WHAT IS THE POINT OF LEAN'S MATHEMATICS LIBRARY?

Kevin Buzzard, Imperial College London
Topos Institute Colloquium, 12th August 2021.

CONTENTS OF THE TALK:

- What is Lean, and what is `mathlib`?
- What is going on in the `leanprover-community`?
- What is the point of it all?

WHAT IS LEAN?

- Lean is an open source computer program, hosted on GitHub.
- Lean was written by Leonardo de Moura and his team at Microsoft Research.

WHAT IS LEAN?

Lean is a way of modelling mathematics on a computer.

WHAT IS BLENDER?

Blender is a way of modelling mathematics on a computer.
Mathematical Idea -> Blender -> application

WHAT IS LATEX?

LaTeX is a way of modelling mathematics on a computer.
Mathematical idea -> LaTeX -> application (e.g. pdf)

WHAT IS LEAN?

Let's see some Lean.

WHAT IS LEAN?

Blender is a way of modelling $\mathbb{R}^3$ in a computer.
LaTeX is a way of modelling human explanation of mathematics on a computer.
Lean is a way of modelling formal classical mathematics on a computer.
So are Coq, Isabelle/HOL, Metamath, Mizar, Cubical Agda and many other computer programs.

WHAT IS `mathlib`?

`mathlib` is a bunch of open source computer code hosted by the Leanprover Community on Github.
`mathlib` is a Lean library containing mathematical definitions and theorems.
```
import topology.continuous_function.basic
```

`mathlib` is a database of classical mathematical definitions and theorems in analysis, topology, algebra, geometry, number theory and other things.
General direction: “Fields Medal Mathematics.”

`mathlib` OVERVIEW (SNAPSHOT 12TH AUG 2021)

Group theory: We are at advanced UG level.
Sylow's theorems, Nilpotent groups (Ines Wright), solvable groups. Fundamental theorem of Galois theory, insolvability of the quintic (Thomas Browning, Kyle Miller, Aaron Anderson).
Coming: representation theory, group cohomology.

Commutative algebra: some stuff at early PhD level. Noetherian rings and modules, localisation and completion (Kenny Lau, Amelia Livingston), Noether normalisation and the Nullstellensatz, Jacobson rings, discrete valuation rings, Dedekind domains, projective and flat modules...(Damiano Testa, Devon Tuma).
Technical typeclass foundational problems solved by Kenny Lau and Eric Wieser.
Topological algebra (Patrick Massot).
Coming: links to homological algebra (final year UG level)

Real analysis: we are at MSc level.
L^p spaces, open mapping theorem, Bochner and Lebesgue integral, Haar measure on topological groups, multivariable calculus...
Coming: Stokes' theorem (Yury Kudryashov).
Unlocks: Complex Analysis.

Complex analysis: we are at 2nd year UG level.
Still can't do a contour integral.
Coming: contour integrals.

Number theory: we are at final year UG level.
Quadratic reciprocity, sums of squares (Chris Hughes), ideals factor into prime ideals in a number field, class group of a number field is finite (Anne Baanen, Ashvni, Sander Dahmen, Filippo A. E. Nuccio).
Coming: adeles (Maria Ines de Frutus Fernandez), elliptic curves (I just made the definition)

Differential geometry: we are at final year UG level. Real manifolds (C^r, C^∞) (Sébastien Gouëzel, Heather Macbeth), Lie groups (Nicolò Cavalleri) and Lie algebras (Oliver Nash). Conformal maps (Yourong Zang) Coming: linear algebra on bundles.

Algebraic geometry:
We have schemes, but not varieties.
We have plans.
Coming: coherent and etale cohomology.

Category theory (Scott Morrison):
Basic theory, abelian categories, derived functors.
Coming: homological algebra.

Mathlib plays two basic roles:

- A comprehensive digitised library of results across undergraduate, MSc and early PhD level pure mathematics;
- A foundation upon which one can build formalised pure mathematics projects, perhaps at modern research level.

Mathlib is open source, which means that anyone can contribute a theorem or a definition to `mathlib`.
Synergy: `mathlib` is being built by mathematicians and computer scientists working together.

Remember: mathematics has been attempted in many systems before `mathlib`.

WHAT IS GOING ON IN THE LEAN COMMUNITY?

We have a website.
https://leanprover-community.github.io/

Much of the community is focussed on pure mathematics.

- People work on `mathlib` itself.
- People do maths which *depends* on `mathlib`.

PROJECTS USING `mathlib`

Many mathematician Lean users work on projects which *use* mathlib rather than *build* it.
Why?

WHY PEOPLE USE MATHLIB AS A DEPENDENCY

They might be beginners.
They might be working on something which is not really mathlib-appropriate.
They might be working on something which is in mathlib already.
They might not have the confidence or the time to contribute (they just want "the buzz").

Example:
I have a summer project student, Tianchen Zhao, working on Euclidean Geometry in Lean.
Their current aim is to make an interactive Euclid book 1.

Example:
I had a former summer project student Luca Gerolla who defined fundamental groups in Lean.
Their code was never “mathlib-ready”.
But it compiled.

Example:
Johan Commelin, Patrick Massot and myself made the definition of perfectoid space in an independent project.

Example:
The breakthrough theorem of Ellenberg--Gijswijt (Annals of Maths, 2017) on subsets of $\mathbb{Z}/p\mathbb{Z}^n$ with no three-term arithmetic progression was formalised by Dahmen, Hoelzel and Lewis in 2019.

Example:
Patrick Massot's Sphere Eversion Project.

Example:
Seven months ago, Fields Medallist Peter Scholze challenged the formal proof community to verify a 2020 theorem he had recently proved with Clausen.
The Lean community was the only formal proof community to respond (so far).
The theorem is a consequence of a technical lemma.
We've proved the lemma, and are developing the homological algebra necessary to deduce the theorem.
Some of the main players: Johan Commelin, Adam Topaz, Riccardo Brasca,...

Should condensed mathematics be part of `mathlib`? We can decide later.

Why was there such noise in the media about this project recently?
After we proved the Lemma, Scholze announced that the thing he *actually* wanted had been done.
Mathematicians are good at skipping the details; Lean did the opposite.

WHY FORMALISE PURE MATHEMATICS?

WHY FORMALISE MATHEMATICS?

Why do mathematics at all?

- Q: "Why prove Fermat's Last Theorem on a computer?"
- A: "Why prove it in LaTeX?"

WHY FORMALISE MATHEMATICS?

I personally find proving theorems on a computer more satisfying.

WHY FORMALISE MATHEMATICS?

Some people think there might be implications in AI. Note that chess and go are *finite*, and mathematics is *infinite*.

WHY FORMALISE MATHEMATICS?

I will be using this stuff for teaching, when the software is fully mature.

WHY FORMALISE MATHEMATICS?

It teaches me new things.

WHY FORMALISE MATHEMATICS?

It's fun! Try it!

HOW TO EXPERIMENT FOR YOURSELF.

- Play the Natural Number Game
- Search the API documentation for mathematics you're interested in. **https://leanprover-community.github.io/mathlib_docs/**
- Download Lean 3 by carefully following the instructions on the community website.
- Now choose something to work on.

THINGS TO WORK ON:

- Download my course on formalising mathematics **https://github.com/ImperialCollegeLondon/formalising-mathematics** and fill in the sorrys.
- Or download the tutorial project **https://github.com/leanprover-community/tutorials** and fill in the sorrys.
- Or make your own project: ask on the Zulip Chat about how to start.
- You can do multiplayer Lean on CoCalc!

WHY DIGITISE MATHEMATICS?

We digitised music and now we have Spotify.
We digitised text and now we have pdfs.
Digitising things makes them more flexible.

WHY DIGITISE MATHEMATICS?

Instant feedback for students who can speak the language.
Computers can mark homework.

WHY DIGITISE MATHEMATICS?

Semantic search.

WHY DIGITISE MATHEMATICS?

“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.”
Peter Scholze.

QUESTIONS?