

Peter van Emde Boas

*Institute for Logic, Language and Computation
Universiteit van Amsterdam
p.vanemdeboas@uva.nl*

In Memoriam Theodorus Jozef Dekker (1927–2021)

Vooraanstaand numeriek wiskundige, en informaticus van het eerste uur

Dirk Dekker was tussen 1971 en 1992 hoogleraar numerieke wiskunde aan de subfaculteit Wiskunde (sinds 1987 subfaculteit Wiskunde en Informatica) van de Universiteit van Amsterdam. In die functie heeft hij een centrale rol gespeeld bij de ontwikkeling van de numerieke wiskunde en programmatuur in Nederland. Daarnaast heeft hij een grote bijdrage geleverd aan de introductie van de informatica als een zelfstandige studierichting. Zijn oud-collega Peter van Emde Boas kijkt terug op zijn leven en werk.

Theodorus Jozef Dekker, in de wandeling Dirk genoemd (of voor intimi Dorus) is geboren op 1 maart 1927 in Heerhugowaard. Hij was de jongste zoon in een katholieke agrarische familie. De oudere kinderen waren drie zusters en een broer. Zijn jeugd-jaren bracht hij door in deze agrarische gemeenschap, maar op de dorpschool viel reeds snel op dat hij beter was in leren dan in werken in het boerenbedrijf. Aldus kreeg de familie het advies van zijn onderwijzer (Meester Buis) om Dirk naar het Petrus Canisius Gymnasium in Alkmaar te sturen. De middelbareschoolperiode viel voor het grootste deel samen met de oorlogsjaren; hij behaalt het eindexamen alfa en bèta in 1945, hetgeen in de praktijk erop neerkomt dat hij het diploma behaalde zonder een eindexamen af te hoeven leggen. Dankzij zijn gunstige leeftijd blijven onaangenaamheden als de Arbeitseinsatz hem bespaard.

Als jongste zoon in een katholiek gezin wordt Dirk na zijn afgesloten gymnasiale

opleiding in navolging van zijn oudere broer op het spoor gezet van de priesteropleiding op het college Hageveld in Heemstede. Maar reeds na een jaar wordt het Dirk duidelijk dat het priesterschap niet zijn roeping was, en verlegt hij zijn pad naar een studie wiskunde aan de Universiteit van Amsterdam. Door gebruik te maken van studieuiststel voor militaire dienst ontkomt hij aan uitzending naar Nederlands Indië. Tijdens zijn studie komt hij in contact met de inspirerende hoogleraar Johannes de Groot en gaat hij diens vakgebied topologie beoefenen.

Na het behalen van het doctoraalexamen in 1954 wacht hem alsnog de militaire dienst. In afwachting van de opkomst gaat hij op advies van professor De Groot een korte periode werken op de Rekenafdeling van het Mathematisch Centrum, waarmee de eerste stappen worden gezet richting zijn latere loopbaan in de informatica. Wetenschappelijk rekenen gebeurde in die tijd nog met zware elektrische tafelrekenma-

chines van het soort dat alleen de oudere lezers nog zullen kennen. Na afloop van de militaire dienst krijgt hij een aanstelling als assistent bij professor De Groot en begint hij aan een promotieonderzoek in de zuivere wiskunde, meer in het bijzonder de topologie. Dit onderzoek resulteert in de dissertatie *Paradoxical Decompositions of Sets and Spaces* [4] onder leiding van pro-



Dirk bij zijn promotie op 17 december 1958



Dirk bij zijn oratie op 25 september 1972

fessor De Groot, dat hij op woensdag 17 december 1958 verdedigt in de aula van de Universiteit van Amsterdam, destijds nog gevestigd in de Oudemanhuispoort.

Dirk is inmiddels zelfstandig gaan wonen in Amsterdam. In 1957 krijgt hij een relatie met Mies (Maria Francisca Huberdina) van Heezik, met wie hij in 1958 trouwt. Zij zullen 56 jaar samenblijven tot haar overlijden in april 2014. Zij krijgen samen twee zonen (Jan 1961 en Niels 1963) en twee dochters (Annemarie 1965 en Eline 1968). Het gezin verhuist in 1964 naar Castricum om vijf jaar later te verhuizen naar het adres in Bakkum waar hij de rest van zijn leven heeft gewoond.



Dirk en Mies bij het emeritaatsymposium op 27 november 1992

Na zijn promotie blijft Dirk verbonden aan het Mathematisch Centrum bij de Rekenafdeling alwaar hij steeds nauwer betrokken wordt bij vroege ontwikkeling en gebruik van elektronische rekenapparatuur. Hij werkt aan programmatuur op de ARMAC en later op de Electrologica X1 en X8. Naast de ontwikkeling van programmatuur voor deze machines geeft hij ook cursussen voor de gebruikers ervan.

In 1969 krijgt Dirk een leeropdracht 'Computer science in het bijzonder de numerieke algebra' aan de Universiteit van Amsterdam. In 1971 wordt dit omgezet in een ordinariaat in de numerieke wiskunde dat hij tot 1992 (modulo enkele organisatorische transformaties) zal bekleden. Tussen 1971 en 1987 had Dirk ook een aanstelling aan de Interfaculteit Actuarial en Econometrie. Op 25 september 1972 houdt Dirk zijn oratie getiteld 'Automatisch Rekenen'. Tussen 1971 en 1992 vervult Dirk een veelheid aan organisatorische taken. Op 27 november 1992 geeft hij zijn afscheidscollege getiteld 'Tussen Wiskunde en Informatica'.

In het Album Academicum van de Universiteit (<https://albumacademicum.uva.nl>) staat Dekker vermeld als promotor van een zestal studenten: Piet Hemker (1977), Jacobus Bus (1980), Johannes Cuppen (1983), Robert Gmelig Meyling (1986), Walter Hoffman (1989) en Theodorus Ripmeester (1996). Maar volgens het Mathematics Genealogy Project (www.mathgenealogy.org) zijn er daarnaast nog twee promovendi: Jan Verwer (1977) en Kees Dekker (1980); deze twee studenten hebben in het Album

Academicum Piet van der Houwen als eerste promotor.

Twee buitenlandse verblijven onderbreken de Nederlandse aanstellingen: eerst een bezoek aan Berkeley in 1963 en later een bezoek aan Bell Labs in 1968.

Ook na zijn emeritaat blijft Dirk een actieve deelnemer aan wetenschappelijke activiteiten, ook na het overlijden van Mies op 19 april 2014. Zijn laatste publieke optreden is in 2018 tijdens een symposium ter gelegenheid van de overdracht van een (gedeeltelijke) X8-machine aan het Boerhaave Museum in Leiden.

Dirk overlijdt op 25 november 2021.

Zuivere wiskunde

Het proefschrift van Dirk handelt over generalisaties van de Banach–Tarski-paradox. Deze paradox handelt over een opsplitsing van een tweedimensionale sfeer, respectievelijk de bol in drie dimensies in een aantal stukken die via meetkundige isometrieën kunnen worden samengevoegd tot twee sferen, respectievelijk bollen. Dit heeft als direct gevolg dat niet alle verzamelingen meetbaar kunnen zijn in de gebruikelijk zin, want ons maatbegrip is invariant onder meetkundige isometrieën.

Het resultaat berust op het bestaan van vrije niet commutatieve ondergroepen van de groep van alle isometrieën van rang groter gelijk twee. Dirks proefschrift handelt dan ook grotendeels over algebra: de expliciete constructie van vrije niet commutatieve ondergroepen met een verzameling generatoren gelijkmatig met de reële getallen. Hiermee laat de stelling van Banach en Tarski zich op diverse wijzen generaliseren. In de driedimensionale ruimte is het mogelijk om transformaties te gebruiken die dekpuntsvrij zijn, waarmee een onelegant onderdeel van de oorspronkelijke constructie (een speciale rol voor een aftelbaar stukje van de ruimte) verdwijnt. In hoofdstuk 2 van zijn proefschrift komen ook topologische eigenschappen van de fragmenten aan de orde.

Vanuit hedendaags gezichtspunt is het een kort proefschrift: 46 pagina's. En uiteraard zijn er bij het proefschrift twaalf stellingen toegevoegd. Ik citeer stelling X:

"De vraag of een automatische rekenmachine kan denken, is in wezen de vraag of het menselijk denken een automatisch proces is. Deze laatste vraag nu kan op wijsgerige gronden ontkennend worden beantwoord."



Algemeen Wiskunde Colloquium, 7 februari 1979

U ziet dat de discussie over de sterke AI geen recent verschijnsel is. De resultaten uit het proefschrift zijn daarnaast gepubliceerd in een achttal artikelen, waarvan drie gezamenlijk met professor J. de Groot.

Aan het einde van zijn loopbaan keert Dirk terug tot de zuivere wiskunde. De ouderen onder u zullen zich wellicht het tafelkleedje herinneren vervaardigd ter gelegenheid van het IMU-congres in 1954 in Amsterdam dat een weergave vertoont van de priemgetallen in de ring van de gehele getallen van Gauss ($\mathbb{Z}[i]$). (De schrijver van dit artikel heeft thuis een parketvloer laten maken van dit patroon.) Deze patronen zijn ook te definiëren voor andere kwadratische getallenringen. Dirk heeft in zijn latere jaren een veelheid van deze patronen uitgerekend en daar plaatjes van gemaakt. Het geval $\mathbb{Z}[\sqrt{2}]$ vormt de basis van het omslag van de gedrukte versie van zijn afscheidscollege. Publicaties hierover zijn verschenen in de *CWI Quarterly* [11] en in 2010 op arXiv [12].

Programmatuurontwikkeling

Twee centrale thema's in het wetenschappelijke rekenen zijn de toepassing van methoden uit de lineaire algebra en het oplossen van differentiaalvergelijkingen. Deze twee onderwerpen vormden en vormen ook nu nog de kern van de numerieke wiskunde en algebra. Dirk heeft zich vanaf het vroegste begin met deze onderwerpen beziggehouden. Dat betreft zowel ontwikkeling van programmatuur en het geven van onderwijs aan gebruikers van

deze programmatuur. Een vroeg voorbeeld daarvan is de cursus programmeren voor de ARMAC uit 1959 [5]. De overgang van de wiskunde achter de gebruikte algoritmen naar de implementatie ervan in een programma roept nu eenmaal een veelheid aan problemen op: de algoritmen zelf zijn soms niet exact maar leveren benaderingen van het gewenste resultaat op, met een al dan niet voorspelbare precisie; de rekenapparatuur werkt met een getalrepresentatie in eindige precisie, waarbij slechts een fractie van de getallen exact wordt gerepresenteerd, en de gebruikte rekenkundige instructies zijn ook niet exact — soms omdat het exacte resultaat niet representeerbaar is, en soms vanwege een ongelukkige implementatie in de hardware van de machine.

Wat het laatste betreft konden wij ons in Amsterdam gelukkig prijzen met de destijds gebruikte Electrologica-machine EL X8 die gebruik maakte van de Grau-aritmetiek [20, 22]. Die leverde gegarandeerd bij ieder bewerking, hetzij het exacte resultaat, hetzij de beste benadering van het exacte resultaat op. Op dit punt waren deze machines destijds uniek in de wereld. Deze eigenschap is ook essentieel voor de correctheid van de methoden die Dirk in 1971 publiceert om de precisie van een computer te kunnen vergroten [8].

In de vroege periode (voor 1968) speelde bij de ontwikkeling van programmatuur een veelheid aan beperkingen een rol waar wij tegenwoordig geen weet meer van hebben. Machines waren langzaam en

hadden maar een zeer beperkt geheugen. Dekker beschrijft in zijn afscheidscollege hoe je bij het bewerken van matrices op de ARMAC zorgvuldig moest nagaan waar de gegevens werden opgeslagen, rekening houdende met de omloopsnelheid van het trommelgeheugen en het gedrag van twee snelle op ferrietkernen gebaseerde caches. Tegen de tijd dat programmatuur werd geschreven in Algol 60 waren dit soort problemen niet meer zo relevant; tegen de tijd dat algoritmen ontwikkeld werden voor parallelle machines kwamen zij in gewijzigde vorm weer boven water.

Een belangrijk aspect bij programmatuurontwikkeling is de keuze van de gebruikte programmeertaal. Als leerling van Van Wijngaarden en lid van de Europese gemeenschap van ontwikkelaars van programmatuur maakte Dekker gebruik van de taal Algol 60 vanaf het moment dat deze taal beschikbaar was in 1960. Dit in een tijd dat de groepen die grootschalig aan het rekenen waren zich primair oriënteerden op Fortran. De voordelen van Algol voor Dirks doelstellingen waren legio: exacte beschrijving, goede semantiek en ook het gebruik van recursieve functies was van groot nut. Helaas was de opvolger van Algol 60, Algol 68, een taal die mislukt is in haar schoonheid; te moeilijk voor de wereld om te begrijpen en te gebruiken.

Centraal bij de ontwikkeling van programmatuur stonden de aspecten van numerieke betrouwbaarheid, correctheid en herbruikbaarheid. Dirk heeft zich al vroeg ingezet voor de ontwikkeling van programmatheken, geïllustreerd door het in twee MC tracts gepubliceerde pakket voor lineaire algebra routines in Algol 60 [6, 7]. Toen uw auteur in 1969 in dienst trad van de universiteit als junior medewerker was een van zijn taken de begeleiding van een practicumgroep bij het college numerieke wiskunde van Dekker, waarbij dit pakket werd gebruikt. Ondanks alle voorzorgen getroffen om de correctheid van de afgedrukte routines te garanderen (zoals onder meer beschreven door Walter Hoffmann in zijn bijdrage in het *liber amicorum* [18]) liep ik in mijn groep toch tegen een fout in een van de routines aan; uiteraard bij het gebruik van de routine op een wijze waarvoor deze niet was bedoeld.

Daarnaast onderzocht Dirk ook de invloed van de eindige precisie van de getalrepresentatie en de gevolgen van de imperfectie van de geïmplementeerde arit-

metiek op de correctheid van programma-tuur [9,10].

Een numerieke procedure gekoppeld aan de naam Dekker is de procedure *zero-in*. De analyse leert dat een continue functie gedefinieerd over een gesloten interval waarvan de functiewaarden op de eindpunten een tegengesteld teken hebben, binnen dat interval een nulpunt moet hebben. De kunst is om het nulpunt te vinden. Bekende methoden hiervoor zijn bisectie, lineaire interpolatie, of (in het geval van een differentieerbare functie) Newtons algoritme. Echter ook hier geldt dat de duvel in de details zit. Je wilt voorkomen dat de routine in een loop raakt omdat het tussenpunt in de computeraritmiek samenvalt met een van de eindpunten, en je wilt ook niet dat de computer door nul gaat delen. Jacques Bus beschrijft in het *liber amicorum* [18] de totstandkoming van de publicatie over deze routine in een ACM tijdschrift [2] en in hetzelfde liber schrijft M. van Veldhuizen over een aanpassing van de procedure om deze nog robuuster tegen onoordeelkundig gebruik te maken.

Na 1980 kwam de ontwikkeling van vectormachines en andere supercomputers. Iedere nieuwe vorm van computerarchitectuur vraagt om een volledige herbezinning van de implementatie van de elementaire bewerkingen in de lineaire algebra. Dekker heeft zich gedurende de laatste tien jaar van zijn aanstelling en tijdens zijn emeritaat intensief bezig gehouden met programmatuur voor computers met parallelisme; zie onder andere de publicatie [23].

In zijn laatste jaren heeft Dekker in samenwerking met Jan Bergstra getracht te komen tot een machine-onafhankelijke definitie van het begrip *algoritme*. Dit heeft geleid tot een aantal manuscripten, maar helaas niet tot een gepubliceerd artikel [3]. Het probleem blijkt gewoon te moeilijk te zijn om tot een overtuigende oplossing te komen. Men kan het erover eens zijn dat een algoritme een voorschrift is om bepaalde handelingen uit te voeren, maar op welk abstractie niveau deze handelingen beschreven moeten worden is minder duidelijk. Als men werkt met toekenningsopdrachten van waarden aan variabelen voert men toch stiekem een abstract model van een computer in. Daarnaast is het duidelijk dat deze handelingen een doel moeten hebben. Een algoritme kan niet los worden gezien van de beoogde relatie tussen begin en eindtoestand, al dan niet

gelardeerd met wenselijke tussentoestan-den en bijbehorende invarianten. Dit type specificatie vormt daarmee een essentieel onderdeel van de beoogde definitie.

Het lastigste probleem betreft de equivalentie van algoritmen. Levert het verwisselen van twee toekenningsopdrachten zonder enige vorm van interactie (die dus in feite ook parallel kunnen worden uitgevoerd) een ander algoritme op? Uiteraard is dit afhankelijk van de vraag of de intermediaire situatie geobserveerd wordt in de vorm van een beschreven tussentoestand. Hiermee wordt de equivalentie afhankelijk van de gebruikte specificatie. Bovendien wil men ook nog een keer dat het equivalentiebegriff een congruentie vormt onder de operaties waarmee men algoritmen wil samenstellen uit deelalgoritmen, en daarvoor is geen passende definitie gevonden.

Algoritme voor wederzijdse uitsluiting

Dekker is in de kerninformatica vooral bekend geworden door de oplossing van een probleem in *concurrency control*; curieus genoeg is dit een onderwerp waarmee hij zich later praktisch niet heeft ingelaten — de oplossing is ook niet door hemzelf gepubliceerd, maar door Edsger W. Dijkstra [14–17]. Pas in 2018 schrijft Dirk er een artikel over [13] naar aanleiding van een voordracht die hij over zijn algoritme heeft gegeven in Leiden. Desalniettemin is Dekkers algoritme een beroemd resultaat dat in vele boeken over concurrency (maar ook over verificatie) is terug te vinden, zoals bijvoorbeeld in [1, hfdst. 3 en 4] of in [19].

De Electrologica X1 was de eerste computer die gebruikmaakte van het interrupt-mechanisme dat de computer in staat stelt om te blijven rekenen terwijl de randapparatuur bezig is met in- en uitvoer van gegevens. In die tijd rond 1960 toen computers nog alles behalve snel waren, leverde dit belangrijke tijdswinst op. Een nadeel was dat het veel lastiger werd om te begrijpen hoe de interactie tussen verschillende actieve processen verloopt: het probleem van de concurrency was geboren. Edsger W. Dijkstra was zich zeer bewust van de complexiteit van deze problemen en heeft zich daar actief mee beziggehouden.

In 1959 (volgens Dekker in 1960) formuleerde Dijkstra het probleem van de wederzijdse uitsluiting van twee naast elkaar lopende processen die beide herhaald een zogeheten kritieke sectie willen doorlopen, waarbij moet worden gegarandeerd dat

deze processen zich nooit tegelijkertijd binnen hun kritieke sectie bevinden. Denk aan een computer waarop twee processen rekenen en resultaten willen afdrucken; je wilt niet dat de resultaten door elkaar heen worden afgedrukt, dus het gebruik van de printer moet afwisselend exclusief aan een van beide processen worden toegewezen.

Ik onderneem een poging om het probleem en de oplossing in woorden te beschrijven; de programmatekst kan u vinden in de genoemde artikelen van Dijkstra, en in het artikel dat Dekker er zelf over heeft geschreven aan het einde van zijn leven [13].

Het is uiteraard nodig dat de twee processen kunnen communiceren. Dijkstra stelt daarvoor gedeelde variabelen beschikbaar met atomaire toegang: beide processen kunnen de variabele lezen en schrijven, waarbij de atomaire toegang garandeert dat de effecten zich altijd laten interpreteren als hadden beide processen hun acties in een of andere volgorde na elkaar uitgevoerd: een leesopdracht levert altijd een eerder door een van beide processen geschreven waarde op, en geen mengsel van twee gelijktijdig uitgevoerde schrijfoopdrachten. En ook in de situatie dat beide processen tegelijkertijd willen schrijven is het resultaat dat een van de bedoelde waarden in de variabele terecht komt (onbepaald welk van de twee), en geen onbedoeld mengsel van twee waarden. Het blijkt in de praktijk voldoende te zijn om slechts gebruik te maken van binnaire variabelen.



Voordracht over wederzijdse uitsluiting, Leiden, 24 maart 2018

De oplossing moet ook aan redelijkheidseisen voldoen. Het verbieden dat een van beide processen ooit toegang krijgt tot de kritieke sectie is een oplossing die wederzijdse uitsluiting garandeert, maar dat is duidelijk niet de bedoeling. Processen die hun kritieke sectie willen betreden moeten daartoe vroeger of later de kans krijgen, ongeacht der relatieve snelheid waarmee de processen hun programma uitvoeren. De processen mogen ook niet blijven hangen in een lus waar zij niet meer uit kunnen komen.

Dijkstra had dit probleem gesteld, nauwkeurig geformuleerd, en voorgelegd bij diverse collega's binnen de secties van de Rekenafdeling van het Mathematisch Centrum. Hij beschrijft diverse voorgestelde oplossingen waarvan hij laat zien dat ze niet werken, en hij twijfelde of het probleem wel oplosbaar was. Tot zijn verbazing kwam Dirk enkele dagen nadat hij over het probleem had gehoord met een oplossing.

Om de oplossing van Dirk te begrijpen is het nuttig om eerst te kijken naar wat niet werkt. Men kan de twee processen om de beurt het recht geven de kritieke sectie te betreden, maar dat is geen correcte oplossing; immers een proces dat zijn kritieke sectie heeft doorlopen en vervolgens het andere proces de beurt geeft kan in de situatie komen dat het opnieuw de kritieke sectie wil betreden terwijl het andere proces in die tijd geen enkele belangstelling heeft getoond om diens kritieke sectie te betreden, en dus ook niet de beurt heeft kunnen teruggeven.

De processen moeten daarom kunnen testen of de kritieke sectie van het andere proces vrij is op het moment dat zij hun kritieke sectie willen binnengaan. Beide processen krijgen daarvoor een binaire variabele die aangeeft dat zij belangstelling hebben; deze variabele is ook voor het andere proces leesbaar. En uiteraard dient een proces dat vaststelt dat de ander belangstelling heeft (en zich dus mogelijk reeds in de kritieke sectie bevindt), wachten tot de ander vertrokken is.

Het proces moet dus alvorens de kritieke sectie te betreden zelf uitspreken geïnteresseerd te zijn, en te controleren dat het andere proces niet geïnteresseerd is. Als deze actie en test samen atomair zouden kunnen worden uitgevoerd (en dus in het geval dat de test oplevert dat op dit moment geen toegang mogelijk is, er niets

wordt veranderd), zou het probleem daarmee zijn opgelost, maar helaas: het zijn twee acties en daartussenin kan het andere proces een willekeurig aantal stappen hebben gezet.

Als beide processen eerst zich aanmelden en dan checken of de kritieke sectie vrij is binnen het andere proces kan de situatie ontstaan dat beide processen vaststellen dat ze niet verder kunnen. Dit levert een dodelijke omarming op doordat beide processen op elkaar staan te wachten. Omgekeerd als de twee processen eerst testen en dan hun interesse publiceren kan de situatie ontstaan dat beide processen elkaar testen (en zien dat de weg vrij is) en daarna beide menen toegang tot de kritieke sectie te mogen claimen. Dit levert de te vermijden onveilige situatie op.

Dijkstra kiest derhalve toch voor het zich eerst aanmelden; echter als na aanmelding de test oplevert dat het andere proces ook geïnteresseerd is moet een extra actie worden ondernomen inhoudende dat het proces zich tijdelijk terugtrekt en wacht tot er wel ruimte is. Echter als beide processen dit doen is niet langer te garanderen dat een van beide processen ooit aan de beurt komt. Dijkstra vergelijkt dit met een tweetal mensen die herhaald elkaar proberen te bellen en elkaar in gesprek vinden en dan weer ophangen. Vroeger of later zal een van beide geluk hebben en de verbinding alsnog tot stand brengen, maar dat is een stochastisch proces en daarmee is het probleem niet op de gewenste wijze opgelost.

De gedachte achter Dekkers oplossing is een combinatie van het mechanisme om de processen elkaar om en om de beurt te geven (via een gedeelde atomaire variabele die zij beide kunnen lezen en schrijven), en de hierboven uitgelegde sluismethode. Echter het hebben van de beurt geeft geen vrije toegang tot de kritieke sectie, maar alleen maar het recht om met handhaving van de claim op de kritieke sectie te wachten tot het andere proces zijn kritieke sectie vrijgeeft. Het proces dat de beurt niet heeft moet zijn claim terugtrekken en daarna wachten tot het de beurt terugkrijgt (waarvoor het andere proces zorg zal dragen als het gereed is met diens kritieke sectie).

In zijn postume publicatie [13], gebaseerd op de voordracht die Dekker 24 maart 2018 in Leiden heeft gehouden, beschrijft hij alternatieve oplossingen die in de literatuur te vinden zijn en de nodige

generalisaties. Dijkstra [15,16] heeft zich door de in de oplossingen optredende wachtlussen laten inspireren tot de introductie van de zogeheten seinpalen (semaforen), waarmee dit type problemen systematisch kunnen worden opgelost.

Organisatie en bestuur

Toen Dekker in dienst kwam bij de Universiteit van Amsterdam waren de wiskundigen verdeeld over twee instituten: het Mathematisch Instituut en het Instituut voor Toepassingen van de Wiskunde. Daarnaast bestond er interfacultair instituut voor grondslagenonderzoek waaraan de logici mede verbonden waren en een instituut voor propeadeutische wiskunde, ooit opgericht door de scheikundigen, biologen, geologen en farmaceuten om hun wiskundeonderwijs te verzorgen; deze groepen waren ontevreden over het onderwijs dat zij vanuit de Subfaculteit Wiskunde en Natuurkunde kregen.

In de jaren zeventig ging de universitaire structuur op de helling. Het resultaat voor wat betreft de wiskundigen was de vorming van een Subfaculteit Wiskunde (dus niet langer gezamenlijk met natuur- en sterrenkunde). Deze subfaculteit kende negen vakgroepen, waaronder een vakgroep numerieke wiskunde en informatica. Als enige voltijds hoogleraar binnen die vakgroep was Dekker vanzelfsprekend de voorzitter van deze vakgroep. Naast hem was Van Wijngaarden als deeltijds hoogleraar actief, en er was een wisselende leeropdracht die in die periode voor vier jaar werd vervuld door Kruseman Aretz.

Uitbreiding van de vaste formatie verliep in de midden jaren zeventig moeizaam. Een extra lectoraat kon slechts verworven worden door dit te delen met de propeadeuten (dit is de positie waarop uw auteur in 1976/77 is aangesteld).

De jaren zeventig zijn ook de jaren van de radendemocratie. Na de invoering van dit systeem kende de Subfaculteit Wiskunde een raad bestaande uit zeven stafleden, zes studenten en één vertegenwoordiger van het (amper aanwezige) technische en administratieve personeel. Een centraal politiek twistpunt was de al dan niet verplichte invoering in het curriculum van een vak *wetenschap en samenleving*, met als afgeleid probleem de nadere invulling van de zetelverdeling in de raad; de zeven zetels voor de staf volgde automatisch uit de omvang van de raad want de wet schreef



Subfaculteitsbestuur Wiskunde 1977–1978, Judith van Witsen, Elly Dobber, Dirk Dekker, Gerard Foekema en Martje Roessingh

voor dat de staf precies 50 procent van de zetels moest bezetten, maar de verdeling tussen studenten en administratieve staf mocht nader worden ingevuld. Dit conflict resulteerde uiteindelijk in een patstelling waarbij geen van de hoogleraren meer bereid was het decanaat te vervullen. Als overgangsmaatregel trad Josef Steenbrink — destijds nog geen hoogleraar — op als voorzitter van een interim bestuur (met steun van de decaan van de faculteit Besters). Dit was een problematische situatie omdat hij bepaalde functies van het decanaat zoals het regelen en voorzitten van promoties niet mocht vervullen.

Dekker is instrumenteel geweest bij het herstel van de normale verhoudingen door zijn bereidheid om in 1977 de taken van Steenbrink te gaan overnemen. Volgens mijn fotocollectie is hij in januari 1978 voor het eerst als voorzitter in de raad opgetreden (zie de foto hierboven). Hij heeft deze functie met veel inzet vervuld tot in 1980.

Een veel fundamenteeler probleem diende zich in de tweede helft van de jaren zeventig aan: de positie van de informatica binnen de academische orde. De mensen die zich met informatica bezighielden waren verdeeld over allerlei instituten. Binnen de Faculteit Wiskunde en Natuurwetenschappen waren de grote computergebruikers te vinden bij natuur en scheikunde (denk aan de groeiende behoefte aan rekenkracht om de experimenten in de hoge energiefysica te kunnen interpreteren). En ook buiten de faculteit waren er informatica-activiteiten — denk aan de bestuurlijke informatica en

het op artificiële intelligentie gerichte werk bij de psychologen.

In de praktijk resulteerde dit vooral in op cultuurverschillen gebaseerde meningsverschillen over zowel de positie die de informatica binnen de universiteit zou moeten krijgen als de inhoud van het vak. Er was onenigheid over de voor het onderwijs te gebruiken programmeertalen (Algol versus Fortran). Was informatica gewoon wiskunde over een nieuw en ander onderwerp (zoals belichaamd in het vak theoretische informatica), stond het programmeren en systeemontwikkeling centraal of moest het zwaartepunt liggen bij de toepassingen, in het bijzonder bij de natuurkunde. Daarnaast speelde de vraag of informatica een aparte opleiding moest worden, dan wel dat binnen de bestaande studierichtingen op informatica gerichte varianten ontwikkeld moesten worden.

Eind jaren zeventig had een facultaire commissie (waarvan ondergetekende als secretaris mocht opdraven) nog geadviseerd tegen een zelfstandige studierichting, maar de landelijke ontwikkelingen haalden ons spoedig in. Er moesten zelfstandige informaticaopleidingen komen, hetgeen de kwestie oprise van het aantal en de plaats van deze nieuwe opleidingen. Oorspronkelijk werd gedacht aan vier plaatsen in het land, waarbij instellingen zouden samenwerken. De Universiteit van Amsterdam zou dan samenwerken met de Vrije Universiteit en de Universiteit Utrecht. Uiteindelijk bleef de samenwerking beperkt tot de UvA en de VU, mede ingegeven door het feit dat

ook met het bijeenzoeken van alle deskundigheid binnen deze twee instellingen het nog maar amper mogelijk bleek een passend curriculum aan te bieden. Opgemerkt dient te worden dat in 1979 en 1980 toen deze plannen moesten worden gerealiseerd binnen onze faculteit geen voltijds vertegenwoordiger te vinden was van wat destijds gezien werd als kerninformatica (programmatuur en bedrijfsystemen); De VU beschikte daar wel over in de personen van Reind van de Riet en Andy Tanenbaum.

De nieuwe studierichting die aan de UvA in 1980 van start ging (met in het eerste jaar nog een vijfjarig programma dat na 1981 werd gereduceerd tot vier jaar conform de vigerende regels) werd verzorgd door een divers gezelschap van docenten en medewerkers uit de subfaculteiten Wiskunde, Natuur- en Sterrenkunde, Scheikunde en Geologie. Dit gezelschap werd ondergebracht in een facultaire vakgroep informatica, waarvan Dekker in 1981 de eerste voorzitter werd. Deze groep behoorde niet tot een subfaculteit en ressorteerde rechtstreeks onder de faculteit. De leden van de vakgroep bleven in het begin officieel aangesteld bij de instelling vanwaaruit zij afkomstig waren. Vanuit deze startpositie kon begonnen worden met de werving van meer informatici. De eerste nieuwe hoogleraar was Bob Hertzberger die een sleutelrol had gespeeld bij de opzet van de nieuwe structuren vanuit zijn positie bij natuurkunde en het CERN. Enkele jaren later werd Jan Bergstra benoemd als programmatuurkundige. Bob Hertzberger nam in 1984 het voorzitterschap van de vakgroep over van Dekker.

In 1984 maakten formatieperikelen bij wiskunde het noodzakelijk om de positie van Dekker (en de mijne) alsnog over te plaatsen naar de facultaire vakgroep. Dit duurde echter niet lang; in 1987 werd de vakgroep weer samengevoegd met de Subfaculteit Wiskunde tot een nieuwe Faculteit Wiskunde en Informatica, waarbij de numerieke wiskunde en algebra werd ondergebracht bij een vakgroep systeemontwikkeling gerichte informatica, naast de vakgroepen wiskunde, logica en theoretische informatica, en programmatuurkunde. Dit was de structuur toen Dekker in 1992 met emeritaat ging. Nadien is in de jaren negentig een fusie gekomen met natuur- en sterrenkunde, om in de jaren nul van deze eeuw terug te keren tot een alomvattende Faculteit der Natuurwetenschappen,



Bezoek IFIP working group WG 2.5 aan Amsterdam op 24 mei 1977, W.S. Brown, P. van der Houwen, Dirk Dekker, B. Einarson, H.J. Stetter en Piet Hemker.

Wiskunde en Informatica — afgezien van de informatica lijkt dit dus op de structuur van vijftig jaar eerder.

Over het karakter van Dekker als bestuurder zijn de meningen van alle betrokkenen gelijklopend: hij was recht door zee, maakte geen vijanden, had een zekere mate van naïeve onhandigheid die soms aanleiding gaf tot verwarring, en hij had nooit een dubbele agenda.

Nog iets over internationale contacten. Dekkers werk op het gebied van numerieke programmatuur, in het bijzonder numerieke lineaire algebra bracht hem in contact met groepen die elders bezig waren met de productie van gestandaardiseerde programmabibliotheken. Te denken valt aan groepen zoals de Numerical Algorithms Group (NAG) die bij gelegenheid in Amsterdam op bezoek kwam.

Een ander internationaal gremium was de IFIP working group WG 2.5, die zich niet alleen bezighield met softwareontwikkeling voor numerieke toepassingen maar ook met de standaardisatie van de aritmetiek. Dekker groeide op in een tijd dat iedere machine er eigen ideeën op nahield over de rekenkundige bewerkingen. Het verhaal ging dat numerici rondliepen met een briefje met een lijstje rekenkundige formules; aan de hand van de uitkomsten bij het uitrekenen van deze formules konden zij de gebruikte machine herkennen. Inmiddels hebben wij de IEEE-standaard die aan deze minder gewenste toestanden een einde heeft gemaakt (zij het niet zo

mooi als de eerder vermeldde Grau-aritmetiek). Dekker was gastheer van een WG 2.5 bijeenkomst in Amsterdam in mei 1977, zie de foto hierboven.

Slotwoord

Ik wil aan het einde van dit artikel nog iets vermelden over onze onderlinge relatie.

Ondanks het leeftijdsverschil van ruim 18 jaar hebben wij een gedeeld verleden in de vorm van de invloed van de hoogleraar Johannes de Groot op onze loopbaan, en daarmee een gedeeld verleden in de topologie. Het was ook De Groot die mij na mijn deeltijds aanstelling aan de universi-

teit in 1969 adviseerde om ter verbreding van mijn werkterrein mijn diensten aan te bieden bij Dekker bij de begeleiding van zijn practica numerieke wiskunde — een vak waarmee ik mij eerder niet had ingelaten. De Groot kon destijds nog niet vermoeden dat ook ik de zuivere wiskunde de rug zou toekeren en zou eindigen als theoretisch informaticus.

Na mijn benoeming tot lector in 1976/77 waren wij collega proximus van elkaar. De beginperiode van de informatica aan de Universiteit van Amsterdam hebben wij gezamenlijk vorm helpen geven, uiteraard vanuit onze eigen verschillende invalshoeken. Na 1984 kwamen wij echter niet alleen in twee verschillende secties van de informatica terecht, maar ook op twee verschillende locaties — ik op het Roeterseiland en Dekker in het Wetenschappelijk Centrum Watergraafsmeer. Tegen de tijd dat alle groepen op de Roetersstraat moesten verhuizen naar het WCW (in 2009) was Dekker reeds lang en breed met emeritaat.

Ondanks onze totaal verschillende werkterreinen denk ik nog altijd met veel plezier terug aan de cursus programmeermethoden die wij gezamenlijk hebben verzorgd in de periode 1975–1977. Verschillende docenten konden daar enkele colleges geven over hun specialisme. Zo sprak Dekker over de invloed van eindige precisie op programmacorrectheid, en ik gaf een van mijn eerste colleges analyse van algoritmen. Intussen mochten de studenten werken aan serieuze programmeerprojecten, zoals het verkiezingsprogramma dat



Dirk verkent de nieuwbouw op het WCW, mei 2009

is ontwikkeld door Gijs Tuynman dat een einde moest maken aan het gebruik van een incorrect en daardoor onwettig pakket overgenomen van de TU Delft ter bepaling van de uitslag van de verkiezingen voor (sub)faculteitsraden volgens het systeem van de Enkelvoudig Overdraagbare Stem. Andere projecten bij deze cursus waren de eerste stappen van Arjen Lenstra (met Ruud Mak en A. Bos) op het gebied van

de polynoomfactorisatie, de implementatie van mijn priority queue door Rob Kaas en Erik Zijlstra en het kraken van het L-spel van de Bono door een viertal studenten waaronder onze latere KdV-directeur en KWG-voorzitter Jan Wiegerinck [21].

Dekker bleef ook na zijn emeritaat actief, en wij hebben elkaar ook regelmatig bij colloquia en congressen getroffen, naast diverse ontvangsten aan huis. Hij

was een zeer prettige collega om mee samen te werken. ☺

Dankwoord

Met dank aan de kinderen van Dirk voor het leveren van materiaal gebruikt voor dit artikel. De foto's van de promotie en oratie zijn door de familie ter beschikking gesteld. Ik ben ook dankbaar voor de commentaren op eerdere versies van dit artikel van Gerard Alberts, Krzysztof Apt en Jan Bergstra.

Referenties

- 1 M. Ben-Ari, *Principles of Concurrent and Distributed Programming*, second edition, Addison-Wesley, 2006.
- 2 J.C.P. Bus en T.J. Dekker, Two efficient algorithms with guaranteed convergence for finding a zero of a function, *ACM TOMS* 1 (1975), 330–345.
- 3 J.A. Bergstra en T.J. Dekker, Definition of algorithm and of equivalence of algorithms (2021), unpublished manuscript.
- 4 T.J. Dekker, *Paradoxical Decompositions of Sets and Spaces*, proefschrift, G. van Soest, Amsterdam, 1958.
- 5 T.J. Dekker, *Programming for the ARMAC, VI, Matrix Complex RAM*, report MR 30 (in het Nederlands), MC, 1959.
- 6 T.J. Dekker, *ALGOL 60 Procedures in Numerical Algebra, Part 1*, Mathematical Centre Tracts 22, 1968.
- 7 T.J. Dekker en W. Hoffmann, *ALGOL 60 Procedures in Numerical Algebra, Part 2*, Mathematical Centre Tracts 23, 1968.
- 8 T.J. Dekker, A floating-point technique for extending the available precision, *Num. Math.* 18 (1971), 224–242.
- 9 T.J. Dekker, Correctness proofs and machine arithmetic, in L.D. Fosdick, ed., *Performance Evaluation of Numerical Software*, North Holland, 1979, pp. 31–43.
- 10 T.J. Dekker, Program correctness and machine arithmetic, in P.C. Messina en A. Murli, eds., *Problems and Methodologies in Mathematical Software Production*, LNCS 142, Springer, 1982, pp. 48–80.
- 11 T.J. Dekker, Prime numbers in quadratic fields, *CWI Quarterly* 7 (1994), 367–394.
- 12 T.J. Dekker, Primes in quadratic fields (2010), arXiv:1001.5214v1.
- 13 T.J. Dekker, History of Dekker's algorithm for mutual exclusion – a personal view, in G. Alberts en J.F. Groote, eds., *Tales of Electrológica; Computers, Software and People*, History of Computing Series, Springer, 2022, pp. 113–123.
- 14 E.W. Dijkstra, Over de sequentialiteit van procesbeschrijvingen, *EWD* 35, ca. 1962–64, <https://www.cs.utexas.edu/~EWD>.
- 15 E.W. Dijkstra, Cooperating sequential processes, *EWD* 123, ca. 1964–67, <https://www.cs.utexas.edu/~EWD>.
- 16 E.W. Dijkstra, Cooperating sequential processes, in F. Genuys, ed., *Programming Languages*, Academic Press, 1965.
- 17 E.W. Dijkstra, Solution of a problem in concurrent programming control, *CACM* 17(8) (1965), 453–455.
- 18 P. van Emde Boas, P.W. Hemker, W. Hoffmann, P.J. van der Houwen en P.R. Pfluger, *Is er nog nieuws?*, liber amicorum aangeboden aan prof. dr. Th.J. Dekker, 1992.
- 19 N. Frances, *Fairness*, Texts and Monographs in Computer Science, Springer, 1986.
- 20 A.A. Grau, On a floating-point number representation for use with algorithmic languages, *CACM* 5(3) (1962), 160–161.
- 21 V.W. Gijlswijk, G.A.P. Kindervater, G.J. van Tubergen en J.J.O.O. Wiegerinck, *Computer Analysis of E. de Bono's L-game*, Report 76–18 MI, Universiteit van Amsterdam, 1976.
- 22 F.E.J. Kruseman Aretz, *Design and Correctness Proof of an Emulation of the Floating-Point Operations of the Electrologica X8: A Case Study*, report TU Eindhoven, 2010.
- 23 H.J.J. te Riele, T.J. Dekker en H.A. van der Vorst, eds., *Algorithms for Solving Numerical Linear Algebra Problems on Supercomputers*, North Holland, 1987.