Diës 1991
Universiteit Utrecht

Jan Bergstra

Filosofische aspecten van digitale systemen

Eredoctoraten:
Dr. Christiane Nüsslein-Volhard
Dr. Vladimir Igorewitsj Arnol'd

Jan Bergstra

Filosofische aspecten van digitale systemen

Filosofische aspecten van digitale systemen

Rede bij het 355-jarig bestaan van de Universiteit Utrecht. Prof. dr. J.A. Bergstra (39) is hoogleraar toegepaste logica bij de Faculteit Wijsbegeerte in Utrecht. Tevens is hij hoogleraar Programmatuur aan de Universiteit van Amsterdam, faculteit Wiskunde en Informatica.

Bergstra is de initiatiefnemer van de Utrechtse studierichting Cognitieve Kunstmatige Intelligentie, die in 1988 van start is gegaan en zich inmiddels in een grote belangstelling onder studenten mag verheugen.

Twee opvattingen betreffende digitale systemen

OMTRENT digitale systemen, een woord dat ik als synoniem zal gebruiken met hedendaagse rekenautomaten en netwerken van rekenautomaten, wil ik twee opvattingen onderscheiden. In de eerste opvatting zijn deze systemen zo complex dat zij als metafoor kunnen dienen voor de hersenen van een mens. Een gecombineerd onderzoek van de cognitieve processen in biologische systemen en van de, als cognitief te definiëren, processen binnen machines is in die opvatting zinvol en productief. Deze opvatting ligt ten grondslag aan de studierichting cognitieve kunstmatige intelligentie welke door de faculteit der wijsbegeerte in 1988 werd ingesteld. Uit het feit dat per september aanstaande niet minder dan drie concurrerende universiteiten de instelling van een soortgelijke studierichting beogen moge blijken dat de studenten door deze eerste opvatting worden aangesproken. De kracht van het concept van de kunstmatige intelligentie ligt in de fundamentele inspiratiebron die deze voor de bouwers en ontwerpers van digitale systemen vormt. Als zwakte van het concept zie ik enerzijds een overschatting van de bruikbaarheid van de vergelijking tussen menselijke hersenen en machine en anderzijds de onderschatting van de geheel eigen problematiek die de digitale systemen met zich brengen.

De tweede opvatting terzake digitale systemen die ik wil noemen benut het feit dat de wereld der digitale systemen in zeer hoge mate

door mensen is gemaakt op basis van langdurige herhaling van een beperkt repertoire van bekende bouwprincipes. Hieruit wordt vervolgens geconcludeerd dat, althans in filosofische zin, een digitaal systeem volledig begrepen moet kunnen worden. Het moet mogelijk zijn een systeem op te splitsen in atomaire delen (waarbij 'atomair' overdrachtelijk wordt gebruikt) en alle samenhangen te inventariseren. Vervolgens moeten mogelijke begintoestanden van een systeem kunnen worden vastgelegd en kan er een overzicht onstaan van de mogelijke levenslopen van het systeem. Het aantrekkelijke van de opvatting zit in de optie op toepassing van het mede door Carnap [C 28] geïntroduceerde logisch atomisme in een kader dat daarvoor geschikt lijkt, namelijk de studie van digitale systemen. In plaats van logisch atomisme zal ik hier liever spreken van logisch reductionisme, omdat niet iedere bouwsteen van een logisch systeem atomair karakter heeft.

Om het onderwerp geheel geschikt te maken voor een logisch reductionistische aanpak zijn twee vereenvoudigingen noodzakelijk. De fysica wordt buitengesloten door systemen te bekijken die bestaan uit bouwstenen waarvan het gedrag door eenvoudige wiskundige modellen wordt beschreven. De idealisering en tevens vereenvoudigende aanname schuilt erin dat deze modellen geheel correct geacht worden te zijn en dat zij niet slechts benaderingen van de werkelijke componenten weerspiegelen, zulks overigens geheel los van de vraag of deze ideale componenten in werkelijkheid kunnen bestaan. Ten tweede wordt de mens als systeemgebruiker buiten beschouwing gelaten. De specificatie, bijvoorbeeld, dat een systeem voor een menselijke gebruiker hanteerbaar moet zijn wordt als zijnde niet verifieerbaar van de hand gewezen. Alleen volstrekt objectiveerbare systeemspecificaties komen in aanmerking.

Mij spreekt de tweede opvatting meer aan, en de te behandelen filosofische aspecten zullen alle betrekking hebben op een logisch reductionistische beschouwing van de levenloze digitale machinerie.

De Turingmachine

In 1936 heeft de Engelsman Alan Turing [T 36] een fundamenteel model voorgesteld van een rekenautomaat. Dit model stamt van voor de tijd van de werkstations en de computernetwerken.

In de Turingmachine is sprake van een programma dat uit niets anders bestaat dan een eindige reeks van toestanden met bij iedere toestand een voorschrift dat aangeeft of het programma in de betreffende toestand is afgelopen (d.w.z. of de toestand een eindtoestand is) en indien niet hoe de volgende toestand bereikt moet worden. Voorts is er een unieke begintoestand aangegeven.

Het programma mag een zeer eenvoudig geheugen gebruiken: een van links naar rechts oneindig doorlopende reeks van vakjes kan gevuld worden met nullen of enen. Dit is de fameuze Turing-tape. Initieel staat in elk vakje een

ongedefinieerde waarde ("blank"). Teneinde toegang te hebben tot dit geheugen heeft de machine een zogenaamde (lees- en schrijf-)kop, deze is steeds bij een van de vakjes gepositioneerd. Bij elke toestandsovergang kan de machine kiezen tussen het verplaatsen van de kop (en wel één positie naar links of naar rechts), het schrijven van een 0 of 1 en het lezen van de waarde in het vak waarbij de kop gepositioneerd is. De te bereiken toestand na toestandsovergang kan afhangen van de gelezen waarde (er zijn drie mogelijkheden: 0, 1 en "blank"). Vanuit een gegeven beginsituatie doet de machine stap na stap totdat een eindsituatie is bereikt. Gebeurt zulks niet dan eindigt het programma niet. Men codeert de invoer voor de machine als een rij van nullen en enen op de tape en plaatst de kop links van de eerste 0 of 1 van deze rij, tevens laat men de machine in z'n begintoestand starten. Vervolgens wacht men tot een eindtoestand is bereikt. De reeks van nullen en enen rechts van de kop bepaalt dan de uitvoer van de machine. Op deze wijze bepaalt de machine een transformatie van reeksen van nullen en enen naar reeksen van nullen en enen. Zo'n transformatie noemen we een berekenbare functie.

Het frappante van dit model is dat niemand er ooit in is geslaagd een model te vinden waarmee meer berekenbare functies kunnen worden geprogrammeerd. In de jaren dertig bleek dat een serie van onafhankelijk ontdekte modellen steeds weer tot dezelfde klasse van berekenbare functies aanleiding gaf. Van deze modellen is de Turingmachine de begrijpelijkste en in filosofische zin de meest overtuigende. Ik zal hier niet ingaan op de op zich niet oninteressante filosofische motivering voor de Turingmachine. Wellicht het noemen waard is het feit dat Turings naam inmiddels is verbonden aan de jaarlijks uitgedeelde Turing Award. Deze Amerikaanse prijs heeft in de informatica inmiddels wereldwijd een onaantastbare reputatie als meest belangrijke bewijs van wetenschappelijke erkening opgebouwd.

Over de Turingmachine bestaat een fundamenteel resultaat [T 36] dat hier genoemd moet worden, omdat het in vele varianten van het model steeds weer terugkeert. Allereerst kunnen we vaststellen dat het mogelijk is de begintoestand van een Turingmachine zelf te coderen in een eindige reeks van nullen en enen. Dit vergt enig werk maar blijkt zeer eenvoudig te zijn. De lengte van deze reeks wordt vanzelfsprekend bepaald door de omvang van het besturingsdeel van de machine en de omvang van de op de tape geplaatste input. Nu zou men deze codering kunnen gebruiken als invoer voor een nog te ontwerpen Turingmachine die tot taak krijgt de gegeven (gecodeerde) Turingmachines te analyseren. Een voor de hand liggend voorbeeld is de vraag te beantwoorden of de gecodeerde machine ooit een eindtoestand zal bereiken (het is immers ook mogelijk dat de machine na opgestart te zijn onbeperkt lang door blijft rekenen). Het fundamentele resultaat is nu dat deze analyse door geen enkele Turingmachine kan worden uitgevoerd. Met andere woorden: te bepalen of een Turingmachine zal termineren is niet op algorithmische wijze mogelijk. Dit resultaat is sterk verwant aan de onvolledigheidsstellingen

van Gödel [G 31] doch veel eenvoudiger te bewijzen. Men noemt dit resultaat doorgaans de onbeslisbaarheid van het 'halting'-probleem. Het geeft een absolute bovengrens aan de mate waarin computers over zichzelf kunnen reflecteren.

De rekenautomatentheorie is uiteindelijk een variant van de theorie van Turing; met deze varianten is het echter vreemd gesteld - er zijn namelijk ook overtuigende varianten waarvoor toepassingen slechts in de logica bekend zijn en niet in de theoretische informatica (zie bv. [N 80]).

Niet-sequentiële machinemodellen

Inmiddels is gebleken dat de Turingmachine zijn absolute status heeft te danken aan een reeks van vereenvoudigende aannamen. Zo laat hij willekeurig omvangrijk geheugengebruik toe en willekeurig grote aantallen rekenstappen tijdens de berekening van een enkele functiewaarde. Daarnaast is het model volstrekt deterministisch en sequentieel. Met deterministisch bedoelt men dat op elk moment slechts (maximaal) één volgende stap mogelijk is. Sequentieel betekent dat er slechts één actief object is. Dit laatste heeft tot consequentie dat in het model van Turing de tijd benodigd voor een stap niet relevant is, omdat er immers geen afstemmingsproblemen tussen verschillende componenten zijn. Tenslotte is de Turingmachine niet interactief, hij beperkt zich in automatiseringsjargon tot 'batch-verwerking'.

Korte inspectie van de in elke faculteit inmiddels geïnstalleerde netwerken van automaten leert dat op elk van de genoemde punten de Turingmachine de wereld te simpel voorstelt. En daarmee ligt de vraag naar het vinden van geschikte modellen voor digitale systemen weer volstrekt open. Voor een logisch reductionistische aanpak van het thema is het vinden van nieuwe modellen dus vereist maar de overgang is bijzonder groot en er is vooralsnog geen sprake van dat men een model heeft kunnen vinden dat binnen de nieuwe randvoorwaarden een met de Turingmachine vergelijkbare claim op algemeenheid heeft.

De eerste stap die men terzake nieuwe modellen heeft gezet is een simpele aanpassing van de Turingmachine in die zin dat men de tape eenvoudig achterwege laat. Zo ontstaat een eindige automaat [RS 59]; bij elke toestandsovergang treedt bovendien een extern zichtbaar effect op. Zo'n effect heet een actie. De automaat heeft evenals het besturingsdeel van de Turingmachine een begintoestand en verschillende eindtoestanden. Vele onderzoekers hebben modellen van dit slag voorgesteld doch het is niet ongebruikelijk er de naam van Moore mee te verbinden. De Mooremachine verlaat op twee punten de idealisering van de Turingmachine. Allereerst is er geen oneindig groot geheugen meer. (Daaruit volgt direct dat we ons over willekeurig lang oplopende rekentijden ook minder zorgen behoeven te maken.) Ten tweede is de machine interactief. Men kan zich voorstellen dat de acties uiteenvallen in twee klassen, die welke door een externe omgeving worden geïnitieerd en die welke door de machine zelf worden ondernomen en op

hun beurt door de omgeving worden opgemerkt.

Een direct volgende stap is die naar netwerken van communicerende Mooremachines. Men spreekt een koppeling af tussen de acties van naburige automaten en direct ontstaat er een model waarin gelijktijdige operatie van vrijwel onafhankelijke componenten optreedt. Dit verschijnsel heet parallellisme. Aan zo'n model kan men dan component voor component weer additioneel geheugen toevoegen in de vorm van een Turing-tape of een beperkte variant daarvan. Aldus ontstaat een enorm scala van modellen. Al deze modellen hebben gemeen dat ze gebaseerd zijn op een of andere vorm van een netwerk van samenwerkende componenten en een of andere vorm van parallellisme omvatten. Ik wil voor zulke machinemodellen de verzamelnaam niet-sequentiële modellen gebruiken. Zoals gezegd is het tot dusverre niet gelukt om binnen de zoekruimte van de niet-sequentiële modellen een machinemodel met universele pretenties te isoleren en de wetenschappelijke aandacht heeft zich dan ook volledig verlegd naar andere zaken:

(1) het vinden van machinemodellen die terzake de klasse van de in het model passende automaten (machinebeschrijvingen) een belangwekkende structuurtheorie opleveren, (2) vragen betreffende de waarneembaarheid van gedrag bij gegeven machinemodellen, (3) vragen betreffende causaliteit en onderlinge (on)afhankelijkheid van acties van verschillende componenten in een niet-sequentiële machine en (4) het bepalen van de steeds complexer wordende betekenis (semantiek) van machinebeschrijvingen in deze nieuwe modellen. Dit leidt tot metamodellen die verschillende equivalentiebegrippen voor de ontwikkelde machinemodellen beschrijven. Bij de Turingmachine is immers duidelijk dat men twee machines bij voorkeur equivalent zal noemen wanneer ze dezelfde transformatie op de invoer realiseren. Hier zijn echter alternatieven mogelijk, omdat het zinvol kan zijn tevens rekening te houden met rekentijd en geheugengebruik. Bij niet-sequentiële machines is tenminste duidelijk dat ze niet equivalent zijn wanneer ze verschillend waarneembaar gedrag hebben. Dit leidt terug tot punt (2) hierboven.

Datastructuur en processtructuur

Een kort intermezzo terzake doel en aard van deze beschouwingen is hier zinvol. Het betreft filosofische bespiegelingen, omdat nog steeds met een groot arsenaal aan idealiseringen wordt getracht de complexiteit van de echte wereld buiten de deur te houden, terwijl tevens concepten worden ontwikkeld die van belang zijn voor de dagelijkse gedachtenwisseling over digitale systemen en hun ontwerp, bouw en feitelijk functioneren.

Het pragmatische doel op de achtergrond is echter wel degelijk een hanteerbaar kader te bereiken dat ook voor de praktijk van belang is en dat van dienst is bij het ontwikkelen van betrouwbare systemen.

Een consequentie van deze overweging is dat het zinvol is om sommige begripsonderscheidingen in de theorie op te nemen die in elke

pragmatische benadering van digitale systemen blijken op te treden niettegenstaande het feit dat deze begripsonderscheidingen zelf slechts moeilijk filosofisch hard te maken zijn. Als meest belangrijke voorbeeld wil ik hier noemen het onderscheid tussen proces en data. Een proces vertoont gedrag in de tijd, data zijn tijdloze vormen met wiskundige eigenschappen zoals symmetrieën, invarianten en representatiestellingen. In alle modellen wordt steeds een onderscheid gemaakt tussen proces en data. Bij de Turingmachine is de tape een datastructuur en genereert de besturingsmachine een proces. Het onderscheid tussen proces en data is doorgaans volledig arbitrair. Grote verwarring ontstaat steeds weer doordat de beschrijving van een proces natuurlijk een datastructuur is terwijl het gebruik van data doorgaans een proces vergt. Men bedenke hierbij dat het in dit kader werkelijk problematisch is om een proces niet met z'n beschrijving te identificeren maar met z'n feitelijke uitvoering door een fysieke machine en evenzo problematisch om een datatype los te zien van het in de tijd uitgestrekte gebruik ervan.

Bij elke opzet onvermijdelijk blijkt echter dat men een (eventueel tijdelijk) verschil tussen proces (gedrag) en data (vorm) hanteert. Ik zal dus niet aarzelen om een dichotomie tussen data en proces eenvoudigweg te postuleren en voor beide een structuurtheorie te kiezen, wetende dat een filosofische onderbouwing van de gekozen dichotomie in feite kansloos is.

Opmerkingen over μCRL, een niet-sequentieel machinemodel

1 Datatypen voor μCRL

Een machinemodel legt een klasse van machines vast die alle volgens een overeenkomstige globale architectuur zijn opgezet. Het begrip architectuur is hierbij echter veel ruimer dan in de bouwkunde. Om de vergelijking door te zetten: zo'n globale architectuur zal vermelden dat een gebouw verschillende delen met elk een welbepaald aantal verdiepingen heeft maar het globale model zegt niets over het aantal delen en het aantal verdiepingen per deel. Ik wil nu een machinemodel aangeven waarvoor een zinvolle structuurtheorie bestaat. De naam van het model is (effectief) μCRL (zie [GP 91], het acronym stamt uit het Europese RACE-project SPECS; het aantal modellen van dit slag en het aantal van de bijbehorende acronymen loopt inmiddels in de honderden).

Ik begin met de keuze van een techniek voor het beschrijven van datastructuren in de wetenschap dat dit nodig zal blijken maar dat de keuze arbitrair is. De keuze valt op meersoortige algebraïsche specificaties voor zogenaamde abstracte datatypen en op (half)complete termherschrijfsystemen voor concrete datatypen. (Het onderscheid tussen abstracte en concrete datatypen is allerminst onomstreden maar vele auteurs maken het, dikwijls verpakt in de terminologie van specificatie versus implementatie.) De betekenis van zo'n beschrijving is een algebra, dat

is een collectie van verzamelingen (ook wel soorten of typen genoemd) van objecten met daarop een reeks wiskundige bewerkingen. Men denke aan een verzameling van geordende lijsten met sorteren en het zoeken van minimum en maximum als bewerkingen. De theorie van deze databeschrijvings-techniek vindt men in [MG 85] fraai samengevat. Het gekozen formaat is voldoende sterk om alle datatypen die door Turingmachines gesimuleerd kunnen worden te beschrijven.

Bij de soorten van een datatype kan een onderscheid worden gemaakt tussen 'klein', als de soort eindig veel elementen heeft en 'groot' wanneer er oneindig veel data-objecten van die soort zijn. Het verschil is dat data uit kleine typen gebruikt kunnen worden zonder enige notie van proces te vooronderstellen, terwijl grote datatypen bij operationeel gebruik in een machine een proces vooronderstellen dat ze toegankelijk maakt. Grote datatypen spelen echter een fundamentele rol bij de beschrijving van processen en, opmerkelijk genoeg, bij de beschrijving van kleine datatypen. In beide gevallen kunnen objecten uit grote datatypen als parameters optreden.

De structuurtheorie van het machinemodel betreffende de zojuist geschetste datatypen-beschrijvingen, is bijvoorbeeld te vinden in [W 90], daarnaast is van toepassing de theorie van de termherschrijfsystemen (zie [D 90] voor een overzicht).

2 Processen in μCRL

Op basis van eerder vastgelegde datatypen wordt in het μCRL model in enkele stappen een wereld van processen beschreven. (De terminologie is ontleend aan de procesalgebra, zie b.v. [BW 90] en [B 91].) Allereerst introduceert een instantie van het model een eindige verzameling van namen van zogenaamde atomaire acties. Een atomaire actie is een gebeurtenis die zich op een ondeelbaar moment in de tijd afspeelt. Het spreekt vanzelf dat het bestaan van atomaire acties een vereenvoudigende aannname is die nauwelijks door de 'sense data' wordt bevestigd. De structuurtheorie van het model wordt in hoge mate geprofileerd door juist deze aanname en het is niet eenvoudig om de voorspellingen die de theorie over een μCRL-machine doet te relateren aan het gedrag van een fysieke implementatie.

De acties mogen worden geparametriseerd door kleine data. Bovendien wordt van elk tweetal acties aangegeven wat er gebeurt wanneer de twee acties op precies hetzelfde moment optreden. Dat levert weer een actie op, te bepalen door de zogenaamde communicatiefunctie.

De kern van μCRL is de beschrijving van een aantal processen die elk langs de volgende lijnen geconstrueerd mogen worden.

(0) Elke atomaire actie is een proces en er is een proces geheten deadlock. Deadlock is een stagnerend proces dat geen enkel gedrag vertoont.

Op drie wijzen kan men uit twee processen een derde maken:

(1) bij p en q is p + q een proces dat zich zowel als p als conform q kan gedragen. De keuze tussen beide gedragingswijzen voor p + q kan gemaakt worden door het proces zelf, door de omgeving van het proces of middels subtiele interactie tussen proces en omgeving. Vast staat dat nadat p + q een eerste atomaire actie heeft volbracht de keuze is gemaakt tussen p en q. Deze operator + is ontleend aan het baanbrekende werk van Milner (zie [Mi 89] voor een recente beschrijving van Milners theorie). De '+' mag ook worden gebruikt als een geparametriseerde som over alle data in een klein datatype.

(2) bij p en q is p . q een proces dat zich eerst conform p en na (eventuele) afloop van p conform q gedraagt. Dit is de sequentiële compositie-operator die in bijna alle programmeertalen voorkomt.

(3) bij p en q is p || q de merge van p en q; deze merge beschrijft de parallelle compositie van p en q. Het is een proces dat afwisselend stappen uit p en q kan doen maar soms ook stappen van p en q op precies het zelfde moment uitvoert; in dat geval communiceren de processen met elkaar volgens de eerder genoemde communicatiefunctie.

Er zijn twee methoden om uit één proces een nieuw te maken:

(4) bij een proces p en een verzameling H van atomaire acties is encaps(H, p) de encapsulatie van H in p. Dit laatste proces is als p met dien verstande dat elke actie uit H wordt verboden. Een proces van de vorm encaps(H, p || q) kan de processen p en q tot communicatie dwingen.

(5) bij een proces p en een verzameling I van atomaire acties is hide(I, p) een proces dat zich gedraagt als p met dien verstande dat de acties uit I voor de buitenwereld van p onzichtbaar worden gemaakt. (Dit mechanisme is de kern van het werk van Milner, het is ongeveer 15 jaar oud).

Tenslotte mag men processen impliciet definiëren als oplossingen van stelsels algebraïsche vergelijkingen van een geschikte vorm:

(6) recursie.

3 Theorie over μCRL

De theorie terzake μCRL wordt met name ontleend aan de procesalgebra zoals beschreven in [BW 90] en [B 90]. Ik wil één aspect bespreken dat mij zeer intrigeert en dat beslist controversieel te noemen valt. Uitgaande van een μCRL-beschrijving is de betekenis een zogenaamd berekenbaar eindigsplitsend toestandsovergangssysteem. Wat dat ook moge betekenen, het kan voorkomen dat een proces p in een toestand verkeert waarin een oneindige reeks van interne (voor de buitenwereld verborgen) stappen mogelijk is. De vergelijkingen van de procesalgebra impliceren nu dat p de neiging heeft om naar vermogen aan deze interne activiteit te ontkomen en een voor de buitenwereld zichtbare stap te ondernemen. Op deze wijze ontstaan belangwekkende voorspellingen over

het systeemgedrag. Het controversiële van deze theorie moge blijken uit een vergelijking met de theorie van Hoare [Ho 85] die aangeeft dat wanneer een proces vanuit enige toestand een onbegrensd aantal inwendige stappen kan ondernemen het systeemgedrag vanaf dat punt juist onvoorspelbaar wordt.

De ontwikkeling van programmeertalen

EEN programmeertaal heeft men nodig zodra een scheiding tussen een fysieke machine en een daarvan deels onafhankelijke besturing is aangebracht. De programmeertaal geeft een syntax voor de besturingsinstructies. Sinds het verschijnen van programmeertalen in de jaren veertig zien we een ontwikkeling naar een steeds hoger nivo. Een hoog nivo taal stelt ons in staat met een beknopt programma veel machinale activiteit teweeg te brengen. Het is inherent aan de notie van een programmeertaal dat er een machine bestaat die alle programma's uit de taal kan uitvoeren. Dit laatste begint nu toenemend een beperking te worden. Er ontstaat een behoefte aan niet-uitvoerbare constructies die kennis over een gewenst gedrag uitdrukken. Zulke programmadelen noemt men (formele) specificaties. Gebruikelijk werden zulke zaken genoemd in zogenaamd commentaar. Het commentaar is echter bedoeld voor de menselijke gebruiker terwijl de formele specificaties bovendien kennis overdragen op de machine.

Het belang van niet-uitvoerbare instructies voor het systeemontwerp is vooralsnog een onderwerp van onderzoek dat zoals alle methodologische kwesties in de programmatuur het gevaar loopt een scheidslijn te worden in een scholenstrijd. Vanuit deze positie is het dan vrijwel ondoenlijk genoemd belang op een zakelijke wijze te onderzoeken.

Onlosmakelijk verbonden met het programmeren is de programmaverificatie. Deze in de jaren '60 en '70 ontwikkelde techniek (zie [Co 90]) zal niet kunnen ontkomen aan enige modernisering. De toen ontwikkelde technieken zijn zeer statisch in die zin dat bewijzen qua vorm volledig overeenkomen met het te verifiëren programma. Mijn conclusie is dat wie over programma's wil redeneren ermee moet kunnen rekenen. Dat wil zeggen dat de gelijkheid tussen programma's een basisbegrip wordt en dat gelijkheidsbehoudende transformaties een hoofdrol gaan spelen bij het leveren van bewijzen over programma's. De uit Polen afkomstige algorithmische logica (door Harel [Ha 84] echter onder de naam dynamische logica wereldwijd gepropageerd) levert zulke mogelijkheden. Langs een geheel andere lijn levert de procesalgebra uit [BW 90] ook een voor verificatiedoelen hanteerbaar programmatransformatiesysteem.

In de filosofie van het systeemontwerp tekent zich inmiddels een tegenstelling af tussen twee invalshoeken. In de ene is een specificatie slechts een technisch hulpmiddel bij verificatie. Men programmeert en geeft van het programma een specificatie die vervolgens via verificatie wordt

gevalideerd. In de tweede opvatting is de specificatie methodologisch van groter belang dan het uitvoerbare programma, omdat het voor ontwerper en gebruiker een begrijpelijkheid heeft die het uitvoerbare programma doorgaans volstrekt mist. Als verklaard logisch reductionist mag ik natuurlijk slechts het verificatiescenario accepteren. Waarneming van de industriële praktijk leert echter dat juist het specificatiescenario op dit moment een grote pragmatische waarde heeft. Deze stand van zaken wordt door veel logisch georiënteerde onderzoekers terzake de methodiek van het programmeren als bijzonder paradoxaal ervaren. De oplossing van de paradox zie ik in de vergelijking van het specificatiescenario met het informeel gebruik van toegepaste wiskunde. Dat kan zeer effectief zijn zonder gebaseerd te zijn op een overtuigende vorm van logisch reductionisme.

Structuur en modulariteit

HET zou een ernstige omissie zijn wanneer ik zou nalaten hier te vermelden dat het een ogenschijnlijk door alle systeemtheoretici gedeelde opvatting is dat systeembeschrijvingen structuur moeten hebben en dat deze structuur zich voornamelijk manifesteert in een inzichtelijke opsplitsing van het systeem in componenten, in de programmatuur doorgaans modulen genoemd. Jammer genoeg is het buitengewoon moeilijk om een modularisering (opsplitsing in modulen) tot het beschreven systeem zelf te herleiden. Modularisering lijkt essentieel niet waarneembaar te zijn. Zo bezien is modularisering een dominant aanwezig principe dat zich onttrekt aan een logisch reductionistische ontologie. Op deze wijze wordt modularisering een moeilijk grijpbare theoretische term die essentieel samenhangt met het feit dat digitale systemen door mensen worden ontworpen en primair aan ergonomische criteria moet worden onderworpen. Terzake modularisering liggen verschillende conceptuele problemen nog volstrekt open. De inmiddels ontwikkelde module algebra (zie [vGV 89] of [J 89] voor toepassingen) lost slechts een fractie van deze problemen op.

Open problemen van conceptuele natuur

(1) EEN machine die informatie bewerkt kan deze informatie vrijelijk hanteren. Bijvoorbeeld het kopiëren van informatie is geen enkel probleem. Een machine die materiële producten bewerkt heeft echter aanzienlijk scherpere randvoorwaarden welke voortkomen uit natuurkundige behoudswetten. De logische status van deze additionele randvoorwaarden is nog allerminst opgehelderd. Zulk inzicht is bijvoorbeeld nuttig voor het ontwerpen van conceptueel adequate modellen voor productieautomatisering.

(2) Men kan de procesalgebra realistischer maken door elke actie te voorzien van coördinaten in de vierdimensionale tijdruimte. Het is dan duidelijk dat de correctheid van een systeem onafhankelijk van de snelheid van een waarnemer zou moeten zijn. Dat vergt bijvoorbeeld een

Lorentz-invariante formulering van de procesalgebra en systeemverificaties daarin. Dit is in beginsel mogelijk gebleken maar toepassing op systemen van enige complexiteit is ogenschijnlijk prohibitief.

(3) Radarwaarneming en radiocommunicatie zijn voorbeelden van mechanismen die men als computationeel zou kunnen opvatten maar waarvoor een modellering in een logisch-formeel kader nog volstrekt ontbreekt. Dit maakt het bijvoorbeeld onmogelijk om de correctheid van satellietcommunicatieprotocollen binnen een homogeen conceptueel kader te formuleren en te bewijzen.

(4) Het is inmiddels duidelijk hoe we een koffieautomaat in de plaatselijke kantine formeel kunnen modelleren (zie b.v. [Mu 90]). Problematisch wordt het wanneer men de steeds wisselende rij van klanten in het model wil betrekken. Een overtuigend model voor de tijdelijke binding tussen automaat en gebruiker bestaat op dit moment niet.

(5) Tenslotte de status van het non-determinisme. Volstrekt onopgehelderd is de vraag in welke mate het voor systeembeschrijvingstalen noodzakelijk is om niet-deterministische mechanismen toe te laten. Als we ons op de filosofisch denkbare positie stellen dat de werkelijkheid deterministisch is dan kan de noodzaak tot het toelaten van niet-deterministische systemen niet worden gemotiveerd met een beroep op de werkelijkheid. Maar dat betekent dat een eventuele noodzaak tot het gebruik van niet-determinisme ligt in de cognitieve eigenschappen van de mens. Dan komt non-determinisme in het zelfde vaarwater als de eerder besproken modularisering. Volstrekt onopgehelderde kwesties ontstaan voorts wanneer men modularisering en niet-determinisme gaat combineren.

Over de kansen van het logisch reductionisme

CONCREET terzake software en systeemontwerp wil ik de these formuleren dat logisch reductionisme haalbaar is mits men dit interpreteert als het consequente streven om de betekenis van systeembeschrijvingen te bepalen middels reductie naar van te voren vastgelegde logisch-wiskundige primitieven. Dit kan leiden tot (falsificeerbare) voorspellingen omtrent het systeemgedrag met bijzonder hoge betrouwbaarheid.

De gebruikte logisch-wiskundige basis is een door mensen gemaakte formele wereld en steeds zal men bij de modellering moeten kiezen tussen verschillende opties. Er zijn vele mogelijkheden en er bestaat geen logisch kookboek. Toch leidt elke keuze tot verheldering (al was het maar het inzicht dat een andere keuze te prefereren valt). Ook in de op de fysica georiënteerde wiskunde is sprake van zulke keuzen, zo kan men desgewenst constructieve wiskunde beoefenen of niet-Archimedische analyse. Het blijkt echter dat wiskundigen in de praktijk weinig verliezen door vrijwel altijd dezelfde keuzen te maken (namelijk voor klassieke logica en Archimedische analyse). Op termijn zou een dergelijke situatie in

de theorie van de digitale systemen ook kunnen ontstaan.

Een volgende opvatting die ik naar voren wil schuiven is dat de inzichten die door een logisch reductionistische benadering van digitale systemen kunnen worden verkregen door de ontwerpers van zulke systemen ook zouden moeten worden nagestreefd. Na vele contacten met programmeurs in de Europese industrie moet ik wel vaststellen dat slechts een enkeling in 'de praktijk' deze opvatting deelt. Volstrekt dominant is een globale benadering waarin reductionistische apecten slechts een ondergeschikte rol spelen. Dit culmineert in de onweerlegbare maar met logisch reductionistische opvattingen volstrekt onverenigbare opvatting dat een programma goed is wanneer de (menselijke) gebruiker ervan tevreden is.

Het is een open vraag of een logisch reductionistische benadering van software uiteindelijk van doorslaggevend belang zal blijken. Zo ja dan moet men wel aannemen dat dit tevens het failliet van het programma van de kunstmatige intelligentie impliceert. Het valt immers voorlopig niet aan te nemen dat de menselijke cognitie via logische reductie profijtelijk verklaard kan worden. De twee opvattingen omtrent digitale systemen die ik in het begin heb geformuleerd staan elkaar ogenschijnlijk naar het leven.

De waarheid zal echter ook hier in het midden liggen. Afhankelijk van de stijl van het systeemontwerp zal een reductionistische aanpak meer kansen hebben en er zullen altijd zeer complexe systemen zijn waarvan betrouwbare voorspelbaarheid buitengewoon gewenst is. Slechts een reductionistische aanpak kan deze voorspelbaarheid garanderen. Voor andere functionaliteiten is deze voorspelbaarheid echter irrelevant, daar heeft een reductionistische aanpak navenant minder belang.

We zien hier een merkwaardig spanningsveld tussen verschillende stijlen van omgang met dezelfde technologie, beide vallende onder het te wijde begrip programmatuur. Mijn verwachting is dat dit spanningsveld zal leiden tot een harde, zeer langdurige en deels ideologisch getinte scholenstrijd die een aanzienlijke plaats voor zich zal opeisen in het toekomstige academisch bestel.

Referenties

[B 90] J.C.M. Baeten (editor), Applications of Process Algebra, Cambridge University Press (1990)

[BW 90] J.C.M Baeten & W.P. Weijland, Process algebra, Cambridge University Press (1990)

[C 28] R. Carnap, Die Logische Aufbau der Welt, Berlin (1928)

[Co 90] P. Cousot, Methods and logics for proving programs, in: [vL 90] pp. 841-994 (1990)

[D 90] N. Dershowitz & J.P. Jouannaud, Rewrite systems, in: [vL 90] pp. 243-320 (1990),

[G 31] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, Monatshefte für Mathematik und Physik, vol. 38, pp. 173-198 (1931)

[GP 91] J.F. Groote & A. Ponse, The syntax and semantics of μCRL, RACE project no. 1046 SPECS / Centrum voor Wiskunde en Informatica, Amsterdam (1991)

[vGV 89] R. van Glabbeek & F.W. Vaandrager, Modular specifications in process algebra with curious queues, in: Springer Lecture Notes in Computer Science 394, pp. 465-506 (1989)

[Ha 84] D. Harel, Dynamic logic, in: Handbook of Philosophical Logic, Vol. 2, eds. D. Gabbay & F. Günther, pp. 496-604, Reidel (1984)

[Ho 85] C.A.R Hoare, Communicating sequential processes, Prentice Hall International (1985)

[J 89] H.B.M. Jonkers, Description algebra, in: Springer Lecture Notes in Computer Science 394, pp. 283-306 (1989)

[vL 90] J. van Leeuwen (editor), Handbook of Theoretical Computer Science, Vol. B, Formal Models and Semantics, Elsevier, The MIT Press (1990)

[Mi 89] R. Milner, Communication and concurrency, Prentice Hall international (1989)

[MG 85] J. Meseguer & J.A. Goguen, Initiality induction and computability, in: Algebrai Methods in Semantics, eds. M.Nivat & J.C Reynolds, pp. 459-541, Cambridge University Press (1985)

[Mu 90] J.C. Mulder, Case Studies in Process Specification and verification, Academisch Proefschrift, Universiteit van Amsterdam (1990)

[N 80] D. Normann, Recursion on the countable functionals, Lecture Notes in Mathematics 811, Springer (1980)

[RS 59] M.O. Rabin & D.S. Scott, Finite automata and their decision problems, IBM Journal of research and development, vol.3, pp. 114-125 (1959)

[T 36] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society 2 vol. 42, pp. 230-265; vol. 43, pp. 544-546 (1936)

[W 90] M. Wirsing, Algebraic specification, in: [vL 90] pp. 675-788 (1990)

Eredoctoraten

Dr. V.I. Arnol'd

Erepromotores: prof.dr. J.J.Duistermaat en prof.dr. D. Siersma

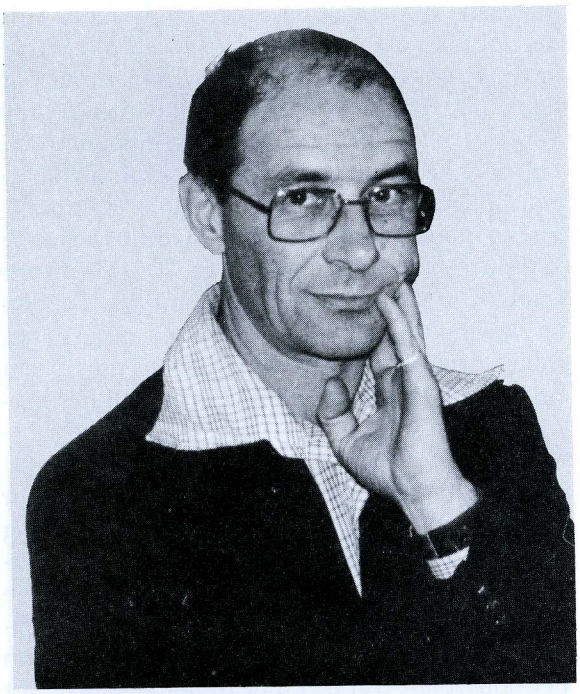
Professor Arnol'd is één van de belangrijkste hedendaagse wiskundigen, met een hoofdrol in minstens twee gebieden van onderzoek: de dynamische systemen en de singulariteitentheorie. Zijn werk is van fundamenteel belang voor vele wiskundigen in de wereld, in Nederland en in het bijzonder in Utrecht. Hij werd zo'n 30 jaar geleden in één keer wereldberoemd met zijn bewijs van een vermoeden van zijn leermeester Kolmogorov. Dit ging over quasi-periodieke bewegingen, in Hamiltonsystemen die in goede benadering integreerbaar zijn. Een voorbeeld is ons planetenstelsel, waar de integreerbare situatie bestaat uit de idealisatie dat de planeten geen massa hebben in verhouding tot de zon. In het bewijs worden moeilijkheden overwonnen die al in de vorige eeuw door Poincaré waren onderzocht.

In de talloze bijdragen van Arnol'd aan de theorie van dynamische systemen vormen bifurcaties een steeds terugkerend thema. Dat zijn plotselinge veranderingen in het gedrag van het systeem, bijvoorbeeld de overgang van een stationaire beweging in een periodieke. Ook past hij Hamiltonsystemen toe in de theorie van oscillatorische integralen. Sedert de latere 60-er jaren ontplooit Arnol'd een indrukwekkende activiteit in de singulariteitentheorie. Hierbij worden families van functies van één of meer variabelen onderzocht bij punten waar de functie een lokaal maximum heeft. Of een lokaal minimum, of een zadelpunt, of nog ingewikkelder.

De Moskouse school, onder leiding van Arnol'd, ontwikkelde systematisch en compleet de theorie van geïsoleerde singulariteiten van complexe functies. Zoals bij al het werk van Arnol'd, worden hierbij algebra, meetkunde en analyse op soepele wijze gecombineerd. Zelf zegt hij gewoonlijk meetkundig te denken en dat hij liever plaatjes tekent dan formules schrijft.

Zowel dynamische systemen, oscillatorische integralen als singulariteitentheorie worden zeer actief bij de vakgroep Wiskunde in Utrecht onderzocht, daarbij sterk geïnspireerd door het baanbrekende werk van Arnol'd. De diesrede van prof. Eckhaus van vorig jaar ging over chaos in dynamische systemen. Een recent Utrechts proefschrift over series van singulariteiten begint met een uitspraak van Arnol'd dat, "hoewel hij er geen definitie van heeft, series van singulariteiten zonder twijfel bestaan." De vakgroep maakt dan ook dankbaar van deze gelegenheid gebruik om de bestaande contacten verder te verrijken.

In zijn vele leerboeken en artikelen is Arnol'd een buitengewoon stimulerende docent. Hij heeft zeer behartigenswaardige, prikkelende meningen over het vak en hoe het beoefend zou moeten worden. Als hij iets niet kan bewijzen, maakt hij graag een skitocht van 40 'a 60 km, vaak in zwembroek, waarbij het probleem zichzelf oplost. Deze opmerkelijke methode zal zeker niet bij eenieder even goed werken.



Vladimir Igorewitsj Arnol'd

Biografie

Vladimir Igorewitsj Arnol'd werd op 12 juni 1937 in Rusland geboren. Hij studeerde bij de afdeling Wiskunde en Mechanica van de Staatsuniversiteit van Moskou, deed doctoraal examen in 1959, zijn eerste doctoraat in 1961 en zijn tweede, vergelijkbaar met de duitse Habilitation, in 1963.

Sedert 1965 is Arnol'd professor aan de Staatsuniversiteit van Moskou en medewerker aan het Steklov Mathematisch Instituut en sedert 1984 is hij geassocieerd lid van de Academie van Wetenschappen van de USSR.

In 1961 ontving hij de prijs voor jonge wiskundigen van de Mathematische Sociëteit van Moskou en in 1965, samen met Kolmogorov, de Leninprijs. In 1982 volgde de Internationale Crafoord prijs van de Zweedse Academie van Wetenschappen. In 1974 werd hij doctor honoris causa aan de Universiteit van Parijs. Buitenlands lid werd hij in 1983 van de US National Academy of Sciences, in 1984 van de Academie van Wetenschappen te Parijs en in 1985 van de American Academy of Arts and Sciences.

Dr. Chr. Nüsslein-Volhard

Erepromotores: prof.dr. J.A.M. van den Biggelaar en prof.dr. S.W. de Laat

Dr. Nüsslein-Volhard heeft gedurende meer dan een decennium een prominente rol gespeeld op de raakvlakken van de moleculaire genetica, genetica en ontwikkelingsbiologie. Door haar grote technische en intellectuele vaardigheden is zij er in geslaagd een concept uit te werken dat een sluitende verklaring kan geven voor de wijze waarop vroeg in de embryonale ontwikkeling complexe structuren uit minder complexe en minder informatierijke voorafgaande organisatiepatronen kunnen voortkomen. Daarmee heeft zij een belangrijke bijdrage geleverd aan het tot stand komen van de zo noodzakelijke synthese tussen de resultaten van de klassiek experimentele embryologie, genetica en moleculaire genetica.

Bij experimenteel embryologen bestond reeds geruime tijd de opvatting dat een embryo niet is voorgevormd in de bevruchte eicel, maar dat de eicel voorzien is van een preliminaire structuur die tengevolge van de wisselwerking tussen de samenstellende componenten geleidelijk complexer wordt. Het is de verdienste van dr. Nüsslein-Volhard dit concept in moleculair genetische zin te funderen. Zij heeft een unieke genetische, embryologische en moleculair genetische analyse uitgevoerd van tal van ontwikkelingsmutanten van Drosophila. Door de resultaten daarvan heeft zij onomstotelijk duidelijk kunnen maken hoe genetisch bepaalde morfogenen op grond van hun localisatie en onderlinge wisselwerking een preliminair patroon genereren, dat vervolgens het fundament vormt voor de realisatie van de latere lichaamscontouren. Het door haar en haar medewerkers ontwikkelde concept geeft de lang verbeide moleculair biologische inhoud aan de gradientveldtheorie uit de klassieke embryologie en het idee van positionele informatie uit de theoretische biologie. De oorspronkelijk gescheiden wegen van de genetici, moleculair biologen en embryologen zijn in haar onderzoek tot een zeer vruchtbare synthese gekomen.

Haar resultaten zijn niet alleen van belang voor een goed begrip van de embryonale ontwikkeling van de fruitvlieg en de wijze waarop die ontwikkeling door moleculair genetische informatie wordt geregeld. Haar concept is van fundamenteel belang voor ons inzicht in de wijze waarop het bouwplan van een dier in het algemeen tot stand komt. Het is daarmee eveneens van belang voor een goed begrip van de vroege ontwikkeling van de gewervelde dieren, de zoogdieren en de mens daarbij inbegrepen.

Het ontwikkelingsbiologisch onderzoek dat binnen het kader van het Centrum voor Ontwikkelingsbiologie te Utrecht wordt uitgevoerd, is in belangrijke mate gericht op de wijze waarop tijdens de vroege ontwikkeling van gewervelde en ongewervelde dieren de lichaamsassen worden bepaald en daarmee de fundamenten voor het

Dr. Christiane Nüsslein-Volhard

bouwplan worden vastgelegd. Er bestaat dan ook een nauwe wetenschappelijke relatie tussen de werkzaamheden van dr. C. Nüsslein-Volhard en een belangrijk deel van het onderzoek dat binnen het Centrum voor Ontwikkelingsbiologie wordt uitgevoerd. De resultaten van haar onderzoek zijn direct van groot belang voor het ontwikkelingsbiologisch onderzoek aan de Rijksuniversiteit te Utrecht en het Hubrechtlaboratorium van de Koninklijke Nederlandse Akademie van Wetenschappen.

Dr. Christiane Nüsslein-Volhard heeft door haar werk niet alleen de genen achterhaald die bij het vastleggen van de contouren van het lichaam een grote rol spelen, maar zij heeft tevens duidelijk gemaakt dat de afzonderlijke genprodukten alleen dan een morfogenetische rol kunnen vervullen wanneer zij in een tijd-ruimtelijk patroon met elkaar in wisselwerking treden, en dat de interactie tussen bepaalde, gelocaliseerde genprodukten weer conditionerend is voor de expressie van een volgende set patroonvormende genen. Daarmee komt haar onderzoek op eenzelfde lijn als het onderzoek naar de rangschikking van patroonvormende genen in een vaste hiërarchische volgorde. Haar onderzoek bevestigt de opvatting dat de activatie van bepaalde genen een noodzakelijke voorwaarde is voor de expressie van lager gerangschikte genen. Het heeft daarmee tevens het inzicht in de organisatie van het genoom belangrijk vergroot.

De reikwijdte van haar resultaten strekt zich

verder uit dan het terrein van de ontwikkelingsbiologie. Haar werkzaamheden oefenen een aanwijsbare invloed uit op de theorievorming binnen het moleculair genetisch, genetisch en evolutiebiologisch onderzoek. Het aanhalen van de banden met mevr.dr. C. Nüsslein-Volhard zal dan ook meerdere groepen van de Utrechtse Universiteit ten goede komen.

Biografie

Dr. Christiane Nüsslein-Volhard werd geboren op 20 oktober 1942 in Magdenburg. Zij volgde haar opleiding in onder andere Biologie en Biochemie te Frankfurt am Main en Tübingen. In 1973 promoveerde zij bij Prof. Dr. Seyffert en Prof. dr. Gierer. Daarna werd mevrouw Nüsslein wetenschappelijk assistent aan het Max Planck Instituut voor Virusonderzoek. Van 1978 tot 1981 bezette zij een onafhankelijke researchpost aan het Europese Laboratorium voor Moleculaire Biologie in Heidelberg.

Sinds 1981 is zij een van de vijf directeuren van het Max Planck Instituut voor Ontwikkelingsbiologie te Tübingen.

Vorig jaar werd mevrouw Nüsslein-Volhard al eredoctor van de Universiteit van Yale; later dit jaar volgt, na Utrecht, een zelfde onderscheiding in Princeton.