Formalization and the future of mathematics

Jared Duker Lichtman

Stanford University

Accelerating Mathematical Discovery
A Crash Course in AI and Computation

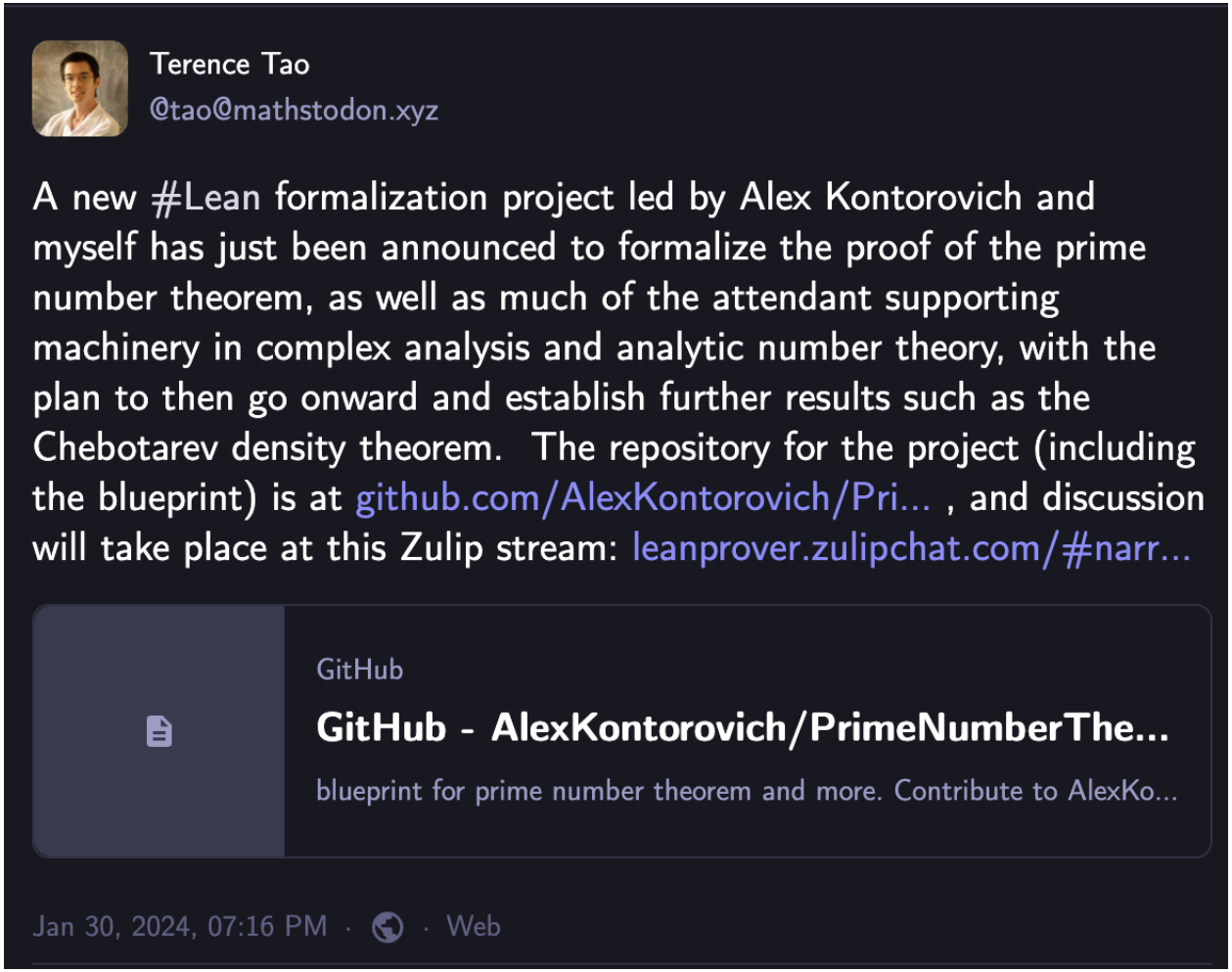
Big Proof

- Let's rewind the clock to summer 2025:
  Deepmind's Silver Medal at IMO was the state-of-the-art
- In July, announced at Big Proof at the Newton Institute
  a protype of the **Gauss** agent formalized de Bruijn's *abc* Theorem
- Generated 3.5K lines of Lean code in 3 weeks
- Human–machine collaboration:
  Intense informal LaTex blueprinting with human (me!)



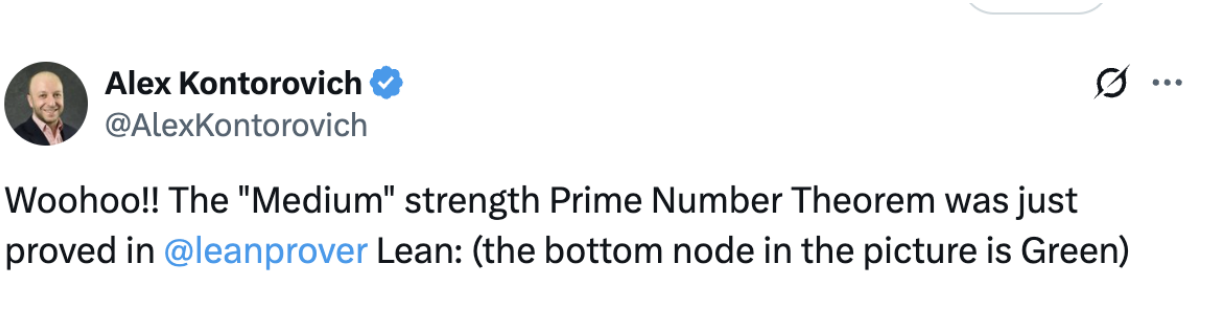
The Prime Number Theorem

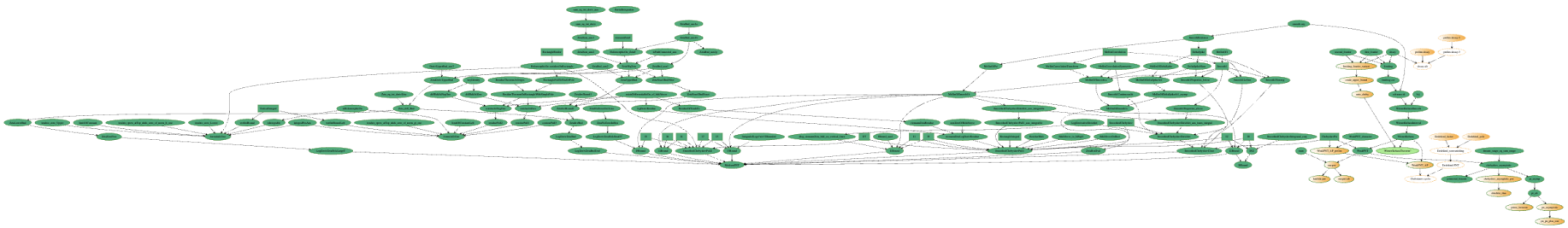
- In January 2024, Terence Tao and Alex Kontorovich announced a project to formalize the PNT in Lean



Medium PNT

- In July 2025, after 18 months, Kontorovich and Tao announced

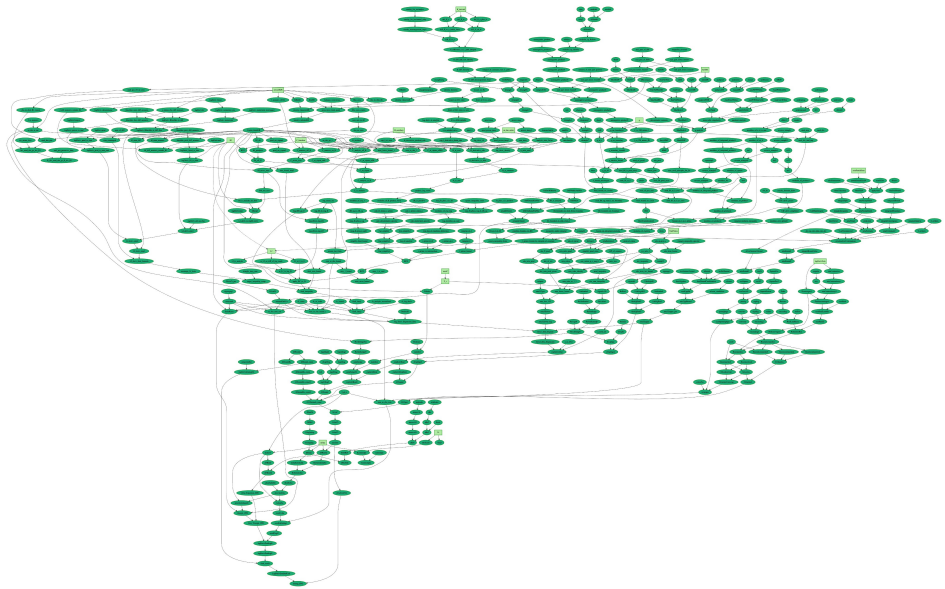




- The 'Medium-strength' PNT, with error $O(x \exp(-c(\log x)^{1/10}))$, was a natural stopping point
- The Medium PNT was useful for applications, but avoided a central difficulty in complex analysis: the zero-free region of $\zeta(s)$

Strong PNT

- In September 2025, we announced Gauss formalized $\zeta(s)$ zero-free region, along with fundamental results in complex analysis
- Intense collaboration with blueprinting in LaTex with human
- Gauss generated 25K lines of Lean code, which combined with Mathlib & PNT+ to complete Strong PNT



https://github.com/math-inc/strongpnt

Sphere Packing

- In May 2024, Maryna Viazovska and Sid Hariharan announced a Lean project to formalize Viazovska's 2022 Fields Medal proof:

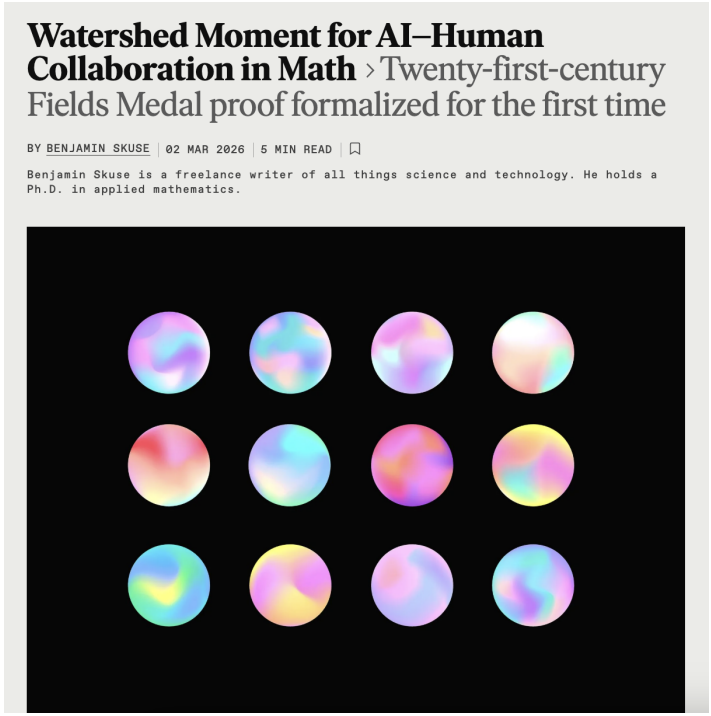
  Optimal sphere packing in dimension 8 for the E_8 lattice
- In July 2025 at Big Proof, she presented progress and ongoing challenges
- By November 2025, the team had made partial progress and welcomed contributions from several companies and community members
- In particular, with Auguste Poiroux (also a PhD student at EPFL) we ran **Gauss** on the repo, and produced several (minor) PRs to the project
- Then in February 2026, **Gauss** produced 50K lines of code to our surprise, completely finishing the rest of the proof in 5 days.

Sphere Packing

- A natural question occurred to me:
  what about Sphere Packing in dimension 24?

Sphere Packing

- A natural question occurred to me:
  what about Sphere Packing in dimension 24?
- Viazovska et al. had also proven optimal sphere packing in dimension 24, by the Leech lattice.
- In another week, **Gauss** produced a complete proof in 500K lines, taking the Annals paper and 20+ arXiv papers from its bibliography



Watershed Moment for AI–Human Collaboration in Math › Twenty-first-century Fields Medal proof formalized for the first time

BY BENJAMIN SKUSE | 02 MAR 2026 | 5 MIN READ |

Benjamin Skuse is a freelance writer of all things science and technology. He holds a Ph.D. in applied mathematics.

https://github.com/math-inc/Sphere-Packing-Lean

Sphere Packing

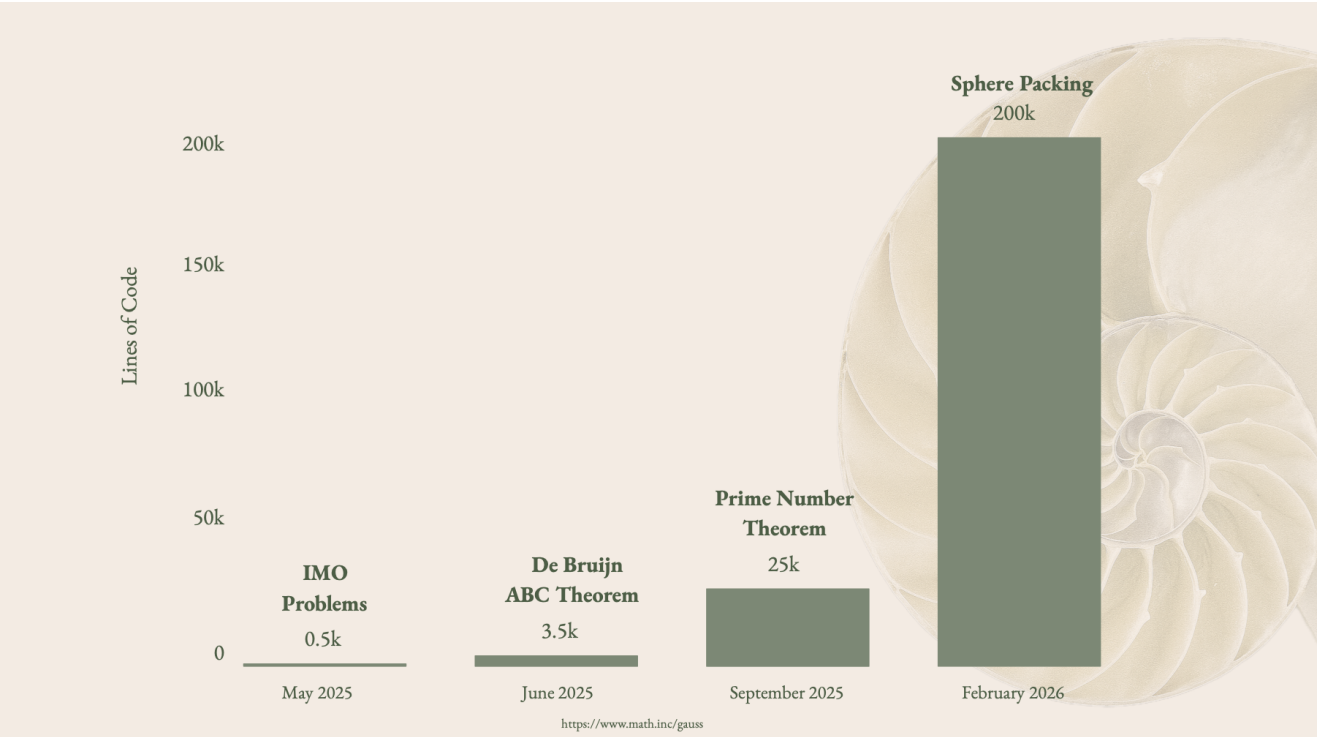
- Previous efforts at this scale would take large teams multiple years
  - Notably, for sphere packing in dimension 3:
    the historic formalization of Hales proof of Kepler's conjecture
  - Flyspeck was 500K lines of code (HOL Light, Isabelle)
  - tour-de-force effort over a decade by team of human experts



- Afterwards, **Gauss** analyzed the dimension 24 proof and found much of the formalized material was not directly needed in the proof
- With further golfing, the dimension 24 proof is down to 200K lines

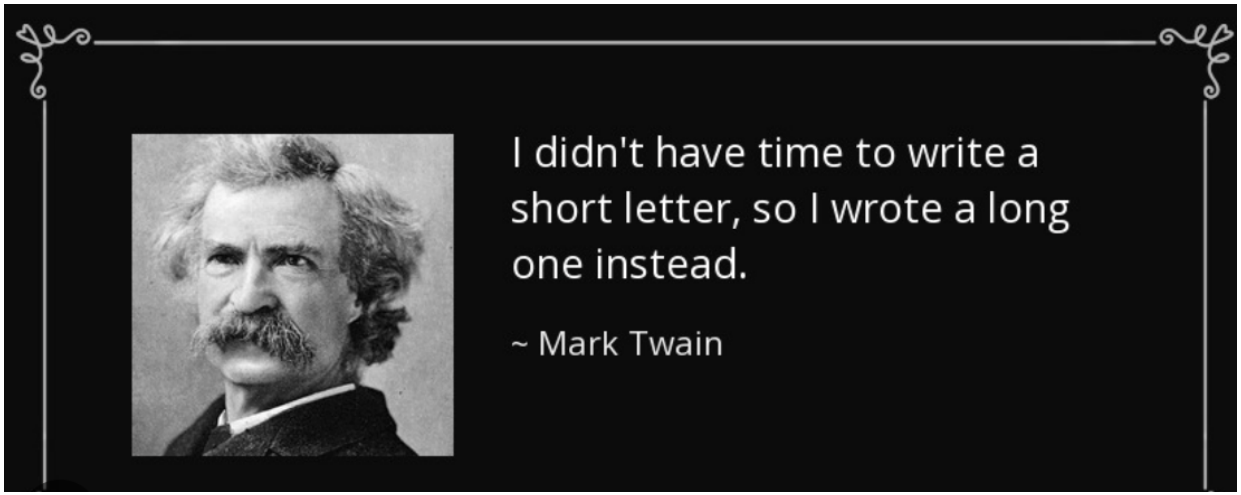
Code Expansion & Compression

- Formal code generation has scaled 1000× in the past year
- Mathlib is 2M lines of code, built carefully over 8+ years
- Code quality will become crucial for scaling up formal libraries
- Compression, refactor, golfing, etc. is also improving



Gradual development

- Historically, mathematics develops gradually over time
- It is a decentralized process across (typically independent) subfields
- The artifacts that are produced mature over decades & centuries
  — notes & preprints
  — journal publications
  — textbooks & course curricula (stable)



Formal libraries

- The objective of Mathlib is to produce an optimized codebase, for compactness and modularity
- produced in 'one-shot' by careful forethought, based on expert knowledge & prior history of the field
- This may be thought of as transcribing the polished final form, rather that reproducing the full history.
- As such it aims to be the idealized 'mature textbook' in form
- However, the codebase is currently limited in coverage, constrained by human time and expertise.



Fermat

- In the period since I last wrote this talk, there have been some remarkable new updates

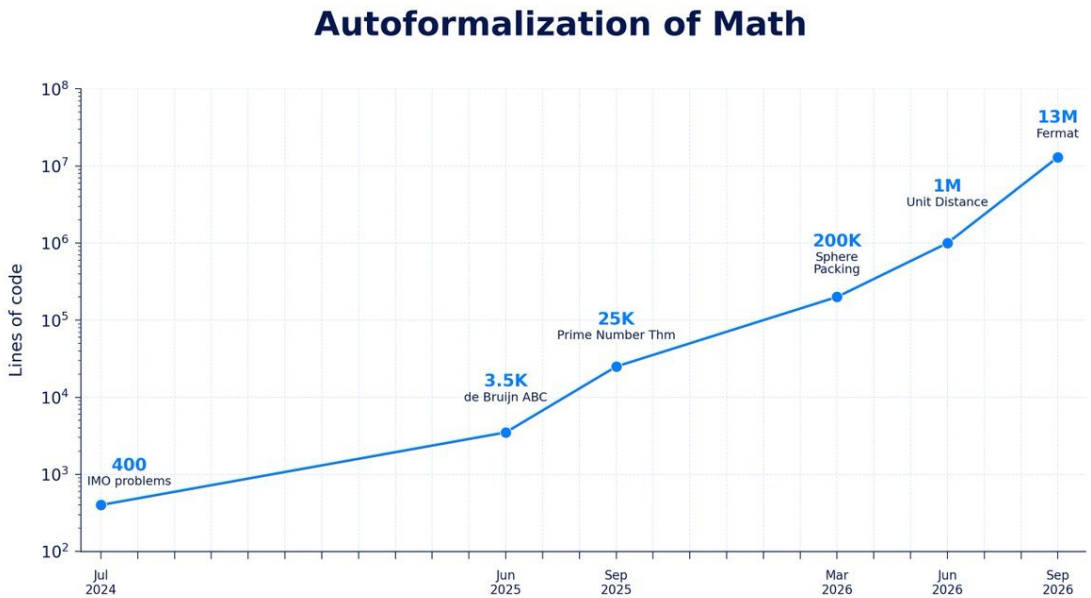
Fermat

- In the period since I last wrote this talk, there have been some remarkable new updates
- Recently, Kevin Buzzard launched a project to formalize Fermat's Last Theorem.
  This was long viewed as a pipe dream, and Buzzard had a £1M grant to do so over 5 years.
- Even so, this would only scratch the surface, since the Fermat proof also depends on diverse math literature from the 20th century.



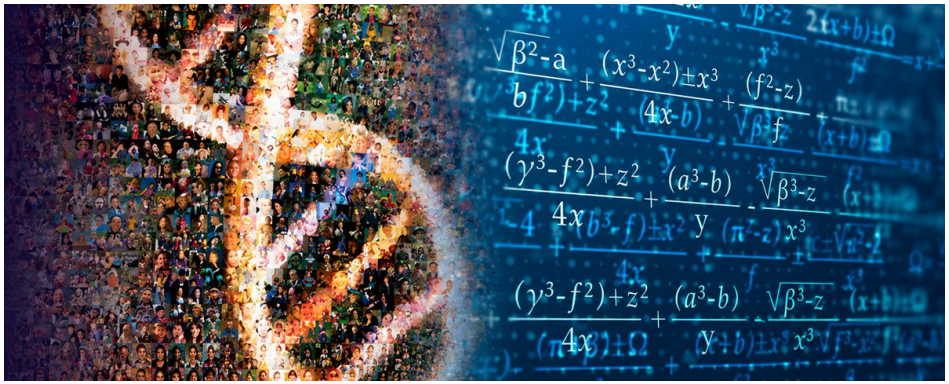
Fermat et Claude

- On September 4th, Anthropic released the formalization of Fermat's Last Theorem.
- Moreover, in just 11 days, Claude wrote 13 million lines of code to autoformalize the Fermat proof, along with ALL the pre-requisite literature.
- For Buzzard, this now unlocks the chance to formalize the literature the Langlands program, which he is concerned might contain subtle errors.



Announcing the MAP

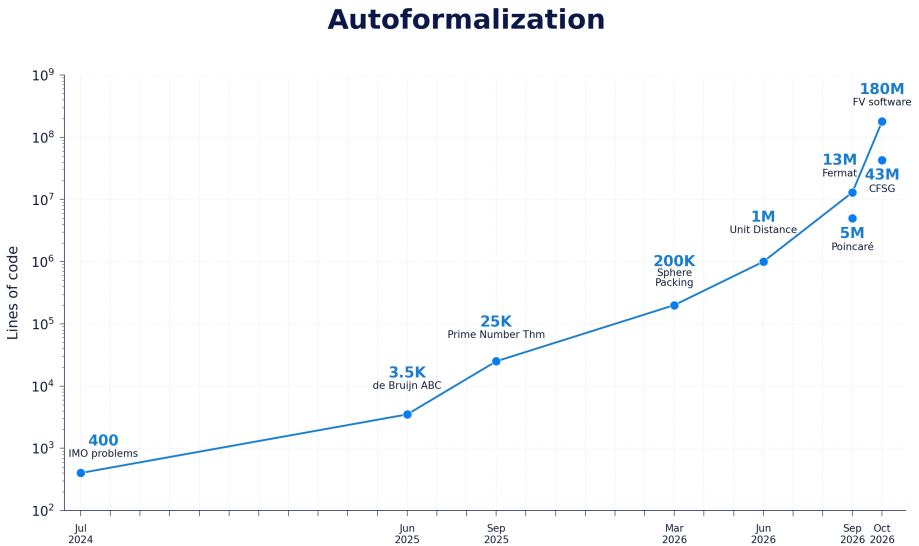
Announcing the Mathematics Autoformalization Project:



- We are now poised to formalize all known mathematics
- Just as AlphaFold tabulated all 200 million proteins known to humanity, the MAP is aimed to do the equivalent in math, in a comprehensive formal database.

The MAP

- Unlike the replication crisis which plagues other sciences, the formalization process can flag errors in the literature, which have occasionally destabilized entire research programs historically.
- We are already seeing small examples of this in recent AI-for-math results (e.g. non-Sofic groups)
- In the time since writing slides last, new formalizations have just occurred: the Poincaré conjecture, the classification of finite simple groups (43M), and Claude's FV of software in Rocq (180M).



The MAP

- Rough estimates on the total mathematical corpus is on the order of 10B lines of code, based Claude's current level of efficiency (to say nothing of future efficiency gains).
- We are now poised to achieve this in the next year, coordinating with academia, industry, government, and philanthropy.
- Once we map out the known territory, since math is cumulative by nature we have a firm base to then set out and explore the mathematical frontier.



Mathematics as a knowledge base

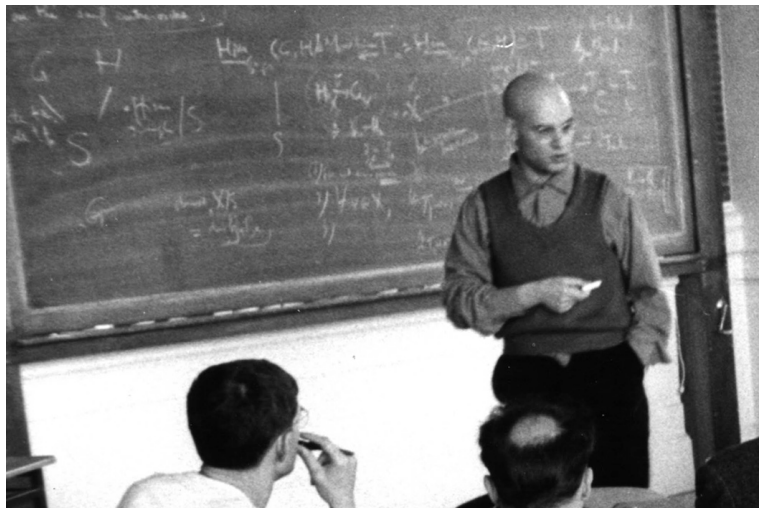
In the remaining time, I would like to discuss emerging views on the structure mathematics.

- Knowledge base accumulated over human history
  each completed proof is permanent
  becomes available for future use
- formalization converts into a compiling formal database
  - (hyper)graph of theorems, definitions, and conjectures
  - edges encode use, logical implication



Dual objectives

- Expansion & Compression
  - in tension, also reinforce each other
- Expansion adds new theorems to the database
  - particularly those influential to the current database
    e.g. the Riemann hypothesis
- Compression synthesizes and refactors the codebase,
  - simple tactics for basic efficiency
  - deep definitions to unify entire domains,
    e.g. Grothendieck



Triad of inferences

The logician and philosopher Charles Peirce categorized (1934) inference based on the structural permutation of a **Rule** (function definition), a **Case** (input), and a **Result** (return value):

- **Deduction** (Rule, Case $\rightarrow$ Result) is the analytic application of a Rule to a Case to predict a Result. It is the only mode that guarantees truth (e.g., executing code to verify output).
- **Induction** (Case, Result $\rightarrow$ Rule) is the synthetic derivation of a Rule from the accumulation of Cases and Results. It validates hypotheses through statistical frequency (e.g., generating a function to satisfy unit tests).
- **Abduction** (Rule, Result $\rightarrow$ Case) is the inference of a Case (or a new Rule) to explain a surprising Result.

The structure of mathematics

- Expansion & Compression are intrinsic to the structure of mathematics itself
- Triad of inferences (Peirce)
  - induction: (stating/conjecturing) retrospective compression
  - deduction: (proving) hierarchical decompression
  - abduction: (defining) representational expansion for prospective compression
- Ground inferences on compressive foundation

Compression is all you will have needed