Reflections on Formalization, Neurosymbolic Reasoning, and the Future of Mathematics

Jeremy Avigad

Department of Philosophy
Department of Mathematical Sciences
Hoskinson Center for Formal Mathematics
Institute for Computer-Aided Reasoning in Mathematics

Carnegie Mellon University

April 7, 2026

Outline

Reflections:

- Formalization (a personal look at how we got here)
- AI for mathematics and neurosymbolic reasoning (thoughts on where we stand)
- The future of mathematics (speculation as to where we are going)

A personal history

I earned my BA in mathematics from Harvard in 1989, worked and traveled for a year, earned my PhD in mathematics from UC Berkeley in 1995, and joined the philosophy department at Carnegie Mellon in 1996.

I started playing with Isabelle and Rocq (then Coq) in the late 1990s.

In 2002, I formalized quadratic reciprocity in Isabelle with students at Carnegie Mellon. The definition of "prime" on the integers implied that there weren't any.

We fixed the definition and the proof went through unchanged.

A personal history

Tom Hales came to the University of Pittsburgh in 2002, and started the Flyspeck project to verify his proof of the Kepler conjecture.

David Gray, Adam Kramer, Kevin Donnelly, and I went on to prove the prime number theorem in 2003.

The library had no complex analysis, and very little analysis, so we used the Selberg–Erdős elementary proof.

Within a few months of finishing the project:

- Georges Gonthier announced a proof of the four-color theorem in Rocq.
- Tom announced a proof of the Jordan curve theorem in HOL Light.

A personal history

Lessons:

- Formalization is fun.
- It's also a pain in the neck.
- It's all about building good libraries.
- Domain specific automation (linear arithmetic) is important.
- Extensible domain general automation (auto, simp) are important.
- Simple type theory is good for reasoning about concrete algebraic structures like the integers as an ordered ring, or the reals as an ordered field, but not algebraic structures that depend on parameters, like the integers modulo p.
- Formalizing a proof requires understanding details that are often glossed over.
- Formalization does not necessarily capture the intuitions: you can formalize a proof without fully understanding it.

A personal history

I attended a meeting of MAP (Mathematics, Algorithms, Proofs) at Dagstuhl in 2005.

Tom was there, as were Assia Mahboubi, Bob Solovay, and Freek Wiedijk.

Tobias Nipkow organized a small sub-meeting of people interested in Tom's *Flyspeck* project. He then spent a sabbatical semester in Pittsburgh working on it.

Around that time, I cleaned up material from the PNT proof and contributed to the Isabelle library.

A personal history

In 2009–2010, I spent a sabbatical year in Paris (at Inria/Saclay) working on Georges Gonthier's project to formalize the odd order theorem.

Assia, Cyril Cohen, and Enrico Tassi were in the same group.

I learned dependent type theory, Rocq, SSReflect, finite group theory, and French.

A personal history

Lessons:

- Dependent type theory can support substantial algebraic reasoning.
- Language design is important.
- A software engineering perspective is valuable.
- Mathematics isn't just software engineering.

A personal history

When I returned from Paris, I decided I was done with formalization. I was more interested in automation.

But then:

- Luke Serafin (an undergraduate) convinced me to work with him on formalizing the central limit theorem in Isabelle.
- Chris Kapulkin (a PhD student with Tom) convinced me to work on a project in homotopy type theory.

A personal history

In 2012, I attended a meeting at the Isaac Newton Institute, and spent time with a friend, Grant Passmore, who was then a postdoc with Larry Paulson.

Grant and I hatched an idea to get mathematicians using automated reasoning tools:

- Write a nice Python front end.
- Connect it to powerful back-end tools.
- Make it part of Sage.

We talked to Leonardo de Moura (then at Microsoft Research) by Skype, and he liked the idea.

A personal history

Cody Roux arrived at Carnegie Mellon for a postdoc in the fall and we began prototyping the system, *Boole*.

The four of us spent a lot of time talking about:

- language and logical foundations (set theory, simple type theory, dependent type theory)
- proof assistants (Isabelle, PVS, ACL2, Agda, HOL Light, Rocq/SSReflect, ...)
- formalizations of mathematics and computer science.

A few months later, Leo announced that he was starting the Lean project, and attention turned to that.

Leo convinced me that even if one is interested in automated reasoning for mathematics, one needs to build it on a system with a good library and precise semantics.

A personal history

I worked on Lean's first libraries and documentation, and the number of students, postdocs, and collaborators joining me grew steadily over the years.

Leo gave a nice overview of the history of Lean at the ItaLean conference in 2025.

Kevin Hartnett has written a book, *The Proof in the Code*, that will be released this summer.

A personal history

2013 Lean 0.1

2014 Lean 2

2016 Lean 3

2017 *Big Proof* at the Isaac Newton Institute.

2017 Mario Carneiro and Johannes Hölzl split off mathlib.

2017 Kevin Buzzard, Patrick Massot, Sébastien Gouëzel, and others get involved.

2019 Rob Lewis and Johannes Hölzl organize *Lean Together* in Amsterdam. The Maintainers Team is formed.

2020 There is another *Lean Together* at CMU, just before the pandemic.

2020 Daniel Selsam proposes the IMO grand challenge.

2020 Johan Commelin and Patrick Massot organize the first *Lean for the Curious Mathematician* (online).

A personal history

2020 The Liquid Tensor Experiment is launched, and Peter Scholze acknowledges a substantial milestone six months later.

2021 Charles Hoskinson endows the Hoskinson Center for Formal Mathematics at Carnegie Mellon.

2022 Lean 4 is released.

2022 *Lean for the Curious Mathematician* at ICERM, and a Mathlib developers' meeting after.

2023 *Machine Assisted Proof* at IPAM. Terry Tao chairs the program committee.

2023 The Lean FRO is launched.

2024 Google DeepMind's AlphaProof and AlphaGeometry score at the level of a silver medal in the IMO.

2025 The Mathlib Initiative is launched.

2025 The Institute for Computer-Aided Reasoning in Mathematics is launched.

A personal history

I have worked on verification related to SAT-solving with Marijn Heule and students (Cayden Codel, Wojciech Nawrocki, Joshua Clune, and James Gallicchio):

- Verifying reductions from mathematical problems to combinatorial problems.
- Verifying reductions from combinatorial problems to propositional logic, including symmetry-breaking.
- Verifying certificates from SAT solvers.

I have also worked on verification related to convex optimization with Alexander Bentkamp and Ramon Fernández Mir:

- Verifying reductions of mathematical problems to convex optimization problems.
- Verifying reductions of convex optimization problems to solver-specific formulations.

A personal history

I have enjoyed working with the blockchain company StarkWare:

- Verifying algebraic encodings of computational claims.
- Verifying the soundness and completeness of library proof procedures.
- Verifying elliptic curve calculations, etc.

Lessons:

- The more verification the better.
- Be practical: pick and choose your battles.
- Figure out where formalization is genuinely helpful.
- Focus on where mistakes typically creep in.

Outline

Reflections:

- Formalization (a personal look at how we got here)
- AI for mathematics and neurosymbolic reasoning (thoughts on where we stand)
- The future of mathematics (speculation as to where we are going)

AI for Mathematics

AI for Mathematics includes:

- Formalization and proof assistants
- Symbolic AI and automated reasoning
- Neural AI and machine learning
  - Mining data for insights
  - Searching for combinatorial objects
  - Physics-inspired neural networks
- Neurosymbolic theorem proving.

Advances in the last domain have been most striking, but should not eclipse the others.

AI for Mathematics

Big tech:

- OpenAI
- Google DeepMind
- Anthropic
- xAI
- Meta
- Mistral
- ByteDance
- DeepSeek

Startups:

- Harmonic
- Logical Intelligence
- Axiomatic AI
- Math Inc.
- Axiom
- Principia Labs

These translate to hundreds of millions of dollars devoted to AI for formal and informal theorem proving.

Neurosymbolic reasoning

Symbolic methods:

- *Pros:* precise, rule-based, logic-based; we know what they are doing and what the results mean; results can be audited.
- *Cons:* search gets lost easily; they are brittle and domain specific

Neural methods:

- *Pros:* can synthesize tremendous amounts of data, detect patterns, see the big picture, perform general reasoning tasks
- *Cons:* we don't know what answers mean, whether they are reliable, or whether they align with our values and interests; they are computationally and environmentally expensive

Combining the two is powerful, and provides the best of both worlds.

Neurosymbolic reasoning

Roles for symbolic methods in AI:

- Computation (numeric, symbolic)
- Input specifications
- Output / answers:
  - models
  - explanations
  - proofs and verifiable certificates
  - descriptions of mathematical objects
  - programs
- Some search:
  - when there are good heuristics (SAT solvers)
  - when they are close to computation
  - when they are small
- Agentic interactions with all the above

Outline

Reflections:

- Formalization (a personal look at how we got here)
- AI for mathematics and neurosymbolic reasoning (thoughts on where we stand)
- The future of mathematics (speculation as to where we are going)

My career as a visionary

Things I got right:

- Formalizing mathematics makes sense.
- Having an expressive language that can handle a large network of algebraic structures is crucial.
- Both domain-general and domain-specific automation are important.
- Computation and classical reasoning are both important, and not at odds with one another.
- Formalization isn't programming; mathematics isn't computer science.
- Formalization isn't just about verification, but also collaboration, exploration, reference, teaching, and having fun.
- Exposition and outreach are important (documentation, tutorials, talks, meetings).
- Building a community is important (social media was key).

My career as a visionary

Things I got wrong:

- In early 1996, a friend explained to me that JavaScript was revolutionary because it would facilitate online shopping. I was skeptical that anyone would want to buy things online.
- In the early days of formalization, with others, I thought that formalizing classical results in mathematics would be compelling to mathematicians. Kevin Buzzard, Patrick Massot, and others did much better by going after fashionable contemporary results.
- In 2017, at Big Proof, I told people that mathematical formalization was "not ready for prime time" because few serious mathematicians would be willing to put up with it. (Tom Hales and Vladimir Voevodsky were outliers.)

My career as a visionary

More things I got wrong:

- In 2017, when Mario Carneiro convinced Leo to split off Mathlib so that he could make it an open source project, I thought it would be a mess, and that mathematicians wouldn't have the skills or patience to develop a clean, maintainable library.
- In 2020, when Daniel Selsam proposed the IMO grand challenge, I thought it was good PR, but nonsense. I wasn't alone; many of us thought ML researchers were making empty promises and seriously misunderstood the nature of mathematics.

My career as a visionary

My success has largely been a matter of being in the right place at the right time.

When I started, formalization was niche. AI for mathematics suddenly appeared around me.

I was lucky

- to have met Leo,
- to be there when he was looking for a project after Z3, and
- that he was interested in supporting mathematics.

I am also lucky to have had so many amazing students and collaborators at Carnegie Mellon.

Contributors to Lean at CMU

Cody Roux (postdoc), Grant Passmore (postdoc), Jason Rute (PhD student), Floris van Doorn (PhD student), Robert Lewis (PhD student), Mario Carneiro (PhD student, then postdoc), Jakob von Raumer (visiting MS student), Johannes Hölzl (postdoc), Andrew Zipperer (MS student), Sebastian Ullrich (visiting MS student), Minchao Wu (MS student), Gabriel Ebner (visiting PhD student, first Hoskinson Center postdoc), Paula Neeley (PhD student), Simon Hudon (postdoc), Bruno Bentzen (postdoc), Seul Baek (PhD student), Edward Ayers (postdoc), Alexander Bentkamp (visiting PhD student and visiting postdoc), Ramon Fernández Mir (visiting PhD student), Bartosz Piotrowski (visiting PhD student), Zhangir Azerbayev (visiting undergraduate), Wojciech Nawrocki (PhD student), Joshua Clune (PhD student), Cayden Codel (PhD student), Hannah Fechtner (PhD student), Chase Norman (PhD student), Patrick Massot (visiting professor), Tomáš Skřivan (postdoc), David Renshaw (local contributor, now innovation engineer), James Gallicchio (MS student), Sidharth Hariharan (PhD student), Thomas Zhu (MS student), Aecus Sheng (MS student), Riyaz Ahuja (undergrad/MS student), Tate Rowney (undergrad/MS student), Shuge Rong (MS student)

Collaborators at CMU and Pitt

Other colleagues and collaborators:

- Steve Awodey
- Ed Clarke
- Thomas Hales (University of Pittsburgh)
- André Platzer
- Marijn Heule
- Sean Welleck
- Bryan Parno

and their students and postdocs (notably Soonho Kong, Jesse Han, and Reid Barton).

Predictions

Mathematics isn't going away.

Key features:

- Mathematics provides precise language for describing the world and abstracting from experience.
- It provides means for solving hard problems, and thinking and reasoning more efficiently.
- It provides means for the mathematical community to come to consensus as to what is true and what is not, and to build on one another's work.
- It provides results and conclusions that are remarkably stable over time.

This is not going to change.

Predictions

What will change: AI will play an increasing role in

- exploring new ideas
- discovering new phenomena
- proving theorems
- verifying correctness

Neural theorem proving and autoformalization are only part of the story.

Future generations will find ways to use AI that we can't even imagine now.

Mathematics will thrive in the long run, but technology will be disruptive in the years ahead.

Concerns

Concerns about AI in general:

- We are trusting AI systems we don't understand.
- We don't know whether the answers are reliable.
- We don't know whether the answers align with our values and interests.
- Data centers are taking a toll on the environment.
- AI is expensive, and the benefits are not distributed equally.

Mathematics is as important as ever in the age of AI:

- We can ask precise questions.
- We can audit and verify answers.
- We can interact with AI systems in a way that is meaningful to us.
- We can use AI to answer questions more efficiently.

Concerns

Concerns about the effects of AI on mathematics:

- AI is getting good at many of the things that mathematicians do, and will eventually be better at many of them.
- Public perception of mathematics may change; why do we need mathematicians when we have AI?
- We need to think about why mathematics is important to us and reasses our roles, but the mathematics community is conservative and slow to change.
- We don't know how to incentivize, evaluate, and reward work that makes use of new technologies.
- Some uses of AI are expensive; mathematics may become more like big science.

Predictions

I have argued that the fundemantal goals and values of mathematics will not change.

What will change:

- the way to pursue the goals
- the questions we think are natural and interesting
- the types of answers we expect
- the kinds of results and activities we find important and valuable

These have all evolved over the history of mathematics, and will continue to evolve.

Predictions

AI will put pressure on mathematicians to change more quickly than we have in the past.

It won't be easy, but we will manage.

We need to support the next generation of mathematicians as they make sense of the new landscape, and learn how to take advantage of the new opportunities.

Questions

As mathematicians, we are theory builders and problem solvers.

AI should enhance that, not replace it.

Mathematicians should be able to use AI to do things that nobody else can do.

How?

Questions

Consider the spectrum from pure to applied:

- Abstract analysis; differential equations and PDEs; applications to physics; engineering; building things.
- Measure-theoretic probability, functional analysis, ergodic theory; applications to probability and statistics; statistical methods in economics, finance, and the sciences; concrete applications.
- Abstract computability theory; computational complexity and algorithms; software engineering; programming.
- Number theory; elementary number theory; cryptography; implementing specific cryptographic protocols.

As we go down each list, the mathematics becomes less pretty but more concretely useful.

Questions

Abstract mathematics provides conceptual framework in which we can think about the applications.

In the face of technology, will "conceptual" mathematics become less important?

Will technology need to use the same conceptual framework? Will we need conceptual mathematics to use technology effectively?

Will there be a greater emphasis on computational techniques than conceptual knowledge?

Questions

The infinite is so much more elegant than the finite!

Pure mathematicians dislike:

- concrete bounds
- questions with numeric answers (how many configurations? is the answer 7 or 8?)

They prefer:

- qualitative results
- asymptotic behavior
- universal/existential questions

Yet these are further removed from experience.

Questions

Technology may provide us with new ways to reason about complex, yet finite, domains.

Will questions about finite objects and domains become more important to mathematics?

Questions

Should we still teach students how to do integration by parts?

- *Con:* AI can do that perfectly well.
- *Pro:*
  - We still teach students basic arithmetic, even though they can use calculators.
  - Working through symbolic calculations helps students understand the associated concepts and apply them.
  - It's part of teaching students how to think and reason mathematically, even in the age of AI.

I tend to the latter view, but we should think about questions like these.

What should we teach our students now, and how should we teach it?

Conclusions

Recent developments in AI for mathematics are both exciting and frightening.

Our challenge is to minimize the disruption and maximize the benefits of the new technologies.