Met de computer wiskundige bewijzen controleren

De QED utopie

In tegenstelling tot wat je misschien zou verwachten gebruiken wiskundigen computers nauwelijks om hun wiskunde mee te doen. Ze gebruiken ze als tekstverwerker (om hun artikelen en boeken mee te schrijven), en ze gebruiken ze voor experimenten (om te kijken hoe speciale gevallen van hun stellingen zich gedragen), maar ze gebruiken ze niet om bewijzen mee te controleren. Wiskundige bewijzen zitten in mensenhoofden of zijn opgeschreven in mensentaal, maar tot nog toe zijn ze bijna nooit zó opgeschreven dat er geen menselijk begrip nodig is om ze te kunnen nalopen.

In 1994 publiceerde een anonieme groep wiskundigen en informatici het *QED Manifesto*.[1] Hierin wordt een toekomst beschreven waarin alle wiskundige bewijzen in de computer zijn gecodeerd. De QED utopie – een wereld waarin de computer wiskundige bewijzen routinematig op correctheid controleert – is het onderwerp van een aantal onderzoeksprojecten. Deze projecten hebben hun doel nog niet bereikt, maar in Nijmegen werkt een groep onderzoekers die hopen dat dat binnenkort zal veranderen.

Automath en zijn opvolgers

De pionier van het *formaliseren* (het met de computer wiskundige bewijzen controleren) is de Nederlander prof. N.G. de Bruijn. Hij leidde in de jaren zeventig in Eindhoven het Automath project. 'De Automath' was zijn naam voor een machine die voldoende preciese wiskundige bewijzen kan begrijpen. In die tijd dachten de meeste wiskundigen dat het volkomen onpraktisch zou zijn om interessante bewijzen op een dergelijke manier in volledig detail uit te werken. Maar prof. de Bruijn heeft als eerste laten zien dat dat wel degelijk kan!

Er zijn tegenwoordig een vijftiental serieuze systemen voor verificatie van wiskunde. De opvolger van Automath is het Franse systeem Coq. Dit systeem is ontworpen voor bewijzen over computerprogramma's (dus niet primair voor wiskundige bewijzen), en is soms wat onhandig omdat het werkt met een beperkt soort wiskunde dat *intuïtionisme* heet, maar het kan ook voor echte wiskunde gebruikt worden. In Nijmegen wordt Coq geschikter gemaakt voor echte wiskunde. Om uit te vinden wat voor verbeteringen er daarvoor aan Coq nodig zijn, vertalen ze daar een aantal niet-triviale bewijzen in de Coq taal.

Een systeem dat wél gemaakt is voor wiskunde is het Poolse systeem Mizar.[2] Dit systeem stamt niet van Automath af, maar is onafhankelijk in de jaren zeventig ontwikkeld. Jarenlang is dit systeem eigenlijk nauwlijks in het westen bekend geweest, maar al die tijd hebben de Polen (en een groep Japanners) bewijzen in de Mizar taal zitten vertalen. Inmiddels heeft de Mizar bibliotheek met gecodeerde wiskunde een indrukwekkende omvang. Het bewijst veertigduizend uitspraken en is ongeveer anderhalf miljoen regels lang.

---

[1] QED is de afkorting van 'Quod Erat Demonstrandum' ('dat wat bewezen moest worden'), traditioneel de manier om een bewijs af te sluiten.

[2] Mizar is de naam van een ster uit het sterrenbeeld de Grote Beer.

Oneindig veel priemgetallen

'Er is geen grootste priemgetal,' in een wiskundeboek wordt die stelling zó bewezen:

> **Stelling** (Euclides) *Er zijn oneindig veel priemgetallen: voor iedere getal n bestaat er een priemgetal p dat groter is dan n.*
>
> **Bewijs** Neem een getal n. Beschouw het getal $k = n! + 1$ (waarbij $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$). Kies een priemgetal p dat k deelt. Dan is $p > n$. Want anders zou $p \leq n$, en dan zou p een deler zijn van $n!$, en dus niet van $k = n! + 1$. Maar we hadden p juist als deler van k gekozen. QED

Dit bewijs kan in deze vorm niet door een computer worden begrepen. Het is geschreven in mensentaal en daardoor zijn de stapjes in het bewijs te vaag. Hier is de Mizar versie, die wel begrijpelijk is voor een computer:

```
theorem Euclides: for n ex p st p is prime & p > n
proof
  let n;
  set k = n! + 1;
  n! > 0 by NEWTON:23;
  then n! >= 0 + 1 by NAT_1:38;
  then k >= 1 + 1 by REAL_1:55;
  then consider p such that
A1: p is prime & p divides k by INT_2:48;
A2: p <> 0 & p > 1 by A1,INT_2:def 5;
  take p;
  thus p is prime by A1;
  assume p <= n;
  then p divides n! by A2,NAT_LAT:16;
  then p divides 1 by A1,NAT_1:57;
  hence contradiction by A2,NAT_1:54;
end;
```

Zoals je ziet lijkt dit erg op een computerprogramma. Het formaliseren van een bewijs lijkt dan ook erg op programmeren. In feite combineert formaliseren het leukste van programmeren en van wiskunde. (Sommige bewijssystemen zijn hierdoor bijna een soort computerspel: de te bewijzen stelling is dan het 'level' dat je aan het spelen bent.)

Om de Mizar versie van het bewijs te kunnen begrijpen moet je weten wat de verwijzingen naar de Mizar bibliotheek betekenen:

```
NEWTON:23     for s holds s! > 0
NAT_1:38      i < j + 1 iff i <= j
REAL_1:55     x <= y & z <= t implies x + z <= y + t
INT_2:48      l >= 2 implies ex p st p is prime & p divides l
INT_2:def 5   p is prime iff p > 1 &
               for n st n divides p holds n = 1 or n = p
NAT_LAT:16    j <= l & l <> 0 implies j divides l!
NAT_1:57      i divides j & i divides j + h implies i divides h
NAT_1:54      0 < j & i divides j implies i <= j
```

Dit zijn dus acht van de veertigduizend uitspraken in de Mizar bibliotheek, de *MML* (Mizar Mathematical Library).

Chips zonder bugs

De QED utopie is helaas nog maar een utopie. Het is momenteel nog onpraktisch veel werk om wiskunde voor een *bewijsassistent* (een systeem om bewijzen mee te controleren) uit te werken. Om een indruk te geven: het kost ongeveer een week van heel hard werken om één pagina uit een wiskundeboek te coderen. Bewijsverificatie is dus nog geen gemeengoed in de wiskunde.

Maar er is een ander vakgebied waar het controleren van bewijzen al wel een hogere vlucht heeft genomen: de informatica. Hierbij gaan de bewijzen niet over wiskundige stellingen, maar over de correctheid van chips of van computer-programma's. Het gaat dan bijvoorbeeld om medische of ruimtevaarttoepassingen, waarbij het van het grootste belang is om zeker te weten dat er geen fouten zijn.

In 1997 was er een versie van de Pentium chip die een bug bevatte. Sommige berekeningen die die chip maakte gaven een fout antwoord. Dit heeft Intel, het bedrijf dat de Pentium maakt, heel veel geld gekost. Daarom hebben ze bij Intel tegenwoordig onderzoekers in dienst die als taak hebben met behulp van bewijsassistenten te *bewijzen* dat er niet meer van dit soort bugs in de Pentium processors zitten. Een van deze onderzoekers heeft de fraaie bewijsassistent HOL Light gebouwd. Dit systeem blijkt ook heel geschikt voor het formaliseren van wiskunde.

In credit cards zit tegenwoordig vaak een chip waarop hele kleine computer-programmaatjes draaien. Sommige van die programmaatjes zijn geschreven in een dialect van de programmeertaal Java met de naam *Java Card*. In Nijmegen wordt met behulp van de bewijsassistent PVS technologie ontwikkeld om te bewijzen dat zulke programmaatjes foutloos zijn. Dat is erg belangrijk omdat zulke programmaatjes potentieel op zeer grote aantallen credit cards draaien: als er dan een probleem mee is hebben daar erg veel mensen last van.

Voor de toepassingen in de informatica werkt een groot aantal onderzoekers aan bewijsassistenten. Dit maakt die systemen ook steeds geschikter voor wiskunde. Dus wellicht dat de QED utopie toch binnenkort werkelijkheid wordt!

Meer informatie

Mocht je geïnteresseerd zijn geraakt in het formaliseren van wiskunde:

- Het QED Manifesto: `http://www.cs.kun.nl/~freek/qed/qed.html`
- Een lijst met systemen voor wiskunde in de computer: `http://www.cs.kun.nl/~freek/digimath/`
- De web-site van het Mizar systeem: `http://mizar.org/` (Mizar is gratis te downloaden en draait zowel onder Windows als Linux.)