Nine Became Eight

A general-reader investigation of nine reported AI-generated Erdős proofs that became eight newly resolved problems after one was reclassified as a rediscovery.

DRAFT PREVIEW DENNIS HEDEGREEN INVESTIGATE DRAFT-LOCAL
HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/NINE-BECAME-EIGHT/

Nine proofs were there.

Nine computer-checked proof files. Nine short written proof explanations. Nine entries connected to 353 mathematical problems attempted by an AI system.

Then I looked more closely.

Nine became eight.

Not because the ninth proof was broken. It passed the project's official computer checks.

It left my count because an older published theorem already implied the answer.

Before going further, here are the four numbers that matter:

- `353` is the number of selected Erdős problems the AI system attempted;
- `9` is the number the paper reported as resolved;
- `8` is the number the audit I built with Codex retained as newly resolved under a strict rule; and
- `846` is the catalogue number of one particular Erdős problem. It is not the number of problems in this experiment.

The Erdős catalogue is a numbered collection of problems associated with the Hungarian mathematician Paul Erdős. Its numbers are labels, not difficulty scores.

That last distinction matters because `846` looks like another result. It is really more like a book's reference number: Erdős problem no. 846.

The question I could honestly ask

Mathematics C at HF is my current level. It is a Danish upper-secondary foundation course. The mathematics in problem 846 is far beyond what I can judge through intuition or training.

I am saying that plainly because the article should not borrow authority I do not have.

But I could still investigate a narrower question.

When an AI project says it solved an open mathematical problem, what exactly should enter the count?

An open problem is simply a question that mathematicians have not yet settled. A theorem is a mathematical statement that has been established by proof. If AI systems begin settling such questions faster, people may reasonably ask whether an old stock of human-written problems is shrinking.

To test that idea, I first needed a trustworthy list of events. I did not need to pretend I could personally certify every proof. I needed a process that could show:

- which files were public;
- which files passed the project's official checks;
- whether the formal statement matched the original question;
- whether two files represented one result or two; and
- whether published mathematics already contained the answer.

That process became a small measuring instrument.

What a computer-checked proof can tell us

In May 2026, the AlphaProof Nexus paper [1] reported that its system had resolved nine of 353 attempted Erdős problems. Its public repository—an online collection of the project's files—[2] contained nine Lean proof files and nine prose PDFs.

Lean is a proof assistant. A person writes a mathematical statement and proof in a formal language. Lean checks whether each step follows from the definitions and rules available inside that formal environment.

This is much stronger than asking readers to trust an announcement. The code can be inspected, and the project can run it through an automated test system.

But Lean checks the question it is given. It cannot tell us whether that formal question is the same one Erdős intended decades earlier. It cannot search all of mathematical history and decide whether the answer is new. It cannot tell us whether a database link points to the right file.

A green computer check means the formal proof works in its stated environment. It does not automatically mean one new open problem has disappeared.

Why the rows became complicated

I used Codex to inspect the nine reported results one by one. The rows quickly stopped behaving like simple rows.

Sometimes one catalogue entry contains several separate questions. That happens with problems 12 and 138. Solving one part does not necessarily solve the whole entry.

Sometimes one ordinary-looking word hides different mathematical definitions. Problems 125 and 741 involve different meanings of density: different ways of describing how much of an infinite set is present. Natural, lower and upper density are not interchangeable. A proof may correctly answer one version while leaving another version open.

Sometimes the claim is correct but the link is wrong. The public material for 741(i) links its positive result to an earlier negative proof about a different density definition. The paper's link for 741(ii) points to problem 12(ii), even though the correct proof exists elsewhere in the same repository.

I cannot evaluate every mathematical line. But a row-level audit can still require the question, proof and link to refer to the same thing before the row is counted.

What happened to problem no. 846

Erdős problem no. 846 asks about infinitely many points in a plane. [5]

Imagine that every finite handful of those points contains a reasonably large group with no three points sitting on one straight line. Does that guarantee that the entire infinite collection can be divided into finitely many groups, each also having no three points on one line?

AlphaProof Nexus produced a proof that the answer is no.

I cannot verify the hundreds of formal steps myself. But I can explain the shape of the argument that the public files describe.

The proof translates the geometry question into a question about a network. The network has points connected by lines. Each connection is then represented as a point in the geometric construction. The coordinates are chosen so that a triangle in the network corresponds to three geometric points on one straight line.

In every finite piece of the network, it is possible to keep at least half the connections while removing all triangles. That produces the well-behaved finite groups required by the original question.

But across the entire infinite network, a theorem about colouring guarantees that any attempt to divide all connections into finitely many groups eventually creates a triangle inside one group. In the geometric version, that becomes three points on one line. The proposed finite division therefore cannot exist.

The short PDF moves quickly over the difficult coordinate work needed to avoid accidental straight lines. The Lean proof spends hundreds of lines on precisely that part. The first located public formal-proof commit and the official full-repository Lean test run are preserved in the source record. [6] [7]

The formal construction exists and passes the project's public checks.

Why remove it from the count?

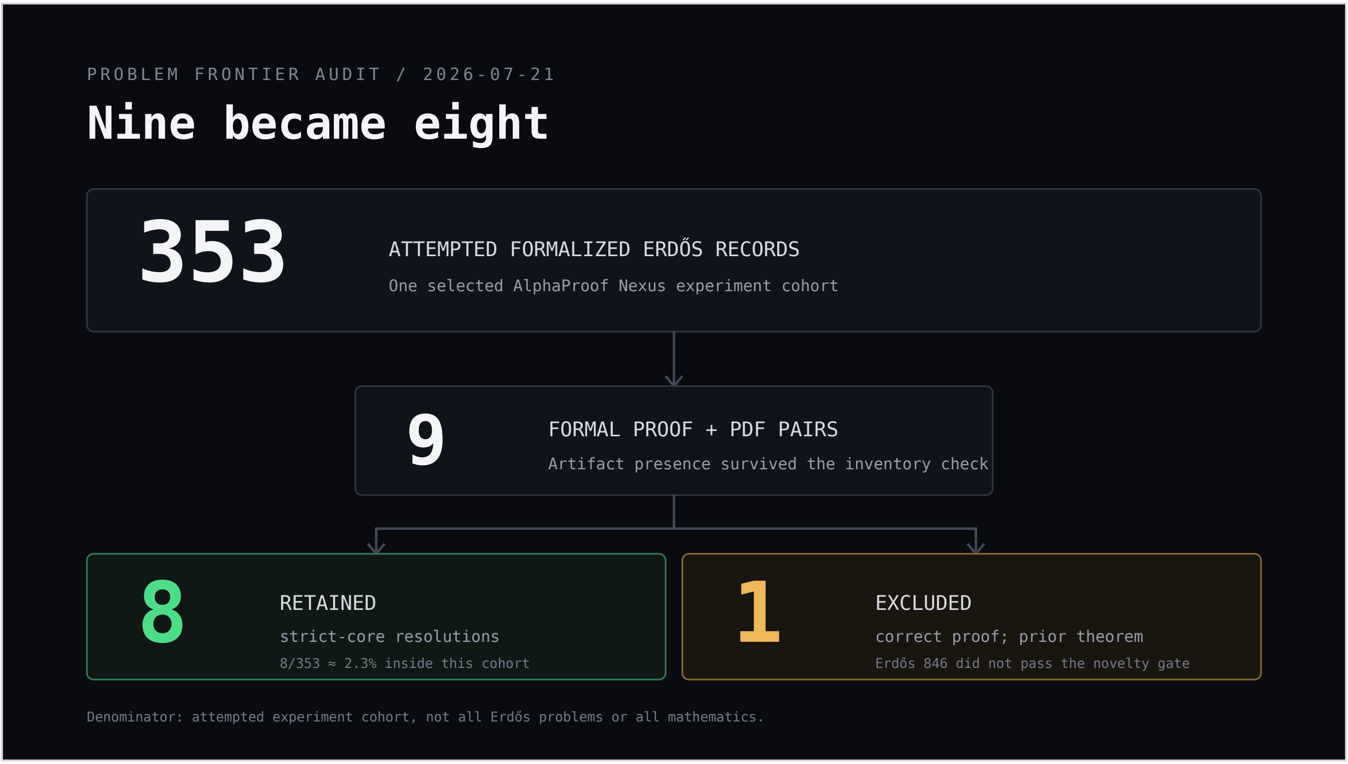
An independent paper produced with an internal OpenAI model [3] reports that mathematician Vojtěch Rödl recognized the answer as a consequence of an older theorem by Reiher, Rödl and Sales. Their work [4] did not name problem 846, but its result already implied the negative answer after the necessary mathematical translation.

The public timeline supports the interpretation that the AI system found its construction independently. That is still interesting. But my strict counter was supposed to measure newly resolved open problems, not new routes to answers that published mathematics already implied.

The proof remained valid.

The row changed category.

Nine became eight.



The AI project attempted 353 selected Erdős problems and released nine proof pairs. This audit currently retains eight as newly resolved under its strict rule. Problem no. 846 is excluded because earlier work already implied its answer. These numbers describe one experiment, not all of mathematics.

Why I kept the wrong first number

The first tracker snapshot had already been saved on 20 July with nine included rows.

A snapshot is a frozen copy of what a system knew at a particular time. Think of it as a dated photograph rather than a live scoreboard.

After the audit produced eight, I could have changed the old snapshot and made the history look cleaner.

I did not.

The original snapshot honestly records the first information state. The working table honestly records the later correction. A future snapshot must show the change from nine to eight instead of pretending the system always knew the answer.

That rule is the beginning of the Problem Frontier Tracker. Each accepted snapshot preserves its files, definitions and sources. Later snapshots can add new results, correct old classifications or record new problems without editing their predecessors.

Objects in Time is the Hedegreen Research register intended to preserve such accepted checkpoints publicly as dated objects.

The tracker calculates.

The register remembers.

The frozen snapshot keeps the original nine. The later audit records eight. The next checkpoint must show what changed without rewriting either state.

What eight out of 353 does not mean

Eight divided by 353 is about 2.3 percent.

It does not mean AI has solved 2.3 percent of mathematics.

The 353 problems were selected partly because someone had already translated them into formal statements a computer system could use. They are not a random sample of mathematical research. They do not represent every kind of creative question, definition, literature search or long collaboration found in real mathematics.

The dates do not show a trend either. Two retained results fall in the first quarter of 2026 and six in the second. That sounds like growth, but it is only one project's work and publication schedule split across two nearby calendar boxes.

There is no second accepted tracker snapshot yet. There is no fair comparison of computing effort. There is no complete public record of failed attempts. I also do not yet have a defensible measure of how many comparable new problems people created during the same period.

Without that incoming flow, I can count subtraction but not depletion.

The second number that did not match

The same paper reports 44 successful OEIS conjectures among 492 attempts.

OEIS stands for the Online Encyclopedia of Integer Sequences. It is a large catalogue of number patterns. A sequence might begin `1, 2, 4, 8` and continue according to a rule. Mathematicians record such sequences, their properties and questions about them. A conjecture is a mathematical claim that appears true but has not yet been proved.

The 492 attempted records can be reconstructed from the public material. The 44 successful results cannot yet be matched one-for-one.

The public repository contains 38 Lean result files representing 37 different OEIS sequence numbers. The repository history, paper files, supplement and public discussion did not reveal six additional results or another counting rule that closes the gap.

This does not prove the reported number 44 is wrong.

It means 44 is what the authors report, while 38 files connected to 37 sequence numbers are what this audit can currently reconstruct publicly.

Maybe AI will create problems too

The whole “running out” idea may use the wrong picture.

Mathematics is not a warehouse where every solved problem leaves one empty shelf. A solution can create a harder version, reveal a new obstruction, join two areas of research or make a better question possible.

AI systems may eventually help create new problems as well as solve old ones. The important question may not be whether the warehouse becomes empty. It may be whether the human-readable edge of unsolved mathematics and a separate machine-generated edge begin moving away from each other.

This is where my own position matters.

Today, the distance is obvious. I am taking Mathematics C. A professional mathematician can understand far more of these proofs than I can. I am not pretending our positions are equal.

But a future AI system might create and solve problems so large or unfamiliar that no single person can hold the complete question, proof and consequence at once. A professional mathematician could then need formal checks, source history and smaller human explanations for the same basic reason I need them now: the complete object exceeds the person inspecting it.

That would not make mathematicians stupid. A thing can exceed one person's understanding without making that person unintelligent.

This article has not shown that such a machine frontier exists. The tracker cannot currently measure it. It is a possibility, not a result.

But it explains why I care about building an instrument ordinary people can inspect. If mathematical work becomes harder for any one human to understand, we will need better ways to see what was claimed, what was checked, what was new and what remains uncertain.

What the instrument needs next

The original question—*Are We Ready to Run Out of Problems?*—remains open.

Before the tracker can say anything useful about it, it needs several accepted snapshots, results from more systems and more kinds of problems, and a fair way to count new problems arriving at the edge of unsolved work. It must preserve corrections and rediscoveries instead of deleting them from the story.

Most of all, it needs time.

The frozen first snapshot says nine.

The audited working table says eight.

Neither contains a trend.

That is a smaller finding than the original question promised.

It is also the first finding I trust.

The measurement system worked because the headline became smaller.

— Dennis Hedegreen, follow the data

Methods and source note

I supplied the question and editorial direction, approved the scope and publication boundary, and remain responsible for the decision to publish.

OpenAI Codex, running GPT-5.6 Sol with reasoning effort set to High, located and inspected the cited public material; compared Lean files, prose PDFs, repository history and original problem records; formalized the working inclusion rules; built the source joins, row audits, frozen tracker, figures, verification scripts and tests; and drafted and revised this article. The resulting eight-row classification has not been independently replicated or peer-reviewed by a mathematician.

The audit used fixed source versions, digital file fingerprints, original sources, public discussion and the project's official automated checks. It did not locally rebuild Lean, inspect private model runs or training data, or conduct an exhaustive literature-priority search across every retained result.

Sources and evidence

1. AlphaProof Nexus paper v2 — open source (https://arxiv.org/abs/2605.22763v2).
2. Pinned AlphaProof Nexus result repository — open source (https://github.com/google-deepmind/alphaproof-nexus-results/tree/0647711a71183c1ea492ad60860776617ce1ea88).
3. Independent OpenAI-model proof — open source (https://arxiv.org/abs/2602.21275v1).
4. Prior Reiher–Rödl–Sales theorem — open source (https://arxiv.org/abs/2311.08556v2).
5. Original statement of problem 846 in the journal's 1992 volume — open source (https://lematematiche.dmi.unict.it/index.php/lematematiche/article/view/587).
6. First located public formal-proof commit for problem 846 — open source (https://github.com/google-deepmind/formal-conjectures/commit/2404258180688283e5141021c75464dc2acfb798).
7. Official full-repository Lean CI run at the pinned result revision — open source (https://github.com/google-deepmind/alphaproof-nexus-results/actions/runs/27019860483).

The frozen tracker, full nine-row audit, OEIS reconciliation and classification history are preserved in a versioned evidence companion:

- [8] Inspect the evidence manifest
- [9] Download the evidence bundle

The companion reproduces the article's central counts and keeps the provisional review boundaries visible. It is not independent mathematical peer review and does not redistribute the cited third-party papers or formal proof repositories.

RELATION MEMORY

NEARBY The Tool Learned to Say Not Yet · Both make a provisional refusal state visible instead of forcing a clean-looking answer.

NEARBY A Pattern Is Not an Explanation · Both separate a small measured sequence from the larger story it might tempt us to tell.

NEARBY The Benchmark Twins · Both ask what a measured capability does and does not establish.