Automath in historisch perspectief

Herman Geuvers

Radboud Universiteit Nijmegen & Technische Universiteit Eindhoven

Symposium ter ere van de 90ste verjaardag van N.G. de Bruijn

Technische Universiteit Eindhoven

5 september 2008

(Met dank aan Freek Wiedijk)

Het Automath project

AUTOMATH is a language intended for expressing detailed mathematical thoughts. It is not a programming language, although it has several features in common with existing programming languages. It is defined by a grammar, and every text written according to its rules is claimed to correspond to correct mathematics. It can be used to express a large part (see 1.6) of mathematics, and admits many ways for laying the foundations. The rules are such that a computer can be instructed to check whether texts written in the language are correct. These texts are not restricted to proofs of single theorems; they can contain entire mathematical theories, including the rules of inference used in such theories.

wiskundige bewijzen doen op een computer

rond 1970 vijf nieuwe systemen / projecten / ideeën

- **Automath** De Bruijn
- **Nqthm** Boyer Moore (Austin, Texas)
- **LCF** Milner (Stanford; Edinburgh)
- **Mizar** Trybulec (Białystok, Polen)
- **Evidence Algorithm** Glushkov (Kiev, Oekraïne)

de rol van bewijzen in de wiskunde

1. een bewijs legt uit: waarom? Doel: begrijpen
2. een bewijs toont aan: is het waar? Doel: verifiëren, overtuigen

Voor (2) kan computerondersteuning heel nuttig zijn.

the *desirability* of mechanical verification. In a short paper by E.W. Dijkstra on a number of processes that might sometimes block one another, the correctness of the algorithm was explained in a paragraph that ended with the remarkable sentence: "And this, the author believes, completes the proof". Indeed, the argument was a bit intuitive. I took it as a challenge and tried to build a proof that would be acceptable for mathematicians. What I achieved was long and very ugly. It might have been improved by developing efficient lemmas for avoiding the many repetitions in my argument, but I left it as it stood. Instead of improving the proof I got the idea that one should be able to instruct a machine to verify such long and tedious proofs. But of course I have to admit that it will be often more elegant and more efficient to try to streamline such an ugly proof before giving it to a machine.

de rol van geformaliseerde bewijzen in de wiskunde

Men moet in het oog blijven houden dat het raamwerk der geformaliseerde wiskundige redeneringen niet hetzelfde is als „de wiskunde". De machine die de geformaliseerde redeneringen heeft doorgewerkt en beaamd, heeft er nog bitter weinig van begrepen. Hij zal misschien in staat zijn het gelezene te onthouden en later te gebruiken om nieuwe teksten te keuren, maar hij heeft niets begrepen van motiveringen en interpretaties. Men zal hem misschien met enige moeite wat creatief vermogen kunnen geven, maar hij zal daarbij niet geleid worden door ideeën uit de aanschouwingswereld, en evenmin door smaak of gevoel voor waarde. Het geformaliseerde raamwerk is slechts een armzalig deel der wiskunde; niettemin hebben de wiskundigen door de eeuwen heen getracht hun beschouwingen in een dergelijk formalisme, vrij van onzekerheden, te laten kristalliseren. De neergeschreven redenering is de afsluiting van het denkproces, en niet het denkproces zelf.

de verschillende fases in een wiskundig bewijs

1. een bewijs vinden
   Alles mag: experimenteren, gokken, versimpelen, ...
   Wordt niet bewaard voor het nageslacht, maar onmisbaar voor studenten om het vak te leren.

2. een bewijs opschrijven
   Bevat uitleg waarom het gestelde geldt en waarom het bewijs is zoals het is, maar ook de bewijsstappen die te samen het gestelde verifiëren.

3. een bewijs presenteren en communiceren
   Uitleggen aan anderen, presenteren in een voordracht. Een bewijs verbeteren, vereenvoudigen, veranderen, generaliseren.

Computer kunnen vooral een rol spelen in (2) en misschien (3)

waarom zouden we een bewijsassistent geloven?

een bewijsassistent is ook maar een programma . . .

Om de betrouwbaarheid zo groot mogelijk te maken:

- Beschrijving van de regels en de logica van het systeem.
- Een kleine "kernel". Alle bewijzen zijn terug te voeren op een klein aantal basis principes. Hoog niveau stappen zijn gedefinieerd in termen van de kleine.

waarom zouden we een bewijsassistent geloven?

een bewijsassistent is ook maar een programma . . .

- Check de checker. Verifieer de correctheid van de bewijsassistent in het systeem zelf of in een ander systeem.
  optie a. Construeer een model van S in S zelf en laat zien dat alle tactieken van S sound zijn in dit model
  optie b. Bewijs in S: als Con(S), dan $\varphi \Leftrightarrow \vdash \varphi$
- Het De Bruijn criterium
  Sommige bewijsassistenten genereren bewijs objecten die onafhankelijk van het systeem geverifieerd kunnen worden door een eenvoudig programma dat een skeptische gebruiker zelf kan schrijven.

Automath 1967

Doelstellingen

Het project AUTOMATH heeft de volgende onderling samenhangende doelstellingen:

(A) Het ontwerpen van een taal waarin de gehele wiskunde zó nauwkeurig kan worden uitgedrukt dat taalkundige correctheid automatisch wiskundige correctheid tot gevolg heeft.

(B) Het maken van programma's die een computer in staat stellen om de in die taal geschreven boeken op taalkundige dus ook op wiskundige juistheid te controleren.

(C) Het leveren van een opbouw in genoemde taal van een stuk wiskunde, voldoende omvangrijk en voldoende bruikbaar om een grote groep van wiskundigen in staat te stellen hun eigen wiskundige teksten in bedoelde taal om te zetten.

(D) Het uitwerken van de gedachten die een dergelijke taal min of meer automatisch opwerpt ten aanzien van opbouw en presentatie van de bestaande wiskunde.

(E) Het bereiken van de situatie dat ingewikkelde en slecht overzichtelijke stukken wiskunde door samenwerking tussen wiskundige en computer op absoluut betrouwbare wijze kunnen worden geproduceerd.

Automath

Formules als Types, Bewijzen als Objecten

De Bruijn vond opnieuw het Curry-Howard formules-als-types principe uit, de nadruk leggend op het bewijzen-als-obecten aspect.

> An important thing I got from Heyting is the interpretation of a proof of an implication $A \to B$ as a kind of mapping of proofs of A to proofs of B. Later this became one of the motives to treat proof classes as types.

Automath

Formules als Types, Bewijzen als Objecten

Isomorphisme T tussen formules en de typen van zijn bewijzen:

$$\Gamma \vdash_{\text{logica}} \varphi \text{ iff } \overline{\Gamma} \vdash_{\text{type theorie}} M : T(\varphi)$$

M codeert (als λ-term) de afleiding van φ.

$\overline{\Gamma}$ bevat

- declaraties $x : A$ van de vrije variabelen
- aannames, van de vorm $y : T(\psi)$
- bewezen lemmas zijn definities, opgeslagen als $y := p : T(\psi)$ (y is een naam voor het bewijs p van ψ).

Automath

Formules als Types, Bewijzen als Objecten

$$\Gamma \vdash_{\text{logica}} \varphi \text{ iff } \overline{\Gamma} \vdash_{\text{type theorie}} M : T(\varphi)$$

Gevolg:

bewijs checken $=$ type checken

Een typerings algorithme voldoet om aan het De Bruijn criterium te voldoen.
Automath systemen hadden een kleine kernel, dus voor die systemen is typering relatief eenvoudig.

Automath

Logisch Raamwerk (Logical Framework)

Automath is een taal voor het omgaan met de basis wiskundige mechanismen zoals substitutie, variabele binding, creëren en uitvouwen van definities etc.

1.2 Properly speaking, the rules of AUTOMATH involve little more than the art of substitution. A text written in AUTOMATH consists of a sequence of lines.

Automath

Logisch Raamwerk (Logical Framework)

Een gebruiker is vrij om de logische regels toe te voegen die hij/zij wenst

$\Rightarrow$ Automath is een logisch raamwerk, waar de gebruiker zijn/haar favoriete logica kan doen (of een ander favoriet formeel system).

> het nu eenmaal niet. In AUTOMATH zijn echter wel dingen als logische connectieven, contradictie en negatie buiten de taaldefinitie gehouden. Het is verstandiger talen te maken die elke gebruiker in staat stellen de door hem zelf gewenste opzet met axioma's in te voeren waaraan hij slechts per boek gebonden is. AUTOMATH is een groot restaurant waar men in iedere stijl kan eten. Wie kosher wil eten doet dat, maar verplicht niemand anders daartoe. Slechts intoleranten kunnen zich ergeren over het feit dat er ook plaats is voor andersdenkenden.

Automath

Logisch Raamwerk (Logical Framework)

Een gebruiker bepaalt zelf de logische spelregels die hij/zij wenst te gebruiken.

De Bruijn's versie van het formules-als-types principe:

$$\Gamma \vdash_L \varphi \text{ iff } \Gamma_L, \overline{\Gamma} \vdash_{\text{type theorie}} M : T(\varphi)$$

waar L een logica is, Γ_L is de context waarin de constructies van de logica L gedeclareerd worden.

Keuze: Welke logische constructie stop je in de type theorie en welke constructies declareer je axiomatisch in de context?

Hierop gebaseerd: Edinburgh LF (1987) en tegenwoordig Twelf (Carnegie Mellon Univ.)
Moderne toepassing: Proof Carrying Code

Na Automath

Constructieve Type Theorie

Formules-als-types isomorphisme relateert bewijzen in constructieve logica en getypeerde λ-termen
Martin-Löf heeft dit uitgebreid naar constructieve type theorie:

- grondslag voor de wiskunde
- inductieve typen en functies gedefinieerd met welgefundeerde recursie zijn de basis principes

Bewijs Assistenten gebaseerd op CTT: Nuprl (Constable, Cornell), Agda, Alf (Gothenburg)
Coq (INRIA, France) is gebaseerd op een mix van ideeën uit Automath, CTT, LCF

Na Automath

Bewijzen als Programma's

In CTT: Extraheer uit een bewijs p van

$$\forall x : A \, \exists y : B \, R(x, y)$$

een functioneel programma

$$f : A \to B$$

en een bewijs q van $\forall x : A.R(x, f(x))$.

De specificatie $\forall x : A.\exists y : B.R(x, y)$, indien gerealiseerd (bewezen) produceert een programma dat eraan voldoet.

Belangrijk ingredient van Coq

Na Automath

Check the Checker

Een type theorie zoals CTT of Coq bevat een functionele programmeertaal. Dus ...

programmeer en verifieer het typeringsalgoritme in het systeem zelf.

Coq in Coq project: Men bewijst in Coq:

$$\Gamma \vdash M : A \Leftrightarrow \mathsf{TC}(\Gamma, M) = A,$$

waar TC het typerings algorithme is:

$\mathsf{TC}(\Gamma, M) = A$ als M heeft type A in Γ

$\mathsf{TC}(\Gamma, M) =$ 'fail' anders.

Dit statement impliceert de consistentie van Coq als een logica, dus kan niet binnen Coq bewezen worden zonder extra aannames.

stand van zaken nu

bewijzen formaliseren net zo veel werk als LaTeX?

As a kind of dream I played (in 1968) with the idea of a future where every mathematician would have a machine on his desk, all for himself, on which he would write mathematics and which would verify his work. But, by lack of experience in such matters, I expected that such machines would be available within 5 years from then. But now, 23 years later, we are not that far yet. Anyway I expected in 1968 that the memory capacity of main frame computers would grow rapidly in the next few years, but that was a deception too. Implementing Automath on the quite advanced computers available to us in the years 1970–1975 was to a large extent a struggle for living with the limitations of fast accessible memory.

stand van zaken nu

100 leuke stellingen, 80 geformaliseerd

1.	The Irrationality of the Square Root of 2	≥ 17
2.	Fundamental Theorem of Algebra	4
3.	The Denumerability of the Rational Numbers	6
4.	Pythagorean Theorem	6
5.	Prime Number Theorem	2
6.	Gödel's Incompleteness Theorem	3
7.	Law of Quadratic Reciprocity	4
8.	The Impossibility of Trisecting the Angle and Doubling the Cube	1
9.	The Area of a Circle	1
10.	Euler's Generalization of Fermat's Little Theorem	4
11.	The Infinitude of Primes	6
12.	The Independence of the Parallel Postulate	0
13.	Polyhedron Formula	1

...

google: 100 theorems

de beste bewijsassistenten

vijf systemen serieus gebruikt voor wiskunde:

HOL	HOL Light	69
	ProofPower	42
	Isabelle	40
	Coq	39
	Mizar	45

HOL Light

in lange traditie:
LCF $\rightarrow$ HOL $\rightarrow$ HOL Light
Stanford, US $\rightarrow$ Cambridge, UK $\rightarrow$ Portland, US

John Harrison

bewijst floating point hardware correct bij Intel
formaliseert wiskunde in zijn vrije tijd

bijzonder elegant systeem
makkelijk uit te breiden

niet gebruikersvriendelijk

Isabelle

soort ‘opvolger’ van HOL

samenwerking tussen twee universiteiten:

Cambridge, UK

vooral: computerveiligheid

München, Duitsland

vooral: wiskunde

gebalanceerd systeem

mooie bewijstaal

krachtige automatisering

Coq

INRIA en Microsoft

Institut National de Recherche en Informatique et en Automatique

systeem met de indrukwekkendste formalisatie tot nu toe

systeem dat wij in Nijmegen gebruiken

geïntegreerde programmeertaal

$\approx$ Haskell

wiskundig expressief

intuïtionistisch

Mizar

Andrzej Trybulec

Białystok, Polen

ook: Nagano, Japan

wiskundigste van de bewijsassistenten

grootste bibliotheek met geformaliseerde wiskunde

2,1 miljoen regels code

gebruikersvriendelijk

soms moeilijk te begrijpen

geformaliseerde stellingen

de hoofdstelling van de algebra

Carl Friedrich Gauss, 1799

Mizar: Robert Milewski, 2000

HOL Light: John Harrison, 2000

Hellmut Kneser, 1940

Coq: Herman Geuvers e.a., 2000

$$x^2 = -1$$

$$x = i = \sqrt{-1}$$

we kunnen nu alle polynomiale vergelijkingen oplossen

$$x^2 = i \ ?$$

de priemgetalstelling

Paul Erdös en Atle Selberg, 1949

Isabelle: Jeremy Avigad, 2004

Jacques Hadamard, Charles Jean de la Vallée-Poussin, 1896

HOL Light: John Harrison, 2008

aantal priemgetallen $\leq$ gegeven getal:

$$\begin{aligned} \pi(1000000000000) &= 37607912018 \\ \frac{1000000000000}{\ln(1000000000000)} &= 36191206825{,}27\ldots \\ \int_2^{1000000000000} \frac{dx}{\ln(x)} &= 37607950279{,}75\ldots \end{aligned}$$

in de limiet is de verhouding 1: $\lim_{n\to\infty} \frac{\ln(n)}{n\,\pi(n)} = 1$

de stelling over Jordan-krommen

Oswald Veblen, 1905

HOL Light: Tom Hales, 2005

Mizar: Artur Korniłowicz e.a., 2005

Jordan-krommen geen Jordan-krommen

Een Jordan-kromme verdeelt het platte vlak in twee stukken

de eerste onvolledigheidsstelling

Kurt Gödel, 1931

Boyer-Moore prover: Natarajan Shankar, 1986

Coq: Russell O'Connor, 2003

HOL Light: John Harrison, 2005

de Gödel-zin:

deze zin is niet bewijsbaar'

als deze zin waar is, zijn er onbewijsbare ware zinnen (onvolledigheid)

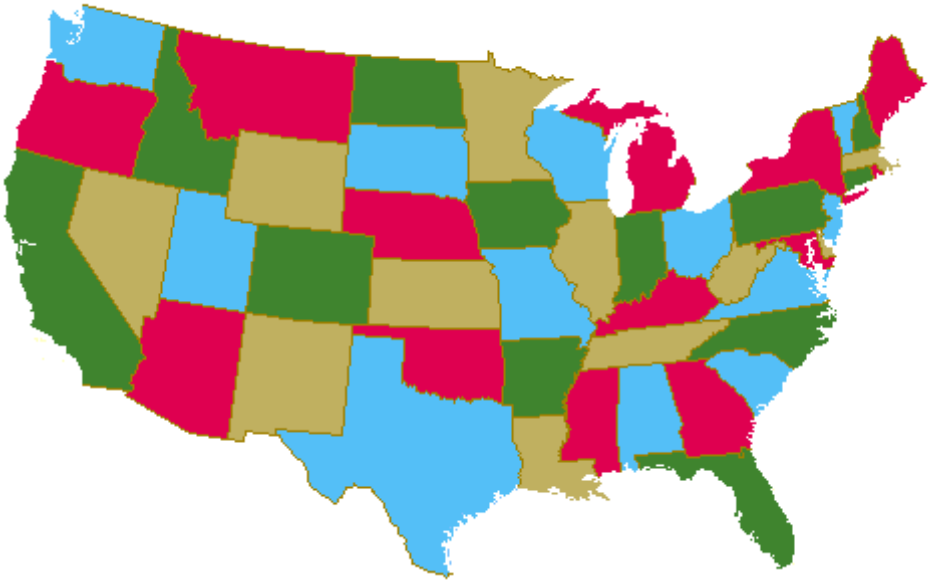
als deze zin onwaar is, zijn er bewijsbare onware zinnen (inconsistentie)

de vierkleurenstelling

Kenneth Appel en Wolfgang Haken, 1976

Neil Robertson e.a., 1996

Coq: Georges Gonthier, 2004



kan *iedere* kaart met maar vier verschillende kleuren worden gekleurd?

filosofische consequenties van Automath

platonisme

Before I started Automath I was slightly anti-platonistic. That is to say that I always sympathized with Kronecker's statement that only the natural numbers *really* exist. But building Automath I rapidly concluded that I *had* to be anti-platonistic. Before that, I had allowed myself to be a bit confused by statements like "3 is not a number, it is the *name* of a number". But if you have to talk to a machine that knows nothing about the real mathematical objects, then you know that you can handle names only. The "real" mathematical objects are irrelevant in the discussion with the machine. In the language used for communicating with

De Bruijn citeert graag Wittgenstein:

> Don't ask for the meaning, ask for its use

filosofische consequenties van Automath

constructivisme vs. formalisme

Remarkably, at that occasion Scott wrote: "de Bruijn had been, of course, personally influenced by Brouwer and wanted to present a suitably constructive notion of proof". I think this is confusing. Not just because I was never influenced by Brouwer's talking or writing, but because "constructive notion of proof" is different from "notion of constructive proof". The latter can be connected to Brouwer, but the former is much closer to Hilbert and his finitistic game with symbols and rigid rules. Brouwer's constructivity is (at least in Heyting's formalized form) is a matter of the *content* of axioms, and not the way these axioms are manipulated.

een zeer korte geschiedenis van de wiskunde

de drie revoluties

Euclid

bewijzen

Cauchy

rigoreus

de Bruijn

formeel