het servetje van Hendrik Lenstra

Freek Wiedijk

Radboud Universiteit Nijmegen

Kaleidoscoop 2 gastcollege

Universiteit Utrecht

2009 03 12, 14 : 15

een krantenknipsel

‘QED zegt de computer’

artikel door Bennie Mols

NRC Handelsblad

wetenschapsbijlage van zaterdag 31 januari 2009

bewijsverificatie met de computer

- ‘voorstander’: informaticus Freek Wiedijk
- ‘tegenstander’: logicus Ulrich Kohlenbach
- ‘neutraal’: wiskundige Hendrik Lenstra

Lenstra herinnert zich een persoonlijke ervaring met computerbewijzen. “Ik stuitte op een stelling die niet uit mijn eigen vakgebied kwam en ik wilde weten of de stelling waar of onwaar was. Ik zag dat een computer het antwoord moest kunnen berekenen. Ik heb het toen voorgelegd aan iemand uit de computeralgebra. Hij programmeerde het probleem en na een dag rekenen gaf de computer het antwoord dat de stelling ‘waar’ was. Een tijd later kwam ik een expert tegen op het terrein van die stelling. Ik legde hem mijn probleem voor en hij zei: ‘dat reken ik zo voor je uit’. Hij pakte een servetje, mompelde wat, krabbelde wat op het servetje en na een tijdje zei hij: ‘de stelling is onwaar’. Toen dacht ik: fijn, ik weet nu nog niet wie gelijk heeft, maar dat servetje kan ik goed gebruiken. Die expert kent de theorie, hij heeft zijn redenering opgeschreven, dus ik kan het narekenen.”
Lenstra nam het servetje mee, ging bestuderen welke redeneringen de expert had gebruikt en ontdekte op het servetje een onnozele rekenfout: “Dat betekende dat de stelling toch waar was. De computer had dus gelijk. Maar belangrijker was dat ik nu begreep waarom, omdat ik de achterliggende theorie begreep. Toen ik de theorie voorlegde aan degene die mijn probleem had geprogrammeerd, kon hij het probleem veel handiger programmeren. In plaats van na een dag rekenen, kregen we na een halve minuut al het antwoord. Echte stappen voorwaarts maak je pas als je het in je hoofd begrijpt.”

het servetje versus bewijsverificatie

computer checkt gevallen	formules op een servetje
	
bewijsverificatie in bewijsassistent	bewijsverificatie in bewijsassistent
(computerassistentie)	(menselijk bewijs)

eerste misverstand: bewijzen met computersupport $\neq$ bewijsverificatie

bewijsverificatie:

meestal: computer controleert *bestaand* bewijs

computer *kan* ook bovenop een bewijsassistent gevallen checken

Lenstra: 'ik weet nu nóg niet wie gelijk heeft'

als je met een bewijsassistent werkt kun je zeker zijn dat het klopt!

(bovenop een bewijsassistent werken kost wel véél meer werk . . .)

tweede misverstand: computeralgebra $\neq$ bewijsverificatie

computeralgebra: geen bewijzen

bewijsverificatie: wél bewijzen

wereld van verschil!

computeralgebra geeft vaak 'foute' antwoorden
gebrek aan semantiek

computeralgebra: niet-triviale antwoorden

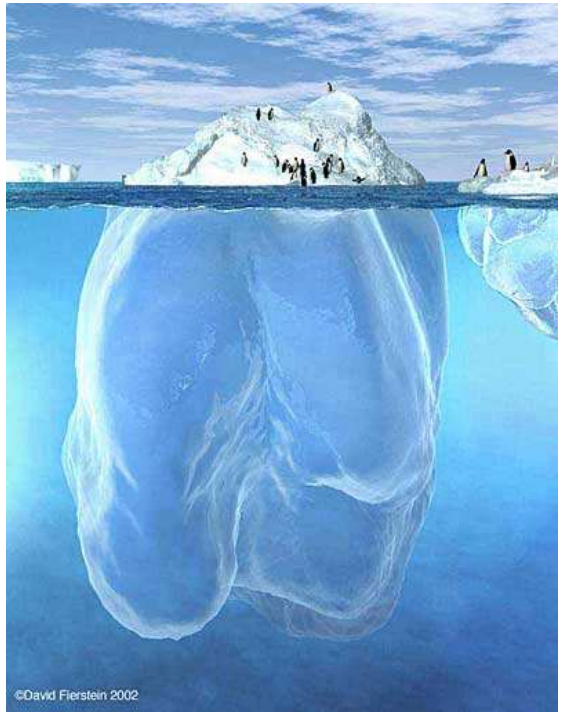
bewijsverificatie (tot nog toe): eenvoudige wiskunde

derde misverstand: traditionele wiskunde $\approx$ bewijsverificatie

bewijsverificatie:

voor mij: tekstverwerker . . .

. . . voor traditionele wiskunde



mens:

stappen die je in een artikel opschrijft

computer:

stapjes ‘onder de bewustzijnsdrempel’

computer moet zich vooral niet tegen het bewijs aan bemoeien!

vier soorten computerwiskunde

- bewijzen met computerberekeningen als onderdeel

  correctheid wordt niet gecheckt

- computeralgebra

  correctheid wordt niet gecheckt

- automatisch stellingenbewijzen

  computer kan dat (nog) helemaal niet: alleen speelgoedwiskunde

- bewijsverificatie

  correctheid wordt gecheckt & realistische wiskunde

twee servetjes uit mijn bureaulade

een ‘servetje’ uit Japan

Michael Beeson
San José State University

schets van het bewijs van de stelling van Liouville

in HOL syntax:

```
!x. algebraic x
    ==> ?n c. c > &0 /\
              !p q. ~(q = 0) ==> &p / &q = x \/
                                 abs(x - &p / &q) > c / &q pow n
```

$f(\alpha) = 0 \qquad f \in \mathbb{Z}[x]$

of degree n

$a^n f(p/q) \in \mathbb{Z}$

$f(p/q) = f(\alpha) + \int_\alpha^{p/q} f'(\xi)\, d\xi$

$= \int_\alpha^{p/q} f'(\xi)\, d\xi$

$1 \le |a^n f(p/q)| \le |p/q - \alpha|\, q^n M$

if $\|f'\| \le M$

$|p/q - \alpha| \ge \frac{1/M}{q^n} \qquad q \ge 0.$

$\therefore \forall \alpha \; \exists c \; \forall p, q \; \left| p/q - \alpha \right| \ge \frac{c}{q^n}$

nog een ‘servetje’

vergeten :

waar vandaan ?

door wie geschreven ?

waar ging het over ?

ondergrens voor $\prod_{p \leq m} p$ zo te zien, maar waarvoor diende dat dan ?

$$\frac{2^m}{(m+1)\, m^{\sqrt{m}}} \leq \prod_{p \leq m} p$$

$$\mathrm{ord}_p(m!) = \sum_{i=1}^{\infty} \left[\frac{m}{p^i}\right]$$

$$2^m = \sum_{i=0}^{m} \binom{m}{i}$$

$$\leq (m+1)\, m^{\sqrt{m}} \cdot \prod_{p \leq m} p$$

$$\mathrm{ord}_p\binom{m}{i} \leq \left[\frac{\log m}{\log p}\right]$$

$$\binom{m}{i} \leq \prod_{\substack{p \leq m \\ p \text{ prime}}} p^{\left[\frac{\log m}{\log p}\right]}$$

$$\leq m^{\sqrt{m}} \cdot \prod_{p \leq m} p$$

wat staat er zoal op servetjes?

- vergelijkingen/ongelijkheden
- simpele uitspraken
- geen lopende tekst!

  wel af en toe een enkel woord

  'of degree n'

  'if'

  'QED.'

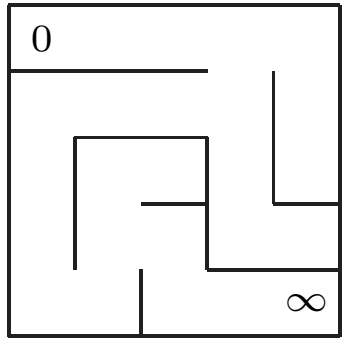
- geïtereerde vergelijkingen/ongelijkheden

$$2^m = \sum_{i=0}^{m} \binom{m}{i} \leq (m+1)m^{\sqrt{m}} \cdot \prod_{p \leq m} p$$

- diagrammen

bewijzen als doolhoven

procedurele versus declaratieve bewijzen



- **procedurele** oplossing

  E E S E N E S S S W W W S E E E

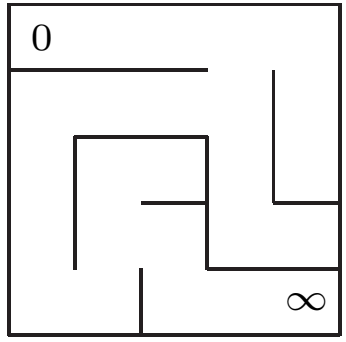
  HOL, Coq, Isabelle

- **declaratieve** oplossing

  (0,0) (1,0) (2,0) (3,0) (3,1) (2,1) (1,1) (0,1) (0,2) (0,3) (0,4) (1,4) (1,3) (2,3) (2,4) (3,4) (4,4)

  Mizar

gestructureerde bewijzen: declaratieve bewijzen procedureel annoteren



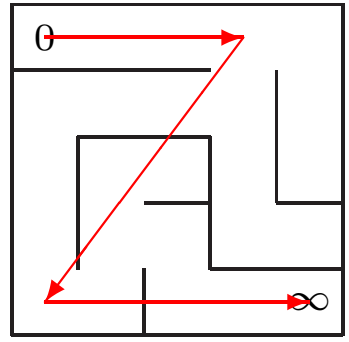
- **gestructureerde** oplossing

  (0,0) E (1,0) E (2,0) E (3,0) S (3,1) W (2,1) W (1,1) W (0,1) S (0,2) S (0,3) S (0,4)
  E (1,4) N (1,3) E (2,3) S (2,4) E (3,4) E (4,4)

  Isabelle

  procedureel en declaratief door elkaar gemengd

‘formal proof sketches’: declaratieve bewijzen indikken



- **‘formal proof sketch’**

  (0,0) (3,0) (0,4) (4,4)

Hardy & Wright, An Introduction to the Theory of Numbers

Theorem 43 (Pythagoras' theorem). $\sqrt{2}$ is irrational.

The traditional proof ascribed to Pythagoras runs as follows. If $\sqrt{2}$ is rational, then the equation

$$a^2 = 2b^2 \tag{4.3.1}$$

is soluble in integers a, b with $(a, b) = 1$. Hence a^2 is even, and therefore a is even. If $a = 2c$, then $4c^2 = 2b^2$, $2c^2 = b^2$, and b is also even, contrary to the hypothesis that $(a, b) = 1$. □

precies hetzelfde in Mizar syntax

theorem Th43: sqrt 2 is irrational :: **Pythagoras' theorem**

proof assume sqrt 2 is rational; consider a, b such that

4_3_1: $a\hat{}2 = 2 * b\hat{}2$

and a, b are_relative_prime; $a\hat{}2$ is even; a is even; consider c such that $a = 2 * c$; $4 * c\hat{}2 = 2 * b\hat{}2$; $2 * c\hat{}2 = b\hat{}2$; b is even; thus contradiction; end;

. . . tot formalisatie

completeren tot volledig Mizar bewijs

```
theorem Th43: sqrt 2 is irrational
proof
  assume sqrt 2 is rational;
  then consider a, b such that
A1: b <> 0 and
A2: sqrt 2 = a/b and
A3: a, b are_relative_prime by Def1;
A4: b^2 <> 0 by A1, SQUARE_1:73;
  2 = (a/b)^2 by A2, SQUARE_1:def 4
    .= a^2/b^2 by SQUARE_1:69;
  then
4_3_1: a^2 = 2 * b^2 by A4, REAL_1:43;
  a^2 is even by 4_3_1, ABIAN:def 1;
  then
A5: a is even by PYTHTRIP:2;
:: lees verder in de volgende kolom
  then consider c such that
A6: a = 2 * c by ABIAN:def 1;
A7: 4 * c^2 = (2 * 2) * c^2
    .= 2^2 * c^2 by SQUARE_1:def 3
    .= 2 * b^2 by A6, 4_3_1, SQUARE_1:68;
  2 * (2 * c^2) = (2 * 2) * c^2 by AXIOMS:16
    .= 2 * b^2 by A7;
  then 2 * c^2 = b^2 by REAL_1:9;
  then b^2 is even by ABIAN:def 1;
  then b is even by PYTHTRIP:2;
  then 2 divides a & 2 divides b by A5, Def2;
  then
A8: 2 divides a gcd b by INT_2:33;
  a gcd b = 1 by A3, INT_2:def 4;
  hence contradiction by A8, INT_2:17;
end;
```

completeerbare formele bewijzen

formal proof sketch = **alleen** 'inference errors'

completeerbaar tot een volledig Mizar bewijs door toe te voegen:

- **stappen**
- **labels/referenties**

formal proof sketches

correctheid: half-beslisbaar

bewijsassistenten voor wiskundigen: zes ideeën

wiskunde versus informatica

- meeste bewijsassistenten ontwikkeld voor **informatica**

  verificatie van software

  verificatie van hardware

  verificatie van communicatieprotocollen

  verificatie van 'security' van systemen

  HOL, Coq, Isabelle, . . .

  niet gemaakt door of voor wiskundigen

- **Mizar** is gemaakt voor **wiskunde**, maar . . .

  niet automatiseerbaar door de gebruiker

  je moet *alle* details zelf uitwerken

idee 1: computerbewijzen door klikken en slepen

'proof by pointing'

voorbeelden:

- CtCoq & PCoq = interfaces voor Coq
- atelier B

formaliseren $\approx$ programmeren

programmeren door klikken en slepen niet praktisch

geen goed idee

idee 2: computerbewijzen in natuurlijke taal

inputtaal voor bewijsassistent: natuurlijke taal in LaTeX syntax

- natuurlijke taal acceptabel parseren erg moeilijk
- natuurlijke taal erg ambigu

formaliseren $\approx$ programmeren

programmeren in natuurlijke taal = COBOL

geen goed idee

idee 3: declaratieve computerbewijzen

- veel meer controle over het bewijsproces
- formal proof sketches mogelijk
- minder systeemspecifiek
  meer 'portable'

experts in bewijsverificatie: declaratieve bewijzen **slecht** idee

Georges Gonthier (= formalisator van de vierkleurenstelling in Coq)

toch goed idee

idee 4: formele formules in een vorm dichterbij die van computeralgebra

- traditionele notatie:

$$\int \frac{1}{x}\,dx = \ln|x| + C$$

- computeralgebra (Mathematica):

```
Integrate[1/x, x] = Log[x]
```

- bewijsassistent (Coq):

```
forall (a:IR) (Ha:Zero[<]a),
  {c : IR | Feq (openl Zero) (([-S-] log_defn_lemma) a Ha)
  (Logarithm{+}[-C-]c)}
```

goed idee: formules zoveel mogelijk in computeralgebra-stijl

idee 5: computer niet-triviale stappen laten nemen

twee concurrerende mentale modellen:

- bewijsassistent als tekstverwerker
- bewijsassistent als research assistent

tweede model duidt op grondig onbegrip van wat een computer kan

kunstmatige intelligentie voor wiskunde bestaat niet binnen afzienbare tijd

geen goed idee

idee 6: automatisering van middelbare schoolwiskunde

bewijsstappen:

$$x = i/n\ ,\quad n = m+1 \quad \vdash \quad n! \cdot x = i \cdot m!$$

$$\frac{k}{n} \geq 0 \quad \vdash \quad \left|\frac{n-k}{n} - 1\right| = \frac{k}{n}$$

$$n \geq 2\ ,\quad x = \frac{1}{n+1} \quad \vdash \quad \frac{x}{1-x} < 1$$

verificatie met huidige technologie: heleboel werk

goed idee: dit soort stappen automatiseren

geïnteresseerd?

afstuderen op bewijsverificatie?

---

bachelor's scriptie

stelling + bewijs

↓

master's scriptie

formalisatie van diezelfde wiskunde?