Computer Wiskunde

Computers en wiskunde

In de jaren 50 werd er nog serieus gedacht dat er in een land als Nederland slechts behoefte zou zijn aan een paar computers. Tien jaar later kwamen er echter langzaam maar zeker steeds meer van die apparaten. Redelijk veel mensen wisten dat computers gebruikt werden voor het doen van berekeningen en het bijhouden van een administratie. Aan wiskundigen werd dan wel gevraagd: 'Jij gebruikt zeker een computer?' De meeste wiskundigen haastten zich te antwoorden dat dat niet zo was: 'Computers maken alleen numerieke berekeningen. Zelfs een getal-theoreticus hoeft nauwelijks met computers te werken.' We zullen zien dat er in de loop van de tijd veel is veranderd.

In het begin van de jaren 60 werden de eerste programma's ontwikkeld voor het uitvoeren van symbolische berekeningen. Een getal als $\sqrt{2}$ kan met behulp van dergelijke programma's exact op een computer gerepresenteerd worden. Dit getal wordt gewoon als symbool gerepresenteerd en hiermee wordt dan 'gerekend'. Deze programmas waren in de jaren 80 zo ver geperfectioneerd dat ze als commerciële pakketten 'Computer Algebra' verkocht werden. Bekende systemen zijn Mathematica, Maple en Reduce. Deze systemen kunnen de meest uiteenlopende vormen van symbolische (en numerieke) berekeningen uitvoeren. Bijvoorbeeld het werken met polynomen van meerdere veranderlijken, zoals het ontbinden in factoren. Het symbolisch oplossen van lineaire vergelijkingen behoort verder tot de mogelijkheden. Ook berekeningen komend uit de analyse kunnen gedaan worden, bijvoorbeeld het vinden van een primitieve functie en verder ook meer ingewikkelde differentiaal vergelijkingen. De mogelijkheden worden slechts beperkt door het al dan niet voorhanden zijn van een efficiënt algoritme voor het doen van de symbolische of numerieke berekeningen. Als een mens het kan uitrekenen, dan ook de computer. De hulpmiddelen zijn nu zo krachtig en gebruikersvriendelijk, dat ze gebruikt worden door veel onderzoekers werkend in de exacte wetenschappen. Ook in de economie en sociale wetenschappen worden systemen voor Computer Algebra gebruikt. Zelfs zuivere wiskundigen, die uit zijn op het ontwikkelen van nieuwe theorie, maken gebruik van deze systemen door deze op de juiste manier in te zetten (onder andere door er experimenten mee te doen).

Onafhankelijk van deze ontwikkeling is er sinds het einde van de jaren 60 gewerkt aan formele talen voor de exacte representatie van willekeurige wiskundige uitspraken en begrippen. In tegenstelling tot de systemen voor Computer Algebra kunnen er in deze talen ook niet-berekenbare begrippen weergegeven worden. In de wiskunde ontstaat niet-berekenbaarheid door de zogenaamde kwantoren $\forall x{\in}\mathbb{N}.Px$ ('voor alle natuurlijke getallen x geldt Px') en $\exists x{\in}\mathbb{N}.Px$ ('er is een x natuurlijk getal zodat Px'), waarbij $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ de verzameling natuurlijke getallen is. Omdat deze verzameling oneindig is, kan een computer niet nagaan of alle elementen een bepaalde eigenschap hebben. Een wiskundige kan echter door middel van *bewijzen* wel met niet-berekenbare begrippen omgaan. Deze bewijzen zullen zodanig weergegeven moeten worden, dat een computer deze kan gebruiken. Hiervoor is het nodig dat het niveau van detail van deze bewijzen voldoende groot is dat het door de machine gevolgd kan worden. Indien dit niveau bereikt is, dan spreekt men wel van bewijs-objecten. Het is niet zo dat een computer deze bewijs-objecten zelf kan vinden. Daarvoor heeft die de samenwerking met een mens nodig. In een interactieve sessie kunnen bewijzen ontwikkeld worden, waarbij de mens de goede ideeën geeft en de computer helpt met het benodigde ambachtelijke vakmanschap. In de jaren 80 zijn er in de USA en Europa verschillende prototype systemen voor bewijs-ontwikkeling en -verificatie geconstrueerd. De reden dat dit nog slechts prototypen zijn is gelegen in het feit dat het genereren van een bewijs-object op dit moment nog veel moeite kost, meer dan een wiskundige bereid is te doen.

Logica en type theorie

De Griekse filosoof Aristoteles heeft een duidelijke invloed gehad op de huidige technologie van de Computer Wiskunde. In zijn werken worden de onderwerpen de axiomatische methode, het zoeken naar de logica en bewijs-verificatie versus -constructie behandeld. Op al deze fronten heeft Aristoteles belangrijk werk geïnitieerd.

De axiomatische methode

Volgens Aristoteles bestaat een wiskundige theorie uit begrippen (zoals bijvoorbeeld in de meetkunde punten, lijnen en incidentie—het liggen van een punt op een lijn) en uit stellingen over die begrippen. Begrippen vormt men door middel van definities uit andere begrippen. Omdat men ergens moet beginnen zijn er primitieve begrippen, die niet nader gedefinieerd worden. Stellingen bewijst men met behulp van redeneringen uit andere stellingen. Weer moet men ergens beginnen: bij de primitieve stellingen ook wel axioma's genoemd. Deze behoeven geen bewijs. Niet lang nadat Aristoteles deze methode beschreven had, heeft Euclides zijn beroemde boeken 'De Elementen' geheel in de stijl van de axiomatische methode geschreven.

De axiomatische methode is zeer krachtig, omdat men alleen uitspraken bewijst op grond van wat er gegeven is aan primitieve begrippen en axioma's. Een groot gedeelte van de wiskunde kan zonder meer axiomatisch opgebouwd worden. Voor enkele gebieden, waaronder de combinatoriek, is dat moeilijker.

Het zoeken naar de logica

Om uit gegeven geldige uitspraken nieuwe af te leiden gebruiken wij redeneringen. De juiste redeneringen waren volgens Plato aan ons bekend omdat wij in contact staan met de zuivere rede. Aristoteles vroeg zich echter af, of alle mogelijke juiste redeneerstappen in kaart konden worden gebracht. Met andere woorden, of het redeneren gevangen kon worden in een eindig aantal regels. Plato vond deze vraagstelling onjuist. Volgens hem was ons contact met de zuivere rede duidelijk genoeg. Ondanks deze waarschuwing maakte Aristoteles een indrukwekkend begin met het zoeken naar de logica, door voor een fragment, de zogenaamde monadische logica, de juiste regels in kaart te brengen. Hierin worden alleen predicaten met één argument beschreven, zoals bijvoorbeeld $E(n) =_{\rm Def} (n \text{ is even})$; binaire predicaten, zoals $G(n,m) =_{\rm Def} (n > m)$, komen in de monadische logica niet voor. Met name gaf Aristoteles regels voor de implicatie $A{\rightarrow}B$ (als A, dan B) en de universele kwantificatie (voor alle x geldt Px). Het duurde meer dan 2300 jaar totdat het plan van Aristoteles voltooid was. Het logische systeem werd gevonden door Frege in 1879 en Gödel bewees in 1930 dat het voldoende sterk was voor alle benodige redeneerstappen voor het ontwikkelen van axiomatische wiskunde.

Verificatie versus constructie van bewijzen

Aristoteles heeft opgemerkt dat het verifieren van de correctheid van een bewijs in principe niet moeilijk is: men volgt het bewijs stap voor stap en kijkt of het klopt. Aan de andere kant wees hij erop dat het vinden van een bewijs wel moeilijk is: bij een gegeven stelling (die weliswaar bewezen is, maar waarbij het bewijs bijvoorbeeld verloren is gegaan) zijn wij niet altijd in staat binnen redelijke tijd een bewijs terug te vinden. (Indien wij de beschikking hebben over willekeurig veel tijd, dan is dit in principe wél mogelijk. Men kan namelijk alle mogelijke bewijzen genereren en kijken of de gevraagde stelling er toevallig uit komt. Op den duur moet dit het geval zijn, want we weten dat er is een bewijs voor de stelling is. Dit is echter onrealistisch, vergelijkbaar met het wachten op een aap aan een tekstverwerker, totdat die op een gegeven moment per ongeluk een gedicht van Paul van Ostaayen zal intikken.) Deze observeringen zijn juist en hebben in onze eeuw tot twee belangrijke corollaria geleid. N.G. de Bruijn heeft rond 1968 een formeel systeem (Automath) ontworpen, waarin bewijzen zodanig formaliseerdbaar zijn, dat deze op efficiënte wijze machinaal geverifieerd kunnen worden en Turing heeft in 1936 bewezen dat voor een willekeurig gegeven uitspraak niet automatisch bepaald kan worden of deze al dan niet volgt uit een gegeven verzameling axiomas.

Brouwer

Ondertussen kwam er uit onverwachte hoek kritiek op de logica van Aristoteles. De jonge Nederlandse wiskundige L. E. J. Brouwer durfde in zijn proefschrift van 1907 de bewering te poneren dat de principes van de logica van Aristoteles onbetrouwbaar waren. Met name voor het principe van het uitgesloten derde

$$A \vee \neg A$$

(A of niet A) was dat volgens Brouwer het geval. Vrijwel niemand wilde hem geloven. Brouwer drukte zulke feiten uit op een voor hem typerende wijze: "Medestanders hebben onderstaande stellingen nog weinig gevonden." Pas nadat Brouwer een drietal jaren hard had gewerkt op een heel ander gebied van de wiskunde en de oplossing van een aantal belangrijke problemen had gevonden, begon men hem ook op het gebied van de grondslagen *au sérieux* te nemen. De analyse van Brouwer kwam er hierop neer, dat het principe van het uitgesloten derde tot gevolg heeft dat men een uitspraak als

$$\exists n \in \mathbb{N}. P(n),$$

kan bewijzen zonder dat een 'getuige' n gevonden kan worden, waarvoor $P(n)$ bewijsbaar is.

....

Beschouw een uitspraak A, waarvan het (nog) niet bekend is of deze geldt of niet. Laat

$$P(n) \ = \ [(n = 0 \ \& \ A) \vee (n = 1 \ \& \ \neg A)],$$

dat wil zeggen (n=0 en A of n=1 en niet A). Dan kunnen we met het principe van het uitgesloten derde bewijzen dat geldt

$$\exists n \in \mathbb{N}. P(n).$$

Immers, als A geldt, dan kunnen we nemen $n = 0$ en als $\neg A$ geldt, dan nemen we $n = 1$. Volgens de logica van Aristoteles geldt $A \vee \neg A$, dus er is in alle gevallen een n zodat $P(n)$ geldt. Maar aangezien we echter niet weten welk van de twee mogelijkheden A of $\neg A$ geldt, kunnen we geen getal n vinden zodat we kunnen bewijzen $P(n)$; want daartoe moeten we eerst A bewijzen of weerleggen.

....

Voorbeeld van een bewijsbare uitspraak $\exists n \in \mathbb{N}. P(n)$ zonder 'getuige' n, waarvoor $P(n)$ bewijsbaar is.

In feite heeft Brouwer een iets andere interpretatie aan de logische operatoren (zoals &, $\vee$, $\rightarrow$, $\forall$ en $\exists$) gegeven. De zogenaamde intuïtionistische logica van Brouwer is een verfijning van de klassieke logica van Aristoteles, omdat in Brouwer's logica een existentiele uitspraak alleen bewijsbaar is, indien er een getuige gevonden kan worden.

Heyting heeft in 1930 voor de geldige redeneringen van de intuïtionistische wiskunde een formeel systeem geïntroduceerd. In 1956 heeft Beth (met klassieke logica in de metataal) en in 1976 hebben Veldman en de Swart (met intuïtionistische logica in de metataal) aangetoond dat hiermee de regels voor het intuïtionistische redeneren volledig in kaart is gebracht. De formalisering van de logica van Frege en van Heyting waren volledig, maar niet erg natuurlijk. Gentzen heeft in 1935 een veel mooiere vorm gegeven aan beide versies van logica. De klassieke logica in de vorm van de 'sequenten calculus' en voor de intuïtionistische logica in de vorm van het systeem van 'natuurlijke deductie'.

Gebaseerd op de intuïtionistische operationele interpretatie van de logische voegwoorden heeft N.G. de Bruijn in 1968 als belangrijke bijdrage aan de Computer Wiskunde de reeds genoemde klasse van Automath talen ingevoerd. De weergave van bewijzen in Automath bestaat in feite uit een natuurlijke deductie bewijs. Hierbij is de zogenaamde type theorie van Russell essentiëel uitgebreid en gebruik gemaakt van de lambda notatie van Church. De correctheid van een op deze manier weergegeven bewijs kan in dit systeem op efficiënte manier geverifieerd worden. De bewijzen in Automath zijn in feite intuïtionistisch. Wil men verder klassiek redeneren, dan kan dat, eenvoudig weg door het principe van het uitgesloten derde als axioma aan te nemen.

Bewijs ontwikkeling en verificatie

Zoals aangegeven door Aristoteles bestaat een belangrijk deel van het doen van wiskunde eruit om in een bepaalde axiomatische context definities te geven voor nieuwe begrippen en bewijzen te geven voor nieuwe stellingen. Wanneer deze definities en bewijzen volledig geformaliseerd zijn, dan kan automatisch geverifieerd worden of de definities welgevormd zijn en of de bewijzen kloppen. In een systeem als Automath is de verificatie zelfs efficiënt. Het is echter behoorlijk lastig om een bewijs volledig te formaliseren. Om deze taak gemakkelijker te maken zijn er zogenaamde 'bewijs ontwikkel systemen' gebouwd. In een interactieve sessie tussen gebruiker en machine worden de complexe formele bewijzen gevormd. Het komt ongeveer neer op het volgende. De gebruiker geeft aan welke stelling bewezen moet worden. Dit wordt het 'doel' genoemd. De machine vraagt dan langs welke weg de gebruiker denkt dat het bewijs mogelijkerwijs loopt. Daarbij worden nieuwe doelen gegenereerd. Wanneer dit proces voldoende vaak en op de juiste wijze herhaald wordt, dan zal het volledig geformaliseerde bewijs voltooid zijn en de computer zal kunnen verifieren of dit correct is of niet. De serie kommando's die de gebruiker moet intikken om een formeel bewijs te genereren heet ook wel een 'taktiek'. Men kan zich afvragen waarom computer ondersteunde theorievorming gewenst is. Immers, er is een goede traditie in de wiskunde om een hoge graad van betrouwbaarheid te bereiken. Als antwoord kan ten eerste opgemerkt worden dat de wiskundige graad van precisie in de loop van de geschiedenis gegroeid is. De axiomatische meetkunde van Euclides is pas door Hilbert geheel rigoureus opgezet en de analyse van Newton en Leibniz is pas door Weierstrass en Dedekind gepreciseerd. De graad van betrouwbaarheid verkregen door computer verificatie is een verdergaande stap in deze ontwikkeling. Ten tweede zijn er door de digitalisering van de informatie technologie belangrijke situaties waarbij de verificatie van bepaalde uitspraken van industrieel belang is. Bijvoorbeeld is dit het geval bij de constructie van hardware en software. De correctheid van een stuk hardware kan vertaald worden naar de correctheid van een omvangrijke uitspraak. Deze uitspraak is wiskundig niet diepgaand, maar door de omvang is computer ondersteuning bij de verificatie onmisbaar. Reeds meer dan een decennium wordt deze methode bij het ontwerpen van hardware gebruikt. De uitspraken die overeenkomen met de correctheid van software zijn moeilijker. Daarom is er voor dit gebied nog geen volledig succes. Voor een klasse van kleine maar belangrijke programma's, de zogenaamde communicatie protocollen, die bijvoorbeeld een afstandsbediening van een TV regelen, wordt er wel vruchtbaar gebruik gemaakt van computer ondersteunde verificatie. In Nijmegen gebeurt dit door collega Vaandrager en zijn groep.

Uitspraken in de wiskunde zijn niet omvangrijk, maar wel diep. Het verschil met uitspraken uit de hard- of software specificatie zit hem erin dat voor de bewijzen van deze laatste categorie een grote hoeveelheid relatief uniforme stappen nodig zijn, terwijl voor wiskundige bewijzen er een relatief kleine verzameling stappen gedaan moet worden, maar wel met een grote diversiteit. Hoewel in de wiskunde (gewone nog niet geformaliseerde) bewijzen onderworpen worden aan een strenge kritiek van vakgenoten ('peer review'), is het ook voor deze discipline nuttig om computer geverifieerde bewijzen te hebben. Het sociologische verificatieproces kan nu gedeeltelijk overgenomen worden door machines. De 'peer reviews' blijven relevant omdat een machine niet kan bepalen wat het belang van een nieuwe stelling is. Verder ligt het in de lijn der verwachting, dat een wiskundige op den duur essentiële hulp zal krijgen van systemen voor bewijs ontwikkeling. De systemen voor Computer Algebra worden, zoals vermeld, op deze manier al gebruikt voor onderzoek in de zuivere wiskunde.

Methodologie

Men kan zich afvragen of een bewijs dat door een computer geverifieerd is, eigenlijk wel correct is. Met andere woorden, de vraag komt op of het programma dat de verificatie uitvoert zelf wel klopt. Deze vraag is op een methodologisch bevredigende manier beantwoord door N.G. de Bruijn, die aantoonde dat het mogelijk is voor bewijzen die in voldoende mate geformaliseerd zijn een zeer kort verifierend programma te gebruiken. Dit heeft ermee te maken dat de correctheid van een redenering kan worden bepaald door voor iedere stap na te gaan of deze voorkomt in de korte lijst van logische stappen die toegestaan zijn. Door nu met de hand dit korte verificatie programma heel goed na te kijken, heeft men de hoogst mogelijke graad van zekerheid. Dit geeft aan wat het belang van geformaliseerde bewijzen is: de skeptische gebruiker kan desgewenst zijn eigen verificatie programma schrijven. Een systeem dat alleen uitspraken doet zonder bewijs erbij, is *a priori* minder betrouwbaar.

Omdat er tijdens bewijzen af en toe gerekend moet worden, zal men soms gebruik willen maken van algoritmen. Om het bewijs nu geloofwaardig te houden moet de correctheid van dit algoritme ook nog bewezen worden. Omdat bij vele efficiënte algoritmen nog geen correctheidsbewijs voorhanden is, wordt er door sommige gebruikers van een verificatie systeem water in de wijn gedaan door deze correctheid axiomatisch aan te nemen. Anderen streven naar de hoogste graad van betrouwbaarheid door alleen een minimum aan niet formeel geverifieerde algoritmen toe te laten. Omdat er verschillende soorten gebruikers zijn, kan een systeem voor bewijsverificatie in principe voorzien worden van een 'joystick', die bepaalt hoeveel algoritmen men om pragmatische redenen aanneemt correct te zijn.

Op weg naar Computer Wiskunde

Onder 'Computer Wiskunde' wordt verstaan het met behulp van systemen voor bewijs ontwikkeling tot stand brengen van gevalideerde uitspraken. De menselijke intuïtie zal hierbij nog wel geruime tijd een rol bij blijven spelen: pas als de gebruiker weet hoe het bewijs loopt, kan samen met het systeem een formeel bewijs ontwikkeld worden. Hoewel er reeds spectaculaire successen geboekt zijn, met name op het gebied van het ontwerpen van hardware met een hoge graad van betrouwbaarheid, zijn er nog vele fronten waarop de methode aanzienlijke verbetering behoeft. De belangrijkste is het gebruikersgemak. Indien het formaliseren van een stuk wiskunde vergelijkbaar wordt met het schrijven ervan op een modern systeem voor wiskundige tekstverwerking, zoals LaTeX, dan zal er een wezenlijke vooruitgang geboekt zijn. Het is mogelijk dat het formaliseren zelfs nog gemakkelijker kan worden, omdat LaTeX 'plat' is, dat wil zeggen geen wiskundige betekenis kent, terwijl de in type theorie geformaliseerde wiskundige bewijzen een rijke structuur bezitten, die bij het intikken van pas kan komen. Om dit te bereiken is onderzoek nodig waarbij fundamenteel logische, wiskundige en implementatie kennis ingebracht zal moeten worden.

Wanneer de systemen voor Computer Wiskunde voldoende ver ontwikkeld zullen zijn, zal de rol van een referee alleen nog bestaan uit de beoordeling van de relevantie van een artikel, want de correctheid is immers al gegarandeerd. Verder kan er internationaal via het internet gewerkt worden aan een grote database van gevalideerde wiskunde. De bewijzen hieruit kunnen dan door andere onderzoekers gebruikt worden voor het vormen van nieuwe theorie.

Omdat alle wiskundige objecten exact op een systeem voor Computer Wiskunde gerepresenteerd kunnen worden, kan men in principe een ingewikkelde integraal selecteren en op een knop drukken om de numerieke waarde in willekeurig veel decimalen weer te geven. Numerieke wiskunde en Computer Algebra zullen geïntegreerd worden met Computer Wiskunde. Uiteraard zullen de specifieke methoden van die twee vakken relevant zijn en blijven.

Tenslotte is Computer Wiskunde van belang voor het onderwijs. Met dit hulpmiddel zullen er interactieve boeken mogelijk worden, waarbij de lezer de opgave kan krijgen om een stuk wiskunde te ontwikkelen en meteen kan zien of het werk goed gemaakt is. Misschien dat hiermee de schade, veroorzaakt door het nagenoeg afschaffen van bewijzen op het VWO, nog ongedaan gemaakt kan worden.

Henk Barendregt
Met dank aan Frans Janssen voor hulp in de formulering.