Machine-Checked Mathematics

Assia Mahboubi, Inria – VU Amsterdam
Ce que se propose pour but essentiel l’axiomatique, c’est précisément ce que le formalisme logique, à lui seul, est incapable de fournir, l’intelligibilité profonde des mathématiques.

The crux of the axiomatic method is precisely what logical formalism by itself cannot provide, which is the profound intelligibility of mathematics.

Nicolas Bourbaki, L’architecture des mathématiques, 1948
"The notion that these conjectures might have been reached by pure thought – with no picture – is simply inconceivable. . . I had my programmer draw a very big sample [Brownian] motion and proceeded to play with it” B. Mandelbrot, 1982
Observation and experimentation

Birch and Swinnerton-Dyer Conjecture

Mathematicians have always been fascinated by the problem of describing all solutions in whole numbers \(x, y, z\) to algebraic equations like

\[x^2 + y^2 = z^2\]

Euclid gave the complete solution for that equation, but for more complicated equations this becomes extremely difficult. Indeed, in 1970 Yu. V. Matiyasevich showed that Hilbert's tenth problem is unsolvable, i.e., there is no general method for determining when such equations have a solution in whole numbers. But in special cases one can hope to say something. When the solutions are the points of an abelian variety, the Birch and Swinnerton-Dyer conjecture asserts that the size of the group of rational points is related to the behavior of an associated zeta function \(\zeta(s)\) near the point \(s=1\). In particular this amazing conjecture asserts that if \(\zeta(1)\) is equal to 0, then there are an infinite number of rational points (solutions), and conversely, if \(\zeta(1)\) is not equal to 0, then there is only a finite number of such points.

This problem is: Unsolved
Proofs

**Computational Details.** Some of the computational details in Sections 12, 13 were carried out in the computer algebra system Magma [7]. The reader can find the Magma scripts for verifying these computations (together with extensive commentary) at:

http://arxiv.org/abs/1310.7088

Our programs make use of several Magma packages. We would like to acknowledge these packages and their authors and main contributors:

- **Hyperelliptic Curves:** Nils Bruin, Brendan Creutz, Steve Donnelly, Michael Harrison, David Kohel, Michael Stoll, Paul van Wamelen;
- **Small Modular Curves Database:** Michael Harrison;
- **Algebraic Function Fields:** Florian Heß, Claus Fieker, Nicole Sutherland;
- **Modular Forms:** William Stein, Kevin Buzzard, Steve Donnelly;
- **Modular Abelian Varieties:** William Stein, Jordi Quer;
- **Matrix Groups over Finite Fields:** Eamonn O’Brien.

[Elliptic curves over real quadratic fields are modular, N. Freitas, B.V. Le Hung, S. Siksek, Invent. Math., 2015]
Larger Scale Computations

Shalosh B. Ekhad  
> Home > Persons

[-] 2010 – today

2013

2011

[-] 2000 – 2009

2009

Robert Brignall, Shalosh B. Ekhad, Rebecca Smith, Vincent Vatter:

2000

Marcin Mazur, Kirit Hanes, Jean Anglesio, M. Benedicty, Shalosh B. Ekhad, N. Lakshmanan, Albert Nijenhuis, John H. Smith:

Wu Wei Chao, Michael Reid, F. Bellot Rosado, Robin J. Chapman, Daniele Donini, Shalosh B. Ekhad, N. Lakshmanan, O. F. Losers, Albert Nijenhuis, Peter Nüesch, C. G. Petalas:

Jean Anglesio, Shalosh B. Ekhad:
In the Age of the Turing Machine

Mathematics in the Age of the Turing Machine

Thomas Hales

(Submitted on 12 Feb 2013)

The article gives a survey of mathematical proofs that rely on computer calculations and formal proofs.

Comments: 45 pages. This article will appear in “Turing's Legacy,” ASL Lecture Notes in Logic, editor Rodney G. Downey
Subjects: History and Overview (math.HO)
Cite as: arXiv:1302.2898 [math.HO]
(or arXiv:1302.2898v1 [math.HO] for this version)

Submission history
From: Thomas Hales [view email]
[v1] Tue, 12 Feb 2013 19:58:52 GMT (3226kb,D)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Link back to: arXiv, form interface, contact.
Interactive Theorem Provers

Formal Logic

Proof Assistant

Proof Checker

Libraries
Interactive Theorem Provers

Formal Logic

Proof Assistant

Proof Checker

Libraries

Type Theory

Coq

MathComp

User
Languages

- Libraries
Languages

- Libraries
- Logical foundations
Languages

- Libraries
- Logical foundations
- Meta-Language(s)
Two Sides of Automation

Filling gaps:
Two Sides of Automation

Filling gaps:
- in proofs
Two Sides of Automation

Filling gaps:
• in proofs
• in statements.
1908


- B. Russell: Mathematical Logic as Based on the Theory of Types, Amer. J. Math. 30 (3): 222–262
Pioneers: AUTOMATH

- Written by Nicolaas Govert de Bruijn (1918 - 2012)
  started circa 1967, still usable
Pioneers: AUTOMATH

• Written by Nicolaas Govert de Bruijn (1918 - 2012)
  started circa 1967, still usable

• Introduced several key ideas still in use
  de Bruijn indices, dependent types, etc.
Pioneers: AUTOMATH

- Written by Nicolaas Govert de Bruijn (1918 - 2012)
  started circa 1967, still usable
- Introduced several key ideas still in use
  de Bruijn indices, dependent types, etc.
- Meant to represent mathematics on computer.
  Edmund Landau’s Foundations of Analysis
Pioneers: Mizar

- Initiated by Andrzej Trybulec (1941 - 2013)
  started circa 1973
Pioneers: Mizar

- Initiated by Andrzej Trybulec (1941 - 2013)
  started circa 1973
- Based on a flavour of ZFC set theory and first order logic
  Tarski-Grothendick set theory
Pioneers: Mizar

- Initiated by Andrzej Trybulec (1941 - 2013) started circa 1973
- Based on a flavour of ZFC set theory and first order logic Tarski-Grothendick set theory
- Meant to represent mathematics on computer.
  Formulated Mathematics
Pioneers: Boyer-Moore Provers

Thm, Nqthm, ACL, ACL2

- Initiated by Robert Stephen Boyer and J Strother Moore
  started circa 1972
Pioneers: Boyer-Moore Provers

Thm, Nqthm, ACL, ACL2

- Initiated by Robert Stephen Boyer and J Strother Moore
  started circa 1972
- First order, quantifier free logic of computable functions
  Lisp
Pioneers: Boyer-Moore Provers

Thm, Nqthm, ACL, ACL2

- Initiated by Robert Stephen Boyer and J Strother Moore
  started circa 1972
- First order, quantifier free logic of computable functions
  Lisp
- Designed and used for (industrial) program verification
  e.g. Oracle’s SPARC processors
Pioneers: Boyer-Moore Provers

Thm, Nqthm, ACL, ACL2

- Initiated by Robert Stephen Boyer and J Strother Moore
- First order, quantifier free logic of computable functions
- Designed and used for (industrial) program verification
- Strongly automated.

E.g. Oracle’s SPARC processors
Pioneers: LCF

HOL, HOL88, HOL4, Isabelle/HOL, HOL-Light, HOL-Zero, etc.

- Initiated by Robin Milner
  circa 1972, after Dana Scott’s 1969 notes
Pioneers: LCF

HOL, HOL88, HOL4, Isabelle/HOL, HOL-Light, HOL-Zero, etc.

- Initiated by Robin Milner
circa 1972, after Dana Scott’s 1969 notes
- Introduced several key ideas still in use
  Higher-order logic, correct-by-type, tactics, etc.
Pioneers: LCF

HOL, HOL88, HOL4, Isabelle/HOL, HOL-Light, HOL-Zero, etc.

- Initiated by Robin Milner
circa 1972, after Dana Scott’s 1969 notes
- Introduced several key ideas still in use
  Higher-order logic, correct-by-type, tactics, etc.
- Initiated the ML family of typed programming languages
  OCaml, Standard ML, etc.
Pioneers: LCF

HOL, HOL88, HOL4, Isabelle/HOL, HOL-Light, HOL-Zero, etc.

- Initiated by Robin Milner
circa 1972, after Dana Scott’s 1969 notes
- Introduced several key ideas still in use
  Higher-order logic, correct-by-type, tactics, etc.
- Initiated the ML family of typed programming languages
  OCaml, Standard ML, etc.
- Originally designed for program verification.
Dependent Type Theory

Coq, NuPRL, Agda, Matita, Lean, etc.

- First prototype by Thierry Coquand and Gérard Huet circa 1984
Dependent Type Theory

Coq, NuPRL, Agda, Matita, Lean, etc.

- First prototype by Thierry Coquand and Gérard Huet circa 1984
- Constructive higher-order logic, dependent types inspired by Automath, Martin Löf, Girard
- Pure functional programs in the logic
Dependent Type Theory

Coq, NuPRL, Agda, Matita, Lean, etc.

- First prototype by Thierry Coquand and Gérard Huet circa 1984
- Constructive higher-order logic, dependent types inspired by Automath, Martin Löf, Girard
- Pure functional programs in the logic
- Designed for the formalization of mathematics
And More

PVS, Metamath, Twelf, MINLOG, Abella, etc.
1

Machine Checked Computational Mathematics
Four Color Theorem

• Conjecture: F. Guthrie (1852)
• Proof: K. Appel - W. Haken (1976)

[Formal Proof—The Four-Color Theorem, G. Gonthier, Notices of the AMS, 2008]
Verified ODE Solver

- Introduced by E. Lorentz (1963)
- Verified ODE in Isabelle/HOL F. Immler (2018).

[A Verified ODE Solver and Smale’s 14th Problem, F. Immler, PhD, 2018]
Flyspeck

- Conjecture: Johannes Kepler (1611)
- Proof strategy, using computer: L. Fejes Tóth (1953)

[A formal proof of the Kepler conjecture, T. Hales et al., Forum of Mathematics, Pi, 2015]
Proof strategy

- Prove exhaustivity of an archive of tame finite graphs
- Solve linear problems
- Solve non-linear problems.
Flyspeck: Linear Problems

- GLPK used as an oracle to find a solution
- (Possibly modified) solutions checked inside the logic
- 43,078 programs, 15 hours on a 2.4GHz computer.

Using HOL-Light.
Flyspeck: Nonlinear Problems

- Non-strict inequalities on rectangular domains
- Algebraic expressions of transcendental functions
- Taylor interval approximations
- 23,000 inequalities, 5000 process-hours in Microsoft’s Azure.

Using HOL-Light.
Flyspeck: Computing in HOL-Light

- Numbers are an equational theory
- Computation is normalization in the logic
- It operates on “logical” floats and integer arithmetics.
Flyspeck: Combination

Issues:

- Time
- Parallelism without proof objects
- Two proof assistants: HOL-Light and Isabelle/HOL.
2

Automated Formal Proofs
Autarkic Proofs

• Automated theorem proving classics
  model-based elimination, tableaux, etc.
Autarkic Proofs

• Automated theorem proving classics
  model-based elimination, tableaux, etc.

• Normalization
  simplifiers, ring tactics, evaluation, etc.
Small Scale Computations

(demo)
Reconstructed Proofs

Sledgehammer (Isabelle/HOL):

- Translation in FOL
- Relevance Filtering of the Context
- Run ATPs
- Reconstruct a proof using an autarkic procedure.

[Three Years of Experience with Sledgehammer, a Practical Link Between Automatic and Interactive Theorem Provers
- L. Paulson, J. Blanchette IWIL’10]
Automatic Asymptotics

\[
F(n + 2) = F(n + 1) + F(n), \quad F(1) = 1 \quad F(0) = 1
\]

\[
F(n) = \mathcal{O}(1, 619^n)
\]

[Verified Solving and Asymptotics of Linear Recurrences, M. Eberl. CPP’2019]
Certificate Based Proofs

Prove that:

\[ c := 9037762929200312168400214710176085810924733654900109067769 \]

is not prime.
Certificate Based Proofs

Prove that:

\[ c := 9037762929200312168400214710176085810924733654900109067769 \]

is not prime.

Proof:

\[ c = 588120598053661 \cdot 260938498861057 \cdot 760926063870977 \cdot 773951836515617 \]
Certificate Based Proofs

- Primality proofs
- Linear programming
- Positivity via decomposition into sums of square
- etc.
Certificate Based Proofs

- The boolean Pythagorean Triples problem.
  [Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer. M. Heule et al., SAT-16]

- Schur Number 5 is 160.
  [Schur Number 5 - M. Heule, AAAI-18]
Certificate Based Proofs

- The boolean Pythagorean Triples problem.
  
  [Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer. M. Heule et al., SAT-16]

- Schur Number 5 is 160.
  
  [Schur Number 5 - M. Heule, AAAI-18]
Apéry’s Constant

Let \( \zeta(3) \) be the real number \( \sum_{k=1}^{+\infty} \frac{1}{k^3} \).

**Theorem** (Apéry, 1978): The constant \( \zeta(3) \) is irrational.
Apéry’s Constant

Let \( \zeta(3) \) be the real number \( \sum_{k=1}^{+\infty} \frac{1}{k^3} \).

Theorem (Apéry, 1978): The constant \( \zeta(3) \) is irrational.

See:
“A proof that Euler missed: Apéry’s proof of the irrationality of \( \zeta(3) \). An informal report.” by A. van der Poorten, 1979.

for the tale of this proof, and of its verification.
Apéry’s mysterious recurrence

The crux in Apéry’s proof is to verify that both \((a_n)_{n \in \mathbb{R}}\) and \((b_n)_{n \in \mathbb{R}}\) verify the recurrence:

\[
(n + 2)^3 y_{n+2} - (17n^2 + 51n + 39)(2n + 3)y_{n+1} + (n + 1)^3 y_n = 0,
\]
Apéry’s mysterious recurrence

The crux in Apéry’s proof is to verify that both \((a_n)_{n \in \mathbb{R}}\) and \((b_n)_{n \in \mathbb{R}}\) verify the recurrence:

\[(n + 2)^3 y_{n+2} - (17n^2 + 51n + 39)(2n + 3)y_{n+1} + (n + 1)^3 y_n = 0,
\]

with:

\[a_n = \sum_{k=0}^{n} \left( \begin{array}{c} n \\ k \end{array} \right)^2 \left( \begin{array}{c} n+k \\ k \end{array} \right)^2, \quad b_n = a_n \sum_{k=1}^{n} \frac{1}{k^3} + \sum_{k=1}^{n} \sum_{m=1}^{k} \frac{(-1)^{m+1} \left( \begin{array}{c} n \\ k \end{array} \right)^2 \left( \begin{array}{c} n+k \\ k \end{array} \right)^2}{2m^3 \left( \begin{array}{c} n \\ m \end{array} \right) \left( \begin{array}{c} n+m \\ m \end{array} \right)}.\]
Computer (Formal) Proof

- Using the Maple/Algolib library (Bruno Salvy)
- Using Coq, and the Maple/Algolib library.

[A Computer-Algebra-Based Formal Proof of the Irrationality of $\zeta(3)$. F. Chyzak, AM, T. Sibut-Pinote, E. Tassi, ITP14]

See also [Formal Proofs of Hypergeometric Sums, J. Harrison. JAR 2015]
3

Elaborating Statements
What Are Perfectoid Spaces?

Peter Scholze has been awarded the Fields medal, at the 2018 International Congress of Mathematicians, for “transforming arithmetic algebraic geometry over p-adic fields through his introduction of perfectoid spaces, with application to Galois representations and for the development of new cohomology theories.”
Odd Order Theorem

Theorem (Feit - Thompson, 1963):

Every finite group of odd order is solvable.

Odd Order Theorem

Theorem (Feit - Thompson, 1963):

Every finite group of odd order is solvable.

- A two-volume revised proof:
  - H. Bender; G. Glauberman, Local analysis for the odd order theorem (1994)
Odd Order Theorem

Theorem (Feit - Thompson, 1963) :

Every finite group of odd order is solvable.

- A two-volume revised proof:
  - H. Bender; G. Glauberman,
    Local analysis for the odd order theorem (1994)
  - Th. Peterfalvi,
    Character theory for the odd order theorem, (1984)

[A machine-checked proof of the odd order theorem. G. Gonthier et al. ITP 2013.]
Bookshelf
“Let $A$ be a square matrix in $\mathcal{M}_n(\mathbb{R})$.

$$\text{Det} \ (A) = \sum_{\sigma \in S_n} \epsilon_{\sigma} \prod_{i} a_{\sigma(i),i}.$$”
“Let $A$ be a square matrix in $M_n(\mathbb{Z}_n)$.  

$$\text{Det } (A) = \sum_{\sigma \in S_n} \epsilon_\sigma \prod_{i} a_{\sigma(i),i}.$$
“Let $A$ be a square matrix in $M_n(\mathcal{I})$. 

$$\det(A) = \sum_{\sigma \in S_n} \epsilon_\sigma \prod_i a_{\sigma(i),i}.$$"
Mathematics in Formal Libraries
Mathematics in Formal Libraries

• In \LaTeX:

\textsf{Det (A)} = \sum_{\sigma \in S_n} \epsilon_{\sigma} \prod_i a_{\sigma (i),i}

• In Coq:

Definition det (R : ringType) n (A : 'M[R]_n) : R :=
\sum_{sigma : 'S_n} (-1)^+ sigma \times \prod_i A i (sigma i).
Mathematics in Formal Libraries

• In \LaTeX:
\textsf{Det (A)} = \sum_{\sigma \in S_n} \epsilon_{\sigma} \prod_i a_{\sigma (i),i}

• In Coq:

Definition det (R : ringType) n (A : 'M[R]_n) : R :=
\sum_(sigma : 'S_n) (-1)^+ sigma * \prod_i A i (sigma i).
This Course

• Basics of computer-assisted proof management (Mon)
• Short introduction to dependent type theory (Wed)
• Automation, in proofs and in statement synthesis (Fri)