AI & PROOFS

Flying Nobita

Feb 6, 2026

ABOUT ME

Independent ZK Research Dev that works on ZK projects from Ethereum and its ecosystem.

Help run local communities in SG:

- ProgCryptoSG
- SoliditySG

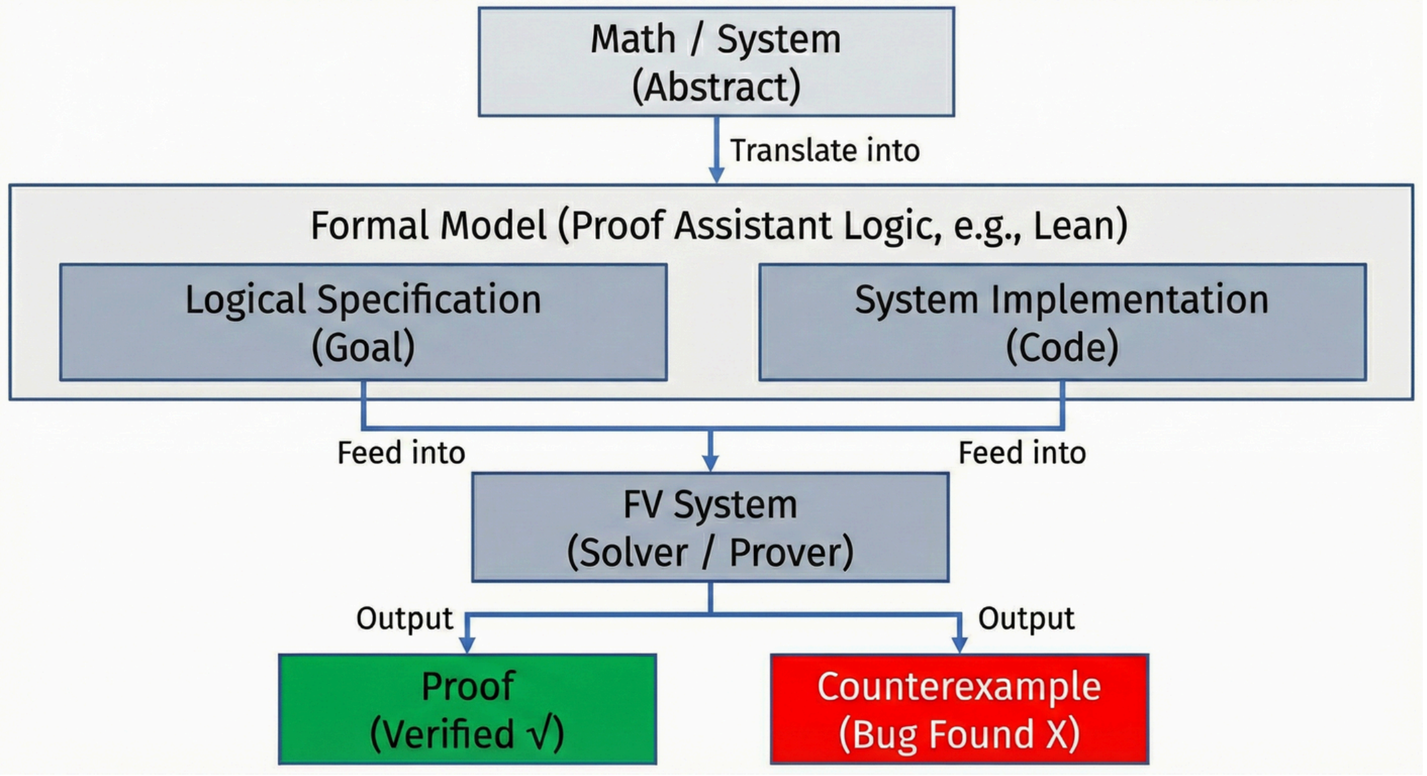
STORY

I went looking for tools to understand some mathematical concepts. I found a fascinating intersection of code and math. Here is my take on exploring AI-assisted proofs with an engineer's mindset.

FORMAL VERIFICATION AND THEOREM PROVERS

Formal verification is a process of proving that a system satisfies its specification.

1. Write the system's specification in a proof assistant's logic (e.g. Lean)
2. Write the system's implementation also in Lean
3. Feed both into the FV System, which tries to prove the specification always holds for the implementation



THEOREM PROVERS

A language that allows one to write formal proofs.

Theorem provers prevent AI hallucinations by using failed compilations as the ground truth.

Examples:
- Lean
- Coq
- Isabelle/HOL
- Acorn

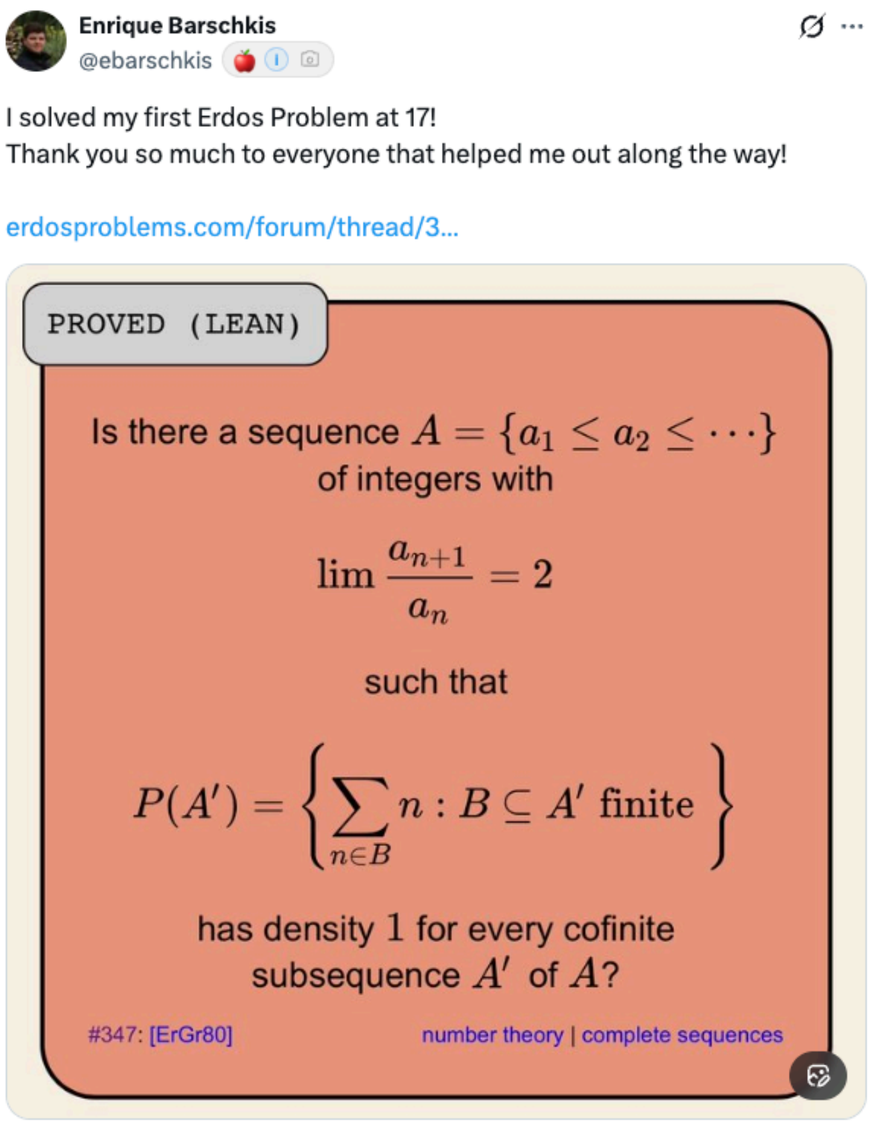
AI PROOF SYSTEM

An evolution of AI + Theorem Provers.

Submit math ideas in English and it formalizes them in Lean

Examples:
- Aristotle by Harmonic
- AlphaProof by Google DeepMind

AI ASSISTED PROVING IS HERE

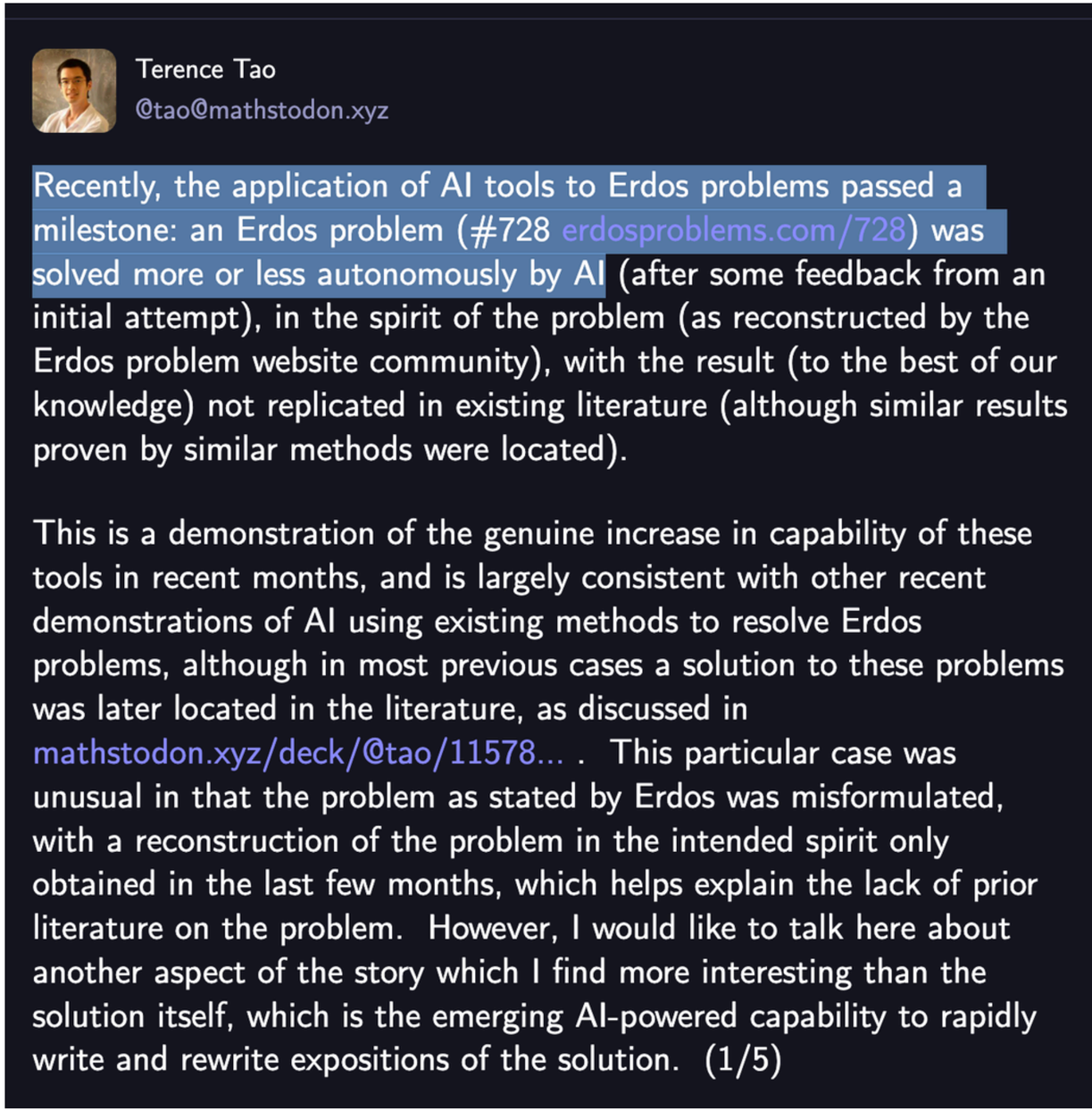


A 17 y/o used AI for help to solve an Erdos problem "in the breaks between high-school classes".

VIBE PROVING ANYONE?

We all know about vibe coding, but what about **vibe proofing**?

Kevin Barreto, a math undergrad, used AI to solve an Erdős problem without knowing much about it.



Terence Tao
@tao@mathstodon.xyz

Recently, the application of AI tools to Erdos problems passed a milestone: an Erdos problem (#728 erdosproblems.com/728) was solved more or less autonomously by AI (after some feedback from an initial attempt), in the spirit of the problem (as reconstructed by the Erdos problem website community), with the result (to the best of our knowledge) not replicated in existing literature (although similar results proven by similar methods were located).

This is a demonstration of the genuine increase in capability of these tools in recent months, and is largely consistent with other recent demonstrations of AI using existing methods to resolve Erdos problems, although in most previous cases a solution to these problems was later located in the literature, as discussed in mathstodon.xyz/deck/@tao/11578... . This particular case was unusual in that the problem as stated by Erdos was misformulated, with a reconstruction of the problem in the intended spirit only obtained in the last few months, which helps explain the lack of prior literature on the problem. However, I would like to talk here about another aspect of the story which I find more interesting than the solution itself, which is the emerging AI-powered capability to rapidly write and rewrite expositions of the solution. (1/5)

HOW TO START?
WITHOUT (TOO MUCH) AI SLOP?

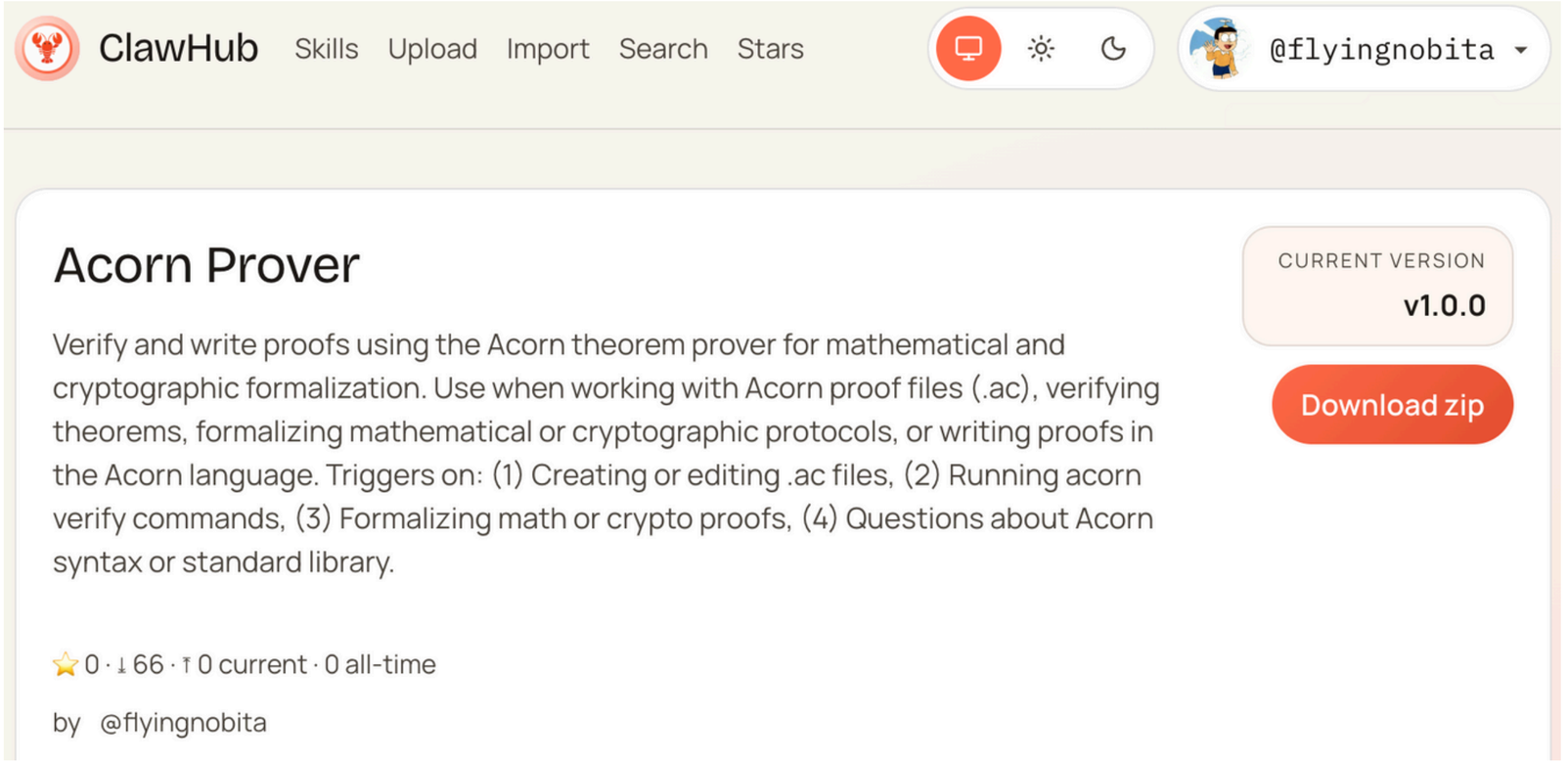
1. Upload the paper to LLM
2. Summarize the theorem in plain English
3. Ask LLM to write the formal proof in Lean
4. Compile Lean code and get error (hints)
5. Fix code based on the errors
6. Repeat 4-5 until proven

ACORN

- Provides a friendlier syntax than Lean.
- has an Assistant that gives you real-time feedback
- still in beta, with its math libraries still being built

OTHER CHANNEL (TREAD CAREFULLY)

You can prove maths in Telegram:



https://clawhub.ai/flyingnobita/acorn-prover

TAKEAWAYS

1. Use AI to explain hard maths (obvious)
2. For best results, use ChatGPT 5.2 or Google Deep Thinking
3. Try to prove an unsolved Erdos problem for **fun**

QUESTIONS?

THANK YOU!