Formal Mathematics and AI

Adam Topaz

University of Alberta

June 14, 2023

Slogan:

Formal methods will act as a *bridge* between AI and mathematics.

Library Building vs. Research Mathematics

Formal Libraries

- Capture some segment of mathematical knowledge in a formal system. (E.g. Undergraduate curriculum.)
- Meant to be *used* in other formalization endeavours.
- Large, many contributors.

Case study (mathlib):

- contributors > 300; 23 maintainers.
- Over a million lines of code (and growing!).
- Algebra, Topology, Category Theory, Analysis, Probability, Geometry, Combinatorics, Dynamics, Data Structures, Logic, ...

Formalizing Research Mathematics

- Formalizes cutting edge theory or result (or both).
- More focused; uses formal libraries.
- Fewer contributors; more direct collaboration.

Case study (The Liquid Tensor Experiment)

- About a dozen main contributors.
- About 100,000 lines of code.
- Algebra, Category Theory, Homological Algebra, Combinatorics, Analysis, Topology.

Key common features:

- *Collaborative.*
- *Asynchronous.*
- *Open.*

“DEMO”

```
instance : Abelian CondensedAb := sorry
```

```
instance : Preadditive CondensedAb := sorry

instance : Abelian CondensedAb where
  normalMonoOfMono := sorry
  normalEpiOfEpi := sorry
  has_finite_products := sorry
  has_kernels := sorry
  has_cokernels := sorry
```

A few reductions later…

```
lemma limitIsSheaf {J : Type (u+1)} [SmallCategory J]
    (F : J ⥤ CondensedAb.{u}) :
    Presheaf.IsSheaf (coherentTopology _)
    (limit (F ⋙ sheafToPresheaf _ _)) := sorry
```

```
lemma limitIsSheaf {J : Type (u+1)} [SmallCategory J]
    (F : J ⥤ CondensedAb.{u}) :
    Presheaf.IsSheaf (coherentTopology _)
    (limit (F ⋙ sheafToPresheaf _ _)) := by
  apply Sheaf.isSheaf_of_isLimit
  apply limit.isLimit
```

How can AI help?

How can AI help? ***Today***?

“AI Collaborator”

- Iterate through all occurrences of `sorry` in the project.
- For each `sorry`, attempt to solve the goal.
- If it succeeds, commit the solution.
- If it fails, move on.

- Based on `GPT-f`, `HyperTree`, `Sagredo`, ...
- Should only attempt proofs.
- Can run as part of a CI process.
- ***It can be useful even if it's not too smart!***

UX

- Search by name, statement, type, etc.
- Natural language.
- **Key**: Editor integration.
- ***It can be useful even if it's not too smart!***

Barriers:

- Cost: Hardware, API fees, ...
- Maintaining infrastructure.
- Culture.

We should implement AI infrastructure in ITPs *right away*. This will be useful even if AI is not too sophisticated. As AI improves, it will be easy to reap the benefits.

Synergy

Mathematical research is diverse and hard to pin down explicitly. Could be split into two interconnected categories:

1. Problem solving.
2. Theory building.

When we *formalize* mathematics (in an ITP or on paper), we don't just solve problems and build theories. Rather:

- We refine and strive to understand the problem solving *process* itself.
- We aspire to find the *right* definitions and abstractions.

These two points (among others) make formalization interesting and scientifically worthwhile.

Parallels with recent trends in AI research:

- Reasoning capabilities of LLMs are augmented using novel prompting techniques (COT, TOT, . . .).
- Getting AI systems to produce useful *definitions* is a huge challenge.

Hopes:

- Expertise gained from formalization of mathematics could help advance AI capabilities.
- Advances in AI could accelerate adoption of formal methods in mathematics.

Hopes:

- Expertise gained from formalization of mathematics could help advance AI capabilities.
- Advances in AI could accelerate adoption of formal methods in mathematics.

We need *increased collaboration* between mathematics and AI.