Teaching proofs to a computer.

K. Buzzard, Imperial College London

20th April 2022, Galatasaray University

Before we start

Two things before we start:

1) Thank you to Özge for the invitation, and thanks to you all for coming!

2) I am not a computer scientist – I am an algebraic number theorist who 5 years ago knew *absolutely nothing* about this stuff.

What is a computer proof assistant?

We're about to see Lean, a free and open source computer proof assistant written principally by Leo de Moura at Microsoft Research.

Many other computer proof assistants exist (Coq, Isabelle/HOL, Agda,...).

Other names for the same thing: proof assistant, computer theorem provers, interactive theorem provers (ITPs), ...

Computers and mathematics.

We all know that computers can be used to *compute*.

We've just seen that they can now be used to *reason*.

Lean turns mathematical questions into levels of a puzzle game.

How did I learn Lean?

I learnt basic Lean skills in 2017 by formalising the proofs in my "introduction to proof" course and asking a bunch of questions on the Lean chat.

The questions were mostly answered by PhD students.

Now I teach a Lean course to the undergraduates at Imperial College London.

Some of the students are better than me at it.

This is really pleasantly humbling.

What else does it do?

We've seen that Lean can verify the proof of a trivial lemma in topology.

Are these systems just toys?

Last year a team led by Johan Commelin and Adam Topaz verified the proof of a 2020 theorem of Clausen and Scholze, and soon we will have verified the proof of an *important* theorem of Clausen and Scholze (showing that the reals are a liquid vector space).

What I want to talk about today: *what is the point of doing maths like this?*

(of course a related question is: what's the point of doing maths the old way?)

The future of mathematicians?

Computers can beat us at chess and at go (and at basically any board game).

Microsoft, Facebook, OpenAI, Google. . . are putting time, money and effort into getting AIs to use sort of software to automatically prove theorems by themselves.

Are we mathematicians all going to be out of business in ten years?

My guess: No.

OK so what then is the point?

I think these things are one day going to be helping us.

But let me say right now that they're not there yet.

How far have we got?

Let me explain more details about where we are.

Lean is just a computer program which knows the rules of logic and type theory.

Type theory is a foundation for mathematics, like set theory.

Set theory only has one thing: sets. Type theory has two things: types and terms.

In Lean, a group is a *type* equipped with a multiplication and satisfying some axioms, and the elements are terms.

Introducing `mathlib`

In the Lean community we are developing a library for doing pure mathematics, called `mathlib`.

Development started in 2017. Right now we have over 35,000 definitions and over 85,000 theorems (typed in essentially by hand by volunteers).

An overview is here.

We still do not have all the pure mathematics theorems in an undergraduate degree.

On the other hand we have quite a lot of 1st year graduate level commutative algebra / category theory.

If you follow the Lean community blog you can see monthly updates.

What is the point of making a massive formalised mathematics library?

It's Bourbaki for the 21st century.

It can be used as a base for other formalisation projects.

Example: in 2019 Dahmen, Hoelzl and Lewis formalised the 2017 Ellenberg–Gijswijt Annals of Mathematics paper on the cap set conjecture.

The paper was a hard theorem whose proof only involves elementary objects.

Typical of the community at the time.

What is the point of making a massive formalised mathematics library?

In 2018 myself and some undergraduates at Imperial College London formalised the definition of a scheme.

We proved some very basic theorems about schemes.

We used `mathlib` (localisation of rings, topology,...) and indeed now schemes are part of `mathlib`.

I now have a publication about this with 4 (at the time) undergraduate co-authors. All of them are now doing PhDs.

Perfectoid spaces.

Peter Scholze won a Fields Medal for his definition and work on perfectoid spaces in 2018.

In 2019, Commelin, Massot and myself defined perfectoid spaces in Lean using `mathlib`.

In contrast to schemes, we proved 0 theorems about them, and made 1 example.

What is the point of formalising perfectoid spaces?

It was a propaganda exercise.

It was a proof that computers could engage with modern research mathematics.

It was also an attempt to break away from the traditional "prove hard theorem which used only elementary objects" approach.

Note that we need research mathematicians to be involved for this.

The propaganda exercise was a success.

In 2019 a bunch of other arithmetic geometers showed up on the Lean Zulip chat (Adam Topaz, Riccardo Brasca, Filippo Nuccio, Damiano Testa, Marc Masdeu…).

In 2020 Peter Scholze asked us if we would try another project – to prove one of his recent theorems (joint with Clausen).

Had there not been an influx of algebraic number theorists in 2019, the answer would have been "no".

Scholze quotes

"I think this may be my most important theorem to date. (It does not really have any applications so far, but I'm sure this will change.) Better be sure it's correct. . . "

"while I was very happy to see many study groups on condensed mathematics throughout the world, to my knowledge all of them have stopped short of this proof. (Yes, this proof is not much fun. . .)"

"With this theorem, the hope that the condensed formalism can be fruitfully applied to real functional analysis stands or falls. I think the theorem is of utmost foundational importance, so being 99.9% sure is not enough."

The Liquid Tensor Experiment

Scholze's challenge was to prove a theorem saying that a certain Ext group vanished.

Scholze and Clausen reduce the theorem to a technical lemma about a collection of pseudo-normed abelian groups, and then prove the lemma.

Two months after the challenge had been made, we had formally stated the lemma in Lean. Project led by Johan Commelin.

Four months later, we had proved the lemma.

In December, we formally stated the theorem.

Give us a few more months (or weeks?) and we'll have proved the theorem.

What did Scholze think?

"While this challenge has not been completed yet, I am excited to announce that the Experiment has verified the entire part of the argument that I was unsure about. I find it absolutely insane that interactive proof assistants are now at the level that within a very reasonable time span they can formally verify difficult original research."

More Scholze

Read what Scholze said!

Digitising mathematics.

In the Lean community we are *digitising mathematics*.

It was trivial to digitise chess and go, and look what happened there.

They digitised music and look what happened there.

Let me list a couple of potential uses of computer theorem provers which are *not* science fiction.

A new kind of mathematical document.

A problem with pdf documents is that it is the author who decides the level of detail.

Achievable goal: formally verified (error-free!) undergraduate or research level texts which enable to you zoom into any argument to any level of detail (ambiguity-free!)

Non-example: the Liquid Tensor Experiment blueprint.

Missing link: infrastructure (i.e. programs).

A tool to help researchers.

The Stacks Project is 7000 pages of algebraic geometry on one website.

As a PhD student I had to learn some algebraic geometry. I resorted to "paging through the textbooks".

Typing in 7000 pages of definitions, theorems and proofs will take a *very long time*.

But most of those 7000 pages is proofs – so what if we just skip them for now?

A combination of a database, and "sledgehammers", give us a tool which is better than google at searching for facts in algebraic geometry.

Other applications

I'd be interested to hear ideas.

Summary

Computer theorem provers have come of age.

They are not hard to learn now, and they're fun to use.

We are digitising mathematics and thus making it better.

If some mathematicians start to learn this stuff, cool new tools will appear.

#ads

- Play the natural number game!
- Lean community website `leanprover-community.github.io` and Zulip chat `leanprover.zulipchat.com`
- My course "Formalising mathematics 2022" just finished: this is a course for mathematicians (including undergraduates) who want to learn Lean.