Formal Assistants and Mathematics Education

Jeremy Avigad

Department of Philosophy
Department of Mathematical Sciences

Hoskinson Center for Formal Mathematics

Carnegie Mellon University

March 26, 2022

Formalization of mathematics

Since the early twentieth century, it has been known that most ordinary mathematics is formalizable, in principle.

With the help of computational proof assistants, mathematics is formalizable *in practice*.

Working with such a proof assistant, users construct a formal axiomatic proof.

Formalization of mathematics

The technology allows us to represent definitions, statements, and proofs in digital formats, with

- complete detail,
- precise semantics, and
- precise rules of use.

This is consistent with traditional aims of rigor and precision in mathematics.

Formalization of mathematics

In this talk, I will focus on one platform, Lean, for formalizing mathematics.

It is one among many. Much of what I say will not be specific to Lean.

I'll start with a demo, to give you a sense of what the field is about.

Formalization of mathematics

Formal technology can help us:

- reason better,
- be precise,
- find and repair mistakes,
- build mathematical libraries,
- find definitions, theorems, and proofs,
- collaborate,
- use computer algebra systems,
- use numeric computation,
- use automated reasoning,
- use machine learning, and
- teach mathematics.

Formal methods in teaching

Cons of using a proof assistant:

- A priori, you need to really know what you are doing.
- A priori, you need to remember a lot.
- Syntax is unforgiving.
- Error messages are unhelpful.
- Trivial steps can be really hard.
- The whole thing can be very demoralizing.

It's hard to subject students to that (unless you are a sadist).

Formal methods in teaching

Pros:

- The language is expressive.
- In principle, anything is possible.
- Feedback is instantaneous.
- It can be a lot of fun.

Formal methods in teaching

We can maximize the pros and minimize the cons:

- choose tasks very carefully
- provide specific information that students will need
- provide interfaces that hide some of the complexity
- provide targeted automation

Questions for educational researchers

Using a theorem prover is like playing a video game:

- Cycles of challenge, frustration, success, challenge, frustration, success, . . .
- One can spend hours in total absorption.

Is this limited to self-motivated, self-directed learners, like mathematics and computer science majors at top universities? Does it work for broader populations?

Questions for educational researchers

We want to teach students to reason mathematically.

We ask them to interact with formal technology.

Do the skills transfer?

Questions for educational researchers

There are two attitudes toward teaching mathematical reasoning.

Attitude #1: Reflection on logic and proof structure is essential.

Students need to understand what hypotheses are in play at each point in a prof. They need to learn how to carry out proof by cases, proof by contradiction, how to recognize the difference between a universal statement and an existential one, how to unpack an existential quantifier, and so on.

Attitude #2: Logic is trivial

Students are smart. Teach them the *mathematics*, and they'll pick up the language and the format.

Which is right?

Teaching with interactive theorem provers

Topics:

- mathematics
- computer science
- logic

Emphasis:

- teaching mathematics, computer science, or logic
- teaching formal methods

We may be interested in teaching anyone from an elementary school student to an expert.

Each combination of attributes determines what we should teach and how we should teach it.

Teaching mathematics with Lean

Teaching mathematics can mean:

- Teaching mathematical calculation and computation.
- Teaching mathematical reasoning and argumentation.

I'll focus on the latter.

Teaching mathematics with Lean

Some efforts I know about:

- Kevin Buzzard at Imperial College London (textbook)
- Heather Macbeth at Fordham University
- Gihan Marasingha at the University of Exeter
- Patrick Massot at Université Paris-Saclay (course, and some natural language experiments)
- Sina Hazratpour (and Emily Riehl) at Johns Hopkins (course)
- Matthew R. Ballard at the University of South Carolina

Also Alexandre Rademaker (Brazil), Benedikt Ahrens (Delft), Paige North (Ohio State).

See also Learning Mathematics with Lean at the University of Loughborough.

Textbooks

Theorem Proving with Lean (with Soonho Kong, Leonardo de Moura, and Sebastian Ullrich)

- audience: undergraduates and graduate students in mathematics, computer science, and philosophy
- an introduction to dependent type theory, Lean syntax and semantics
- the course followed up with formalization, metaprogramming.

Logic and Mechanized Reasoning (with Marijn Heule and Wojciech Nawrocki)

- audience: second-year students in computer science
- theory, implementation, and application of logic and automated reasoning

I *won't* talk about these.

Textbooks

Logic and Proof (with Robert Lewis and Floris van Doorn)

- audience: freshmen and sophomores from a variety of backgrounds; no prerequisites beyond high-school mathematics
- an introduction to symbolic logic, informal mathematical proof, and formal proof.

Mathematics in Lean (with Kevin Buzzard, Robert Lewis, and Patrick Massot)

- audience: undergraduate mathematics majors to professional mathematicians
- formalizing mathematics

Logic and Proof

Goals of the course:

- Teach students to write ordinary mathematical proofs.
- Teach students how to use symbolic logic (to make assertions, prove assertions, and specify properties).
- Teach students to use Lean, in service to the other two goals.

Students could find the textbook and do exercises online:

https://leanprover.github.io/logic_and_proof/

Logic and Proof

Mathematical topics: sets, relations (order, equivalence relations), functions, induction, combinatorics, probability, elementary analysis (the real numbers and limits), axioms of set theory

Logic topics: propositional logic, natural deduction, first-order logic, truth assignments and models (informally), higher-order quantifiers

Lean exercises: e.g. propositional and first-order logic, set-theoretic identities, showing that the composition of surjective functions is surjective, or proving the commutativity of multiplication by induction.

Each strand was standard. The main novelty was in combining them.

Logic and Proof

Question. Let $\equiv$ be an equivalence relation on a set A. For every element a in A, let $[a]$ be the set of elements $\{c \in A \mid c \equiv a\}$, that is, the set of elements of A that are equivalent to a. Show that for every a and b, $[a] = [b]$ if and only if $a \equiv b$.

Student answer. Assume $[a] = [b]$. This means that the set of elements $c \in A$ such that $c \equiv a$ is equal to the set of elements $c \in A$ such that $c \equiv b$. Thus everything that is equivalent to a is also equivalent to b. But by transitivity of $\equiv$, $a \equiv c$ and $c \equiv b$ implies that $a \equiv b$.

Moreover, assume $a \equiv b$. Then any $c \in A$ that is $\equiv$ to a will also be $\equiv$ to b by transitivity, and any $c \in A$ that is $\equiv$ to b will also be $\equiv$ to a by transitivity. So $[a] = [b]$.

Logic and Proof

Question. Let f be any function from X to Y, and let g be any function from Y to Z. Show that if $g \circ f$ is injective, then f is injective.

Give an example of functions f and g as above, such that $g \circ f$ is injective, but g is not injective.

Student answer (to first part). Assume that $g \circ f$ is injective. Then by definition, for all $a, b \in X$, we have that $(g \circ f)(a) = (g \circ f)(b) \implies a = b$.

Now assume that there exist some $x, y \in X$ such that $f(x) = f(y)$. Then we have $(g \circ f)(x) = (g \circ f)(y)$, which implies $x = y$ by the injectivity of $g \circ f$. So f is injective by definition.

Logic and Proof

```
variables A B C : Type
variables (f : A → B) (g : B → C)

example (h : injective (g ∘ f)) : injective f :=
assume x1: A,
assume x2: A,
assume h1: f x1 = f x2,
have h2: g (f x1) = g (f x2), by rw h1,
show x1 = x2, from h h2

example (h : surjective (g ∘ f)) : surjective g :=
assume z : C,
exists.elim (h z) $
assume x : A,
assume h1: (g (f x)) = z,
exists.intro (f x) h1
```

Logic and Proof

Observations:

- Make it clear that there are three distinct languages:
  - ordinary mathematics
  - symbolic logic
  - formal proof languages

  Students will not get them confused.
- The parallel developments seemed to help. Students could "see" an exists elimination or an or elimination in an informal proof.
- Students liked the course. There was no clear favorite among the topics: some liked Lean more than the other parts, some less.

Mathematics in Lean

We will likely use this for a meeting this summer, Lean for the Curious Mathematician 2022, at ICERM.

I also plan to use it for an undergraduate course in the fall, for mathematics majors at Carnegie Mellon.

The textbook is designed to be read alongside examples and exercises in Lean.

Goal: Teach readers to formalize mathematics as quickly as possible, to the point where they can contribute to mathlib.

Mathematics in Lean

Design decisions:

- Don't worry about theory, logic, foundations, or explaining how Lean works.
- Start with basic skills, and build up gradually.
- Introduce information as it is needed.
- Focus on mathematical examples.
- Build text around exercises.

Early on, we provide readers with the library facts they will need, and we ask them to fill in small inferences.

Gradually, we show them how to find the facts they need, and expect them to become more independent.

Mathematics in Lean

The table of contents mirrors an introductory "concepts" course:

- basics (calculation, using theorems and lemmas)
- logic (quantifiers, proof by cases, negation)
- sets and functions
- elementary number theory
- abstract algebra

Crowdsourcing

The Lean community web pages and Zulip chat are a tremendous resource.

More seasoned Lean users help newcomers, who then pay the favor forward.

If we can figure out how to expand this cycle to mathematics education more broadly, it can become a powerful force.

Summary

Interactive proof assistants can be a powerful teaching tool.

They provide instantaneous feedback, and allow independent experimentation and exploration.

Questions:

- How can they best be used with different audiences?
- Do they really teach the skills we want to teach?

Conclusions

Formal methods are bringing about a digital revolution in mathematics.

There is a lot of potential for pedagogy:

- interactive feedback
- libraries and online resources
- user communities

We need to be careful.

If we do it right, it can have a big impact.