

Mathematics and the Formal Turn

Jeremy Avigad

Department of Philosophy
Department of Mathematical Sciences
Hoskinson Center for Formal Mathematics

Carnegie Mellon University

July 15, 2024

Formal methods in mathematics

Formal methods are a body of logic-based methods used in computer science to

- write specifications (for hardware, software, protocols, and so on), and
- verify that artifacts meet their specifications.

They rely on:

- formal languages
- formal semantics
- formal rules of inference.

Formal methods in mathematics

There are:

- tools for automated reasoning
- tools that support robust user interaction.

Most domains require a combination of the two.

Formal methods can also be used for mathematics.

I will try to explain how, and why they are useful.

Interactive theorem provers

We have known since the early twentieth century that mathematics can be formalized:

- Mathematical statements can be expressed in formal languages, with precise grammar.
- Theorems can be proved from formal axioms, using prescribed rules of inference.

With the help of computational proof assistants, this can be carried out in practice.

In many systems, the formal proof can be extracted and verified independently.

Interactive theorem provers

Proof assistants are now used for

- hardware, software, and systems verification
- mathematics and the mathematical sciences

Some proof assistants for mathematics:

- Mizar (1973, set theory)
- Isabelle (1986, simple type theory)
- Rocq (1989, dependent type theory)
- HOL Light (1994, simply type theory)
- Lean (2013, dependent type theory)

In this talk, I will focus on Lean.

The Lean interactive proof assistant

I will give a demonstration.

You can run a similar demonstration [in your browser](#).

(The red highlighting signifies a link you can click.)

Notes:

- Put your cursor on any keyword with a squiggly blue underline to see the response from Lean in the information window to the right.
- Move your cursor through any proof to see the proof state change.
- Hover over identifiers and symbols to see popup documentation.

Lean and Mathlib

Very few mathematicians were using formal methods in 2017.

Things have changed dramatically since then:

- **Mathlib** has almost **1.6 million lines of code**.
- The **Lean Zulip channel** has 9.5K members, about 850 active in any two-week period.
- There have been a number of celebrated successes.
- There have been a number of articles in the general media.
- There are **meetings, workshops, and summer schools** related to Lean.
- There is growing interest and enthusiasm in the mathematical community.

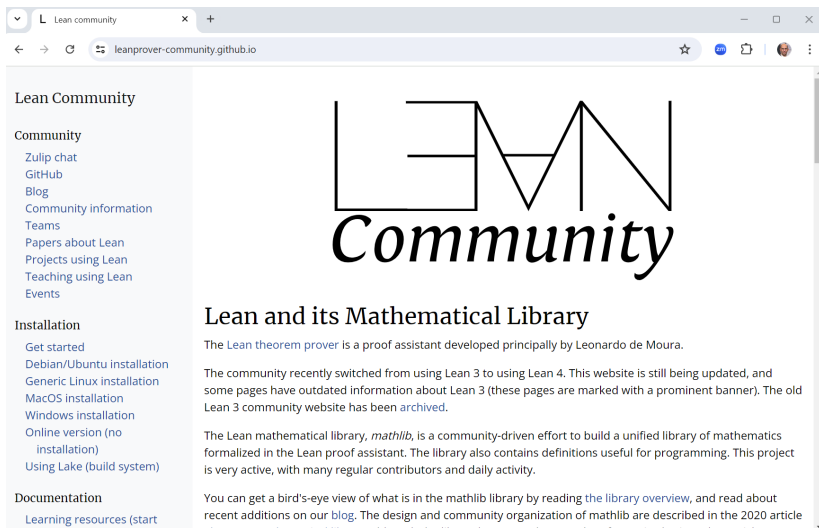
Lean and Mathlib

Some history:

- The Lean project was launched in 2013 by Leonardo de Moura, with Soonho Kong.
- In 2017, Mathlib was separated from the main repository, and the **Lean Community** was born.
- Lean 4 was officially released in 2023, with de Moura and Sebastian Ullrich as principal developers.
- The **Lean Focused Research Organization** was launched in 2023.

It's an open source project: many people have contributed code, libraries, tooling, infrastructure, documentation, teaching materials, and more.

Lean and Mathlib

A screenshot of a web browser displaying the Lean Community website. The browser's address bar shows 'leanprover-community.github.io'. The website has a light blue header with the text 'Lean community'. On the left, there is a sidebar with navigation links under the heading 'Lean Community'. The main content area features a large logo for 'LEAN Community' and a section titled 'Lean and its Mathematical Library' with several paragraphs of text.

Lean community

leanprover-community.github.io

Lean Community

Community

- [Zulip chat](#)
- [GitHub](#)
- [Blog](#)
- [Community information](#)
- [Teams](#)
- [Papers about Lean](#)
- [Projects using Lean](#)
- [Teaching using Lean](#)
- [Events](#)

Installation

- [Get started](#)
- [Debian/Ubuntu installation](#)
- [Generic Linux installation](#)
- [MacOS installation](#)
- [Windows installation](#)
- [Online version \(no installation\)](#)
- [Using Lake \(build system\)](#)

Documentation

- [Learning resources \(start](#)

LEAN Community

Lean and its Mathematical Library

The [Lean theorem prover](#) is a proof assistant developed principally by Leonardo de Moura.

The community recently switched from using Lean 3 to using Lean 4. This website is still being updated, and some pages have outdated information about Lean 3 (these pages are marked with a prominent banner). The old Lean 3 community website has been [archived](#).

The Lean mathematical library, *mathlib*, is a community-driven effort to build a unified library of mathematics formalized in the Lean proof assistant. The library also contains definitions useful for programming. This project is very active, with many regular contributors and daily activity.

You can get a bird's-eye view of what is in the mathlib library by reading [the library overview](#), and read about recent additions on our [blog](#). The design and community organization of mathlib are described in the 2020 article [The Lean mathematical library](#), although the library has grown by an order of magnitude since that article.

Lean and Mathlib

Some achievements:

- a **formalization of perfectoid spaces** (Buzzard, Commelin, and Massot)
- the **liquid tensor experiment** (Commelin, Topaz, and many others)
- a **formalization of the sphere eversion theorem** (Massot, Nash, and van Doorn)
- Mehta's **formalization** of Campos, Griffiths, Morris, and Sahasrabudhe's lower bounds in Ramsey theory
- the **formalization** of the Gowers, Green, Manners, and Tao proof of the Polynomial Freiman-Ruzsa conjecture
- a **formalized consistency proof** for Quine's NF by Holmes and Wilshaw.

Lean and Mathlib

Good press:

- *Quanta*: “Building the Mathematical Library of the Future”
- *Quanta*: “At the Math Olympiad, Computers Prepare to Go for the Gold”
- *Nature*: “Mathematicians Welcome Computer-Assisted Proof in ‘Grand Unification’ Theory”
- *Quanta*: “Proof Assistant Makes Jump to Big-League Math”
- *New York Times*: “A.I. is Coming for Mathematics Too”
- *Quanta*: “‘A-Team’ of Math Proves a Critical Link Between Addition and Sets”
- *Scientific American*: “AI Will Become Mathematicians’ ‘Co-Pilot’”

Lean and Mathlib

A.I. Is Coming for Mathematics, Too

For thousands of years, mathematicians have adapted to the latest advances in logic and reasoning. Are they ready for artificial intelligence?

 Share full article



Karlsruhe connections

- Sebastian Ullrich
 - master's thesis under Avigad and Gregor Snelting
 - PhD under Snelting
 - Co-developer of Lean 4
 - Head of Engineering and co-founder of the Lean FRO
- Jakob von Raumer
 - master's thesis under Avigad, Awodey, and Snelting
 - member of the programming paradigms group
- Marc Huisinga
 - master's thesis under Ullrich
 - Research Software Engineer, Lean FRO
- Markus Himmel
 - bachelor's thesis under Ullrich
 - Research Software Engineer, Lean FRO

Formal methods in mathematics

The *Bulletin of the American Mathematical Society* just ran two consecutive special issues on new technologies for mathematics, *Will Machines Change Mathematics?*

The collection explored:

- formalization and proof assistants
- AI for mathematics
- automated reasoning for mathematics
- social, ethical consequences of the new technologies.

Formal methods in mathematics

MATHEMATICS AND THE FORMAL TURN

JEREMY AVIGAD

ABSTRACT. Since the early twentieth century, it has been understood that mathematical definitions and proofs can be represented in formal systems systems with precise grammars and rules of use. Building on such foundations, computational proof assistants now make it possible to encode mathematical knowledge in digital form. This article enumerates some of the ways that these and related technologies can help us do mathematics.

INTRODUCTION

One of the most striking contributions of modern logic is its demonstration that mathematical definitions and proofs can be represented in formal axiomatic systems. Among the earliest were Zermelo's axiomatization of set theory, which was introduced in 1908, and the system of ramified type theory, which was presented by Russell and Whitehead in the first volume of *Principia Mathematica* in 1911. These were so successful that Kurt Gödel began his famous 1931 paper on the incompleteness theorems with the observation that “in them all methods of proof used today in mathematics are formalized, that is, reduced to a few axioms and rules of inference.” Cast in this light, Gödel's results are unnerving: no matter what mathematical methods we subscribe to now or at any point in the future, there will always be mathematical questions, even ones about the integers, that cannot be settled on that basis—unless the methods are in fact inconsistent. But the positive

Applications

Some applications:

- verifying mathematics
- building mathematical libraries
- collaborating
- verifying mathematical computation
- using automated reasoning
- using machine learning
- synthesizing neural and symbolic AI
- teaching

Verifying mathematics

On December 5, 2020, Peter Scholze **challenged** anyone to formally verify some of his recent work with Dustin Clausen.

Johan Commelin led the response from the Lean community. On June 5, 2021, Scholze acknowledged the achievement.

“Exactly half a year ago I wrote the Liquid Tensor Experiment blog post, challenging the formalization of a difficult foundational theorem from my Analytic Geometry lecture notes on joint work with Dustin Clausen. While this challenge has not been completed yet, 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.”

Verifying mathematics

On November 9, 2023, W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao announced a proof of the PFR conjecture.

On November 18, Tao asked for help verifying it.

Close to 30 members of the Lean community joined him. The formalization was done by December 5.

See:

- the [project page](#)
- the [article in Quanta](#)

Verifying mathematics

Quine introduced his *New Foundations* system (NF) in 1937.

Its consistency has been a longstanding open question.

Randall Holmes claimed a proof in 2015. Jamie Gabbay also claimed one. The technical details were overwhelming.

In April 2024, Holmes announced that Sky Wilshaw, a Part III (masters) student at Cambridge, formally verified Holmes' proof.

Verifying mathematics

“I have been convinced for a long time that I saw the path to proving the result. The problem is that the argument is insanely detailed and any paper text has something wrong with it. This can be attributed partly to deficiencies of mine as an expositor, but since I was the only one who saw it, I had to do the writing. It is also intrinsic: there is a lot of elaborate and not necessarily intuitive bookkeeping in the argument, and its very easy to write things down wrong. I am sure now that (1) the Lean proof is correct, I read the statements of the conclusions, and it proves the right thing and (2) the paper as it stands is converging to the right thing, because Sky is advising me where what I do differs from what is done in the formal proof, and she appears to follow what I have written now fairly happily.” (Holmes)

Building mathematical libraries

Lean's Mathlib currently has almost 1.6 million lines of code.

We can look at:

- the repository
- library statistics
- API documentation
- instances of the ring class
- classes that the real numbers are instances of

This provides a searchable digital reference and repository of knowledge.

Collaboration

Digitizing mathematics is a collaborative effort.

The Lean community is a self-governing grassroots organization.
See:

- the web pages
- the community teams

The Lean Zulip channel currently has

- More than 9.5K subscribers
- About 850 active in any 15-day period
- about 1,000 messages every day

Collaboration

The liquid tensor experiment was a model for digital collaboration.

- The formalization was kept in a shared online repository.
- Participants followed an informal blueprint with links to the repository.
- Participants were in constant contact on Zulip.
- Lean made sure the pieces fit together.

Patrick Massot developed *Blueprint* software to support collaborative projects like this.

A number of projects, including the sphere eversion project and the polynomial Freiman-Rusza conjecture project follow this methodology.

Collaboration

Blueprint for the Liquid Tensor Experiment

Introduction

1 First part

1.1 Breen–Deligne data

1.2 Variants of normed groups

1.3 Spaces of convergent power series

1.4 Some normed homological algebra

1.5 Completions of locally constant functions

1.6 Polyhedral lattices

1.7 Key technical result

2 Second part

3 Bibliography

Section 1 graph

Section 2 graph

1.2 Variants of normed groups

Normed groups are well-studied objects. In this text it will be helpful to work with the more general notion of *semi-normed group*. This drops the separation axiom $\|x\| = 0 \iff x = 0$ but is otherwise the same as a normed group.

The main difference is that this includes “uglier” objects, but creates a “nicer” category: semi-normed groups need not be Hausdorff, but quotients by arbitrary (possibly non-closed) subgroups are naturally semi-normed groups.

Nevertheless, there is the occasional use for the more restrictive notion of normed group, when we come to polyhedral lattices below (see Section 1.6).

In this text, a morphism of (semi-)normed groups will always be bounded. If the morphism is supposed to be norm-nonincreasing, this will be mentioned explicitly.

Definition 1.2.1 ✓

Let $r > 0$ be a real number. An r -normed $\mathbb{Z}[T^{\pm 1}]$ -module is a semi-normed group V endowed with an automorphism $T: V \rightarrow V$ such that for all $v \in V$ we have $\|T(v)\| = r\|v\|$.

The remainder of this subsection sets up some algebraic variants of semi-normed groups.

Definition 1.2.2 ✓

A *pseudo-normed group* is an abelian group $(M, +)$, together with an increasing filtration $M_c \subseteq M$ of subsets M_c indexed by $\mathbb{R}_{\geq 0}$, such that each M_c contains 0, is closed under negation, and $M_{c_1} + M_{c_2} \subseteq M_{c_1+c_2}$. An example would be $M = \mathbb{R}$ or $M = \mathbb{Q}_p$ with $M_c := \{x : |x| \leq c\}$.

A pseudo-normed group M is *exhaustive* if $\bigcup_c M_c = M$.

All pseudo-normed groups that we consider will have a topology on the filtration sets M_c . The most general variant is the following notion.

Definition 1.2.3 ✓

A pseudo-normed group M is *CH-filtered* if each of the sets M_c is endowed with a topological space structure making it a compact Hausdorff space, such that following maps are all continuous:

- the inclusion $M_{c_1} \rightarrow M_{c_2}$ (for $c_1 \leq c_2$);
- the negation $M_c \rightarrow M_c$.

Verifying mathematical computation

Proof assistants can be used to verify the correctness of mathematical results obtained by computation.

There are efforts to use Lean:

- to verify reductions for optimization problems
- as a scientific programming language

Broader applications of formal verification:

- hardware and software
- cyber-physical systems
- network protocols
- privacy and security
- blockchain and decentralized finance

Automated reasoning

Automated reasoning tools hold promise for solving combinatorial problems in mathematics.

For example, Joshua Brakensiek, Marijn Heule, John Mackey, and David Narváez used a SAT solver to resolve Keller's conjecture:

Quanta, "[Computer Search Settles 90-Year-Old Math Problem](#)"

The SAT solver output a proof that was checked with a verified proof checker.

Josh Clune [verified](#) the key mathematical reduction in Lean.

Automated reasoning

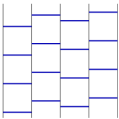


Figure 1: Two-dimensional tiling

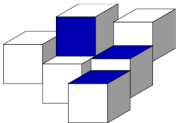


Figure 2: Three-dimensional tiling

Figure 1: a gap-free tiling of the two-dimensional space with equal-sized square tiles. The bold blue edges denote that two tiles are fully connected.

Figure 2: a partial tiling of the three-dimensional space with equal-sized cubes. The only way to tile the entire space would result in a fully face-sharing square at the position of the blue squares.

Keller graphs

A crucial step in proving Keller's conjecture in the seventh dimension is a reformulation of the problem as a property of Keller graphs, an invention by Corradi and Szabo in 1990. The Keller graphs are constructed using two parameters: the dimension n and the shift s . Each vertex in a Keller graph can be considered a dice with n dots such that each dot is colored using a palette of $2s$ colors. The colors come in s pairs of opposite colors. For example, black and white are opposite colors. Red and green are opposite colors as well. Two vertices (dice) are connected if 1) they have at least two dots that differ in color and 2) they have at least one dot with opposite colors.

Let's consider the graph with $n=2$ and $s=2$. For the two pairs of opposite colors we will use black/white and red/green. Figure 3 shows this graph. All 16 different dice are shown. The top dice (black + white) is connected to the left-most dice (red + black) because both dots are different (requirement 1) and the color of their second dot is opposite (white versus black, thus requirement 2). The top dice is not connected to the dice with two red dots: The colors of both dots differ, but they don't have a dot with opposite colors.

Corradi and Szabo showed that Keller's conjecture is false for dimension n if there exists a Keller graph with dimension n and some shift s such that $2^n n$ dice are fully connected. Keller's conjecture would have been false if there were 4 dice that were fully connected in the shown graph. However, observe that there are not even 3 dice that are fully connected.

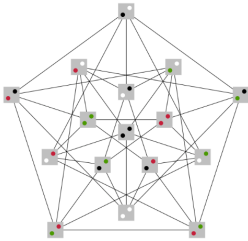


Figure 3: a Keller graph

Automated reasoning

In recent years Kisielewicz and Lysakowska made significant progress regarding Keller's conjecture. In short, they

Automated reasoning

The *Happy Ending Theorem* (Erdős, Szekeres, and Klein) says that for every positive n , any sufficiently large finite set of points in general position contains a convex n -gon.

One can also ask about *empty* n -gons.

There are infinite sets of points with no empty convex 7-gon.

In 2024, Heule and Scheucher used a SAT solver to **show** that 30 points guarantee the existence of an empty hexagon, but not 29.

Also in 2024, Subercaseaux, Nawrocki, Gallicchio, Codel, Carneiro, and Heule **verified** the reduction to a SAT problem in Lean (and ran a verified checker on the SAT solver's proof).

Machine learning

Applications of machine learning to mathematics are a new frontier.

There have been important machine-learning projects using Mizar, HOL Light, Metamath, Isabelle, Coq, Lean, and others.

OpenAI got a neural theorem prover for Lean to solve problems from the International Mathematics Olympiad.

Searching for formally checkable content provides a clear signal.

Using symbolic AI and machine learning

Symbolic AI and machine learning have complementary strengths. It's an important challenge to synthesize the two.

Mathematics is the best place to start. AI can be used for mathematical discovery as well as verification.

See:

- [This](#) survey of automated reasoning for mathematics, and the references to machine learning there.
- [LLMLean](#)
- [Lean Copilot](#)

Teaching

Proof assistants offer a lot of potential for teaching mathematics.

Interaction provides:

- immediate feedback and positive encouragement
- error messages and correction
- information about the current state of a proof
- means to search, experiment, and explore
- increased student engagement

There is a lot of helpful information on the [teaching page](#) of the Lean community web site.

Teaching

Some of my favorite teaching resources:

- The Natural Number Game
- The Set Theory Game
- The Mechanics of Proof
- Verbose Lean
- How to Prove it With Lean
- Mathematics in Lean
- Logic and Mechanized Reasoning

Lean's **widgerts** library offers promising opportunities.

Why formal methods

Formal technology can help us:

- verify results,
- build mathematical libraries,
- explore new concepts,
- collaborate,
- teach mathematics,
- carry out mathematical computation more rigorously,
- explore applications of automated reasoning and machine learning, and
- discover new mathematics.

Digital mathematical technology is transformative, and has a lot to contribute to mathematics.