Proof assistants and mathematics education

Jeremy Avigad

Department of Philosophy
Department of Mathematical Sciences
Hoskinson Center for Formal Mathematics

Carnegie Mellon University

May 3, 2024

The Lean programming language and proof assistant

I'll start with a demonstration of Lean.

A brief overview

Proof assistants are now used for

- hardware, software, and systems verification
- mathematics and the mathematical sciences

Some proof assistants for mathematics:

- Mizar (1973, set theory)
- Isabelle (1986, simple type theory)
- Rocq (1989, dependent type theory)
- HOL Light (1994, simply type theory)
- Lean (2013, dependent type theory)

Applications for mathematics

Some applications:

- verifying proofs, finding errors
- building mathematical libraries
- collaborating
- verifying mathematical computation
- using symbolic AI and machine learning
- teaching

See also the special issue of the *Bulletin of the American Mathematical Society*, Will Machines Change Mathematics?

Verifying mathematics

Some landmarks:

- The verification of the Feit-Thompson theorem (Gonthier et al., 2012)
- The Flyspeck project (Hales et al., 2014)
- The Liquid Tensor experiment (Commelin et al., 2021)

Some recent landmarks:

- The verification of the polynomial Freiman-Rusza conjecture (Tao et al., 2023)
- The verification of the consistency of Quine's NF (Holmes and Wilshaw, 2024)

Verifying mathematics

On November 9, 2023, W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao announced a proof of the PFR conjecture.

On November 18, Tao asked for help verifying it.

About 30 members of the Lean community joined him. The formalization was done by December 5.

See:

- the project page
- the article in Quanta

Verifying mathematics

Quine introduced his *New Foundations* system (NF) in 1937.

The original version was inconsistent. In 1950, Hao Wang suggested a fix, which Quine published in 1951.

The consistency of the latter system has been a longstanding open question.

Randall Holmes claimed a proof in 2015. Jamie Gabbay also claimed one. The technical details were overwhelming.

A few weeks ago, Sky Wilshaw, a Part III (masters) student at Cambridge, formally verified Holmes' proof.

Verifying mathematics

“I have been convinced for a long time that I saw the path to proving the result. The problem is that the argument is insanely detailed and any paper text has something wrong with it. This can be attributed partly to deficiencies of mine as an expositor, but since I was the only one who saw it, I had to do the writing. It is also intrinsic: there is a lot of elaborate and not necessarily intuitive bookkeeping in the argument, and its very easy to write things down wrong. I am sure now that (1) the Lean proof is correct, I read the statements of the conclusions, and it proves the right thing and (2) the paper as it stands is converging to the right thing, because Sky is advising me where what I do differs from what is done in the formal proof, and she appears to follow what I have written now fairly happily.” (Holmes)

Building mathematical libraries

Lean's Mathlib currently has more than 1.5 million lines of code.

We can look at:

- the repository
- library statistics
- API documentation
- instances of the ring class
- classes that the real numbers are instances of

Collaboration

Digitizing mathematics is a collaborative effort.

The Lean community is a self-governing grassroots organization.
See:

- the web pages
- the community teams

The Lean Zulip channel currently has

- 9,000 subscribers
- 700 active in any 15-day period
- about 1,000 messages every day

Patrick Massot's *Blueprint* software supports collaborative projects like the PFR formalization.

Verifying mathematical computation

Proof assistants can be used to verify the correctness of mathematical results obtained by computation.

There are efforts to use Lean:

- to verify reductions for optimization problems
- to use Lean as a scientific programming language

Broader applications of formal verification:

- hardware and software
- cyber-physical systems
- network protocols
- privacy and security
- blockchain and decentralized finance

Using symbolic AI and machine learning

Symbolic AI and machine learning have complementary strengths. It's an important challenge to synthesize the two.

Mathematics is the best place to start. AI can be used for mathematical discovery as well as verification.

See:

- This survey of automated reasoning for mathematics, and the references to machine learning there.
- LLMLean
- Lean Copilot

If time allows, I'll demonstrate Github Copilot with Lean.

Teaching

Proof assistants offer a lot of potential for teaching mathematics.

Interaction provides:

- immediate feedback and positive encouragement
- error messages and correction
- information about the current state of a proof
- means to search, experiment, and explore
- increased student engagement

There is a lot of helpful information on the teaching page of the Lean community web site.

Teaching

I will talk about:

- The Natural Number Game
- The Set Theory Game
- The Mechanics of Proof
- Verbose Lean
- How to Prove it With Lean
- Mathematics in Lean
- Logic and Mechanized Reasoning

I will also show you some Lean widgets.

Teaching

Lean was designed as a tool for research and industrial applications.

Teaching undergraduate students in mathematics and computer science how to formalize is reasonably straightforward.

Using a proof assistant to teach university and high school students how to write mathematics is more challenging.

Using interactivity to teach elementary, middle, and high school students how to think mathematically is a greater challenge still.

Caveats

The fine print:

- The field is young and evolving rapidly.
- The core technology is not easy to use.
- Most members of the Lean community contribute voluntarily, in their spare time.
- Documentation and expository materials are scarce.
- There isn't a lot of central organization or institutional support.
- Educational research is needed to assess claims as to the pedagogical benefits.

Caveat emptor!

Caveats

Digital technology creates new access points.

Children can find material online and learn how to do mathematics the same way that children find material online and learn how to code.

But the children who benefit the most are those with parents, teachers, and financial resources to support their exploration.

If we are not careful, the digital divide may exacerbate the mathematical divide.

We need to think long and hard about how to ensure that everyone benefits.

Conclusions

Digital mathematical technology is transformative.

It's like the digitization of language (in email, on the web, in databases) but better: mathematics is inherently formal.

The interest and enthusiasm among young people is encouraging.

Let's give them opportunities to make the most of the new technologies.