Bewijsassistenten

Computerondersteund redeneren: de boekhouder steunt de denker

Bewijsassistenten moeten ervoor zorgen dat complexe softwaresystemen correct werken. Geen eenvoudige klus, zegt Herman Geuvers, die vandaag zijn inaugurele rede houdt. Het is nog vrijwel onmogelijk formele wiskunde tussen verschillende bewijsassistenten uit te wisselen.

Computers maken fouten. Soms zijn dat vervelende fouten waardoor gebruikers zich ergeren of werk verliezen, maar soms zijn het grote fouten, waardoor grote sommen geld verloren gaan of mensen zelfs hun leven kunnen verliezen. Hoe voorkomen we deze fouten? Door de software en hardware op een gedegen manier te ontwikkelen en de gewenste eigenschappen te verifiëren. Er zijn veel methoden om software en hardware te verifiëren, bijvoorbeeld door middel van testen. Testen is het gestructureerd zoeken naar fouten door bij bepaalde invoer te kijken of de uitvoer klopt. Hiermee kunnen we fouten vinden, maar nooit alle en als we geen fouten vinden hebben we geen garantie dat die er ook niet zijn. De ultieme vorm van correctheid is een correctheidsbewijs: een wiskundig bewijs dat een bepaald stuk computercode aan bepaalde eigenschappen voldoet.
Software of hardware correct bewijzen is ingewikkelde materie, omdat de systemen groot zijn. Daarom gebruiken we computerprogramma's om ons daarbij te helpen, de stellingbewijzers of bewijsassistenten (zie kader). Het woord 'stellingbewijzer' suggereert dat het systeem automatisch stellingen voor ons bewijst. Dat is echter niet zo en daarom is 'bewijsassistent' een betere term. De bewijsassistent gaat na of we de goede syntax gebruiken, houdt de bewijstoestand bij (wat moeten we nog doen) en kan bewijsstappen suggereren, maar het bewijs moeten we als gebruiker zelf leveren.
Het idee dat je software correct kunt bewijzen heeft de laatste jaren in de informatica op verschillende plekken ingang gevonden, onder andere ook bij Microsoft. Het blijkt dat veel van de fouten in Windows niet worden veroorzaakt door het besturingssysteem zelf, maar door slecht geschreven drivers voor randapparatuur die interfereren met het besturingssysteem. De Static Driver Verifier kan bij het compileren nagaan of drivers aan bepaalde regels voldoen. NASA gebruikt bewijsassistenten om software voor de luchtverkeersleiding te verifiëren en Intel gebruikt bewijsassistenten bij het verifiëren van nieuwe chips. Het correct bewijzen van software of hardware met behulp van een bewijsassistent is niet eenvoudig. We moeten de bewijzen heel precies maken en de bewijsstappen in veel detail geven. Bij het precies maken van de bewijsstappen en het nagaan of ze samen een correct bewijs vormen kunnen we goed een computer gebruiken. Dit is het boekhoudershetwerk dat we aan een machine kunnen overlaten.

Formele bibliotheek

We zijn nog lang niet zover dat we met de huidige bewijsassistenten eenvoudig een flink deel van de wiskunde – definities, bewijzen, berekeningen – kunnen formaliseren. Een belangrijk onderdeel van zo'n formalisatie is de bibliotheek van basiswiskunde waarop gebruikers nieuwe dingen kunnen doen. In dit kader is het interessant na te denken over de hoeveelheid werk die hiermee gepaard zou gaan. Een gemotiveerde berekening van Freek Wiedijk komt uit op 140 manjaren die nodig zijn om het standaardcurriculum van een wiskundestudie te formaliseren. Dat gaat de onderzoeksbudgetten op onze universiteiten ver te boven.
Het creëren van een grote bibliotheek van geformaliseerde wiskunde kost ontzettend veel tijd en mankracht en lijkt onmogelijk. Maar ontwikkelingen als Linux en Wikipedia laten zien dat een gedistribueerde goed georganiseerde opzet met een lichte basistechnologie veel kan bewerkstelligen. Men zou kunnen denken dat zo'n benadering voor het formaliseren van wiskunde niet werkt, maar dat vermoedde men van Wikipedia ook: Wikipedia is typisch iets dat in theorie niet werkt, maar in de praktijk wel. Het probleem is dat we de 'lichtgewicht basistechnologie' voor het formaliseren van wiskunde nog niet hebben. Er wordt wel toegewerkt naar zo'n Wikipedia voor formele wiskunde, bijvoorbeeld door bestaande geformaliseerde bibliotheken te presenteren op het internet.
Een voorbeeld hiervan is de Hypertextual Library of Mathematics (HELM), ontwikkeld aan de universiteit van Bologna. Voor de bewijsassistent Coq, ontwikkeld bij INRIA in Parijs, is er in Nijmegen een webinterface gemaakt waarmee iedereen die een internetverbinding heeft op eenvoudige wijze – zonder Coq te installeren – kan bijdragen aan één gezamenlijke bibliotheek die op een centrale server staat. Het plan is om deze webinterface uit te breiden tot een zogenaamde 'MathWiki'. Het grote probleem blijft dat het tot nu toe vrijwel onmogelijk is formele wiskunde tussen verschillende bewijsassistenten uit te wisselen. Ook op dit gebied zijn er veel ontwikkelingen, onder andere OMDoc, dat beoogt een standaard te worden voor wiskundige documenten op het web. OMDoc is ontwikkeld in Duitsland als een markupformaat en een datamodel dat zich richt op het representeren van de inhoud van wiskundige formules, berekeningen en bewijzen. In de VS wordt gewerkt aan 'Logosphere', een formele digitale bibliotheek voor wiskunde, waarbij de wiskunde kan worden uitgewisseld tussen verschillende bewijsassistenten.

Grote bibliotheek van geformaliseerde wiskunde lijkt onmogelijk

Hybride systemen zijn toepassing bewijsassistenten in informatica

Hybride systemen

Hybride systemen zijn een toepassing van bewijsassistenten in de informatica waaraan sinds kort wordt gewerkt. Zo'n systeem bevat zowel continue componenten, zoals een klok, een thermometer of een snelheidsmeter, als discrete componenten, zoals een gaskraan die in drie standen gezet kan worden. De besturingssoftware moet op basis van invoergegevens van sensoren de gaskraan aansturen zodat de temperatuur of de snelheid binnen een bepaalde bandbreedte blijft. Interessant hieraan is dat we voor het correct bewijzen van de besturingssoftware ook de omgeving moeten modelleren, door middel van differentiaalvergelijkingen. Verder is de toestandsruimte oneindig. We kunnen dus alleen na een discrete abstractie onze geautomatiseerde tools, zoals 'model checkers', toepassen. Om de hoogste graad van correctheid te verkrijgen willen we dit hele proces formaliseren in een bewijsassistent, gebruik makend van een formele bibliotheek van reële getallen. Hieraan wordt onder andere in Nijmegen en Eindhoven gewerkt.

Herman Geuvers
AG • 09-03-'07

Prof.dr. Herman Geuvers is als hoogleraar Computer-ondersteund redeneren verbonden aan het Institute for Computing and Information Science (ICIS) van de Radboud Universiteit Nijmegen (Faculteit der Natuurwetenschappen, Wiskunde en Informatica). Het artikel is een verkorte versie van de oratie die hij vandaag zal houden, met als titel: Computer-ondersteund redeneren: de boekhouder steunt de denker.

Twee rollen

Een bewijs speelt in de wiskunde twee rollen.
A. Een bewijs overtuigt de lezer van de correctheid van het gestelde.
B. Een bewijs legt uit waarom het gestelde geldt.
Rol (A) kan prima door een computer worden overgenomen. Zodra we een bewijs formeel opgeschreven hebben, kan een computerprogramma dit bewijs checken, als de boekhouder, die één voor één de stapjes nagaat en kijkt of het klopt. De Nederlandse wiskundige De Bruijn wist dit al in de jaren zestig van de vorige eeuw. In het door hem geleide Automath-project werden de eerste bewijscheckers ontwikkeld. Bij Automath typte de gebruiker een bewijs in, in de speciale syntax van het systeem, en de Automath-bewijschecker checkte dan of het bewijs klopte. De huidige systemen werken volgens hetzelfde idee, maar nu helpt het systeem ook om het bewijs te maken en daarom spreken we nu van een bewijsassistent. De bewijsassistent checkt de syntax, doet een paar stappen, probeert een paar stappen en houdt de bewijstoestand bij.

Mexicaanse hoed

Voor het opzetten van een bibliotheek van geformaliseerde wiskunde werkt de zogenaamde 'Mexicaanse Hoed'-benadering goed. Hierbij richten we ons op een grote hoofdstelling, maar resultaten die we bewijzen formuleren we zo algemeen mogelijk en stoppen we in een algemene bibliotheek. Zo ontstaat een brede basis (de rand van de hoed) en een piek met technische ad-hoclemma's voor onze hoofdstelling (de piek van de hoed). Idealiter ontstaan er na verloop van tijd meerdere pieken op de rand en wordt de rand steeds dikker. Doel is dus om de rand zo dik mogelijk en de pieken zo smal mogelijk te houden.