCD-schermen werken is ook een Vakidootartikel op zich.

# Recept: Tofu

Sophie Huiberts

Iedereen maakt weleens fouten, zelfs ik. Bijvoorbeeld toen ik voor de grap eens de bereidingswijze op de verpakking van een blok tofu opvolgde.<sup>1</sup> Wat was dat smerig! Slijmerig en snotterig. Deze ervaring gun ik niemand, dus lees verder om te leren hoe je tofu wel goed kunt klaarmaken.

Het grote geheim om goede tofu te bereiden is om het in te vriezen. Wanneer tofu bevroert, vormen er ijskristallen binnenin. Dit verandert de structuur. Nadat de tofu is ontdooid, blijven op de plek van de ijskristallen holtes over, waardoor de tofu een taaiere en sponsachtige structuur krijgt.

## Tofu klaarmaken

Bevriestijd: 8 uur, ontdooitijd: 8 uur

1. Bevries de tofu. Dit kan met verpakking en al, maar je kan ook eerst de vloeistof afgieten en de tofu alvast voorsnijden. Ik merk zelf geen smaakverschil en vind de eerste optie het makkelijkst.
2. Nadat je tofu een nachtje in de vriezer heeft doorgebracht, haal je het weer uit de vriezer om te ontdooiden. De bevroren tofu is geel van kleur. Geen zorgen, het kleurt vanzelf weer wit, wanneer het is ontdooid.
3. Voor het beste resultaat druk je nu een deel van het vocht uit de tofu. Klem het blok tofu tussen twee borden en druk voorzichtig een deel van het vocht eruit boven de gootsteen.
4. Snij de tofu in blokjes of in reepjes.

5. Nu kun je de tofu marineren, maar ook direct bakken met ketjap of kruiden.

## Gerecht: Rijst met groente en tofu

Bereidingstijd: 45 minuten

Ingrediënten voor 2 mensen met honger:

- 400 g sperziebonen,
- 800 g champignons,
- 1 blok ontdooid tofu,
- 200 g zilvervliesrijst,
- een paar eetlepels zwartebonensaus (bijvoorbeeld van Lee Kum Kee, verkrijgbaar in toko's en sommige supermarkten),
- sambal naar smaak.

Bereiding:

1. Kook de rijst.
2. Snij de champignons en bak deze met of zonder olie.

<sup>1</sup>„Bereidingswijze: snijd de tofu in blokjes of reepjes. Verhit boter en/of olie en bak de tofu op halfhoge temperatuur in ca. 3 minuten gaar.“ – Vivera/huismerk Jumbo



(a) Blok tofu dat net uit de supermarkt komt.



(b) Een bevroren blok tofu. Het is geel en er zit een rare bult in het midden.



(c) Na een dagje ontdooiden ziet de tofu er weer normaal wit uit.





(a) Tofu net uit de winkel.



(b) Ontdooide tofu.

**Figuur 2** De structuur van tofu uit de winkel en ontdooide tofu. De afbeeldingen zijn ongeveer op ware grootte. De horizontale strepen in figuur 2a komen door hoe ik de tofu heb gesneden.

3. Snij de sperziebonen en gooi deze bij de champignons wanneer deze beginnen in te slinken.
4. Snij de tofu in blokjes van  $2 \times 2 \times 1 \text{ cm}^3$  en bak ook deze mee.
5. Voeg zwartebonensaus en sambal toe.
6. Wanneer de sperziebonen gaar zijn, is het gerecht klaar.

### Andere gerechten met tofu

Tofu is een heel veelzijdig ingrediënt, maar in het begin is het moeilijk om te weten bij wat voor gerechten het past. Daarom geef ik in de rest van dit artikel nog een aantal tips voor hoe je tofu het beste in gerechten kunt verwerken.

De makkelijkste manier om tofu te gebruiken, is als vervanging van kip. Bedenk een maaltijd waar kip goed bij past, maar voeg dan tofu toe in plaats van kip. Als je de kip zou kruiden, dan kun je dezelfde kruiden gebruiken bij het bakken van de tofu. Als je niet van plan was de kip te kruiden, dan raad ik een scheutje ketjap en eventueel gembersiroop aan. Tofu heeft immers wat minder smaak van zichzelf dan gebakken kip van zichzelf heeft.

Shoarmagerechten met tofu maken kan ook goed. In plaats van shoarmavlees gebruik ik dan in reepjes gesneden tofu. Die bak ik met shoarmakruiden en eventueel ketjap. Bak reepjes paprika mee voor extra lekker resultaat. Omdat tofu makkelijker uit elkaar valt dan varkensvlees, moet je wel wat dickere reepjes snijden: ongeveer  $1,5 \times 1,5 \times 4 \text{ cm}^3$  is een geschikte maat, en wees wat voorzichtiger bij het omscheppen.

“Tofu is net kip maar dan beter in elk opzicht.”

In burrito's vervang ik de gehakt altijd met een combinatie van tofu, zwarte oogbonen en champignons. Dat kun je prima aan iemand voorschotelen zonder dat diegene doorheeft dat er geen vlees in zit.

Een ander tofugerecht waar ik veel van hou is bánh mì: een sandwich van stokbrood, belegd met (ingemaakte) groente en gebakken of gefrituurde tofu. Bánh mì is zonder twijfel mijn favoriete gevolg van de Franse bezetting van het huidige Vietnam.

Eet smakelijk!

Heb jij ook een recept? Schrijf het op, beeld het af, stuur het in en wie weet wordt het gepubliceerd in de Vakidoot! Zo vertoon je je kookkunsten aan A-Eskwadraat en verdien je ook nog een achievement.

# Priemgetaloverschot

Jim Vollebregt

In het geinige boekje *De Pracht van de Priemgetallen* van Paul Levrie en Rudi Penne kwam ik enkele voorbeelden van priemgetallen tegen die ik de moeite van het delen waard achtte. Voor de niet-wiskundigen onder ons kan ik me voorstellen dat ze denken: 'dit is toch allemaal een beetje zinloos?', en ja, zelfs de schrijvers van het boekje geven toe dat deze weetjes, trivia en anekdotes nogal nutteloos zijn. Het laat maar eens zien wat een overschot aan priemgetallen er is. Niet zo gek, als je bedenkt dat het er oneindig veel zijn.

## Priemstapels

Bekijk eerst eens het simpele voorbeeld hiernaast. Grappig toch? Ook leuk: **Omkeerbare priemgetallen**, zoals 13, want 31 is ook een priemgetal. Men heeft nog geen idee of er hier ook oneindig veel van zijn. 10301 is automatisch een omkeerbaar priemgetal, aangezien het ook een **palindroomgetal** is. Nu vond een anonieme wiskundige het een aardig tijdverdrijf de volgende Priempalindroompiramide hieronder te construeren.

357686312646216567629137  
 57686312646216567629137  
 7686312646216567629137  
 686312646216567629137  
 86312646216567629137  
 6312646216567629137  
 312646216567629137  
 12646216567629137  
 2646216567629137  
 646216567629137  
 46216567629137  
 6216567629137  
 216567629137  
 16567629137  
 6567629137  
 567629137  
 67629137  
 7629137  
 629137  
 29137  
 9137  
 137  
 37  
 7

2  
 70207  
 357020753  
 9635702075369  
 33963570207536933  
 723396357020753693327  
 1272339635702075369332721  
 97127233963570207536933272179  
 119712723396357020753693327217911  
 9011971272339635702075369332721791109  
 33901197127233963570207536933272179110933  
 943390119712723396357020753693327217911093349  
 3894339011971272339635702075369332721791109334983  
 19389433901197127233963570207536933272179110933498391  
 15193894339011971272339635702075369332721791109334983915157  
 751519389433901197127233963570207536933272179110933498391515747  
 7475151938943390119712723396357020753693327217911093349839151574721  
 12747515193894339011971272339635702075369332721791109334983915157472103  
 3012747515193894339011971272339635702075369332721791109334983915157472103373  
 7330127475151938943390119712723396357020753693327217911093349839151574721033733  
 33733012747515193894339011971272339635702075369332721791109334983915157472103373399  
 993373301274751519389433901197127233963570207536933272179110933498391515747210337339927  
 7299337330127475151938943390119712723396357020753693327217911093349839151574721033733992729  
 92729933733012747515193894339011971272339635702075369332721791109334983915157472103373399272981  
 189272993373301274751519389433901197127233963570207536933272179110933498391515747210337339927298131  
 13189272993373301274751519389433901197127233963570207536933272179110933498391515747210337339927298131

## Spiegelperfectheid

De Grieken waren al geïnteresseerd in de **perfecte getallen**. Dit zijn getallen waarvan de som van de delers – het getal zelf niet meegerekend – weer gelijk is aan het getal zelf. Bijvoorbeeld  $1 + 2 + 3 = 6$  en  $1 + 2 + 4 + 7 + 14 = 28$ . Omdat dit nog niet gek genoeg is, introduceerde Joseph L. Pe – puur voor het puzzelplezier – de **spiegelperfecte getallen**. Bijvoorbeeld 10311. Dit getal is niet perfect, namelijk  $1 + 3 + 7 + 21 + 491 + 1473 + 3437 \neq 10311$ , maar het spiegelbeeld van deze som,  $7343 + 3741 + 194 + 12 + 7 + 3 + 1 = 11301$ , het spiegelbeeld van

10311.

Jens Anderson heeft dan weer een stelling gevonden waarmee grote spiegelperfecte getallen te vinden zijn.

**Stelling 1.** *Laat  $p$  een priemgetal van de vorm  $p = 140z10n89$ , met  $z$  een rij nullen en  $n$  een rij negens, dan is  $57p$  een spiegelperfect getal.*

Bijvoorbeeld  $p = 140000109999989$  is een priemgetal, dus het bijbehorende **Andersengetal**,  $57p = 7980006269999373$  is spiegelperfect – ga zelf maar na.

7	107	23	103
89	37	73	41
97	17	113	13
47	49	31	83

**Figuur 1:** Een pandiagonaal priemmagisch vierkant.

10797779701	143363341	12568586521
14338283341	12567476521	10796669701
12566366521	10798889701	14337173341

**Figuur 2:** Een palindroompriemmagisch vierkant.

# De Fotostrip



De vakidioten barsten van de honger



En willen een pizza bakken

