

Terence Tao

@tao@mathstodon.xyz

In a day or so, the mathematicians behind the [#1stproof](#) challenge at [1stproof.org/](#) will reveal their solutions to the 10 challenge problems they posted recently. (I am not directly involved in this challenge, although I know most of the authors personally and approve of their experiment.) It seems likely that there will be many claims, both trustworthy and dubious, of proofs of these problems by various AI-generated means.

The Erdos problem web site, having dealt with this type of thing for several months now, has come up with several guidances on how to increase confidence in the correctness of an AI-generated proof: [github.com/teorth/erdosproblem...](#) The wording there is specific to Erdos problems, but much of the advice can be applied more broadly.

I would like to highlight in particular the additional correctness guarantees provided by formalizing the argument in Lean. When used correctly, a Lean formalization of a proof can provide extremely high confidence that a given proof correctly proves the desired claim. However, if the Lean proof is itself AI-generated without supervision from an expert in Lean, there are still ways in which a supposed "Lean certificate" of correctness is unsatisfactory or even worthless. These include:

1. A Lean proof that adds additional axioms in the proof beyond the standard three, or which relies on malicious metaprogramming.

2. Subtle errors in the formalization of the *\*statement\** of the result to be proved, that allows the claim to be proven on a technicality. (This is a particular risk if this statement formalization is also AI-generated.)

See [leanprover-community.github.io...](#) and [lean-lang.org/doc/reference/la...](#) for best practices on guarding against such issues.

1stproof.org

📄

First Proof | Research-Level Math for...

A set of research-level mathematics questions to assess ...

Feb 12, 2026, 07:35 PM · 🌐 · Web

Last edited Feb 12, 07:35 PM

14 boosts · 1 quote · 39 favorites

↶

↺

☆

🔖

⋮

Aditya Khanna

@adityakhanna

🌐 18h

@tao whoa I am really curious about these metaprogramming techniques. Is there a simple example?

↶ 1

↺

☆

🔖

⋮

Terence Tao

@tao

🌐 16h...\*

@adityakhanna One example is given at [leanprover.zulipchat.com/#narr...](#)

The issue is not with Lean per se, but rather with various tricks used to significantly speed up Lean typechecking, for instance by trusting that precompiled Lean files are sound and not tampered with, and that the native kernel can be trusted to perform certain types of unfolding of Lean types. These are not an issue for non-adversarially written code, but can be exploited by malicious use of the more advanced metaprogramming features of Lean.

Zulip

Public view of Lean | Zulip tea...

Browse the publicly accessible channels in Lean ...

↶ 0

↺

☆

🔖

⋮

marilyn

@marilyn

🌙 9h

@tao  
Very curious to see the results of this challenge 🍿

↶ 0

↺

☆

🔖

⋮

thilakan

@thilakan

🌐 10h

@tao I just watched Bubeck’s IPAM talk on GPT-5 coming up with a proof of a nontrivial extremal combinatorics problem. He talks about having designed a specific harness for the AI to work long term and that producing several thousands of pages of intermediate steps that could possibly contain novel ideas. What are your thoughts on this experiment?

🔍 Search

🌐

Live feeds

📘

About

Mastodon is the best way to keep up with what's happening.

Follow anyone across the fediverse and see it all in chronological order. No algorithms, ads, or clickbait in sight.

Create account

Login



 Live feeds

 About

**Mastodon is the best way to keep up with what's happening.**

Follow anyone across the fediverse and see it all in chronological order. No algorithms, ads, or clickbait in sight.

Create account

Login