Formalising mathematics in Lean

A GlaMS course

Monica Abu Omar, Simone Castellan, Adrián Doña Mateo, & Patrick Kinnear

a.k.a. **The Lean Team**

January 30th, 2024

Introduction

What is mathematics?

- Is it not just formal logic?
- To non-experts, it may seem hidden behind complex definitions and results, losing its essence of "formal logic."

Mathematics has become so needlessly complicated. Can we reclaim its simplicity and logical nature? Can we make it *accessible* to all, restoring the *beauty* that lies within the subject?

What is Lean?

To quote "Mathematics in Lean":

- Put simply, Lean is a tool for building complex expressions in a formal language known as *dependent type theory*.

Having said this, Lean can be used as a proof verification system.

Demo some "Type theory"

What can proof assistants do?

A proof assistant uses logical and computational methods to verify precise mathematical expressions.

- **Verification:** A proof assistant can verify your proof.
- **Interactivity:** It makes your proof interactive, enhancing teaching and learning engagement.
- **Focus:** It helps you focus on one goal at a time.
  (off-topic debate: Can Lean assist mathematicians with ADHD?)

What can proof assistants not do?

- A proof assistant will **not** automatically prove something for you;
- It is **not** a mathematician.

Patrick's slides from last year's version of this course:

> A calculator can add, but it doesn't know the meaning of addition
>
> -an undergraduate overheard in JCMB, Edinburgh

Specific examples

The four-color theorem:

- In 1879 and 1880, Alfred Kempe and Peter Guthrie Tait proposed proofs of the four-color theorem which were both widely acclaimed. It took 11 years on both accounts to show that the proofs were incorrect!
- In 1989, a book (over 400 pages long) claiming a complete and detailed proof was released.
- In 2005, it was finally formally and reliably verified inside the Coq proof assistant (`https://github.com/coq-community/fourcolor`).

To summarise, proof assistants can help us verify theorems by removing the risk of human-error.

Why Lean?

An ordinary mathematician might ask:

> *"Why should I be interested in Lean? I'm only really interested in mathematical research."*

Contemporary mathematical research has become incredibly specialized, making verification too complex for non-experts (hence, "peer reviews"). This **exclusivity** hinders collaboration and the effective teaching of advanced mathematics.

Why Lean? (cont.)

HOWEVER, thanks to Lean, this can change:

- Formalizing work makes mathematics more accessible, formal, and reliable.
- No need for peer reviews[1]. It's already verified.
- Collaboration becomes easier. People from other fields can build on your formalized code, advancing research in new ways.
- Teaching advanced mathematics becomes more inclusive and engaging.

Lean is our hope to make advanced mathematics as inclusive as possible.

[1] But probably will require some sort of reviews of the code structure, implementation, etc...

About this course

We have but one learning outcome:

- to use Lean to prove mathematical results

Similarly to how one needs to **do** mathematics and not just simply read mathematics:

- Lean is about **doing** mathematics **interactively**, not just simply reading about it.

Because of this, to feel comfortable with Lean and enjoy the niceties of interactive theorem proving:

- there is weekly homework that you are expected to do (Side note: it would be nice and productive to do them together, and this can be in any form you guys want. Go to channel `#socials-info` on Discord for more on this.)

About this course (cont.)

- We will mostly be reading through the interactive book "Mathematics in Lean" for the first 5 weeks to get the basics down,
- then we will have our own set of fun and engaging exercises for you to delve yourself into.

About this course (cont.)

For people wanting to get credit[2] for this course, you will need to complete a (group) project:

- Pick a topic you know well. Familiarize yourself with the relevant section in `Mathlib` (likely in weeks 3/4 onwards).

More details about potential projects will be discussed in week 5. Once you have something in mind, please come talk to us so that we can discuss the feasibility and all. Go to the `#projects` channel for more information.

[2] counts as much as any other SMSTC/GlaMS course!

About this course (cont.)

Last, but not least, the *Sporadic Seminars in Formalising Mathematics* (SSFM):

- These seminars allow anyone to share progress in formalization.
- Presentations can be quick 5-minute talks or longer 20-minute talks.

This is in order to encourage each other to formalise mathematics!

This can take place online, in-person with an agreed time, or before/after each weekly tutorial session (check Discord channel `#ssfm` for details).

What to do now?

- Join the **Discord** server if you haven't already! (if you don't have the link, please email us)
- Create a **GitHub** account and join our organization (link in Discord).
- Make a **Zulip** account, and join the leanprover Zulip chat (link is also in Discord)
- Now head to Discord channel `#session-1` for the next task!