Teaching with Lean

Jeremy Avigad

Department of Philosophy
Department of Mathematical Sciences

Hoskinson Center for Formal Mathematics

Carnegie Mellon University

April 6, 2022

Formalization of mathematics

Contemporary *digital mathematical assistants* are now being used to formalize substantial mathematical results.

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 demonstration.

Formalization of mathematics

Formal methods are important in computer science and engineering, for verification of hardware, software, networks, security protocols, cryptographic protocols, cyber-physical systems, financial technology, and more.

The technology is also useful for mathematics.

- It provides precise statements of mathematical results.
- It supports verification.
- It supports collaboration.
- Libraries can be used for exploration and search.
- It can serve as a front end to systems for numerical and symbolic computation.
- It opens doors to automated reasoning, machine learning, and new means of discovery.
- It can be used for teaching.

Teaching mathematics with Lean

Proof assistants can be used to teach various aspects of logic and computer science.

I will focus on teaching *mathematics*.

Teaching mathematics with Lean

Why use Lean to teach mathematics?

- Students get immediate feedback.
- It's fun; it can keep students engaged.
- The technology will likely be important for mathematics.
- The technology is already important for other things.

Teaching mathematics with Lean

Caveats:

- Proof assistants are hard to use; they are not designed for teaching.
- They require a lot of knowledge that is incidental to the mathematics.

Strategies:

- Choose tasks very carefully.
- Provide the students the information they need to succeed.
- Use targeted automation and well-designed interfaces.

Teaching mathematics with Lean

How can we incorporate Lean in the classroom?

Three models:

1. Use Lean in an introductory course to teach mathematical concepts and proof.
2. Use Lean in a second-year (or third-year) course to teach students specifically how to formalize mathematics.
3. Use Lean in as either an optional or required add-on to a conventional course.

I will focus on the first two. The third option is much easier if the first two are in place.

Two teaching models

	Model 1 (intro to mathematics)	**Model 2 (intro to ITP)**
Background:	minimal	students who know some mathematics
Audience:	broad	self-selected majors
Purpose:	to teach mathematics	to teach formalization
Justification:	formal methods help understand mathematics	formal methods are important *per se*

Teaching mathematics with Lean

Some efforts I know about:

- Kevin Buzzard at Imperial College London (textbook)
- Gihan Marasingha at the University of Exeter
- Heather Macbeth at Fordham University
- 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).

All but Kevin's class are model 1.

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 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.

Logic and Proof

Debatable features:

- There is an explicit emphasis on symbolic logic.
- The class relies on a very declarative style.

Macbeth and Massot do not make logic explicit, and rely on tactics more (though they still enforce a declarative style).

Another approach

This summer, at the Hoskinson Center, we will start working on a proof checker for the introductory "Concepts of Mathematics" classes at Carnegie Mellon.

Rules of the game:

- We have to work with the existing course material.
- Formal checking will supplement existing exercises.
- Formal syntax for *statements* is o.k.
- A regimented proof language is o.k., with identifiers like *have*, *show*, *assume*, *by induction*, *by cases*
- Justification should be things like "lines 3, 5, and the definition of continuity."

We hope to use very targeted automation to check the justification.

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.

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.