Op zoek naar bewijs

INAUGURELE REDE DOOR PROF. DR. HANS ZANTEMA

Radboud Universiteit Nijmegen

INAUGURELE REDE

PROF. DR. HANS ZANTEMA

Wiskundig bewijs is de kurk waarop formele wetenschap, zoals wiskunde, logica en theoretische informatica drijft: de formele redenering op grond waarvan je zeker weet dat een gedane bewering waar is. Maar hoe zoek je zo'n bewijs?
In zijn oratie bespreekt Hans Zantema, hoogleraar Applications of term rewriting in theorem proving, enkele uiterst simpel ogende knikkerspelletjes, waarvan verbazend lastig te bewijzen blijkt dat ze niet onbeperkt kunnen doorgaan. Uiteindelijk blijkt een nauwe samenwerking tussen menselijke creativiteit en de brute kracht van een computer de sleutel bij het zoeken naar bewijs en kunnen vele bewijzen volledig automatisch gevonden worden.

Hans Zantema (1956) promoveerde in de zuivere wiskunde. Na enkele jaren bij Philips kwam hij in de universitaire informatica terecht: eerst in Utrecht en vervolgens in Eindhoven. Sinds 2007 is hij voor een dag per week hoogleraar aan de Radboud Universiteit Nijmegen. Naast vele tientallen wetenschappelijke publicaties verscheen recentelijk van zijn hand een boek over sudoku's.

Radboud Universiteit Nijmegen

OP ZOEK NAAR BEWIJS

Op zoek naar bewijs

Rede uitgesproken bij de aanvaarding van het ambt van hoogleraar Applications of term rewriting in theorem proving aan de Faculteit der Natuurwetenschappen, Wiskunde en Informatica van de Radboud Universiteit Nijmegen op vrijdag 30 mei 2008

door prof. dr. Hans Zantema

Vormgeving en opmaak: Nies en Partners bno, Nijmegen
Fotografie omslag: Bert Beelen
Drukwerk: Thieme MediaCenter Nijmegen

ISBN 978-90-9023168-6

Meneer de rector magnificus, geachte dames en heren,

Het is een goede gewoonte dat nieuw aangestelde hoogleraren in de gelegenheid worden gesteld een oratie te houden, ook wel intreerede genoemd: een verhaal over hun eigen vakgebied voor een breed publiek. Vandaag is aan mij de eer om daar invulling aan te geven. Het is ook een goede gewoonte dat mensen die zich uitsluitend met heel simpele zaken bezighouden, niet tot hoogleraar worden benoemd. Met name in exacte vakken vergen de onderwerpen van onderzoek veel specialistische kennis en zijn ze daardoor moeilijk voor een breed publiek toegankelijk te maken. Deze twee goede gewoontes lijken dan ook op gespannen voet met elkaar te staan.

Hoe red je je daar dan uit? Het is prachtig te zien hoe anderen met deze spagaat zijn omgegaan. Sommigen plaatsen hun vakgebied in historische context, en weten daarbij een perfecte balans te vinden tussen de degelijkheid van jaartallen en de luchtigheid van anekdotes. Anderen leggen de nadruk op het maatschappelijk belang van hun bezigheden, en doen hun uiterste best een beeld te schetsen waarin hun vakgebied de spil is waar alles om draait om de wereld verder te helpen. Weer anderen weiden uit over de grote uitdagingen in hun vakgebied, of ontvouwen hun visie over beleidszaken. Bij al deze invalshoeken voel ik mijzelf echter niet thuis. Niet dat ik dergelijke opvattingen over mijn vak niet zou hebben, maar ik wil iets van mijn vak zelf laten zien: echt iets van de inhoud, ook al is het maar een klein snippertje. Ik zal mij richten op een stukje van mijn onderzoeksgebied dat zich leent om aanschouwelijk te maken, en waar in versimpelde vorm iets over te vertellen valt zonder veel voorkennis te vereisen. Natuurlijk brengt dat wel enige vertekening van het onderwerp met zich mee, nou ja, dat is dan maar zo. Het doen van wetenschappelijk onderzoek is een soort ontdekkingstocht. Zelf heb ik mogen ervaren dat je onderweg allerlei spannende en verrassende dingen tegenkomt. Graag wil ik proberen een tipje van de sluier van enkele van die verrassingen op te lichten en met u te delen.

HERSCHRIJVEN, TERMINATIE EN BEWIJZEN

Laten we ons een heel simpele eendimensionale wereld voorstellen. Deze wereld bestaat alleen maar uit een eindig rijtje knikkers: witte en blauwe. Ook de ontwikkelingen in deze eenvoudige wereld voltrekken zich volgens een eenvoudig patroon: als er zich rechts van een blauwe knikker een witte knikker bevindt, dan mogen die blauwe en witte knikker van plaats verwisselen. De evolutie in deze wereld laat zich beschrijven als een *herschrijfregel*: we zoeken een patroon, de *linkerkant* van de regel, in dit geval blauw-wit, en vervangen dat door de *rechterkant* van de regel, in dit geval wit-blauw. Zo'n herschrijfregel schrijven we met een pijl, met links van de pijl de linkerkant, en rechts van de pijl, u raadt het al, de rechterkant. Als we wit en blauw afkorten tot w en b ziet dit er als volgt uit:

bw → wb

De vraag die we ons nu gaan stellen is de volgende: kan dit onbeperkt doorgaan of zal dit uiteindelijk altijd een keer stoppen? In het gegeven voorbeeld ziet u dat na een tijdje alle witte knikkers naar links geschoven zijn en alle blauwe naar rechts, en het proces stopt, want nergens staat meer een witte knikker rechts van een blauwe:

bbww
bwbw
wbbw
wbwb
wwbb

Maar is dit nu altijd het geval, onafhankelijk van de vraag hoe we beginnen, en welke strategie van herschrijven we volgen? Het lijkt er wel op: steeds schuiven witte knikkers naar links en blauwe naar rechts, en dit gaat door tot we een groep witte knikkers overhouden met rechts daarvan alleen maar blauwe, en we niet verder kunnen herschrijven. Als het proces altijd stopt, onafhankelijk van de vraag met welke eindige rij knikkers we beginnen of welke strategie we volgen, dan zeggen we dat het systeem *termineert*.

Laten we een klein stapje verder gaan. We houden de eenvoudige wereld met witte en blauwe knikkers, maar de herschijfregel wordt ietsje ingewikkelder: de linkerkant blijft dezelfde maar de rechterkant bestaat nu uit twee witte knikkers gevolgd door een blauwe knikker.

bw → wwb

Herschrijven is dus nu niet alleen het omwisselen van een blauwe knikker gevolgd door een witte, maar ook nog het toevoegen van een extra witte knikker uit een reservevoorraadje knikkers. Ook hier zien we bij een voorbeeld dat dit termineert, stopt, en het argument is hetzelfde als zonet: de witte knikkers schuiven naar links en de blauwe naar rechts net zolang tot we alleen nog witte knikkers hebben gevolgd door blauwe en we niet verder kunnen herschrijven.

Vol goede moed doen we er nog maar een schepje bovenop. De linkerkant blijft blauw-wit, maar de rechterkant wordt wit-wit-blauw-blauw.

bw → wwbb

Herschrijven betekent nu dus dat je niet alleen blauw-wit verwisselt, maar dat je links een extra witte en rechts een extra blauwe knikker toevoegt. Gezien het patroon wit-wit-blauw-blauw verwachten we dat ook nu de witte knikkers naar links schuiven en de blauwe naar rechts net zolang tot we alleen nog witte knikkers hebben gevolgd door

blauwe en we niet verder kunnen herschrijven, het systeem dus termineert. Maar is dat nu wel zo? Laten we eens beginnen met blauw-wit-wit, dan zie je dat je na twee stappen terechtkomt op een rijtje waarin datzelfde patroon blauw-wit-wit weer voorkomt.

bww
wwbbw
wwbwwbb

Op deze manier kun je het herschrijven dus onbeperkt voortzetten, en zien we dat het systeem *niet* termineert.

Wat leren we hier nu van? Dat als we zeker willen weten of een systeem wel of niet termineert, we niet kunnen vertrouwen op een plausibel maar vaag argument van knikkers die naar links of rechts schuiven, maar dat we echt op zoek moeten naar een hard *bewijs*. Dit geldt overigens niet alleen voor herschrijven en terminatie, maar voor elke bewering in de wiskunde of informatica waarvan we zeker willen weten of die waar is of niet. Ik weet dat het moeten geven van wiskundige bewijzen bij sommigen minder prettige herinneringen oproept aan onbegrijpelijke proefwerken waar met veel moeite een vijf of een zes uit geperst moest worden. Toch is het geven van een bewijs het enige gereedschap dat er bestaat om een precies geformuleerde bewering helemaal hard te maken. Het mooie van een bewijs is dat het iedereen, zowel jezelf als ieder ander, op een volstrekt objectieve manier kan overtuigen van de geldigheid van de bewering, en dat er nooit aanleiding zal zijn om die geldigheid in twijfel te trekken. We gaan op zoek naar bewijs, eigenlijk net zoals een rechter of detective dat doet, maar we stellen wel de hoogst mogelijke eisen aan de kwaliteit van zo'n bewijs, veel hoger dan in de rechtspraak haalbaar is.

TECHNIEKEN VOOR TERMINATIEBEWIJZEN

Hoe kunnen we nu een bewijs geven van terminatie van onze herschrijfsystemen? Dan hebben we technieken nodig waarmee we kunnen concluderen dat iets niet oneindig lang door kan gaan. Een van die technieken is gebaseerd op het principe van een vermageringsmethode, erg populair in deze tijd waarin overgewicht veel voorkomt. Stel je een vermageringsmethode voor waarbij je elke maand gegarandeerd minstens een kilogram afvalt. Laten we ervan uitgaan dat deze bewering klopt, wat hebben we daar dan aan? We kunnen dan concluderen dat dit termineert: dit kan niet oneindig lang doorgaan. Hoe groot het aanvankelijke gewicht ook is, ook al ben je zwaarder dan een olifant, je kunt maar een eindig aantal keren minstens een kilogram afvallen, want je gewicht blijft altijd minstens nul kilogram. Dit principe gaan we nu op herschrijven toepassen. We gaan een gewicht toekennen, een of ander natuurlijk getal, aan een rijtje knikkers, en wel zo dat elke keer als je een herschrijfstap doet, het gewicht minstens een afneemt.

Net als bij de vermageringsmethode kan dit dan maar eindig vaak gedaan worden. Als het ons lukt zo'n gewicht te definiëren, hebben we daarmee een *bewijs* dat het herschrijfsysteem termineert. Laten we ons eerste voorbeeld er weer eens bij pakken: blauw-wit gaat naar wit-blauw. Het gewicht van een rijtje knikkers berekenen we als volgt:

- we doorlopen de knikkers van rechts naar links, beginnend met gewicht 0;
- elke witte knikker voegt 1 toe aan het gewicht;
- elke blauwe knikker verdubbelt het gewicht.

Het totale gewicht is dan het verkregen gewicht als we helemaal links zijn aangekomen.

Omdat we het gewicht bepalen van rechts naar links, en blauw altijd verdubbelt, wordt de witte knikker in het patroon blauw-wit wel verdubbeld en in het patroon wit-blauw niet. Op grond hiervan kun je laten zien dat het gewicht altijd kleiner wordt als je blauw-wit vervangt door wit-blauw. En dit kun je helemaal hard maken, echt in een bewijs omzetten. Dit geeft een terminatiebewijs voor ons eerste voorbeeld. Hier zien we een voorbeeld van het afnemen van gewicht: **wbww** heeft gewicht 5 als we rechts beginnen met 0: dan maakt de meest rechtse **w** daar 1 van, de w daarnaast maakt daar een 2 van, de b verdubbelt dat tot 4, en tenslotte maakt de meest linker **w** daar 5 van. Als we dit herschrijven naar **wwbw** blijkt het totale gewicht van het resultaat 4 te zijn: inderdaad kleiner dan 5.

Ook voor ons tweede voorbeeld, blauw-wit vervangen door wit-wit-blauw, is zo'n bewijs te maken, het enige verschil is dat we dan blauw niet moeten laten verdubbelen, maar verdrievoudigen. Voor het derde voorbeeld lukt het echter met geen mogelijkheid, en dat is maar goed ook, want we hadden juist gezien dat terminatie daarvoor niet geldt: daar bestaat dus ook geen terminatiebewijs voor.

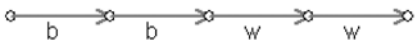
Zou dat nu altijd lukken: als we een terminerend herschrijfsysteem hebben, is er dan ook altijd een eenvoudig terminatiebewijs met gewichten voor te vinden? Nee, dat lukt niet. Je kunt zelfs bewijzen dat dit *onbeslisbaar* is: je kunt bewijzen dat er geen programma bestaat dat een herschrijfsysteem inleest en dan altijd vaststelt of het wel of niet termineert. Een geruststellende gedachte als dit je onderzoeksgebied is: je weet zeker dat je je pensioen kunt halen met het steeds maar weer zoeken en vinden van steeds maar weer sterkere technieken om terminatie te bewijzen. Ik zal nu twee eenvoudig ogende herschrijfsystemen laten zien die wel termineren, maar waarvoor dat lastig te bewijzen is, en ik zal schetsen wat er bij die bewijzen komt kijken. De onderliggende ideeën zijn in samenwerking met diverse collega's ontwikkeld; in het bijzonder wil ik in dit verband noemen: Alfons Geser, Dieter Hofbauer en Johannes Waldmann.

BEWIJZEN MET AUTOMATEN

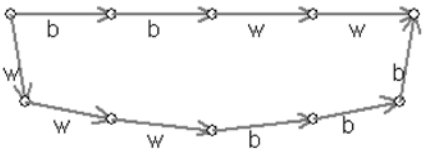
De eerste van de twee voorbeelden oogt net zo als de voorbeelden die we al gezien hebben: een blauw-wit patroon wordt vervangen door een wit-blauw patroon. Nu is de regel: twee blauwe-twee witte wordt vervangen door drie witte-drie blauwe.

bbww → wwwbbb

Hiermee vermeerdert elke stap zowel het aantal witte als het aantal blauwe, en hierdoor lukt het niet om met eenvoudige gewichten een terminatiebewijs te vinden. Maar toch termineert dit voorbeeld. We zullen een andere techniek nodig hebben om dat te bewijzen. Een techniek die daarvoor werkt en die ik nu ga schetsen is gebaseerd op *automaten*: de *match bound*-techniek. Bij automaten kun je denken aan een koffieautomaat: je kunt op allerlei knopjes drukken, en als je dat doet springt de automaat van de ene toestand naar de andere. De hele automaat laat zich dan beschrijven als een stel rondjes (de toestanden) met pijlen ertussen, en bij elke pijl staat de naam van een knopje. Bij het naspelen van herschrijven met deze regel heb je twee knopjes: een wit en een blauw. We zeggen dat de automaat *gesloten* is onder herschrijven als altijd als je van een toestand naar een andere toestand kan komen door de linkerkant van een regel te doorlopen, dat ook kan door de rechterkant te doorlopen. Zo is de volgende automaat niet gesloten onder herschrijven:



want je kunt van links naar rechts lopen waarbij je **bbww** doorloopt, maar niet waarbij je **wwwbbb** doorloopt. Als we nu als volgt een paar toestanden en pijlen toevoegen:



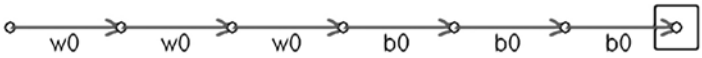
is de automaat wel gesloten onder herschrijven met betrekking tot ons voorbeeld.

Maar wat helpt dat nu om een terminatiebewijs te vinden? Daartoe moeten we nog twee uitbreidingen doen van het idee van het naspelen van herschrijven in een automaat. We gaan uiteindelijk aannemen dat het herschrijfsysteem een oneindige berekening toelaat en willen daar een tegenspraak uit afleiden. Eerst gaan we die hypothetische oneindige berekening een beetje oppoetsen. Je kunt laten zien dat je dan ook een oneindige berekening hebt met een meest linkse stap die ertoe doet. Alles wat links

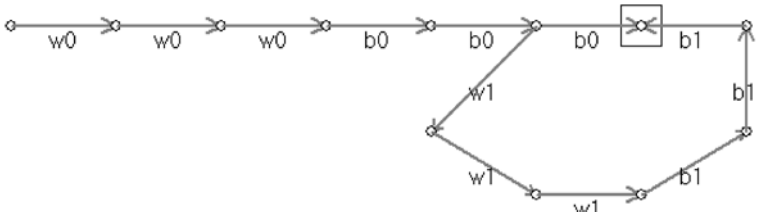
daarvan gebeurt kunnen we negeren, en alles wat rechts daarvan gebeurt creëren we onderweg. Dat betekent dat we een witte bron invoeren, genoteerd als een blokje □. Zo'n *witte bron* kan tijdens het herschrijven desgewenst witte en blauwe knikkers tevoorschijn toveren, en is daarmee een soort tegenovergestelde van een zwart gat. We modelleren dit door een paar regels toe te voegen: naast **bbww → wwwbbb** hebben we ook nog:

bbw□→ wwwbbb
bb□→ wwwwbbb
b□→ wwwbbb

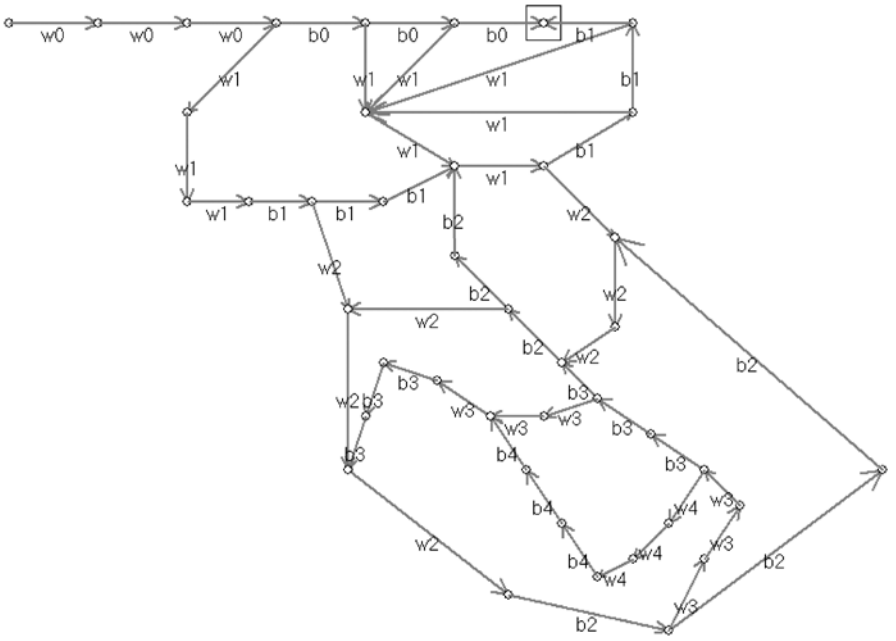
De observatie is nu dat als het systeem niet termineert, er een oneindige berekening is die begint met de rechterkant: drie witte knikkers gevolgd door drie blauwe, gevolgd door de witte bron. Om hier uiteindelijk een terminatiebewijs mee te kunnen geven moeten we nog een technische aanpassing doen: we moeten bijhouden hoe oud de knikkers zijn. Hierbij beginnen we met knikkers van leeftijd nul, en elke keer als we een stap doen, wordt de leeftijd van elk van de nieuw gecreëerde knikkers eentje hoger dan de jongste van de knikkers die we vervangen hebben. Nu beginnen we met de automaat die alleen de rij beschrijft bestaande uit drie blauwe knikkers gevolgd door drie witte, allemaal van leeftijd nul, gevolgd door de witte bron. Deze willen we uitbreiden tot hij gesloten is onder herschrijven. Dat kunnen we doen door op zoek te gaan naar patronen die overeenkomen met linkerkanten, en dan te checken of er ook een pad is overeenkomend met de rechterkant. Zo niet, dan voegen we dat toe. In ons voorbeeld begint dat als volgt:



Rechts zien we **bo** gevolgd door een □, met herschrijven wordt dit vervangen door **w1w1w1b1b1b1**, dit kunnen we doen door de automaat als volgt uit te breiden:



Als we dit systematisch voortzetten blijkt dat we na een flink aantal stappen een automaat hebben die gesloten is onder herschrijven, met in totaal 42 toestanden:



Hierin is de hoogst voorkomende leeftijd vier: de *match bound*. Wat kunnen we hier nu mee? De hypothetische oneindige berekening is geheel in deze automaat na te spelen, en daarom weten we ook dat in deze oneindige berekening een knikker nooit een leeftijd krijgt van meer dan vier. En uit deze begrensdheid van leeftijd kunnen we uiteindelijk bewijzen dat het herschrijfsysteem termineert [2,3].

Is zo'n bewijs nou mensenwerk of is een computer er essentieel voor? Het ontwikkelen van de theorie, met automaten, witte bronnen en leeftijden, is puur mensenwerk. Als je dit voor het eerst ziet is het best ingewikkeld en zeker niet tot in detail te volgen, al was het alleen maar omdat ik die details ook niet gegeven heb. Maar ik hoop toch dat ik een gevoel heb kunnen geven dat als je hier dieper in duikt, deze dingen voor je gaan leven, dat je er iets bij ziet, compleet met menselijke intuïtie en emotie. Maar het uiteindelijk bouwen van de automaat is iets wat een computer veel beter kan dan een mens: sneller en betrouwbaarder. In dit voorbeeld met 42 toestanden zou je je nog kunnen voorstellen dat je het met de hand doet, maar er zijn ook voorbeelden waarbij de automaten veel groter zijn. Bovendien wil je niet voor elk voorbeeld opnieuw aan het werk moeten, maar wil je een implementatie van de algemene techniek, en het zoeken en opbouwen van de automaat door een computer laten doen. Diverse tools, waaronder het door mijzelf ontwikkelde TORPA [3], vinden het zonet geschetste bewijs volledig

automatisch in een fractie van een seconde, en doen hetzelfde kunstje ook automatisch voor een groot aantal andere herschrijfsystemen. Het zonet geschetste bewijs met die automaat met 42 toestanden heb ik helemaal niet zelf gevonden, nee, ik heb het resultaat van TORPA geanalyseerd en daaruit de opbouw van de automaat gereconstrueerd. Erop terug kijkend is het eigenlijk best verbazend dat er zo'n ingewikkelde techniek voor nodig is om terminatie van zo'n uiterst simpel ogend herschrijfsysteem van slechts een regel te bewijzen. Er zijn wel andere bewijzen voor bekend, maar die zijn allemaal ook vrij ingewikkeld, en om te automatiseren is deze *match bound*-techniek toch het meest geschikt.

DE MATRIXMETHODE

Dan komen we nu aan het tweede voorbeeld van een herschrijfsysteem waarvan terminatie lastig te bewijzen is. Nu zijn er drie kleuren knikkers: naast blauw en wit ook nog rood. Er zijn drie herschrijfregels: elke keer als twee knikkers van dezelfde kleur naast elkaar staan, mogen ze vervangen worden door twee knikkers van de andere twee kleuren, in de volgorde van onze vlag. Dus: rood-rood wordt wit-blauw, wit-wit wordt rood-blauw, en blauw-blauw wordt rood-wit.

rr → wb
ww → rb
bb → rw

Termineert dit of niet? Tot voor enkele jaren was dit onbekend. Hoewel er toen ook al krachtige technieken waren om terminatie te bewijzen, en krachtige tools om deze technieken automatisch toe te passen, faalden alle technieken en tools op dit voorbeeld. Inmiddels is er een techniek waarmee het wel lukt, bedacht door Dieter Hofbauer en Johannes Waldmann [4,5]: een variant op het idee van gewichten waarmee we begonnen. Voor dit voorbeeld is het gewicht niet een getal, nee, het is een rijtje van vier getallen. Verder houden we wel het principe aan dat we terminatie gaan bewijzen door te laten zien dat elke keer als we een herschrijfstap doen, het gewicht kleiner wordt. Maar wat betekent kleiner als het gewicht niet een getal is maar een rijtje van vier getallen? We spreken af dat we vooral kijken naar het eerste van de vier getallen, dat echt kleiner moet worden, en dat de andere drie niet groter mogen worden. Dus:

$(x_1,x_2,x_3,x_4) > (y_1,y_2,y_3,y_4) \quad \leftrightarrow$
$x_1 > y_1$ en $x_2 \geq y_2$ en $x_3 \geq y_3$ en $x_4 \geq y_4$.

Ook nu geldt dat 'kleiner maken' niet onbeperkt door kan gaan: kijk maar naar het eerste van de vier getallen. Dit is precies wat we nodig hebben om terminatie uit afname van gewicht te kunnen concluderen.

Als we nu de gewichten **[r]**, **[w]**, **[b]** voor **r**, **w**, **b** als volgt berekenen:

[r] $(x_1,x_2,x_3,x_4) = (x_1 + 3x_4 + 1, 2x_3 + x_4, x_2 + x_4 + 1, 0)$
[w] $(x_1,x_2,x_3,x_4) = (x_1 + 2x_2, 2x_2 + x_4 + 2, x_2, 0)$
[b] $(x_1,x_2,x_3,x_4) = (x_1 + x_4 + 1, x_4, x_2 + x_4 + 3, 2x_2)$

blijkt dat op magische wijze voor elk viertal getallen (x_1,x_2,x_3,x_4) geldt:

[r] [r] $(x_1,x_2,x_3,x_4) >$ **[w] [b]** (x_1,x_2,x_3,x_4),
[w] [w] $(x_1,x_2,x_3,x_4) >$ **[r] [b]** (x_1,x_2,x_3,x_4),
[b] [b] $(x_1,x_2,x_3,x_4) >$ **[r] [w]** (x_1,x_2,x_3,x_4),

waarmee de sleutel voor het terminatiebewijs gegeven is.

De uitdrukkingen voor **[r]**, **[w]**, **[b]** kunnen we ook schrijven met matrices:

$$[r]x = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad [w]x = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

$$[b]x = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

vandaar dat dit ook wel *matrixinterpretatie* genoemd wordt. In deze matrixnotatie hebben we:

$$[r][r]x = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 2 \\ 2 \\ 1 \\ 0 \end{pmatrix} > [w][b]x = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

voor elke vector x van positieve getallen, en net zo **[w][w]** $x >$ **[r][b]** x, en **[b][b]** $x >$ **[r][w]** x voor elke x. Voor het vinden van de goede matrices, dus die uiteindelijk linkerkant > rechterkant opleveren, is er nog wel het een en ander te kiezen, die vind je bepaald niet zomaar. Als je zomaar wat invult lukt het van geen kanten; de juiste keuze ligt

echt heel subtiel. In ons geval kiezen we voor elk van de drie symbolen een 4 bij 4 matrix plus nog vier getallen, dat zijn twintig getallen per symbool, dus 60 in totaal. Elk getal is 0, 1, 2 of 3, vier mogelijkheden dus. Als we van tevoren deze dimensie 4 vastleggen, en de vier mogelijkheden voor elk van de getallen, zijn er dus 4^{60} mogelijkheden om een matrixinterpretatie te kiezen. Dat is een ontzettend groot getal, veel te groot om al deze mogelijkheden uit te kunnen proberen. Hoe is bovenstaande matrixinterpretatie dan gevonden?

Uiteindelijk is dit gedaan met SAT: *satisfiability*, vervulbaarheid, een techniek om te kijken of een waarheidsformule in heel veel variabelen waar is of niet. Uiteindelijk blijkt het hele gereken met matrices, uitmondend in de eis dat elke linkerkant groter is dan de bijbehorende rechterkant, daarin uit te drukken. En niet alleen uit te drukken, de moderne SAT-tools zijn dusdanig krachtig dat die hiervoor geheel automatisch een vervulling vinden. Deze vervulling is dan weer terug te vertalen naar getallen in de matrices. Zo zijn de getallen in de bovenstaande matrices dan ook gevonden. Dit is een methode die alleen maar iets oplevert in hechte samenwerking met de brute kracht van een computer. En ook deze methode werkt niet alleen voor dit ene voorbeeld, maar is bruikbaar voor heel veel voorbeelden.

REFLECTIE OP DE METHODEN

We hebben nu twee vrij ingewikkelde terminatiebewijzen gezien: eentje met automaten en eentje met de matrixmethode. Voor allebei was het zoeken van het bewijs met de computer gedaan, terwijl de onderliggende theorie puur mensenwerk was. Vorig jaar stond mijn collega Herman Geuvers op deze zelfde plek zijn oratie te houden, en hij noemde twee redenen waarom je wiskundige bewijzen geeft:

(1) om zeker te weten dat je bewering waar is, en
(2) om te begrijpen waarom je bewering waar is.

Bij het soort bewijzen dat hier vandaag langsgekomen is staat dat eerste punt helemaal overeind: met zulke bewijzen weten we inderdaad heel zeker dat de gedane bewering waar is. Maar van de tweede reden blijft niet veel over: een bewijs dat gevonden is door de brute kracht van een computer in te zetten kan helemaal goed zijn, maar geeft vaak helemaal geen inzicht waarom de bewering waar is. Je zou computergegenereerde bewijzen kunnen vergelijken met computerschaak: logisch klopt het allemaal wel, maar de stijl van een menselijke strategie is ver te zoeken.

Er zijn nog veel meer technieken om terminatiebewijzen te geven en met de computer te zoeken. Vakgenoten zijn misschien verbaasd dat ik hier de basistechnieken *padordeningen* en *dependency pairs* niet eens heb genoemd. Welnu, ik zou deze technieken met alle plezier willen uitleggen, maar als ik dat nu zou doen komen we niet op tijd bij de receptie, en dat wil ik u niet aandoen. En zoals gezegd, hoeveel technieken we ook hebben, altijd zijn er voorbeelden waar alle bestaande technieken op falen. Het is

verrassend dat dat ook voor heel kleine systemen geldt. Een voorbeeld hiervan is het volgende:

bbw → wbwbb
ww →

De laatste regel heeft een lege rechterkant, en beschrijft dus dat twee opeenvolgende witte knikkers weggehaald mogen worden. Van dit zeer kleine herschrijfsysteem weet momenteel niemand ter wereld of dit termineert of niet.

Waarom doen we dit soort dingen allemaal? Is dit alleen maar gepuzzel, net zoals het oplossen van een sudoku? Eerlijk gezegd, toen ik in de jaren negentig hiermee begon [1] vond ik dit vrijetijdsgepuzzel het eigenlijk niet waard om resultaten over te publiceren. Maar toen het, gestimuleerd door collega´s, uiteindelijk toch tot publicaties kwam, bleken die door veel anderen opgepakt te worden. Er is veel voor te zeggen een onderwerp met veel diepgang dat diverse onderzoekers aanspreekt, alleen op grond daarvan als onderzoeksonderwerp te rechtvaardigen. In de geschiedenis van de wetenschap wemelt het van de voorbeelden van belangrijk onderzoek dat zo is begonnen en pas later zijn toepassingen heeft gekregen.

Toch bevindt dit terminatie-onderzoek zich wel in een breder kader. De voorbeelden die we hier gezien hebben zijn *stringherschrijfsystemen*: de regels gaan over rijtjes, strings, van knikkers, of wat voor symbolen dan ook. Er zijn nog veel meer smaken herschrijven. Dit stringherschrijven kunnen we zien als de vanillesmaak: de meest basale vorm. We kunnen ook termen gaan herschrijven of grafen. We kunnen condities aan de regels toevoegen of de regels prioriteiten geven of hogere orde regels bekijken. Deze rijkere soorten herschrijven zijn ook uitgebreid onderzocht en kunnen gezien worden als manieren om programma's te beschrijven. Terminatie van het herschrijfsysteem komt dan overeen met terminatie van het programma. Iedereen die wel eens programmeert maakt wel eens een foutje waardoor het programma in een eindeloze lus terechtkomt: het programma *hangt*, het termineert niet. Het zou mooi zijn als dit automatisch vast te stellen was. Net zoals je bij het compileren van een programma gewezen wordt op syntaxfouten, zoals het vergeten van een puntkomma, zou het uitermate prettig zijn als de aanwezigheid van dit soort eindeloze lussen ook automatisch vastgesteld kon worden. Het onderzoek op dit gebied staat nog in de kinderschoenen, maar de essentie hiervan is nauw verwant met terminatie van herschrijf–systemen.

Ook heel andere verbanden zijn er te geven. Een van de hoofdonderwerpen van de groep Grondslagen hier in Nijmegen, waarvan ik deel uit maak, is het met de computer verifiëren van wiskundige bewijzen. Traditioneel is een wiskundig bewijs een stuk tekst dat een redenering aangeeft, net zoals wij zonet een aantal bewijzen hebben gegeven, en dat vervolgens door een mens wordt gelezen, en die dat dan wel of niet accepteert als

een bewijs. Maar hoe weet je nu zeker dat zo'n bewijs goed is? Er zijn technieken voor om dat met de computer te checken: dan moet je zo'n bewijs in kleine stukjes opsplitsen, en die op de een of andere manier formaliseren.

Een heel belangrijke bewijstechniek is inductie: je hebt een notie van groot en klein, en je bewijst dat een eigenschap altijd geldt door aan te nemen dat de eigenschap voor kleinere instanties altijd geldt. Zo'n inductieprincipe is alleen maar correct als je die instanties van de eigenschap niet onbeperkt kleiner kunt maken: het proces van kleiner maken moet zelf termineren. Op deze manier is het geven van een terminatiebewijs een heel wezenlijk en natuurlijk onderdeel van de correctheid van een bewijs waarin inductie een rol speelt, en dat is op een heel breed scala van terreinen. Het toepassen van herschrijftechnieken op het met de computer formaliseren van bewijzen is dan ook de opdracht die ik bij mijn aanstelling hier aan de Radboud Universiteit heb meegekregen.

ONDERWIJS

Zowel mijn aanstelling hier aan de Radboud Universiteit als mijn aanstelling aan de Technische Universiteit Eindhoven behelst niet alleen onderzoek, maar ook onderwijs, en daarnaast nog een bijdrage aan bestuurlijke taken. Deze combinatie van onderwijs en onderzoek ervaar ik als een zeer vruchtbare. Er is geen meer gedegen manier om je de fundering van onderwerpen eigen te maken, of je die nu wel of niet in je onderzoek tegenkomt, dan door er les over te geven, en het geeft zeer veel voldoening om op deze manier bij te dragen aan de ontwikkeling van gemotiveerde en getalenteerde jonge mensen. Ik prijs mij dan ook zeer gelukkig met mijn huidige functie.

DANKWOORD

Tot slot van deze rede wil ik enkele mensen bedanken die hebben bijgedragen aan het hele proces op grond waarvan ik hier nu sta, en wel in chronologische volgorde. Allereerst wil ik mijn ouders noemen die mij gestimuleerd en in de gelegenheid gesteld hebben te studeren, terwijl ze zelf allebei na hun veertiende al van school af moesten. Voor veel mensen is het kunnen studeren tamelijk vanzelfsprekend, maar ook nu zijn er mensen die heel graag willen studeren maar voor wie de mogelijkheid daartoe allerminst vanzelfsprekend is, bijvoorbeeld onder vluchtelingen. Ik gun deze mensen van harte dezelfde kansen die ik heb gehad, vandaar dat ik voor deze bijzondere dag geen cadeau voor mezelf vraag, maar een gift voor UAF, een stichting die zich voor deze vluchtelingstudenten inzet.

Na mijn middelbare school ben ik in Groningen wiskunde gaan studeren, waar een wereld voor me open ging, en waarvoor ik de docenten daar nog zeer erkentelijk ben. Mijn promotie heb ik aan de Universiteit van Amsterdam gedaan onder begeleiding van Hendrik Lenstra aan wie ik veel verschuldigd ben. Als ik nu werk van studen-

ten rood becommentarieerd teruggeef, denk ik nog vaak aan hoe hij dat bij mij deed. Op dat moment was ik daar niet altijd blij mee, maar ik heb er veel van geleerd. Na een paar jaar bij Philips in Apeldoorn kwam ik bij de universitaire informatica in Utrecht terecht. Op zoek naar een interessant onderzoeksonderwerp kwam ik via het regelmatig bezoeken van het Term Rewriting Seminar aan de VU in Amsterdam in aanraking met het onderzoek in herschrijven, dat mij zeer aansprak en waarin ik me verder heb ontwikkeld. In het bijzonder wil ik Jan Willem Klop en Aart Middeldorp bedanken, hier beiden aanwezig, de laatste helemaal uit Oostenrijk overgekomen. Verder ben ik veel dank verschuldigd aan Jan Friso Groote, die mij eerst naar het CWI haalde en vervolgens in het jaar 2000 naar de Technische Universiteit Eindhoven.

De laatste stap is mijn benoeming tot hoogleraar voor een dag per week hier aan de Radboud Universiteit. Ik dank graag de Radboud Universiteit voor het in mij gestelde vertrouwen, in het bijzonder Bart Jacobs, Herman Geuvers en Henk Barendregt. Tenslotte wil ik mijn thuisbasis noemen. Tineke, Marieke en Wouter, wat jullie voor mij betekenen is niet in woorden uit te drukken.

Ik heb gezegd.

REFERENTIES

1 H. Zantema and A. Geser, 'A complete characterization of termination of $0^p 1^q \rightarrow 1^r 0^s$', in: *Proceedings of the 6th Conference on Rewriting Techniques and Applications*, 1995, editor J. Hsiang, Springer Lecture Notes in Computer Science, volume 914, pages 41-55

2 A. Geser, D. Hofbauer and J. Waldmann, 'Match bounded string rewriting systems' in: *Applicable Algebra in Engineering, Communication and Computing*, 2004, volume 15, pages 149-171

3 H. Zantema, 'Termination of String Rewriting Proved Automatically', in: Journal of *Automated Reasoning*, 2005, volume 34, pages 105-139

4 J. Endrullis, J. Waldmann and H. Zantema, 'Matrix Interpretations for Proving Termination of Term Rewriting', in: *Proceedings of the 3rd International Joint Conference on Automated Reasoning* (IJCAR '06), Seattle, USA, 2006, Springer Lecture Notes in Artificial Intelligence 4130, pages 574-588

5 D. Hofbauer and J. Waldmann, 'Termination of {aa → bc, bb → ac, cc → ab}' in: *Information Processing Letters*, 2006, volume 98, pages 156-158