

# Hartstocht om een ontbrekend argument

Wim H. Hesselink

23 februari 2011

## Geachte toehoorders,

Sommige dingen zijn waar. Dat zie je direct. Maar waarom is het waar? Kun je het bewijzen, weet je het intuïtief, of moet je het maar geloven?

Het is net zo met computerprogramma's. Sommige voldoen aan hun specificatie. Je kunt dat geloven. Je kunt het testen. Maar als je zeker wil zijn, heb je ook een bewijs nodig.

Ik heb deze problematiek als kernthema gekozen voor deze rede, die u verder kunt opvatten als een selectief verslag van mijn werkzame leven tot mijn 65ste verjaardag, die vandaag valt.

Wat is een bewijs? Een bewijs bestaat uit een rij van argumenten. Als er één ontbreekt, kan het hele bewijs als een kaartenhuis in elkaar storten. Vandaar de titel van deze voordracht:

*Hartstocht om een ontbrekend argument.*

## 1 Om te beginnen: helden uit het verleden

Sinds de griekse wiskundige Euclides in de derde eeuw vóór Christus de meetkunde systematisch opgebouwd heeft, weten we: als je iets in de wiskunde bewijzen wil, moet je axioma's hebben om op te bouwen.

Dit geldt ook voor de rekenkunde. Het favoriete axiomasysteem daarvoor is dat van Peano uit 1889. In 1931 bewees Kurt Gödel echter, dat elk axiomasysteem voor de rekenkunde onvolledig is in de zin dat er ware rekenkundige beweringen zijn, die niet met dat axiomasysteem bewezen kunnen worden [13, 3].

In het verlengde hiervan bedacht Alan Turing [15] in 1936 de Turingmachine als een denkmodel van de *computer* (een computer was in die tijd een mens die een berekening uitvoert volgens een gegeven voorschrift). Hij bewees dat er problemen zijn waarvoor je wel een voorschrift (computerprogramma) kunt schrijven dat in alle gevallen dat het antwoord "ja" moet zijn uiteindelijk "ja" laat afdrukken, terwijl er voor dat probleem geen voorschrift is dat in alle gevallen dat het antwoord "nee" moet zijn uiteindelijk "nee" laat afdrukken. Op deze manier weten we dus al 75 jaar dat computers sommige dingen **niet** kunnen.

Dit zuiver-wetenschappelijk onderzoek heeft toch maar mooi de computer opgeleverd, waarmee we ongelofelijk veel dingen **wel** kunnen doen.

## 2 De mythe van Sisyphus

Krap een jaar geleden nam Peter van Emde Boas afscheid als lector Mathematische Informatica aan de Universiteit van Amsterdam. In zijn afscheidsrede legde hij een één op één relatie tussen zijn diverse onderzoeksprojecten en de werken van Heracles, de held uit de Griekse mythologie. Toen ik deze vergelijking hoorde, bekwam mij het gevoel dat ik mijn activiteiten eerder met die van Sisyphus zou vergelijken. Sisyphus, voormalig koning van het Griekse Ephyra, was omdat hij de goden vaak te slim af was geweest veroordeeld om in de onderwereld ten eeuwige dagen tevergeefs een rotsblok een berg op te rollen.

Het is een bekend gevoel voor de ervaren docent: je probeert de studenten iets bij te brengen, je hebt dat vorig jaar ook gedaan, en nu kun je opnieuw beginnen.

## 3 Hoe het voor mij begon

Tussen 1970 en 1975 deed ik in Utrecht promotieonderzoek bij prof. Springer in de algebraïsche meetkunde. Dit is een uitgebreid vakgebied. Ik heb toen mijn hartstocht voor het ontbrekende argument moeten overwinnen. Wilde ik in dat gebied iets nieuws bereiken, dan moest ik voortbouwen op wat men al bewezen had, en niet alle bewijzen zelf willen nagaan.

Het hoogtepunt van mijn wiskundige carrière was in het najaar van 1979 de publicatie [7] van een artikel van mij in de *Inventiones*, een vooraanstaand wiskundetijdschrift. In januari 1980, echter, stierf ons derde kind Mark Hessel, nog voordat hij een maand oud was. Dit verlies raakte mij zo zeer, dat ik de aansluiting met het researchfront in mijn tak van wiskunde verloor. Dit was de aanleiding tot een carrièreswitch.

Ik volgde in 1982/1983 enkele informaticacolleges om computerprogramma's te kunnen schrijven voor de oplossing van een algebraïsch classificatieprobleem. Ik bemiddelde tevens als voorzitter van de onderwijscommissie van onze subfaculteit bij een conflict over het vak Voortgezet Programmeren, dat ik zelf dat jaar gelopen had. Studenten vonden namelijk, dat het vak te veel werk vergde.

Het jaar daarop zou dit vak gegeven worden door de nieuw aangestelde hoogleraar Bron. Enkele maanden na aanstelling bleek deze echter overspannen te zijn. Uit arren moede vroeg men mij, die het vak immers het jaar daarvoor gevolgd had, om dit vak te geven. Ik ging de uitdaging aan. Het was hard werken om de studenten voor te blijven, maar zeer bevredigend. De studenten waren veel meer betrokken bij dit vak dan ik gewoon was bij de wiskundecolleges waar ik ervaring mee had. Ze wilden het leren omdat zij hun practicumopgaven wilden

kunnen maken.

Hoewel mijn studenten er geen probleem mee hadden, bevatte de collegestof voor dit vak een stuk over abstracte datatypen, dat mij niet zinde. Ik heb mijn hartstocht voor het ontbrekende argument in deze eenvoudiger omgeving uit kunnen leven: het werd mijn eerste informatica-publicatie [8].

## 4 De overstap

Ik stapte in 1985 officieel over van wiskunde naar informatica. Ik durfde dit aan vanwege het elan van Jan van de Snepscheut die hier in 1984 hoogleraar informatica geworden was. De overstap is bezegeld met een sabbatical year in 1986/1987 aan de University of Texas at Austin bij de bekende Nederlandse informaticus Edsger W. Dijkstra, Jan's leermeester.

Ik had in juli 1985 een kennismakingsgesprek met Dijkstra in zijn huis in Nuenen. Ik trachtte hem in dit gesprek ertoe te brengen de informatica in te delen in deelgebieden, zoals ik dat zelf bij de wiskunde kende. Hij liet zich hiertoe niet verleiden: de informatica was voor hem ondeelbaar. De centrale vraag voor de informatica was om de complexiteit van onze eigen maaksels te bedwingen.

In het jaar dat ik in Austin was gaf Dijkstra een honours course “wiskundige methodologie” aan undergraduates. Ik heb die cursus gelopen. Voor mij als gepromoveerd wiskundige was dit een eigenaardige ervaring. Dijkstra wist weinig van wiskunde, maar hij dacht wel als een wiskundige. Een centraal punt was het vermijden of uitstellen van gevalsonderscheid.

Mijn wekelijks hoogtepunt dat jaar was Dijkstra's Tuesday Afternoon Club, met Dijkstra, Jay Misra, Tony Hoare (die dat jaar ook in Austin was), en diverse anderen. Ik herinner mij bv. nog hoe Amir Pnueli er zijn temporele logica verdedigde tegen de gecombineerde scepsis van Dijkstra en Hoare.

## 5 Programmeermethodologie

Tussen 1975 en 1990 hebben Dijkstra en zijn medewerkers in Eindhoven en Austin een techniek ontwikkeld om algoritmen of programma's *af te leiden* uit hun specificatie. Ik heb het van Jan van de Snepscheut geleerd. Ik kan het alleen voor sequentiële programma's, maar het kan dan zeer effectief zijn.

Toen Jan naar Caltech vertrokken was, heb ik een cursus programmacorrectheid opgezet waarin ik de mondeling overgeleverde kennis van programmaderivatie vastlegde. Ik breidde dit uit met recursieve procedures met parameters zoals in de programmeertaal Pascal. De zorgvuldige semantische onderbouwing van deze behandeling heeft mij diverse wetenschappelijke publicaties gekost (of opgeleverd) [9, 10, 11].

De techniek van programmaderivatie wordt nu al meer dan 20 jaar aan onze

studenten in het eerste of tweede jaar geleerd. Hartstocht om de ontbrekende details is hier leerdoel geworden. De computerprogramma's in dit vak zijn namelijk ingewikkeld genoeg om foutief te zijn als de argumentatie ontbreekt, terwijl het argumenteren erover wel degelijk te leren en te onderwijzen is.

## 5.1 Euclidische distance transformatie

Eén voorbeeld van de effectiviteit van programmaderivatie: rond 1997 kwam Arnold Meijster op een middag bij mij langs omdat hij een programma wilde hebben om de Euclidische distance transformatie efficiënt te berekenen. Wat dat is, doet hier niet ter zake, wel dat dit begrip al rond 1965 ingevoerd was, dat er in 1980 een goed, benaderend algoritme ontworpen was, maar dat een efficiënt exact algoritme ons in 1997 onbekend was. Ik kon het die middag niet maken; in feite had Arnold mij dezelfde vraag een paar maanden daarvoor ook gesteld, en toen kon ik het ook niet; maar de aanhouder wint.

Ik ging er die avond nog een keer goed voor zitten, en loste het met de techniek van programmaderivatie op. De volgende morgen bleek, dat Arnold de vorige middag na zijn bezoek aan mij ook bij mijn toenmalige promovendus Rutger Dijkstra was gaan buurten. Rutger had het dezelfde avond met dezelfde techniek ook opgelost. Later bleek dat we helaas *twee jaar te laat* waren. Efficiënte algoritmen voor de Euclidische distance transformatie waren reeds in 1995 gepubliceerd. We hebben om die reden ons algoritme alleen in conferentieproceedings [14] gepubliceerd en de submittie aan het toonaangevende tijdschrift PAMI teruggetrokken. Enkele jaren later zagen wij tot onze verbazing en ergernis, dat er toch een groep auteurs in geslaagd was om hun versie van dit algoritme in PAMI te publiceren.

## 6 Bestuur en organisatie

In het bestuurlijke circuit is er maar zelden belangstelling voor volledige argumentatie. Ik heb mijn hartstocht voor *ontbrekende argumenten* hier dus nauwelijks laten blijken.

Er was wel hartstocht om kwaliteit. Kwaliteit van onderwijs en onderzoek, kwaliteit van de docenten, kwaliteit van de arbeidsverhoudingen voor de staf, kwaliteit van de relatie tussen studenten en staf, en kwaliteit als leerdoel in het informaticaonderwijs. Uit hartstocht om kwaliteit heb ik met Sisyphus menig rotsblok de berg op gewenteld.

### 6.1 Een eerste visiting committee

Ik was in 1979/1980 secretaris van het bestuur van de subfaculteit Wiskunde, met Hendrik Hoogstraten als voorzitter. Omdat de faculteit van Wiskunde en

Natuurwetenschappen geen zicht had op wat er bij wiskunde gebeurde, werden we in januari 1980 onderworpen aan een zogenaamd “visiting committee”, een bestuurlijke noviteit die later op veel ruimer schaal is ingezet.

Het oordeel van het visiting committee was zeer positief. Opvallend was de aanbeveling om informatica uit de subfaculteit te halen en tot een facultaire vakgroep te maken. Dit is inderdaad gebeurd. Het had het goede gevolg dat de noodzakelijke groei van informatica niet uitsluitend ten koste van wiskunde ging. Het is overigens spoedig daarna weer teruggedraaid, maar de lucht was toen gezuiverd.

## 6.2 Informatica op en neer

Na mijn terugkomst uit Texas in 1987 was er veel veranderd. Roland Backhouse was erbij gekomen als hoogleraar, en Jan en Roland hadden de organisatie op zich genomen van een wetenschappelijke conferentie ter gelegenheid van het 375-jarig bestaan van de universiteit. Het werd “Mathematics of Program Construction”, juni 1989 [16]. De conferentie werd een succes: het was het begin van een rij van tot nu toe 10 MPC-conferenties.

Jan van de Snepscheut was echter intussen zo gefrustreerd geraakt door wat hij ervoer als gebrek aan medewerking hogerop in de universiteit dat hij in september 1989 naar Caltech vertrok. Zijn vertrek was een groot verlies voor de vakgroep Informatica. Hij was een inspirerende leider en een enthousiaste docent en onderzoeker.

In september 1990 vertrok ook Roland Backhouse, naar Eindhoven. De vakgroep was weer min of meer terug bij **af**. We waren beangstigend klein geworden.

## 6.3 Informatica uit het dal

Informatica kreeg in 1991 versterking van de nieuwe hoogleraren Nicolai Petkov en Gerard Renardel, in 1993 gevolgd door Ben Spaanenburg.

Na mijn benoeming tot hoogleraar in 1994 werd ik voorzitter van de vakgroep Informatica. Als ik er nu op terugkijk, was dit een goede tijd. Natuurlijk niet probleemloos, maar we hadden voldoende staf en goede arbeidsverhoudingen waarin goed onderwijs gegeven kon worden. En een min of meer democratische structuur waar ik me goed in thuis voelde. Het onderzoek was de verantwoordelijkheid van het IWI, daar kom ik zo op.

Mijn vakgroepsvoorzitterschap duurde tot 1997, toen de vakgroepen opgeheven werden en er een verticale bestuursstructuur werd ingevoerd. Dit maakte me erg ongelukkig. Nog niet eens zozeer omdat ik mijn invloed kwijt was, als wel, omdat de communicatie verdwenen was: ik wist niet meer wat er speelde, en niemand leek het te weten. De oude communicatiestructuren waren opgeheven en nieuwe moesten nog uitgevonden worden.

## 6.4 Onderzoekscholen

Het idee van de onderzoekscholen dateert uit 1991. Rond 1995 werd het serieus. Niemand wist precies de consequenties, maar het idee bestond dat je buiten de onderzoekscholen geen onderzoek meer zou kunnen doen. Ik werd dus lid van de Onderzoeksschool IPA, die in 1995 werd opgericht. Het IPA was (en is) vooral goed en inspirerend voor de deelnemende promovendi, maar ook voor mij was het een goede gelegenheid voor contacten met collega's in het land en deelname aan inspirerende conferenties.

In 1995 werd hier in Groningen het onderzoeksinstituut van wiskunde en informatica IWI opgericht. Deze plaatselijke en landelijke samenwerkingsverbanden hebben elkaar nooit in de weg gezeten.

## 6.5 Commissie Van Lint

De onderzoeksvisitatie 1997 verliep voor het IWI zo slecht dat het College van Bestuur van de universiteit najaar 1998 een Commissie Van Lint instelde om te adviseren over hoe het verder moest. De commissie had natuurlijk tijd nodig. Het is achteraf gezien maar een jaar geweest, maar in mijn herinnering duurde het langer dan mijn driejarig vakgroepsvoorzitterschap. Er was grote onzekerheid bij de staf, en ik kon daar niets mee doen, omdat ik zelf geen informatie had.

Het rapport van de commissie kwam uit in juni 1999. Het bevatte een tiental aanbevelingen. Het opvallendst was wellicht dat er gewerkt moest worden aan een versterking van de *esprit de corps* binnen het IWI. Ter versterking van het onderzoek was het noodzakelijk nieuw talent aan te trekken, maar hierbij dienden strenge selectiecriteria te worden gehanteerd. Onderzoekers moesten alleen op grond van meetbare indicatoren als *fellow* tot het IWI worden toegelaten. Niet toegelaten onderzoekers zouden slechts lid kunnen worden van het onderwijsinstituut, en dan slechts een zeer beperkte onderzoekstaak overhouden. Ik kom nog terug op de gevolgen hiervan.

## 6.6 Opleidingsdirecteur

Ik was in het jaar 2005 adjunct-opleidingsdirecteur van Informatica. De onderwijsvisitatie van 2006 kwam dreigend op ons af. Eén van de eisen die daarbij aan ons onderwijs gesteld zouden worden, was dat het onderwijs ook in de bachelorfase gegeven dient te worden door docenten die bij het onderzoek betrokken zijn. Dit conflicteerde natuurlijk met de beslissing om docenten die niet tot het IWI waren toegelaten, een veel grotere onderwijslast te geven.

Los daarvan kwam ik eind 2005 in een conflict terecht. Omdat ik in mijn rol als kwaliteitsbewaker van het lopend onderwijs niet de erkenning kreeg waarop ik recht meende te hebben, zag ik me genooddaakt terug te treden.

Het was voor mij, en niet alleen voor mij, uiterst pijnlijk en onaangenaam. Jos Roerdink heeft, met de nodige hulp van anderen, de krachttoer verricht de zelfstudie zo te herschrijven dat we de onderwijsvisitatie 2006 goed doorgekomen zijn. Gerard Renardel is mij opgevolgd als adjunct-onderwijsdirecteur.

De docenten die niet tot het IWI zijn toegelaten, zijn inmiddels merendeels vertrokken of doorgestroomd naar andere functies binnen de universiteit. We zitten dienegevolge nu in een fase dat belangrijke bachelorvakken verzorgd worden door ingehuurde docenten. Het onderwijs moet namelijk wel gegeven worden en de onderzoekende staf heeft zijn handen vol, aan onderzoek en onderwijs, en aan het schrijven van researchproposals die een zeer kleine kans hebben om geld op te leveren.

## 6.7 Examencommissie

Ik weet niet meer precies wanneer ik toetrad tot de dagelijkse commissie voor de examens van wiskunde en informatica. Het moet vóór 2001 geweest zijn. In augustus 2001 was er namelijk een student die beroep aantekende tegen het cijfer  $6\frac{1}{2}$  dat zijn docent voor zijn afstudeerwerk gegeven had. Ik moest, als het enige aanwezige lid van de dagelijkse commissie, hem verwijzen naar het College van Beroep voor de Examens. Tevens moest intern gepoogd worden tot een schikking te komen. Dat laatste is uiteindelijk gelukt. Het cijfer  $6\frac{1}{2}$  is blijven staan.

Beroepszaken zijn voor de Examencommissie de krenten uit de pap. Het meeste werk bestond rond 2000 uit het beoordelen of een student die onder steeds wijzigende studieprogramma's resultaten behaald had, uiteindelijk voor een diploma in aanmerking kwam. Tegenwoordig zit er nog meer werk in het beoordelen van verzoeken van buitenlandse studenten voor toelating tot de master. Nieuwe regelgeving over de kwaliteitsbewaking van toetsen maakt de taak nog zwaarder, maar ik heb het voorzitterschap van de examencommissie dit najaar aan Michael Biehl overgedragen, zodat ik deze beker gelukkig aan mij voorbij kan laten gaan.

## 7 Onderzoek

Het is bij onderzoek dat de waarde van het ontbrekende argument het meest naar voren komt. Naast de hartstocht om het ontbrekende argument is er echter ook de liefde voor het uitmuntende argument. Wat dat betreft is het hoogtepunt in mijn herinneringen een lezing van George Kempf in Les Plans sur Baix, Zwitserland, in 1977. Zijn uitleg was heel warrig, zodat het merendeel van het publiek zeer wazig keek, maar ik zag plotseling de kern van zijn argument.

Dat was als volgt. Wilt u zich een aardappel voorstellen. We noemen de aardappel *convex* als voor elk tweetal punten van de aardappel het hele lijnstuk dat de twee punten verbindt ook in de aardappel ligt. De aardappel is dus “gevuld” en heeft geen inhammen. Het is dus geen banaan of peer, en zelfs een

appel heeft meestal een inham bij het steeltje. Kempf had het natuurlijk over willekeurige meetkundige figuren en niet speciaal over aardappels.

Maar goed, neem dus zo'n convexe aardappel in gedachten. Voor een punt  $P$  buiten de aardappel is er dan precies één punt in de aardappel dat het dichtst bij  $P$  ligt. Want als er twee punten  $A$  en  $B$  in de aardappel zijn die even dicht bij  $P$  liggen, dan ligt het middelpunt van het lijnstuk  $AB$  nog dichter bij  $P$ .

Ik kan me goed voorstellen dat u hiervoor niet uit uw dak gaat. In de context waarin Kempf dit gebruikte, was het echter een prachtig meetkundig argument met ver-reikende consequenties.

Dat ik het met Edsger Dijkstra goed kon vinden, komt ongetwijfeld doordat ik de liefde voor het zorgvuldige argument met hem deel. Dijkstra was wel kieskeuriger dan ik. Een argument moest mooi zijn, terwijl ik soms tevreden ben met een doeltreffend maar lelijk argument. We hebben in 1990 voor Dijkstra's 60ste verjaardag een boek geschreven onder de titel "Beauty is our business" [4]. Ik heb mijn bijdrage destijds zorgvuldig op Dijkstra's schoonheidseisen toegesneden.

## 7.1 Betekenis van programma's, engelen en demonen

Toen ik overstapte naar informatica was mijn eerste vraag naar de betekenis van programma's. Het ging mij om het opbouwen van een wiskundig fundament voor programmeermethodologie. Het is bij het programmeren belangrijk voortijdige ontwerpbeslissingen uit te stellen, maar dit leidt tot onzekerheid, zg. nondeterminisme. Er zijn twee vormen van nondeterminisme: engeliek nondeterminisme waarin we veronderstellen dat er verderop een zo gunstig mogelijke keuze gemaakt zal worden, en demonisch nondeterminisme waarin we op het ergste voorbereid moeten zijn. Je mag bij *engeliek* nondeterminisme verwachten, dat de nondeterministische keuze door je persoonlijke *beschermengel* gemaakt zal worden. Bij *demonisch* nondeterminisme kun je te maken krijgen met een *demon* die je dwars wil zitten.

Ik heb me hier vooral tussen 1986 en 1994 mee bezig gehouden. Rond 2007 kreeg ik een artikel van twee Engelsen (Morris en Tyrrell) te lezen waarin engeliek en demonisch nondeterminisme gecombineerd werden op een manier waarvan ik dacht: zoiets heb ik jaren geleden ook geprobeerd en het is me toen niet gelukt. Toen ik enkele maanden later opnieuw een artikel in deze geest te zien kreeg, ben ik mijn oude papieren gaan doorsnuffelen. Ik vond inderdaad een doodlopend conceptartikel in deze richting, maar ook, zowaar, een stuk uit januari 1987 waarin ik hetzelfde resultaat bereikte. Ik zat toen bij Dijkstra in Austin, met voortdurend nieuwe ideeën om me heen. Dit stuk was kennelijk ondergesneeuwd geraakt.

## 7.2 De leesgroep

Toen Jan van de Snepscheut in 1989 naar Caltech vertrok nam ik zijn Tuesday Afternoon Club over. Het was een groep van staffleden en studenten die wekelijks



bijeen kwam. In Dijkstra's Tuesday Afternoon Club ging het er om gezamenlijk al discussiërende onderzoek te doen. Deze invulling vraagt echter grote en onvermoeibare creativiteit van de organisator, iets wat Dijkstra kon, Jan eigenlijk net niet, en ik zeker niet. Ik heb het daarom de invulling gegeven van het gezamenlijk lezen van wetenschappelijke artikelen, meestal van elders, maar soms ook eigen artikelen, al dan niet in wording.

Deze leesgroep heeft 20 jaar bestaan. Ik heb er veel aan gehad. We hebben er stukken gelezen die mij tot nieuw onderzoek geïnspireerd hebben, en ik heb veel eigen conceptstukken eerst aan de snijtafel van deze leesgroep onderworpen. Het is voor een schrijver heel leerzaam en soms frustrerend om te zien hoe een ander probeert jouw tekst voor te lezen.

### 7.3 Concurrency

Sinds 1992 heb ik mij gespecialiseerd in het ontwerp en de correctheid van concurrent algoritmen.

Bij concurrency gaat het erom dat meerdere computers (of meerdere processen op één computer) met elkaar communiceren, bv. om een gemeenschappelijk doel te bereiken of om een gemeenschappelijke resource, zoals een database, te gebruiken en in stand te houden. Dit onderzoeksgebied wordt tegenwoordig steeds belangrijker, omdat de nieuwste computers multicore computers zijn. Dit wil zeggen dat ze eigenlijk uit verscheidene computers bestaan die hun taken efficiënt moeten verdelen. Dit maakt het programmeren lastiger omdat je nooit weet welke computer of welk proces de volgende stap zal doen.

De manier om toch zekerheid over aspecten van het gedrag van zo'n gedistribueerd systeem te krijgen is te analyseren welke eigenschappen *door geen enkele stap van het systeem onwaar gemaakt worden*. Deze eigenschappen heten de invarianten van het systeem. Omdat het gedistribueerde systeem op elk moment een groot aantal verschillende stappen kan doen, vergt het bewijs van invariantie het onderscheiden van veel verschillende gevallen.

Mensen zijn daar niet erg goed in, omdat we geneigd zijn onschuldig ogende gevallen over het hoofd te zien. Ik gebruik daarom een stellingbewijzer, dit is een computerprogramma waarmee je wiskundige stellingen kunt bewijzen. Het heeft geen echte intelligentie, maar wel een goed geheugen. Het kan dus uitstekend administreren wat bewezen is en wat nog bewezen moet worden.

Mijn koerswending naar concurrent programmeren kwam doordat we in de leesgroep een artikel [6] van Herlihy lazen, dat we niet helemaal konden volgen. Ik bestudeerde het toen zo diepgaand dat ik uiteindelijk tot een bewijs van een ander algoritme kwam dan dat van Herlihy. Ik schreef dit op en diende het in ter publicatie bij een tijdschrift. Eén van de referees van dat artikel uitte zoveel twijfel aan mijn bewijsvoering, dat ik het bewijs nog veel grondiger ben gaan aanpakken. Dit leidde weer tot een bewijs, maar de details waren voor mij zodanig dat ik telkens als ik aan het eind gekomen was niet meer wist of ik in

het begin wel alle gevallen behandeld had.

Dit heeft mij ertoe gebracht om een stellingbewijzer te gaan gebruiken. Ik had in Austin in 1986 college van J Moore gelopen over zijn stellingbewijzer NQTHM, maar ik had die stellingbewijzer nooit “aangeraakt”. Dat zou nu anders worden. Ik gaf mezelf zes maanden om NQTHM te gaan beheersen en er mijn algoritme mee te bewijzen. Gelukkig bleek J Moore in Austin steeds weer bereid mijn per e-mail gezonden vragen hierover te beantwoorden. Dankzij deze coaching op afstand gelukte het me inderdaad het algoritme te bewijzen.

Dat ik het prettig vind met een stellingbewijzer te werken, heeft natuurlijk alles te maken met mijn hartstocht voor het ontbrekende argument. Tussen collega’s informatici komt het maar weinig voor dat je een bewijs tot de laatste snik moet uitwerken. Er zijn zo vaak andere prioriteiten. Een stellingbewijzer is echter pas overtuigd als het goede argument geleverd is.

## 7.4 Jan Friso Groote en Gao Hui

Rond 1997 ontstond een samenwerking met Jan Friso Groote uit Eindhoven, eerst rond het thema van wachtvrij geheugenbeheer [12]. We probeerden daarna wachtvrije hash-tables te implementeren. Dit laatste project verzandde, toen ik de eerste helft met de stellingbewijzer NQTHM bewezen had en Jan Friso de tweede helft met zijn stellingbewijzer PVS wilde doen, maar vervolgens onderwijsdirecteur werd.

In de zomer van 2000 kwam Gao Hui uit China naar Groningen omdat zijn vrouw een hier aio-plaats bij scheikunde gekregen had. Gao wilde bij mij promoveren in de informatica, maar ik had geen positie voor hem. Hij wist toen een promotieplaats in de statistiek te krijgen. Het jaar daarop vertrokken zijn beide begeleiders echter naar het buitenland. Er is toen geregeld dat Gao zou omzwaaien naar de informatica om aan een promotie bij mij te werken.

Tijdens zijn promotieonderzoek heeft Gao onder meer het werk aan de wachtvrije hash-tables van Jan Friso en mij voltooid. Hij promoveerde in april 2005 op een mooi proefschrift met Jan Friso als tweede promotor en Maurice Herlihy in de promotiecommissie. We hadden hem voor vandaag uitgenodigd naar Groningen te komen, maar dat is helaas door visumproblemen mislukt.

## 7.5 Het voorspellen van berekeningen

Mijn eerste confrontatie met profetievariabelen was als ik me niet vergis tijdens de inaugurele rede [5] van Jan Friso, waarin hij het vermoeden uitte dat een correctheidsbewijs van Bloom’s algoritme profetievariabelen nodig zou hebben. Dat leek mij onzin, want ik had dit algoritme reeds zonder profetievariabelen bewezen.

Jan Friso bleek achteraf vermoedelijk gelijk te hebben, omdat ik in mijn bewijs uitgegaan was van een bewijsbeginsel waarvan ik later de correctheid met

profetievariabelen bewezen heb.

Wat zijn profetievariabelen? Het zijn hulpvariabelen die aan een computer-programma worden toegevoegd om als het ware de toekomstige ontwikkeling van de berekening te voorspellen. Ze zijn nuttig voor de bewijsvoering omtrent het programma, maar spelen geen rol in de berekening. Ze zijn ingevoerd in 1991 door Abadi en Lamport [1]. Ik had ze in 2002 nodig toen ik iets wilde bewijzen over transacties van databases. De profetievariabelen van Abadi en Lamport waren echter te beperkt voor het doel dat ik ermee had. Ik heb toen iets nog beters moeten verzinnen.

Het nadenken over deze problematiek leidt tot speculaties over predestinatie, met een opperwezen dat in één oogopslag de ontwikkeling van zijn schepping van begin tot eind kan overzien. De ontwikkeling van de schepping staat daarbij voor de executie van het programma. Ik werd hierbij geïnspireerd door de boeken over Dune van Frank Herbert. Eigenlijk is het helemaal niet moeilijk: je kunt een oneindige rij getallen heel goed als één ding beschouwen, zonder er mee rekening te houden dat de getallen één voor één worden opgeleverd door een sensor af te lezen of een dobbelsteen te werpen. De conceptuele moeilijkheid zit erin dat je voorstelt tijdens de ontwikkeling van het proces het uiteindelijk resultaat al te kennen.

## 7.6 Wederzijdse uitsluiting

*My latest collaboration with someone outside the institute is with Alex Aravind from British Columbia. I have e-mail contact with him since August 2008, but I met him yesterday for the first time, and he is now in the audience.*

*In 2007, Alex submitted a paper to Information Processing Letters. The paper contained a new interesting algorithm for mutual exclusion, but it was rejected by the journal because it had no convincing proof of correctness. I was one of the anonymous referees, but when the paper was rejected, I contacted the author, because I found it a beautiful algorithm and I could prove its correctness [2].*

*It is a good collaboration based on complementarity. Alex has very good ideas for algorithms, and I am often able to prove or refute these ideas. Let me explain his first algorithm in Dutch.*

Ik zal zijn eerste algoritme aan u uitleggen als een kinderpartijtje. De kinderen spelen in de tuin. Als ze dorst hebben, kunnen ze naar binnen om limonade te drinken. Er is echter maar één glas. Het probleem is te organiseren dat de kinderen het glas niet tegelijk pakken. Dit heet *wederzijdse uitsluiting* en het is voor computers (niet voor kinderen) in 1965 bedacht door de eerder genoemde Dijkstra.

Alex Aravind bedacht de volgende oplossing. Stel dat er  $N$  kinderen zijn. Zorg dan dat de kamer met het glas  $N$  hoeken heeft, genummerd van 0 tot  $N - 1$ . In hoek 0 staat het glas. Elk kind in hoek 0 mag het glas pakken. Als het kind gedronken heeft, gaat het weer buiten spelen om pas terug te komen als het weer

dorst heeft. Elk kind dat de kamer binnen komt, telt het aantal kinderen in de kamer. Als een kind in de kamer ziet, dat er niet meer dan  $k$  andere kinderen in de kamer zijn, mag het naar hoek  $k$  gaan. Een kind dat geen andere kinderen in de kamer ziet, mag dus direct naar hoek 0 gaan en drinken. Maar stel nu eens, dat drie kinderen min of meer gelijk binnen komen, alle drie dus twee andere kinderen zien en naar hoek twee gaan. Er is dan geen voortgang meer mogelijk. Alex loste dit op door in elke hoek met nummer 1 of hoger een stoel te zetten. Een kind in die hoek mag op de stoel gaan zitten. Als het dat doet, duwt het de huidige bezetter van de stoel af. Een kind dat van de stoel met nummer  $k$  afgeduwd is, mag naar hoek  $k - 1$ .

Deze organisatie heeft tengevolge, dat er nooit meer dan één kind tegelijk in hoek 0 is. Kinderen kunnen voordringen, maar zelfs een onhandig kind wordt nooit meer dan  $N$  keer gepasseerd.

Ik heb de bewijzen hiervoor geleverd. Het zijn ingenieuze bewijzen. Niet van grote schoonheid, zoals Dijkstra eigenlijk zou willen, maar wel doeltreffend en geverifieerd met de stellingbewijzer PVS.

## 8 Dankbetuigingen

Collega's, medewerkers, en studenten van informatica, wiskunde en kunstmatige intelligentie in Groningen, ik ben dankbaar dat ik hier 35 jaar met jullie heb mogen werken.

Ik ben het Groninger Universiteits Fonds dankbaar dat ik dit 16 jaar als hoogleraar heb mogen doen.

Ik kan de velen met wie ik in de loop der jaren heb samengewerkt, niet allen bedanken. De lijst is te lang. En dan nog zou achteraf blijken, dat ik mensen met wie ik intensief heb samengewerkt, vergeten was.

Eén uitzondering moet gemaakt worden: voor Gerard Renardel. Beste Gerard, je kwam in 1992 als hoogleraar naar Groningen om leiding te geven aan de fundamentele informatica, en dus ook aan mij. We hebben 18 jaar vruchtbaar en conflictloos samengewerkt. Ik ben je dankbaar voor deze goede samenwerking en voor de ruimte die je mij gegeven hebt.

Nog één uitzondering moet gemaakt worden: voor mijn vrouw Marijke, met wie ik over drie dagen 40 jaar getrouwd ben. Lieve Marijke, ik dank je zeer, zonder jou was dit alles niet mogelijk geweest.

## 9 Ten slotte

Ook het de hand schudden van een afscheid nemende hoogleraar is een wederzijds uitsluitingsprobleem. Ik stel voor dat we dit concurrent en nondeterministisch oplossen. Gaat u maar liever niet in de rij staan, maar neemt u drinken als er

drinken te krijgen is en schudt u mij de hand als die vrij is en u daar behoefte aan hebt.

## Referenties

- [1] M. Abadi and L. Lamport. The existence of refinement mappings. *Theor. Comput. Sci.*, 82:253–284, 1991.
- [2] A.A. Aravind and W.H. Hesselink. A queue based mutual exclusion algorithm. *Acta Inf.*, 46:73–86, 2009.
- [3] A. Doxiadis, C.H. Papadimitriou, A. Papadatos, and A. di Donna. *Logico-mix*. Dutch Media Uitgevers, 2009.
- [4] W.H.J. Feijen et al., editors. *Beauty is our business, a birthday salute to Edsger W. Dijkstra*. Springer, 1990.
- [5] J.F. Groote. We moeten software leren beheersen, 1999. Intreerede, Technische Universiteit Eindhoven.
- [6] M. Herlihy. Wait-free synchronization. *ACM Trans. Program. Lang. Syst.*, 13:124–149, 1991.
- [7] W.H. Hesselink. Desingularizations of varieties of nullforms. *Inventiones math.*, 55:141–163, 1979.
- [8] W.H. Hesselink. A mathematical approach to nondeterminism in data types. *ACM Trans. Program. Lang. Syst.*, 10:87–117, 1988.
- [9] W.H. Hesselink. *Programs, Recursion and Unbounded Choice, predicate transformation semantics and transformation rules*. Cambridge University Press, Cambridge, 1992. (Cambridge Tracts in Theoretical Computer Science 27).
- [10] W.H. Hesselink. Proof rules for recursive procedures. *Formal Aspects of Comput.*, 5:554–570, 1993.
- [11] W.H. Hesselink. Predicate transformers for recursive procedures with local variables. *Formal Aspects of Computing*, 11:616–636, 1999.
- [12] W.H. Hesselink and J.F. Groote. Wait-free concurrent memory management by Create, and Read until Deletion (CaRuD). *Distr. Comput.*, 14:31–39, 2001.
- [13] D.R. Hofstadter. *Gödel, Escher, Bach: an eternal golden braid*. Vintage Books, 1979.

- [14] A. Meijster, J.B.T.M. Roerdink, and W.H. Hesselink. A general algorithm for computing distance transforms in linear time. In J. Goutsias, L. Vincent, and D.S. Bloomberg, editors, *Mathematical morphology and its applications to image and signal processing (Proc. 5th Int. Conf.)*, pages 331–340. Kluwer, 2000.
- [15] A.M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proc. of the London Mathematical Society* 2, 42:230–265, 1936.
- [16] J. van de Snepscheut, editor. *Mathematics of Program Construction*, volume 375 of *LNCS*. Springer V., 1988.