Trusting AI with the Lean Theorem Prover

Sebastian Ullrich
Head of Engineering, Lean FRO

NVIDIA, Nov 17, 2025

How can we ensure confidence in statements made by humans or AI, and that every proof of correctness is independently verifiable?

Lean is a Development Environment for formal verification

Mathlib > RingTheory > Finiteness.lean

```lean
theorem FG.stabilizes_of_iSup_eq {M' : Submodule R M} (hM' : M'.FG) (N : ℕ →o Submodule R M)
    (H : iSup N = M') : ∃ n, M' = N n := by
  obtain ⟨S, hS⟩ := hM'
  have : ∀ s : S, ∃ n, (s : M) ∈ N n := fun s =>
    (Submodule.mem_iSup_of_chain N s).mp
      (by
        rw [H, ← hS]
        exact Submodule.subset_span s.2)
  choose f hf using this
  use S.attach.sup f
  apply le_antisymm
  · conv_lhs => rw [← hS]
    rw [Submodule.span_le]
    intro s hs
    exact N.2 (Finset.le_sup <| S.mem_attach ⟨s, hs⟩) (hf _)
  · rw [← H]
    exact le_iSup _ _
```

▼Finiteness.lean:365:2

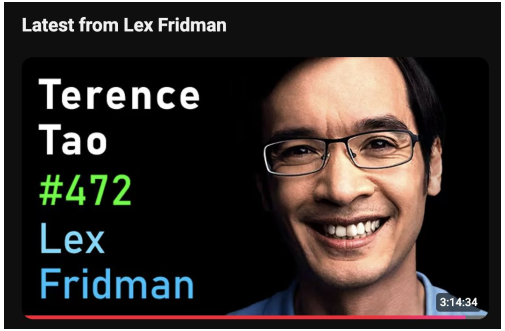
▼Tactic state

1 goal

```
▼case intro
R : Type u_1
M : Type u_2
inst✝² : Semiring R
inst✝¹ : AddCommMonoid M
inst✝ : Module R M
M' : Submodule R M
N : ℕ →o Submodule R M
H : iSup ⇑N = M'
S : Finset M
hS : span R ↑S = M'
f : { x // x ∈ S } → ℕ
hf : ∀ (s : { x // x ∈ S }), ↑s ∈ N (f s)
⊢ ∃ n, M' = N n
```

Lean is Taking Mathematics by Storm

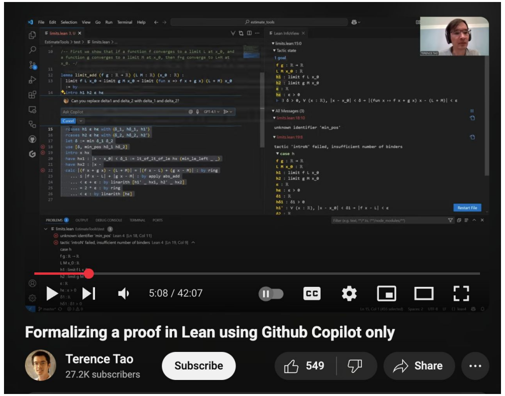
*"**Lean enables large-scale collaboration** by allowing mathematicians to break down complex proofs into smaller, verifiable components. This formalization process ensures the correctness of proofs and facilitates contributions from a broader community. **With Lean, we are beginning to see how AI can accelerate the formalization of mathematics, opening up new possibilities for research.**"* — *Terence Tao*



Fermat's Last Theorem – Kevin Buzzard

Carleson's Theorem – Floris van Doorn

…and many more!



Lean is Taking *Math+AI* by Storm

17 out of 29 *AI for Math Fund* winners reference Lean

Renaissance Philanthropy and XTX Markets Launch New $9 million AI for Math Fund

Renaissance Philanthropy | [XTX] MARKETS

2025 Winners

- An AI-Focused Tactic for Language Learning — Explore project
- A Dataset of Modern Formalized Theorem Statements — Explore project
- A Principled Approach to Proof Search with Applications to Siderenko's Conjecture — Explore project
- A Structured Representation of Tactics for Machine-Assisted Theorem Proving — Explore project
- Bridging AI, Proof Assistants, and Mathematical Data (BRIDGE) — Explore project
- Bridging Complexity and Automation to Advance Automatic Theorem Proving — Explore project
- Bridging Proof and Computation — Explore project
- Constraining LLMs for Theorem Proving — Explore project
- Copilots for Isabelle — Explore project
- Crowdsourcing and Reinventing the Next Generation of Dynamic and Scalable Math Benchmarks — Explore project
- Databases of Structured Motivated Proofs — Explore project
- DEEPER — Explore project
- Document-Level
- Domain Specific
- Game Over or QED?
- GNN-SMT

renaissancephilanthropy.org/ai-for-math-fund-projects

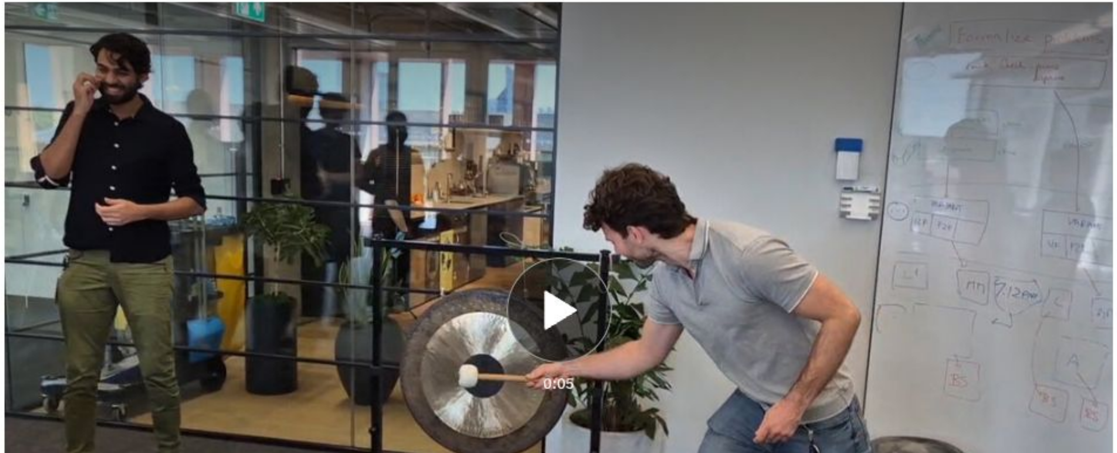
The New York Times

Artificial Intelligence › | A.I.'s Math Problem | A.I. Training Data Disappears | Microsoft's Risk-Taker | Fine Print Changes | Quiz: Fake or Real Images?

Move Over, Mathematicians, Here Comes AlphaProof

A.I. is getting good at math — and might soon make a worthy collaborator for humans.

Share full article | 47



AlphaProof & the International Math Olympiad

Determine all real numbers α such that, for every positive integer n, the integer

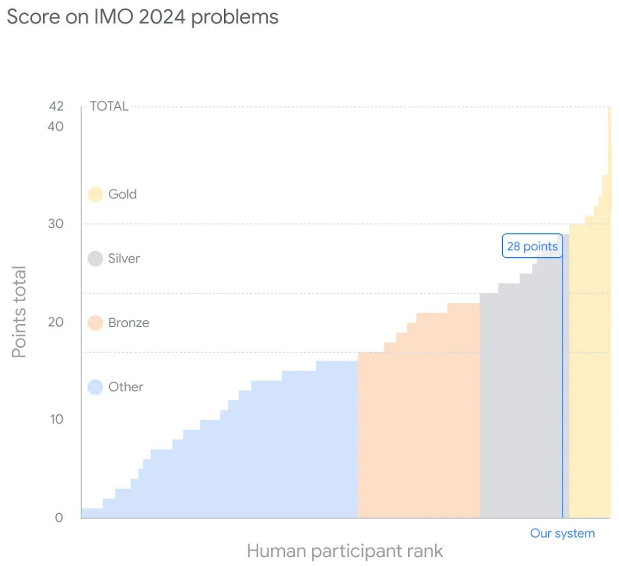
$$\lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is a multiple of n. (Note that $\lfloor z \rfloor$ denotes the greatest integer less than or equal to z. For example, $\lfloor -\pi \rfloor = -4$ and $\lfloor 2 \rfloor = \lfloor 2.9 \rfloor = 2$.)

Solution: α is an even integer.

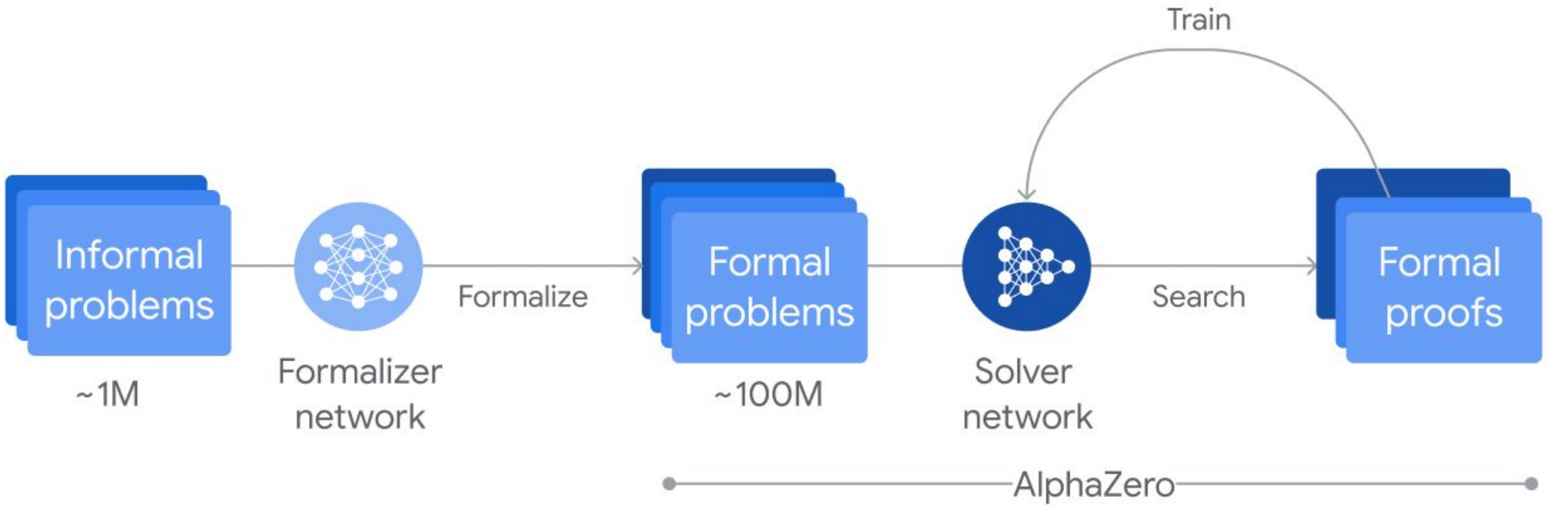
```
open scoped BigOperators

theorem imo_2024_p1 :
    {(α : ℝ) | ∀ (n : ℕ), 0 < n → (n : ℤ) | (∑ i in Finset.Icc 1 n, ⌊i * α⌋)}
    = {α : ℝ | ∃ k : ℤ, Even k ∧ α = k} := by
  rw [(Set.Subset.antisymm_iff ), (Set.subset_def), ]
  /- We introduce a variable that will be used
     in the second part of the proof (the hard direction)
```



deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level

Learning à la DeepMind



deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level

Learning à la DeepMind

Article | Published: 12 November 2025

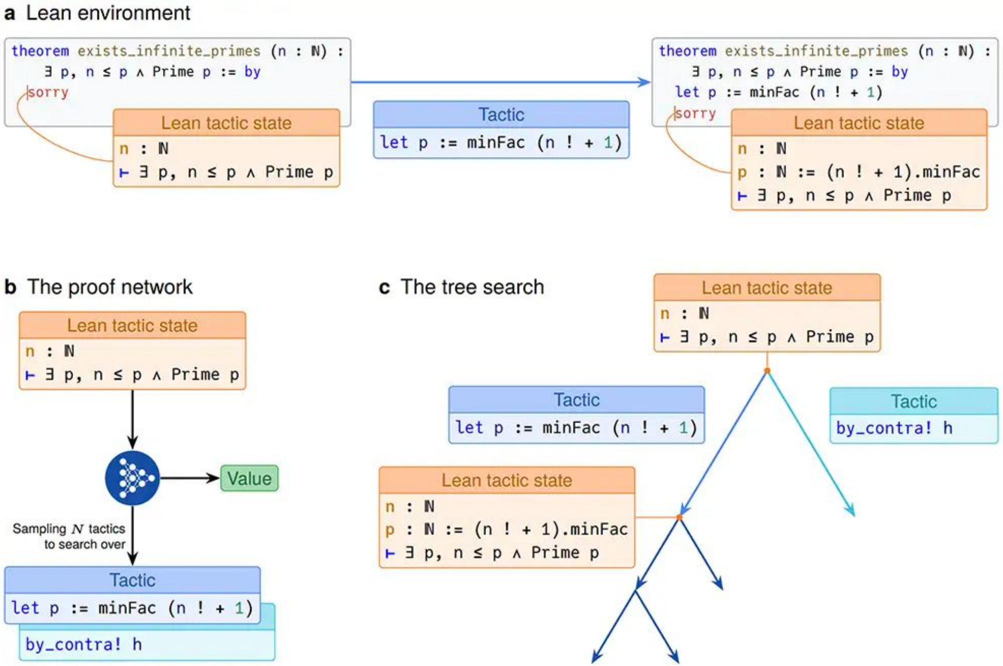
Olympiad-level formal mathematical reasoning with reinforcement learning

Thomas Hubert ✉, Rishi Mehta, Laurent Sartran, Miklós Z. Horváth, Goran Žužić, Eric Wieser ✉, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, Ottavia Bertolli, Tom Zahavy, Amol Mandhane, Jessica Yung, Iuliya Beloshapka, Borja Ibarz, Vivek Veeriah, Lei Yu, Oliver Nash, Paul Lezeau, Salvatore Mercuri, Calle Sönne, Bhavik Mehta, Alex Davies, Daniel Zheng, Fabian Pedregosa, Yin Li, Ingrid von Glehn, Mark Rowland, Samuel Albanie, Ameya Velingker, Simon Schmitt, Edward Lockhart, Edward Hughes, Henryk Michalewski, Nicolas Sonnerat, Demis Hassabis, Pushmeet Kohli & David Silver — Show fewer authors

Nature (2025) | Cite this article

nature.com/articles/s41586-025-09833-y

Learning à la DeepMind



julian.ac/blog/2025/11/13/alphaproof-paper

IMO 2025: 3 Gold, 1 Silver

OpenAI (informal)

Alexander Wei
@alexwei_

1/N I'm excited to share that our latest @OpenAI experimental reasoning LLM has achieved a longstanding grand challenge in AI: gold medal-level performance on the world's most prestigious math competition—the International Math Olympiad (IMO).

DeepMind (informal)

Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

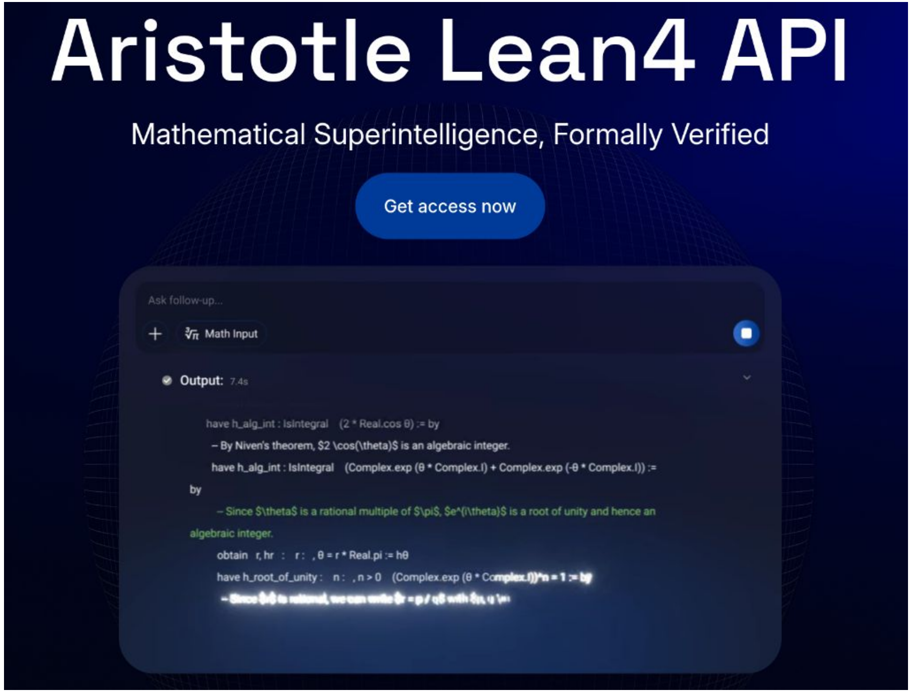
Harmonic (Lean)

Aristotle Achieves Gold Medal-Level Performance at the International Mathematical Olympiad, iOS App Beta Launch

ByteDance (Lean)

ByteDance Seed Prover Achieves Silver Medal Score in IMO 2025

IMO-Level Proof Search as a Service



AlphaProof Interest Form

Thanks for your interest in AlphaProof. To deliver a reliable user experience, we are carefully managing our compute resources to maintain system stability. We will be reviewing applications and prioritizing access over time as we scale the system. We appreciate your patience and understanding.

To help you get the most from this process, we encourage you to first explore the AlphaProof Blog Post, the AlphaProof Nature Paper, and the **AlphaProof Tool Demos [1][2]**. Understanding AlphaProof beforehand will empower you to complete this form more effectively and confidently.

aristotle.harmonic.fun

deepmind.google/alphaproof

Highly Active, Healthy Competition on Putnam Bench

Lean (out of 660)

#	Model	num-solved	compute
1	Hilbert	462	avg pass@1840
2	Seed-Prover	329	MEDIUM
3	Ax-Prover💙	91	pass@1, avg. 100 tool calls
4	Goedel-Prover-V2💚	86	pass@184
5	DeepSeek-Prover-V2💚	47	pass@1024
6	DSP+💚	23	pass@128
7	Bourbaki💚	14	pass@512
8	Kimina-Prover-7B-Distill💚	10	pass@192
9	Self-play Theorem Prover💚	8	pass@3200
10	Goedel-Prover-SFT💚	7	pass@512

UC San Diego/Apple, Sep 2025
ByteDance, Jul 2025
Axiomatic_AI et al, Oct 2025
Princeton/NVIDIA et al, Aug 2025
DeepSeek-AI, Jul 2025
Microsoft Research et al, Jun 2025
Huawei et al, Jul 2025
Numina/Kimi, Apr 2025
Stanford, Mar 2025
Princeton et al, Apr 2025

💚 fully open-sourced
💙 partially open-sourced

trishullab.github.io/PutnamBench/leaderboard.html

Startups Mentioning Lean+AI

axiommath.ai

caj.al

harmonic.fun

logicalintelligence.com

morph.so + math.inc

…and more to come

Extensibility and Introspection

"At Google DeepMind, we used Lean to build AlphaProof, a new reinforcement-learning based system for formal math reasoning. ***Lean's extensibility and verification capabilities were key in enabling the development of AlphaProof****."* — *Pushmeet Kohli, Vice President, Research Google DeepMind*

Lean 4 is implemented in Lean, allowing for unprecedented extension and introspection by users.

leanprover-community/repl is a simple Lean program for communicating with the Lean compiler via JSON.

AI labs have customized it in many ways, especially around parallelized proof tree search.

stanford-centaur/PyPantograph is the state of the art for open-source Lean REPLs.

Autoformalization

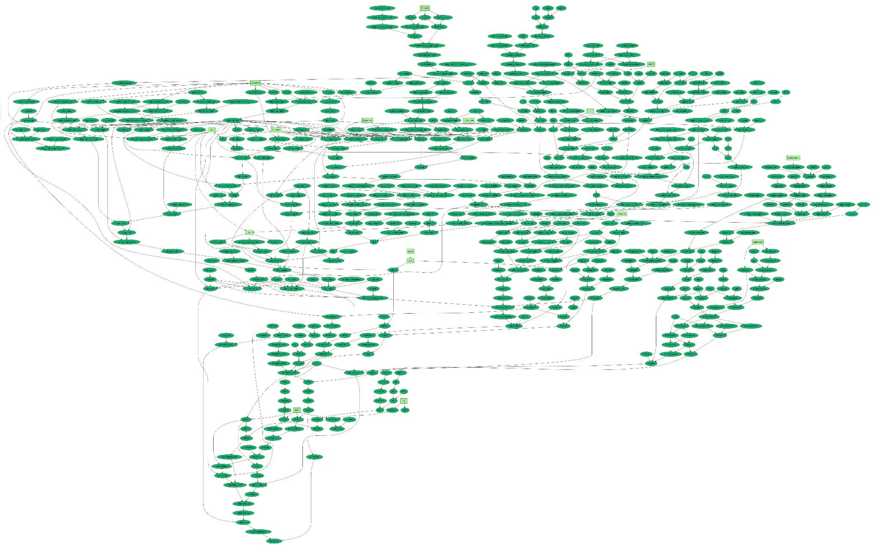
Auguste Poiroux

Dear all,

We, at Math, Inc., are excited to introduce Gauss, a new autoformalization agent designed to assist mathematicians in their Lean formalization.

Using Gauss, and building over the recent medium PNT progress by Kontorovich et al., we formalized the **strong Prime Number Theorem** in Lean. Over the course of three weeks, Gauss produced ~25k lines of Lean code and over 1,000 theorems and definitions. To put these numbers in perspective, our previous iteration, three months ago, produced ~3.5k lines of Lean code and 100 theorems and definitions to formalize the abc conjecture almost always.

Our goal is to make Gauss accessible and easy to use. If you are interested, you can register for early access to Gauss here. We are happy to discuss about Gauss and Math, Inc. in the discussion thread.



github.com/math-inc/strongpnt

Autoformalization Inspection (Demo)

```
  have h := zetacnot0 t
  exact norm_pos_iff.mpr h

lemma Zeta1_Zeta_Expand :
    ∃ A > 1, ∃ b > 1,
    ∀ (t : ℝ) (ht : |t| > 2)
    (r1 r R1 R : ℝ)
    (hr1_pos : 0 < r1) (hr1_lt_r : r1 < r)
    (hr_pos : 0 < r) (hr_lt_R1 : r < R1) (hR1_pos : 0 < R1) (hR1_lt_
    let c := (3/2 : ℂ) + I * t;
    ∀ (hfin : (zerosetKfRc R1 c riemannZeta).Finite),
    ∀ z ∈ closedBall c r1 \ zerosetKfRc R1 c riemannZeta,
    ‖logDerivZeta z - ∑ p ∈ hfin.toFinset,
      ((analyticOrderAt riemannZeta p).toNat : ℂ) / (z - p)‖ ≤
      (16 * r^2 / ((r - r1)^3) +
    1 / ((R^2 / R1 - R1) * Real.log (R / R1))) * (Real.log |t| + Real
  -- Apply the three lemmas mentioned in the informal proof
  obtain ⟨b, hbgt1, hb⟩ := zeta32upper
  obtain ⟨A, hAgt1, hA⟩ := zeta32lower_log

  -- Provide the constants A, b as required
  refine ⟨A, hAgt1, b, hbgt1, ?_⟩
  intro t ht r1 r R1 R hr1_pos hr1_lt_r hr_pos hr_lt_R1 hR1_pos hR1_
  intro c hfin z hz

  -- Apply log_Deriv_Expansion_Zeta
  have hexp_lemma := log_Deriv_Expansion_Zeta t ht r1 r R1 R hr1_pos

  -- Set B = b * |t| as mentioned in informal proof
  have htpos : (0 : ℝ) < |t| := by linarith [ht]
  have hBgt1 : b * |t| > 1 := by
    have hb_pos : (0 : ℝ) < b := by linarith [hbgt1]
```

Mathematical significance:

This is likely part of a proof related to the Prime Number Theorem, where you need precise control over how well the logarithmic derivative of the zeta function can be approximated by summing over its zeros. The bound includes both geometric terms (involving the radii) and growth terms (logarithmic in `|t|`).

The proof strategy mentioned in the comments indicates it combines results from `zeta32upper` and `zeta32lower_log` lemmas to establish the required constants and bounds.

What is the definition of the logarithmic derivative here?

PNT3_RiemannZeta.lean:4428-4440

I'll search for the definition of `logDerivZeta` in this file to show you how the logarithmic derivative is defined here.

Searched text for `logDerivZeta` (`**/StrongPNT/PNT3_RiemannZeta.lean`), 20 results

Perfect! I found the definition. Let me read the context around line 3495 to give you the complete picture:

✓ Read PNT3_RiemannZeta.lean, lines 3490 to 3500

Vibe Proving

Forbidden Sidon subsets of perfect difference sets, featuring a human-assisted proof

Boris Alexeev ChatGPT* Lean† Dustin G. Mixon‡§

Abstract

We resolve a \$1000 Erdős prize problem, complete with formal verification generated by a large language model.

In over a dozen papers, beginning in 1976 and spanning two decades, Paul Erdős repeatedly posed one of his "favourite" conjectures: every finite Sidon set can be extended to a finite perfect difference set. We establish that $\{1, 2, 4, 8, 13\}$ is a counterexample to this conjecture.

During the preparation of this paper, we discovered that although this problem was presumed to be open for half a century, Marshall Hall, Jr. published a different counterexample three decades *before* Erdős first posed the problem. With a healthy skepticism of this apparent oversight, and out of an abundance of caution, we used ChatGPT to vibe code a Lean proof of both Hall's and our counterexamples.

borisalexeev.com/pdf/erdos707.pdf

Autonomous Exploration

MATHEMATICAL EXPLORATION AND DISCOVERY AT SCALE

BOGDAN GEORGIEV, JAVIER GÓMEZ-SERRANO, TERENCE TAO, AND ADAM ZSOLT WAGNER

ABSTRACT. `AlphaEvolve` [223] is a generic evolutionary coding agent that combines the generative capabilities of LLMs with automated evaluation in an iterative evolutionary framework that proposes, tests, and refines algorithmic solutions to challenging scientific and practical problems. In this paper we showcase `AlphaEvolve` as a tool for autonomously discovering novel mathematical constructions and advancing our understanding of long-standing open problems.

1.5. **Building a pipeline of several AI tools.** Even more strikingly, for the finite field Kakeya problem (cf. Problem 6.1), `AlphaEvolve` discovered an interesting general construction. When we fed this programmatic solution to the agent called `Deep Think` [148], it successfully derived a proof of its correctness and a closed-form formula for its size. This proof was then fully formalized in the Lean proof assistant using another AI tool, `AlphaProof` [147]. This workflow, combining pattern discovery (`AlphaEvolve`), symbolic proof generation (`Deep Think`), and formal verification (`AlphaProof`), serves as a concrete example of how specialized AI systems can be integrated. It suggests a future potential methodology where a combination of AI tools can assist in the process of moving from an empirically observed pattern (suggested by the model) to a formally verified mathematical result, fully automated or semi-automated.

arxiv.org/abs/2511.02864

Autonomous Exploration (Inspection Demo)

```lean
-- Inside this block, when we write `(K1 p)`, Lean uses the `p` from the context.
lemma card_K1 (hp_odd : p > 2) : (4 * (K1 p).card : ℤ) = p^3 + 2*p^2 + p := by
    have := (Fact.out : (p.Prime)).odd_of_ne_two fun and' =>by simp_all
    rcases this with ⟨x, hx⟩
    subst hx
    delta K1
    rw_mod_cast[Finset.card_filter]
    trans(4)*∑S:ZMod (2*x+1) ×_ × _,ite (IsSquare (S.1^2+4* S.2.1) ∧IsSquare (S.1^2+4* S.2.2)) (1) 0
    · exact (congr_arg _) (Fintype.sum_equiv ⟨ fun and=> (and 0, and (1), and 2), fun and=>![ _,_, _]
    · trans(4)*∑S:ZMod (2*x+1),(( Finset.univ.image fun and=> (and* and-S^2 :) /4) ×ˢ.image (@ fun ar
      · rw[ Fintype.sum_prod_type, Fintype.sum_congr _ _ fun and=>(Finset.card_filter _ _).symm.trans
        field_simp[IsSquare, sub_eq_iff_eq_add'.trans (comm),show (4 :ZMod (2*x + 1))≠0 from(( (ZMod.:
      · push_cast[sq, add_assoc,.>., Finset.card_product, true, Finset.mul_sum]at *
        trans∑S:ZMod (2*x+1), 4*((( Finset.univ.filter fun and=>and.val≤and.val).image fun p=>(p*p-S
        · push_cast only [le_refl, true, Finset.filter_true_of_mem, implies_true,sq]
        · trans∑S:ZMod (2*x+1), 4*(( Finset.range (x + 1)).image fun p : ℕ=>(p*p-S* S:ZMod (2*x + 1)
          · refine Fintype.sum_congr _ _ fun and=>congr_arg _ ((congr_arg (.^2) (congr_arg _ (Finset.
            use Finset.mem_image.2 (if h: a.val≤x then⟨ _, Finset.mem_range_succ_iff.2 h,by simp_arit
          · rewrite[ Finset.sum_congr rfl fun and x =>(congr_arg _) ((congr_arg) ( ·^2) ( Finset.card
            · simp_all[show (2*x+1)*(4*(x+1)^2) = (2*x+1)^3+ (2 *((2*x+1)*(2*x + 1))+ (2*x + 1))by ri
            · field_simp[show (4 :ZMod (2 *by bound + 1))≠0 from((ZMod.isUnit_iff_coprime _ _).2 (Odd
              use (by valid:).elim (by valid ∘ (ZMod.val_cast_of_lt (by valid)).symm.trans.comp (.▸ Z
```

Prospective: AI+CS as the Next Frontier is Imminent

leanprover/cslib founded, supported by Amazon, Google & U of Southern Denmark

Companies are starting to pivot their math-proven AIs towards program verification

▸ Beyond Math

While the IMO benchmark highlights Aristotle's mathematical ability, the implications reach much further. The same architecture that can solve Olympiad problems also supports:

- Code generation that doesn't require human verification
- Engineering tasks where precision is mission-critical
- Scientific research where progress depends on rigorous mathematical understanding

harmonic.fun/news#blog-post-aristotle-tech-report

Ilya Sergey
@ilyasergey

The proof of the last remaining Lean theorem in our upcoming conference submission has now been completed with the help of AI.

The research community's perception of program verification is about to change irreversibly.

8:32 PM · Nov 13, 2025 · **15.8K** Views

Spec autoformalization + program synthesis + verification as a service

A vibrant community of users at leanprover.zulipchat.com

Machine Learning for Theorem Proving Machine Learning and AI or theorem proving. HOList, AI for…

Standard view | Filter

- MCP Tools for LLMs and Agentic Mathematics
- Building an Autoformalizer on Analysis
- ✓ Executing Conv and Calc in Pantograph
- Dataset to rule them all?
- REPL: automated incremental state reuse across commands
- Better way for tracing tactic states
- Blind Speculation about IMO 2025
- Claude 4 agent in VS Code
- DeepMind and Navier Stokes
- LeanTool feature: Sorry Hammer
- AI tutorial for LEAN beginner
- Proof or Bluff
- Autoformalization of the probabilistic abc-conjecture
- Illusion of Thinking
- Tool for Lean code verify
- Gemini 2.5 Pro 06/05
- Current state of AI for Mathematics and what could come n...
- System card: Claude Opus 4 & Claude Sonnet 4
- DeepSeek-Prover V2
- I was pleasantly surprised by DeepSeek

How to Get Involved

- Participate in community channels
- Spread the word! Active research lives from making connections
- Consider new combinations of agents
- Consider UX improvements to make systems more accessible
- Sponsor our work :)

Lean FRO: Shaping the Future of Lean Development

The Lean Focused Research Organization (FRO) is a non-profit dedicated to Lean's development.

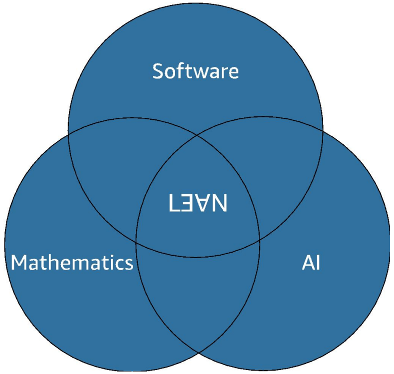
Founded in **August 2023**, the organization has 18 members.

Its mission is to enhance critical areas: **scalability**, **usability**, **documentation**, and **proof automation**.

It must reach **self-sustainability in August 2028** and become the **Lean Foundation**.

We are very grateful for all philanthropic support we have received.

Conclusion



Lean is a development environment for formalized mathematics and program verification.

Lean's rigor allows for *trustless* collaboration between humans and AI

- AI for accelerating Lean development
- Lean for verifying AI output

More research making Math+AI systems stronger and more autonomous arrives weekly.

Thank You

https://leanprover.zulipchat.com/
x: @leanprover
LinkedIn: Lean FRO
Mastodon: @leanprover@functional.cafe
#leanlang, #leanprover

https://www.lean-lang.org/