

Precisie, computers en logica

Gerard R. Renardel de Lavalette

20 oktober 1992

Mijnheer de Rector, dames en heren,

met de wiskundigen hebben wij logici de betreuenswaardige omstandigheid gemeen dat het vrijwel ondoenlijk is leken een indruk te geven van de inhoud van ons onderzoek. Dat is de prijs die wij moeten betalen om in deze vakgebieden te vertoeven. Ik zal dan ook geen poging doen u uit te leggen wat voor een interessant onderzoek ik momenteel verricht, samen met Dick de Jongh en Lex Hendriks van de Universiteit van Amsterdam, naar exacte modellen voor eindige fragmenten van de intuïtionistische propositielogica. Dat is jammer, want het is een geslaagde symbiose tussen logische theorievorming en computerexperimenten die bovendien een aantal aardige plaatjes oplevert.

In plaats daarvan zal ik in meer algemene termen spreken over de logica (om precies te zijn: de mathematische logica) en u tevens kort vertellen hoe ik daarin terechtgekomen ben. Ook zal ik ingaan op de vraag welke rol de logica speelt en kan gaan spelen in andere gebieden van wetenschap en intellectuele bezigheid.

Een waarschuwing vooraf: mocht u een enkel punt in mijn betoog niet kunnen volgen, bedenk dan dat ik enige vaktermen toegevoegd heb met het doel ook mijn vakgenoten bij de les te houden. Zij zullen overigens opmerken dat ik op diverse plaatsen de wetenschappelijke feiten vereenvoudigd heb om de grote lijn van het verhaal niet voor scherpslijperij te hoeven onderbreken.

1 Precisie

Alvorens over de logica te spreken wil ik het even hebben over het begrip precisie. Hierbij doel ik niet op numerieke precisie ('cijfers achter de komma'), maar op wat ik begripsmatige precisie noem: zorgvuldigheid en nauwgezetheid bij het definiëren en beschrijven van begrippen. Bezie als voorbeeld de volgende definitie:

A is broer van B als A een man is en dezelfde ouders heeft als B.

Dit lijkt een aardige definitie, maar hij is onvolledig: er dient toegevoegd te worden dat A ongelijk is aan B, want anders is iedere man een broer van zichzelf! Dit lijkt een beetje flauw: immers, in de alledaagse praktijk zal de onvolledige definitie geen problemen opleveren. Anders wordt het als deze definitie wordt gebruikt in een computersysteem, bv. een adviessysteem voor het erfrecht. Dat kan tot ongewenst gedrag van het systeem leiden, doordat de computer de definitie letterlijk zal nemen. Denk maar aan de situatie van een gezin met drie zonen, waar bepaald moet worden hoeveel broers de oudste zoon heeft. Wij mensen hoeven daar nauwelijks over na te denken: twee natuurlijk! Echter, bij letterlijk gebruik van de definitie zal de computer de oudste zoon zelf ook meetellen en op drie uitkomen.

Het lijkt er aldus op dat het de computer is die de mens dwingt tot het hanteren van een grotere precisie dan voor intermenselijk gebruik vereist is. Het opbrengen van deze extra precisie wordt vaak als een last gezien, een inspanning die nu eenmaal

geleverd moet worden om die o zo domme computer duidelijk te kunnen maken wat er van hem verwacht wordt. In het vervolg van deze oratie zal ik beargumenteren dat precisie meer kan opleveren dan alleen computerprogramma's en dat de logica het vakgebied bij uitstek is om deze begripsmatige precisie vorm te geven. Maar eerst zal ik u verhalen hoe ik in dat vakgebied terecht ben gekomen.

2 Levensloop (deel 1)

2.1 Wiskunde

Op het gymnasium wist ik al vrij snel dat ik wiskunde wilde gaan studeren. Het ging me gemakkelijk af en ik beschouwde het als een eigen wereld waarin het goed toeven was: alles was helder en duidelijk, en als je heel goed keek kon je soms plotseling prachtige nieuwe dingen zien, een geweldige sensatie. Logica was nog niet in beeld: wat ik er over las in populair-wetenschappelijke pocketboeken kon mij niet bekoren, met uitzondering van de drogredenen in het boek van Stanley Jevons, bv.:

Niets gaat boven wijsheid,
droog brood is beter dan niets,
dus gaat droog brood boven wijsheid!

Veel plezier beleefde ik ook aan het wiskundetijdschrift Pythagoras voor middelbare scholieren. Het bracht mij in contact met wiskunde die op school niet gegeven werd. De getaltheorie was al snel mijn favoriet en ik knutselde uitgebreid met de rij van Fibonacci modulo p (p een priemgetal) en met formules voor sommen van machten van natuurlijke getallen. Tijdens mijn studie wiskunde aan de Universiteit van Amsterdam ontdekte ik dat de nieuwe stellingen en resultaten die ik gevonden meende te hebben vrijwel alle reeds lang bekend waren: in het beste geval vond ik ze terug in een leerboek, weggestopt in een halve alinea. De voorliefde voor getaltheorie taande.

2.2 Mathematische Logica en het Intuïtionisme

Om mijn kandidaatsexamen rond te krijgen deed ik nog even een tentamen Inleiding Mathematische Logica, een college gegeven door professor Löb. Een verrassend vak, anders dan de andere wiskundevakken, handelende over de propositiologica (de logica van beweringen en uitspraken) en de predikatenlogica (de logica van eigenschappen van objecten en relaties tussen objecten).

In mijn eerste doctoraaljaar besloot ik meer logicavakken te volgen: pas veel later zou ik beseffen dat de ruime keuzemogelijkheden op dat gebied uniek waren in Nederland. Het meest intrigeerde mij het vak Inleiding Intuïtionisme van professor Troelstra. Dat was even schrikken: had ik net bij Löb de klassieke tweewaardige logica geleerd (alles is waar of onwaar, kon het logischer?), blijkt er ene L.E.J. Brouwer uit Amsterdam in het begin van deze eeuw bedacht te hebben dat het anders moet! Hij had overigens, als een der grootste wiskundigen van zijn tijd, wel enig recht van spreken. Zijn argument loopt ongeveer als volgt.

Wiskunde, zo stelde hij, gaat over gedachtenconstructies zoals getallen, driehoeken, functies en dergelijke. Deze zaken bestaan niet in de reële wereld: het zijn constructies die wij in onze geest opbouwen. Wiskundige beweringen zijn niet *a priori* waar of onwaar: een bewering A is pas waar als we er een bewijs voor hebben geconstrueerd; A is pas onwaar als we een bewijs hebben geconstrueerd dat A tot tegenspraak leidt. Vanuit dit perspectief is het niet meer evident dat elke wiskundige bewering waar of onwaar is: immers, dat zou alleen het geval zijn als we over een universele methode beschikken om alle wiskundige problemen op te lossen, iets wat

bepaald niet het geval is. Gevolg: de klassieke tweewaardig logica is niet geschikt om het wiskundig redeneren te beschrijven.

Dit is, zeer in het kort, de filosofie achter het Intuïtionisme van Brouwer en ook van het sterk verwante Constructivisme. Overigens hebben zijn denkbeelden geen grote omwenteling in de wiskunde teweeg gebracht: constructieve wiskunde is momenteel een respectabel maar niet centraal deelgebied van de wiskunde. Meer belangstelling is er vanuit de logica en de laatste tijd ook vanuit de theoretische informatica.

Mij fascineerde vanaf de eerste kennismaking de intuïtionistische logica: een interessante en rijke theorie, complexer dan zijn klassieke tweewaardige tegenhanger. Mijn doctoraalscriptie was er aan gewijd en bevatte enkele bescheiden nieuwe resultaten. Na mijn doctoraalexamen verrichte ik door ZWO betaald promotieonderzoek bij Troelstra op het gebied van de intuïtionistische metamathematica: wiskunde over intuïtionistische wiskunde. Een nadere uitleg van dit interessante onderwerp laat ik hier maar achterwege. In plaats daarvan wil ik met u een blik werpen op de geschiedenis van de logica.

3 Geschiedenis van de logica tot 1930

3.1 Aristoteles

We beginnen bij Aristoteles. Hij beschouwde syllogismen, eenvoudige redeneringen zoals:

alle mensen zijn sterfelijk
Socrates is een mens
dus: Socrates is sterfelijk

Hierbij ging het hem niet om de conclusie (die immers al enige decennia daarvóór feitelijk aangetoond en door Plato indringend beschreven was), maar om het *mechanisme* dat hier aan het werk is. Dit syllogisme is een speciaal geval van een algemeen schema, te weten

alle A zijn B
C is een A
dus: C is een B

Wat we ook voor A, B en C invullen, steeds krijgen we een geldig syllogisme, bv.

alle katten zijn aaibaar
Tom Poes is een kat
dus: Tom Poes is aaibaar

Dat is de fundamentele ontdekking die op naam van Aristoteles staat: de correctheid van velerlei redeneringen is onafhankelijk van de inhoud van de termen die er in voorkomen en wordt geheel bepaald hun vorm. Onthoud dit onderscheid tussen vorm en inhoud, we komen er nog op terug. Aristoteles slaagde er overigens niet in alle redeneringen in syllogismen te vatten.

3.2 Frege

Het opmerkelijke van de geschiedenis van de logica is dat er tussen Aristoteles en de helft van de vorige eeuw betrekkelijk weinig nieuws te melden is. Tot die tijd heeft de logica niet veel meer techniek in huis dan de reeds door Aristoteles beschreven syllogismen en is zij daardoor nog lang niet in staat de verwachting waar te maken een enigermate volledige en bruikbare theorie van het redeneren te zijn. Vanaf

ca. 1850 kwam hier verandering in. Wiskundigen als De Morgan, Boole en Peirce leverden substantiële bijdragen aan het vak, maar hun werk wordt overschaduwd door dat van de Duitse wiskundige Gottlob Frege aan het einde van de vorige eeuw. Hij is de uitvinder van zowel de propositie- als de predikatenlogica, waarvan de laatste tot op heden een centrale plaats in het vakgebied inneemt. Frege breidde de predikatenlogica uit in de stijl van de verzamelingenleer — het geesteskind van een andere Duitse wiskundige, Georg Cantor — en hij gaf aan hoe de gehele wiskunde in dit logisch systeem gereconstrueerd kan worden. Een prestatie van formaat, ook na de dramatische ontwikkeling die nu volgt.

De Engelse wiskundige, logicus en wijsgeer Bertrand Russell ontdekte een fundamentele fout in het gedeelte van Freges systeem dat over verzamelingen ging. Deze fout hangt direct samen met de zogeheten Paradox van Russell waarvan ik u slechts de eveneens van Russell afkomstige parafrase noem. Er is een dorp met een kapper die alle mannen scheert die zichzelf niet scheren. Vraag: scheert deze kapper zichzelf of doet hij dat niet? Welk antwoord u ook kiest, er volgt tegenspraak.

3.3 Het programma van Hilbert

In dezelfde periode werden dergelijke paradoxen ook ontdekt in andere delen van de wiskunde waar algemene begrippen met grote abstractie behandeld werden. Men zou kunnen spreken van groeistuipen van een zich snel ontwikkelend vak in de vorm van interne tegenspraken. Dit bracht de wiskunde, bij monde van David Hilbert in 1900, tot introspectie: een zelfonderzoek moest uitwijzen dat de ‘gewone’ wiskunde, ontdaan van te ver doorgesloten abstractie, vrij van tegenspraak was. Dit zelfonderzoek bestond uit twee delen. Ten eerste een formalisering van de diverse deelgebieden van de wiskunde, d.w.z. een precieze beschrijving van alle wiskundige begrippen en toegestane bewijsprincipes. Voor deze formalisering bleek de predikatenlogica (uiteraard zonder Freges gewraakte uitbreiding) zeer geschikt. De tweede stap van het zelfonderzoek diende te bestaan uit een consistentiebewijs van alle geformaliseerde wiskundige theorieën, d.w.z. een bewijs dat tegenspraken niet voorkomen. Zo’n consistentiebewijs mocht slechts gebaseerd zijn op de meest elementaire wiskundige principes wier consistentie evident was. Dit grootse plan van formaliseren en de consistentie bewijzen staat bekend als Hilberts programma.

De afloop is bekend: het eerste deel (de formalisering van de wiskunde) werd een groot succes, het tweede deel (het bewijzen van de consistentie) een fiasco. Dit laatste is het gevolg van de Onvolledigheidsstellingen van de Oostenrijkse wiskundige Gödel, door hem in 1930 op 25-jarige leeftijd bewezen. De strekking van deze Onvolledigheidsstellingen is: elke fatsoenlijke wiskundige theorie is onvolledig, d.w.z. laat een aantal uitspraken onbeslist; een van die uitspraken betreft de consistentie van die theorie. Gevolg hiervan is: wil je de consistentie van die theorie toch bewijzen, dan heb je een nieuwe theorie nodig die sterker is (en dus een grotere kans op inconsistentie oplevert). Aldus raak je met het zoeken naar consistentiebewijzen alleen maar verder van huis: steeds sterkere theorieën wier consistentie met nog krachtiger middelen bewezen moet worden. De reductie tot iets eenvoudigs wordt daarmee definitief onbereikbaar.

3.4 Formalisering van de wiskunde

Laten we de succesvolle eerste stap van deze onderneming nog even bezien: de formalisering van de wiskunde in de logica. Het moge voor een buitenstaander verrassend lijken dat er aan de wiskunde nog iets te preciseren valt: immers, als de wiskunde al niet volkomen precies is, wat dan nog wel? Toch is de wiskunde niet altijd zo exact als het nu lijkt. Zo heeft het ca. twee millennia geduurd voordat Euclides’ axiomatisering van de meetkunde uitgebreid werd tot een axiomastelsel

waaruit alle stellingen van de meetkunde daadwerkelijk kunnen worden afgeleid. Ook het gebruik van de in veler ogen wat schimmige infinitesimalen door Newton en Leibniz bij het differentiëren en integreren van functies heeft twee eeuwen op een consistente formulering moeten wachten.

Deze problemen waren echter opgelost toen de formalisering van de wiskunde ter hand werd genomen, onder anderen door Russell en Whitehead in de *Principia Mathematica*. Daarbij bleek dat de paradoxen van Russell en anderen terug te voeren waren op vermenging van wiskundige objecten en de taal waarin over die objecten gesproken wordt. Een effectieve methode om deze bron van tegenspraken bloot te leggen bestaat uit het aanhouden van het onderscheid tussen vorm en inhoud dat we bij Aristoteles al zagen: vorm slaat hierbij op de taal van een wiskundige theorie, inhoud op de betekenis die aan die taal wordt toegekend. Dit wordt ook wel aangeduid als het onderscheid tussen taal en meta-taal: in de meta-taal kan over de taal worden gesproken, bv. om aan te geven hoe de betekenis wordt toegekend.

De huidige stand van zaken t.a.v. de formalisering van de wiskunde is als volgt: vrijwel de gehele wiskunde kan gereduceerd worden tot de verzamelingentheorie, deze verzamelingentheorie is naar tevredenheid geaxiomatiseerd in de predikatenlogica: deze axiomatisering staat bekend onder de naam ZF, naar de wiskundigen Zermelo en Fraenkel. Dit laat zich samenvatten in het adagium *kan het niet in ZF dan is het geen wiskunde* (dat u wel met een korreltje zout moet nemen).

Hier past overigens een relativerende opmerking. U kunt van de hiervoor geschetste ontwikkelingen in de grondslagen van de wiskunde licht de indruk overhouden van een crisis in de gehele wiskunde. Zo ernstig was het bepaald niet: voor het overgrote deel van de wiskunde was er niets ernstigs aan de hand en niemand maakte zich zorgen of $2 + 2$ nog wel 4 was en of de stelling van Pythagoras in gevaar verkeerde. Je zou daarom kunnen stellen dat het er wel op lijkt dat de grondslagen van de wiskunde zich niet onder maar naast de wiskunde bevinden.

De hierboven geschetste ontwikkelingen zijn dan ook van meer betekenis geweest voor de logica dan voor de wiskunde. De logica heeft niet alleen met succes de wiskundige methode overgenomen, ook diende zich in de wiskunde een geschikt, niet te weerbarstig object van studie aan om de messen der logische analyse op te slijpen.

4 Formele talen

Tot het precisie-instrumentarium dat in de logica wordt gehanteerd behoren de zogeheten formele talen. Formeel wordt hier gebruikt in de letterlijke betekenis: de vorm betreffend, zonder de connotatie stijf en vormelijk. Formele talen zijn nauwkeurig en ondubbelzinnig gedefinieerde kunstmatige talen, waarvan ik u een voorbeeld zal schetsen: de formele taal die de basis vormt van de propositielogica. Deze taal bestaat uit een aantal formules, en elke formule is een eindige rijtje symbolen. Als symbolen nemen we de letters van het alfabet en verder enkele logische symbolen waarvan we voor het gemak alleen de pijl in het verhaal betrekken. Met deze symbolen gaan we op een systematische manier formules maken, en wel als volgt. Ten eerste zijn alle letters formules. Verder spreken we af: uitgaande van twee reeds gevormde formules kunnen we een nieuwe formule maken door de ene formule achter de andere te schrijven, met een pijl ertussen. Aldus krijgen we een welomschreven verzameling formules, en we kunnen zelfs van elk rijtje symbolen vaststellen of het een formule is of niet. Sterker nog: het is zo elementair dat we het een computer kunnen uitleggen.

Dit is een eenvoudig voorbeeld van een formele taal: een systematisch gedefinieerde verzameling van eindige tekenrijtjes. Tot nu toe lijkt het op een curieus abstract

spelletje, leuk voor een computer, en de vraag dringt zich op: betekent dit ook nog iets, ofwel: welke inhoud hoort bij deze vorm? Antwoord: dat heb je zelf in de hand door de definitie van de betekenis (meestal semantiek genoemd) naar eigen inzicht in te richten (modulo enkele randvoorwaarden die ik buiten beschouwing laat). Deze scheiding tussen vorm en inhoud, ofwel tussen syntax en semantiek vormt de crux van formele talen.

Laten we deze formele taal eens vergelijken met een niet-formele taal: het Nederlands. We nemen daartoe de verzameling van alle correcte zinnen van het Nederlands in gedachten. Net als bij de formele taal hebben we dan te doen met eindige rijtjes symbolen: letters, spaties, leestekens. Het verschil met onze formele taal is echter dat we niet beschikken over een definitie van het Nederlands waarmee we goede van verkeerde zinnen kunnen onderscheiden. Het opmerkelijke is dat wij allemaal dat wel kunnen, althans tot op zekere hoogte. Geef ons, sprekers van de Nederlandse taal, een rijtje letters, spaties en leestekens en we zeggen in vrijwel alle gevallen direct en eenstemmig of het wel of niet een correcte Nederlandse zin is. Een enkele keer weten we het niet, zijn we het onderling niet eens of veranderen we na verloop van tijd van mening. Een ander verschil met formele talen is dat wij Nederlandse zinnen vrijwel altijd in direct verband met hun betekenis zien, een betekenis die dan ook niet achteraf daaraan is toegekend.

Deze verschillen verklaren ook het grote verschil tussen onze taalvermogens en die van een computer, geprogrammeerd volgens de huidige stand der techniek: wij onderscheiden gemakkelijk goede van verkeerde zinnen en begrijpen de betekenis, computers slagen daar nog nauwelijks in maar zijn — zoals iedere gebruiker van een tekstverwerker weet — wel goed in een aantal zuiver syntactische bewerkingen zoals het opzoeken en tellen van woorden.

Met een formele taal kunnen we ook formeel redeneren, en wel door een redeneersysteem vast te leggen. Interessant wordt het nu als we naar de relatie tussen betekenis en redeneersysteem kijken, in het geval dat de betekenis in termen van waarheid en onwaarheid is gegeven. Het ligt voor de hand de verzameling van ware formules te vergelijken met de verzameling van formules die conclusie zijn van formele redeneringen. Dit leidt tot twee complementaire vragen.

- Zijn alle conclusies van formele redeneringen waar?
- Kunnen alle ware formules verkregen worden als conclusie van een formele redenering?

De eerste vraag betreft de gezondheid van het formele redeneersysteem, de tweede de volledigheid. Idealiter is een formeel systeem gezond en volledig en komen waarheid en afleidbaarheid dus overeen. In de praktijk blijkt de volledigheid niet altijd haalbaar, maar is gezondheid wel een vereiste.

Over het bewijzen van deze eigenschappen voor gegeven formele systemen valt nog te vermelden dat gezondheid i.h.a. eenvoudig volgt door de axioma's en redeneerregels te inspecteren: de axioma's moeten waar zijn en de regels moeten waarheid behouden, d.w.z. van ware premissen tot ware conclusies leiden. Volledigheid is vaak meer werk: bij elke ware uitspraak moet een formele redenering worden gevonden. Dit kan soms alleen indirect aangetoond worden, met een argument uit het ongerijmde: toon van elke formule die niet formeel afleidbaar is dan hij niet waar is.

Tot zover mijn uiteenzetting over formele talen. Alvorens u te vertellen hoe ik zelf van de logica in de informatica terecht ben gekomen, maak ik deze overstap in mijn historisch overzicht.

5 Van Logica tot Informatica

De situatie in de jaren dertig is: de logica is in de voorliggende halve eeuw door wiskundigen getransformeerd van een uithoek in de wijsbegeerte met meer curiositeitswaarde dan praktische betekenis tot een volwassen vakgebied waarin met succes de formalisering van de wiskunde was voltoerd en waarin tevens nauwkeurig de beperkingen van zo'n formalisering konden worden aangegeven. Hier bleef het niet bij. Het begrip berekenbaar werd bestudeerd en op diverse manieren geformaliseerd door Alonzo Church, Alan Turing en anderen. Al deze formalisering, uiteenlopend in verschijningsvorm, bleken inhoudelijk gelijkwaardig: er lijkt dus sprake van de formalisering van het begrip berekenbaar, die heden ten dage nog gehanteerd wordt. Interessant is de benadering van Turing: hij maakte voor zijn definitie van berekenbaarheid gebruik van een abstracte en geïdealiseerde machine, later Turingmachine genoemd. Hiermee schiep hij tevens een wiskundig model voor wat toen nog niet bestond: de programmeerbare computer. Zo stond de logica, samen met de elektrotechniek, aan de wieg van de Informatica.

Formele talen spelen in de Informatica een belangrijke rol, hetgeen niet verrassend is als men bedenkt dat de precisie waarmee formele talen en hun afleidingssystemen gedefinieerd zijn machinale verwerking zonder menselijke tussenkomst in principe mogelijk maken. Daarvoor geschikte machines bestonden niet toen de formele talen bedacht werden, inmiddels zijn ze er in overvloed: computers. Alle programmeertalen zijn dan ook formele talen; sterker nog, alle vormen van symbolische interactie met een computer laten zich bij uitstek met behulp van formele talen beschrijven.

Dit is echter niet het enige gebruik van formele talen in de Informatica. Computers werden gaandeweg sneller en krachtiger, programma's werden steeds groter. Al snel werd de omvang van de programmatuur een probleem, in het bijzonder bij het schrijven van foutloze programma's. Nu is de computer de programmeur vaak zeer behulpzaam bij het vinden van fouten, zolang het vergrijpen tegen de grammaticale regels van de programmeertaal zijn. Hiermee worden de meeste verschrijvingen en een aantal eenvoudige denkfouten gedetecteerd: met dit laatste bedoel ik onder andere bewerkingen uitvoeren op de verkeerde categorie van objecten (in parafrase: appels bij peren optellen).

Hiermee hebben we echter lang niet alle fouten gehad, in het bijzonder niet de fouten die voortkomen uit discrepanties tussen de bedoeling van de programmeur en het geprogrammeerde. Zulke fouten kunnen aan het licht komen als de programmeur het programma laat uitvoeren door een computer en kijkt wat er gebeurt: testen dus. Dit is echter een notoir gebrekkige methode, aangezien het vaak bij lange na niet uitvoerbaar is om het programma voor alle mogelijke invoergegevens en omstandigheden te testen. Veel beter is het als de bedoeling van de programmeur ook is vastgelegd, bij voorkeur zo precies mogelijk. Hier dreigt natuurlijk een gevaar: als de beschrijving van die bedoeling even omvangrijk is als het programma zelf, dan is het probleem van het foutenvrij krijgen slechts verplaatst maar niet gereduceerd. Dit kan ondervangen worden door te streven naar beschrijvingen op een hoger niveau van abstractie. Nu zijn er vele manieren om door het weglaten van zekere aspecten abstractie te realiseren, maar een daarvan is in deze context aparte vermelding waard: weglating van alle interne details van het programma, waarna uitsluitend het gedrag van het programma overblijft. Anders gezegd: alleen het *wat* (het gedrag), niet het *hoe* (de wijze waarop dit gedrag wordt gerealiseerd). Zo'n beschrijving van het gedrag van een programma wordt een specificatie genoemd.

Even een voorbeeld: sorteren. Als u iemand vraagt een stapel adreskaartjes op naam alfabetisch te sorteren, dan hoeft u er niet bij te vertellen hoe dat moet gebeuren: u bent slechts in het resultaat geïnteresseerd en dat valt eenvoudig te beschrijven (een stapel kaartjes alfabetisch geordend op naam), eenvoudiger dan de

methode om de stapel gesorteerd te krijgen.

Binnen de Informatica houdt de Software Engineering (in het Nederlands wel aangeduid als Programmatuurkunde) zich bezig met ontwikkelingsmethoden voor programmatuur. Een programma heet correct als het aan zijn specificatie voldoet, d.w.z. als het zich gedraagt zoals in de specificatie is aangegeven. Is de specificatie formeel, dan is correctheid een relatie tussen twee formele expressies en daardoor in principe met logisch-wiskundige methoden en hulpmiddelen te bewijzen. Zo kunnen we dan bewijzen dat een programma doet wat de bedoeling was en is testen overbodig geworden. Dat is natuurlijk een interessant perspectief, gezien de zekerheid die een wiskundig bewijs biedt. Probleem bij de praktische realisatie is onder andere de schaalfactor: op kleine schaal is deze aanpak inmiddels praktisch uitvoerbaar, maar voor grote specificaties en programma's voorlopig nog lang niet.

Onderzoek richt zich op computerondersteuning bij het opstellen en verifiëren van correctheidsbewijzen en op het structureren en modulariseren (zeg maar: in samenhangende stukken knippen) van specificaties en programma's. Ander onderzoek houdt zich bezig met ontwikkelingsmethoden waarbij specificaties stapsgewijs in programma's worden getransformeerd: als we daarbij slechts gebruik maken van transformaties die de correctheid bewaren, dan krijgen we de correctheid van het eindresultaat cadeau. In alle hier genoemde gevallen is sprake van formele specificatietalen die gebruikt worden om het gedrag van programma's te specificeren. Deze formele specificatietalen vormen een geschikt punt voor het weer opnemen van de draad van mijn levensloop.

6 Levensloop (deel 2)

Na mijn promotie-onderzoek heb ik twee jaar gewerkt bij de afdeling Automatisering Informatiesystemen van Hoogovens. Heel wat anders dan de universitaire omgeving: een groot bedrijf met continuproductie waar gewoon geld verdiend moest worden. Een nuttige ervaring om daar gewerkt te hebben, maar het enige op het gebied van de wetenschap dat ik daar geleerd heb is dat wetenschappelijk onderzoek mij toch meer aanspreekt dan het schrijven van programma's. Wel heb ik in die periode meer oog gekregen voor de toepasbaarheid van wetenschappelijke onderzoeksresultaten.

Blij was ik dan ook toen ik in 1985 als universitair docent bij de Faculteit der Wijsbegeerte in Utrecht aan de slag kon, bij de sectie Toegepaste Logica in opbouw onder de inspirerende en doortastende leiding van Jan Bergstra. Die toepassing van de logica vond voornamelijk plaats in de informatica: met mijn collega Karst Koymans reisde ik wekelijks naar Philips NatLab in Eindhoven, waar wij contractonderzoek verrichten.

7 Specificatietalen

In het kader van het door de Europese Gemeenschap gefinancierde ESPRIT-project METEOR werd bij Philips gewerkt aan de ontwikkeling van een zogeheten breed-spectrum specificatietaal. Daarmee wordt aangeduid een specificatietaal waarin niet alleen specificaties kunnen worden opgesteld, maar waarin ook de stapsgewijze ontwikkeling van programma's (d.w.z. de overgang van het wat naar het hoe) kunnen worden beschreven. De motivatie voor een bedrijf als Philips om dergelijk onderzoek uit te voeren wordt duidelijk als u beseft hoeveel programmatuur er komt kijken bij de ontwikkeling van telefooncentrales, televisietoestellen en andere elektronica, en hoe ernstig de aanwezigheid van fouten in dergelijke programmatuur kan zijn.

De naam van de onderhavige taal is COLD (Common Object-oriented Language for Design), en de ideeën ervoor waren goeddeels afkomstig van Hans Jonkers. Karst

en ik waren als logici ingehuurd om, bij het samenvoegen van die ideeën tot een enkele taaldefinitie, de onderlinge samenhang te waarborgen met behulp van een in logische termen geformuleerde semantiek. Dat had meer voeten in de aarde dan we aanvankelijk dachten. Evenals in de wiskunde is onderzoek in de logica meer gericht op diepgaande analyse van in isolatie bestudeerde problemen dan op synthese van verkregen resultaten tot een groter geheel. Dat laatste werd evenwel van ons verwacht en daar zijn we ook redelijk in geslaagd. Het resultaat is een rapport van meer dan 200 bladzijden met de definitie van COLD en zijn semantiek, aangevuld met speciaal voor dat doel ontwikkelde theorie. Delen ervan hebben hun weg in de internationale wetenschappelijke literatuur gevonden en aanleiding gegeven tot nieuw onderzoek en publicaties. Inmiddels is de toepassing van COLD binnen Philips gaande: zonder problemen gaat dit niet (de afstand tussen theorie en praktijk is altijd weer groter dan je denkt) maar de resultaten zijn bemoedigend. Het is zeer wel mogelijk dat binnen afzienbare tijd consumentenelektronica op de markt verschijnt die programmatuur bevat die met behulp van COLD is ontwikkeld.

Bij het ontwikkelen van COLD lag, enigszins in tegenstelling tot hetgeen ik hiervoor betoogde, de nadruk niet op het correct bewijzen van programma's maar op het vervaardigen van specificaties, m.a.w. op het formaliseren. Ervaring met reeds langer bestaande specificatietalen zoals VDM en Z wijst uit dat alleen al het formaliseren van een probleem dat m.b.v. programma's moet worden opgelost een zeer nuttige bezigheid is: het dwingt je zeer precies over het probleem na te denken en het resultaat van al dat denkwerk helder en nauwkeurig vast te leggen, wat controle op denkfouten mogelijk maakt. Dit leidt door het toegenomen inzicht in de probleemstelling tot de eliminatie van veel programmeerfouten, ook als correctheidsbewijzen achterwege blijven.

Een deel van het positieve effect van het gebruik van formele talen ligt in de beperking die het gebruik ervan oplegt. In die zin is er verwantschap met de dichtkunst, waar eisen van rijm, metrum en alliteratie beperkingen opleggen. Zoals de beperking door vormvereisten de dichter brengt tot wendingen en vondsten die zonder die beperkingen niet bij hem waren opgekomen, zo is ook een formele taal in staat de gebruiker ervan te sturen in de richting van heldere structurering en kernachtige formulering die anders niet gevonden zou zijn.

8 Gemodulariseerd taalontwerp

Het werk aan COLD is van wezenlijke invloed geweest op mijn ideeën over de mogelijke rol van logica in de informatica en andere wetenschappen. Uitgangspunt is de constatering van een toenemende vraag naar formele talen (programmeertalen, specificatietalen, ontwerptalen, beschrijvingstalen, kennisrepresentatietalen), voortvloeiend uit de toenemende mogelijkheden van automatische verwerking van kennis en informatie. Bij gebleken behoefte worden die talen gedefinieerd, en dat gebeurt dan vaak op weinig systematische wijze: men heeft wat leuke ideeën, heeft wel eens wat gezien, neemt een bestaande taal, haalt wat weg, voegt wat toe en dat is het dan. Er is een duidelijke parallel met programmeren, dat zeker vroeger en ook nu nog wel met even weinig systeem bedreven werd. Deze toenemende vraag doet de behoefte ontstaan aan theoretische en praktische kennis over de mogelijkheden bij het definiëren van zulke talen. Voor het exploreren en verwerven van die theoretische kennis is de logica het juiste adres.

Mij staat de ontwikkeling van theorie voor ogen die de mogelijkheden van gemodulariseerd taalontwerp onderzoekt: het samenstellen van formele talen uit kleine bouwstenen, elk overeenkomend met een taalkundig idee. Zulke bouwstenen dienen niet alleen uit een stukje taaldefinitie te bestaan, maar ook de bijbehorende semantiekdefinitie en eventueel redeneer- of rekenregels te bevatten. Verder zullen er

ook verschillende bouwstenen zijn die met eenzelfde idee overeenkomen, aangezien er verschillende eisen aan notatie kunnen worden gesteld: een wiskundige heeft vaak behoefte aan compacte notatie, een beginnend gebruiker van een specificatietaal aan informatieve namen.

9 Levensloop (deel 3)

Ik keer met u in gedachten weer terug naar Utrecht. Een andere activiteit waar ik bij betrokken raakte was de nieuwe studierichting CKI (Cognitieve Kunstmatige Intelligentie), een idee van Jan Bergstra, ingegeven door de observatie dat we hier te maken hebben met een nieuw en interdisciplinair vakgebied waar in Nederland nog geen vierjarige universitaire opleiding bestond. Na een jaar van voorbereiding ging deze studie in 1988 van start met een propedeuse waarin colleges afkomstig van psychologie, taalkunde, informatica, logica en kennisleer. CKI is inmiddels zijn vijfde jaar ingegaan met elk jaar tussen de zestig en tachtig eerstejaars. Een aardig succes waaraan ik met veel plezier heb meegewerkt.

Reeds voor ik officieel in dienst trad van de Universiteit alhier raakte ik betrokken bij een soortelijk initiatief, te weten de voorbereiding van een bovenbouwstudie Technische Cognitiewetenschap, toegankelijk via een propedeuse psychologie, wijsbegeerte, natuurkunde of informatica. De verwachting is dat binnen enkele maanden het groene licht voor realisatie van de plannen kan worden gegeven, hetgeen naar verwachting zal leiden tot verbreding van het scala hier reeds bestaande technische studierichtingen.

10 Kunstmatige Intelligentie en Cognitiewetenschap

In de kunstmatige intelligentie en de cognitiewetenschap gaat het om het ontwikkelen van theorieën over menselijke intelligentie en cognitie en het toepassen van die theorieën bij het ontwikkelen van systemen die intelligent gedrag vertonen, hetgeen weer leidt tot toetsing van de toegepaste theorieën. Nu is er bij het ontwikkelen van dergelijke programmatuur nog nauwelijks sprake van het gebruik van specificaties: de extra inspanning die het vervaardigen van die specificaties vereist lijkt ook niet zo zinvol aangezien de correctheid en betrouwbaarheid niet primair staat: het gaat dikwijls om experimentele programma's, als er wat mis gaat is dat geen ramp en wordt het programma even aangepast. Het tij is overigens aan het keren: dit is te zien op de plaatsen waar kennis en methoden van de kunstmatige intelligentie hun weg vinden naar bedrijfsgerichte toepassingen. Zo is aan de Vrije Universiteit te Amsterdam de specificatietaal DESIRE ontwikkeld voor het specificeren van kennisystemen en wordt aan de Universiteit van Amsterdam gewerkt aan de formele specificatie van het expertisemodel dat aan de kenniselicitiemethode KADS ten grondslag ligt. Momenteel bestudeert Rix Groenboom, als AIO onder mijn leiding werkzaam bij de vakgroep Informatica, formele specificatie van SOAR, een succesvol en veel gebruikt programma dat gebaseerd is op een model van menselijke cognitie.

Het belang van deze ontwikkeling ligt in het feit dat daardoor de mogelijkheid zal worden geboden beschrijvingen van intelligentie en cognitie te vervaardigen die de precisie van het formele combineren met het abstractieniveau van wetenschappelijke theorieën. Zulke beschrijvingen zullen de huidige kloof vullen tussen enerzijds de informele theorie en anderzijds het formele maar door talloze implementatiedetails onleesbare programma. Vanuit die positie zullen die beschrijvingen door hun explicietheid en abstractie bijdragen aan zowel de theorie als aan de kwaliteit van de programmatuur.

We zien dat aldus de computer als breekijzer fungeert bij het openleggen van de cognitieve psychologie en de cognitiewetenschap voor de introductie van nieuwe manieren om de theorieën te beschrijven. Bij de taalwetenschap is deze ontwikkeling al enige tientallen jaren aan de gang en is er inmiddels sprake van een der rijkste toepassingsgebieden van de logica. Bij deze vakgebieden zal het overigens niet blijven. De informatietechnologie vindt niet-triviale kennisintensieve toepassing in de exacte wetenschappen, econometrie, rechten (juridische adviesystemen) en medicijnen (medische expertsystemen); waar het niet slechts kwantitatieve modellen en berekeningen betreft, daar zullen formele talen, in het kielzog van de informatietechnologie, op termijn gevolgen hebben voor de begripsvorming en de theorievorming binnen de desbetreffende disciplines.

Ook hier zal een beroep gedaan worden op de flexibiliteit die bij het definiëren van formele talen in principe mogelijk is. Er zal behoefte ontstaan aan het ontwikkelen van formele talen die zijn toegesneden op zeer specifiek gebruik, bv. voor het beschrijven van verkeersregels, genen in het menselijk DNA of heuristische bij het reconstrueren van chemische structuurformules uit spectroscopische gegevens.

11 Conclusie

Concluderend kan gesteld worden dat de opmars der formele talen, begonnen in de wiskunde en voortgezet in de informatica, zich zal voortzetten in vele andere gebieden van wetenschap en samenleving, in het kielzog van de opmars van de computer. Zij zullen worden toegepast voor het opstellen van precieze, heldere en ondubbelzinnige beschrijvingen van wetenschappelijke en andere kennis. Die beschrijvingen zullen verdere analyse, toepassing en theorievorming mogelijk maken. Bij de verwezenlijking van een en ander zal behoefte ontstaan aan op maat gesneden formele talen. Ontwikkeling van de hiervoor benodigde theorie is gewenst maar nog amper aangevat.

De logica speelt hierbij een belangrijke rol, overigens lang niet alleen op de hier door mij beschreven wijze. Het vak zit onmiskenbaar in de lift, en het is dan ook een goede zaak dat dit weerspiegeld wordt in de organisatie van dit vakgebied in een land met een respectabele logische traditie. Niet alleen is er al bijna een halve eeuw een bloeiende Vereniging voor Logica en Wijsbegeerte der Exacte Wetenschappen, er is nu ook een Onderzoekschool Logica die enige weken geleden de begeerde erkenning van de Koninklijke Nederlandse Academie van Wetenschappen heeft ontvangen. Laat het deze Universiteit tot eer strekken dat drie onderzoeksgroepen via het lokale BCN in deze Onderzoekschool participeren, en dat de drijvende kracht achter deze Onderzoekschool gevormd wordt door Johan van Benthem, momenteel hoogleraar aan de Universiteit van Amsterdam maar daarvoor hoogleraar aan deze Universiteit.

12 Woorden van dank en goede wensen

Tot zover het inhoudelijk gedeelte van mijn oratie. Mij rest gelukkig nog enige tijd voor woorden van dank en goede wensen.

Het College van Bestuur van deze Universiteit dank ik voor mijn benoeming, evenals alle commissies, besturen, raden en colleges die het voorstel daartoe opgesteld dan wel gesteund hebben. Ik doe mijn best het gestelde vertrouwen waardig te zijn.

Mijn ouders dank ik voor mijn opvoeding, in het bijzonder voor de wijze waarop zij mij behoed hebben voor de eenzijdige ontwikkeling die bij wiskundige aanleg kan optreden.

Collega Anne Troelstra dank ik voor alles wat ik van hem geleerd heb dan wel getracht heb te leren: intuïtionisme, het verrichten van onderzoek, maar vooral het stellen van hoge eisen aan jezelf en je werk.

Collega Jan Bergstra dank ik voor zes jaren begeleiding op afstand en het voorbeeld dat hij is geweest in het ontwikkelen van eigen ideeën en het met gerechvaardigd zelfvertrouwen realiseren daarvan.

De leden van de vakgroep Informatica wens ik een voortzetting van de voorspoedige samenwerking, in het bijzonder ten aanzien van de plannen voor een afstudeerrichting Technische Informatica die wij samen met de op korte termijn te benoemen hoogleraar Technische Informatica voornemens zijn te realiseren.

Mag ik u tenslotte verzoeken, in navolging van mijn collega Van Baak, hoogleraar Slavische taal- en letterkunde die hier vorig jaar zijn oratie uitsprak, om bij het verlaten van de Aula de boven de uitgang aangebrachte inscriptie 'Favete linguis' (koestert de talen) in acht te nemen, onder toevoeging — in gedachten — van het adjectief 'formalibus'.

Ik heb gezegd.