Formalizing Mathematics in Lean.

1. Logic

María Inés de Frutos Fernández

Universidad Autónoma de Madrid

18/07/2023

6th EACA International School on Computer Algebra and its Applications

Santiago de Compostela

Mathematical Formalization

Mathematical formalization (supported by a proof assistant) consists on digitalizing mathematical definitions, statements and proofs, using a language that can be interpreted by a computer.

Why formalize mathematics?

- Check proof correctness.

Why formalize mathematics?

- Check proof correctness.
- We CAN formalize current mathematical research:
  - Liquid tensor experiment (Commelin, Topaz, et. al., 2022).

Why formalize mathematics?

- Check proof correctness.
- We CAN formalize current mathematical research:
    - Liquid tensor experiment (Commelin, Topaz, et. al., 2022).
- Semantic search tools.

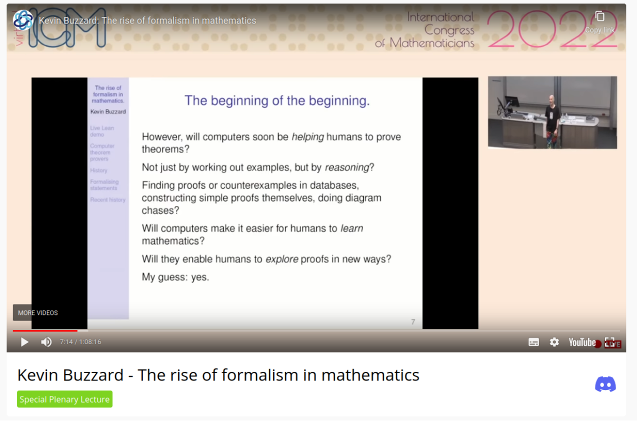
Why formalize mathematics?

- Check proof correctness.
- We CAN formalize current mathematical research:
  - Liquid tensor experiment (Commelin, Topaz, et. al., 2022).
- Semantic search tools.
- Artificial intelligence.

Why formalize mathematics?

- Check proof correctness.
- We CAN formalize current mathematical research:
  - Liquid tensor experiment (Commelin, Topaz, et. al., 2022).
- Semantic search tools.
- Artificial intelligence.
- Teaching and communication.

Why formalize mathematics?



The beginning of the beginning.

However, will computers soon be *helping* humans to prove theorems?

Not just by working out examples, but by *reasoning*?

Finding proofs or counterexamples in databases, constructing simple proofs themselves, doing diagram chases?

Will computers make it easier for humans to *learn* mathematics?

Will they enable humans to *explore* proofs in new ways?

My guess: yes.

7

Kevin Buzzard - The rise of formalism in mathematics

Special Plenary Lecture

`https://www.youtube.com/watch?v=SEID4XYFN7o`

Lean

- Lean is an interactive theorem prover.
- Based in dependent type theory.
- It is also a functional programming language.
- Developed by Leonardo de Moura (Microsoft) since 2013.
- Other proof assistants: Coq, Isabelle/HOL, HOL Light, Agda, Metamath, Mizar, ...

Lean's timeline

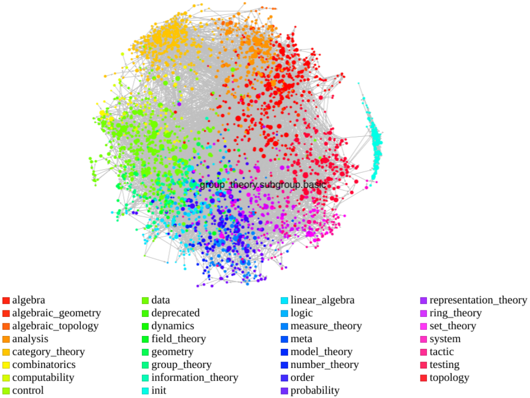
- 2013: Lean starts being developed by Leonardo de Moura (Microsoft).
- 2017: Lean 3.
- 2017: `mathlib` as an independent library.
- 2021: Lean 4.
- 2023: Mathlib 4 (port finished July 15).

Mathlib

- `Mathlib` is Lean's mathematical library (since 2017).
- Open source.
- Decentralized (310 contributors).
- Monolithic ($\sim$45k definitions, $\sim$113k theorems, $>$ 1.1 million lines).

https://leanprover-community.github.io/mathlib_stats.html

mathlib



https://eric-wieser.github.io/mathlib-import-graph/

Course Contents

Goal: introduction to mathematical formalization using Lean 4.

- Tactics
- Structures
- Classes
- Variables (implicit, explicit, inferred)
- Mathlib

Course Plan

1. Logic (tactics)
2. Functions (more tactics)
3. The Algebra Hierarchy (variables; structures and classes)
4. Examples from Number Theory (put everything together)

Tactics

Tactics: instructions to build a proof.

- sorry
- intro
- exact
- apply
- cases'
- constructor
- left
- right
- exfalso
- by_contra
- rfl
- rw
- have
- use
- ...

- Descriptions and examples in tactics.lean.
- `https://github.com/mariainesdff/EACA_School/blob/master/1_logic/tactics.lean`

Learning Resources (I)

Course Repository:

- https://github.com/mariainesdff/EACA_School

Natural Number Game:

- https://adam.math.hhu.de/#/g/hhu-adam/NNG4

Mathematics in Lean:

- https://leanprover-community.github.io/mathematics_in_lean

MSRI Summer School on Formalization of Mathematics:

- https://www.msri.org/summer_schools/1021

Learning Resources (II)

Theorem Proving in Lean 4:

- https://leanprover.github.io/theorem_proving_in_lean4/

More Tutorials/Books/Videos:

- https://leanprover-community.github.io/learn.html

Zulip (Lean community chat):

- https://leanprover.zulipchat.com/