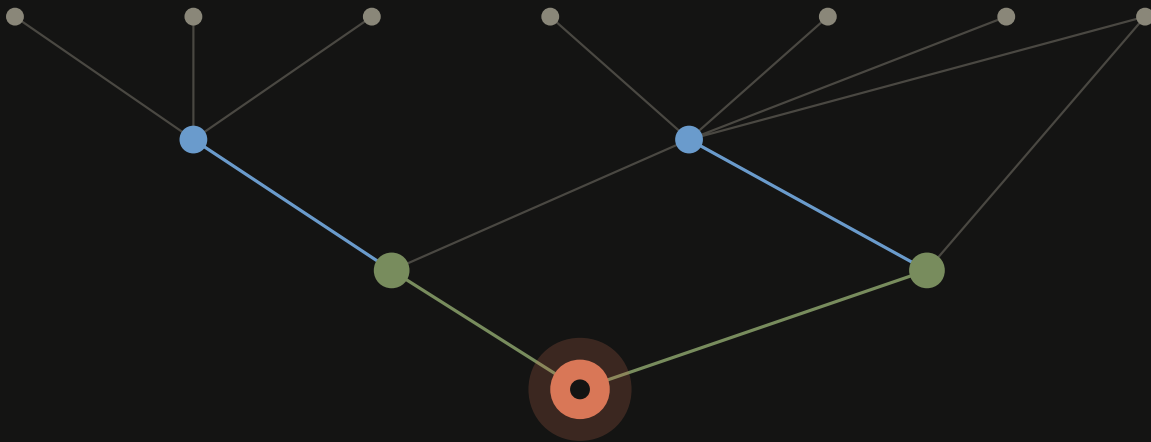


INFLECTION

THE WEEKLY MAGAZINE OF INNOVATION

ISSUE 02 • 11 SEPTEMBER 2026

DEEP DIVE — MACHINE-CHECKED MATHEMATICS



THE COORDINATION UNLOCK

The Bottleneck Was Never the *Brain*

Claude machine-checked Fermat's Last Theorem in eleven days — 13 million lines of Lean, 29,500 theorems, one general-purpose model. Its first attempt failed outright. What changed between the failure and the proof was not intelligence. It was a data structure.

SIGNALS

A pancreatic cancer drug doubles median survival

SIGNALS

Taipei admits the bottleneck is copper, not compute

AGAINST THE GRAIN

Why the referee queue just got worse, not better

03 The Bottleneck Was Never the Brain

Eleven days, 13 million lines, and a graph that held the whole thing together

04 How the Swarm Actually Worked

Statements split from proofs; agents talk to the artifact, not each other

05 From Lab to Market

What a durable, verifiable substrate is worth outside mathematics

06 Against the Grain · The Contrarian

The proof is real. The bottleneck simply moved somewhere worse

07 Signals

Oncology's undruggable target falls · Copper's last mile · A machine wins the coding olympiad

08 By the Numbers

Seven figures that size the week

09 The Long View · Sources

Why the cheap thing turned out to be the scarce one

DISPATCH · FROM THE EDITOR

The failed run was not wasted

There is a sentence buried in Anthropic's write-up of the Fermat formalization that deserves more attention than the headline number. The first attempts, using an ordinary multi-agent harness, failed: the agents "quickly lost track of the project's state and stopped collaborating effectively." Then comes the line. Those failed efforts "contributed ~7% of the non-boilerplate lines in the final proof."

Read that twice. The run that did not work still deposited a permanent, load-bearing seventh of the answer. Most agent systems we build are transactional: a run succeeds, or it rolls back and its output is discarded as unreliable. Here, failure was *banked*. Every theorem an agent managed to close was checked by Lean's kernel

and became a fact the next agent could stand on — regardless of whether the run that produced it ever reached its goal.

That is the lens for this issue. We spend enormous effort making models smarter and almost none making their output *durable*. The Fermat result suggests the ratio is backwards. Same model, same weights, two runs: one produced chaos, one produced the largest formal proof ever written. The variable was a shared graph and a checker that could tell truth from plausibility.

Everything else in this issue — the contrarian column, the three Signals, the numbers — is a variation on that theme. Read them asking the same question: what is this system's real constraint, and is anyone measuring it?

The Bottleneck Was Never the *Brain*

A 350-year-old theorem, a proof the mathematical community expected to spend five years formalizing, and eleven days of machine time. The interesting part is not that it worked. It is why the first try didn't.

BY THE INFLECTION DESK • 11 SEPTEMBER 2026

On the night of 17 August, at 02:00:57 UTC, a node in a dependency graph flipped from open to proved. The node was the root. Above it sat 29,500 verified theorems, 13 million lines of Lean code, and a chain of reasoning running back through Wiles and Taylor, Ribet, Serre and Frey to a marginal note Fermat scrawled around 1637, claiming a proof his margin was too narrow to contain. Anthropic published the result on 4 September: the first complete, end-to-end computer-checked proof of Fermat's Last Theorem. Kevin Buzzard, the Imperial College London number theorist leading the community effort to do exactly this, compiled the repository, ran the comparator tool against Mathlib's canonical statement, and confirmed it holds "with no assumptions other than the axioms of mathematics."

The obvious reading is that a model got smart enough to do Fermat. That reading is wrong, and the way it is wrong is the most useful thing a technical leader can take from this week.

Start with what was *not* new. The mathematics is Wiles's, following the 1995 Darmon–Diamond–Taylor exposition; Buzzard is blunt that "mathematically this work of Anthropic tells us essentially nothing." The model was a general-purpose internal research build, roughly comparable to a shipped product. Mathlib, the library it stands on, is the accumulated volunteer labour of hundreds of mathematicians. Anthropic's own repository credits 106 upstream files to Buzzard's project and to flt-regular.

The ingredients were all on the table. What was missing was a way to stop dozens of agents colliding for two weeks.

The failure mode nobody prices in

The first attempts used a conventional multi-agent harness. They failed — not for lack of capability, but for lack of memory. Thirteen million lines and thousands of intermediate results fit in no context window. Without an external record of what has been proved, what is in progress, and what remains, agents duplicate work, overwrite each other, and eventually cannot tell whether a task is done. This is not a model problem. It is a distributed-systems problem wearing a model problem's clothes.

Anthropic's fix was Prove2Me, an open platform built by Tianyi Peng's group at Columbia. It does one conceptually simple thing: it maintains a directed acyclic graph of every theorem statement in the project. Nodes are claims; edges are dependencies. An agent joining the effort queries the graph rather than a colleague.

How the swarm actually worked

1 • The graph is the only shared truth

Multi-agent systems usually coordinate through conversation: agents message each other, or a supervisor holds the plan. Both approaches put project state inside a context window, which means state degrades exactly as the project grows large enough to need it. Prove2Me inverts this. Coordination happens through the artifact. No agent needs to remember the project; the project remembers itself.

Crucially, the graph is not advisory. Because every node carries a machine-checkable claim, an agent cannot bluff its way to "done." Lean's kernel is the arbiter, and it does not accept an argument on the strength of the argument's confidence.

2 • Statements were split from proofs

A quiet engineering decision did outsized work. Prove2Me keeps theorem *statements* in different files from their *proofs*, with the dependency links maintained externally. Lean recompiles only what changed — so an agent refining one proof does not force millions of lines to rebuild. Compilation time is the tax on iteration speed in formal mathematics, and this cut the tax dramatically.

Each node also carried a natural-language description, letting agents search for a prior result semantically instead of scanning code. In effect the swarm got an index of its own accumulated knowledge.

3 • Failure became an asset

This is the part that should reshape how you think about agent economics. Roughly 7% of the non-boilerplate lines in the final proof came from the runs that failed. In a

DAG-coordinated, kernel-verified system, an agent that misses its goal still leaves behind verified lemmas — permanently reusable, because their correctness is not a matter of opinion. Exploration becomes append-only. The system ratchets.

4 • Supervision was almost absent

Human mathematical input amounted to occasional nudges from Peng: "Jacobian as a scheme sounds high priority," "push the Mazur theorem to be done soon." The run launched overnight on 7 August and closed on the 17th, consuming roughly six billion output tokens. Dozens of agents worked in parallel without a central planner — the graph did the scheduling.

5 • And then they did it on a consumer plan

The detail that should unsettle anyone budgeting for agent infrastructure: in a follow-up, researchers used *three personal Claude Max subscriptions* to formalize Vinogradov's Three Primes Theorem in three days, collaborating entirely through Prove2Me. If the scaffold is right, the compute bill is not the gate.

"Same model. Same weights. One run produced chaos; the other produced the largest formal proof ever written. The variable was where the state lived."

THE INFLECTION DESK

From lab to market

Strip the mathematics away and what remains is a general recipe with three parts: decompose the goal into a dependency graph of independently checkable claims; give every claim a mechanical verifier; let agents coordinate through the graph rather than through each other. Wherever those three conditions hold, this week's result says the work is now tractable at a scale that was not tractable before.

The conditions hold in more places than mathematics. Large-scale code migrations decompose into modules with type checkers and test suites as verifiers. Formal hardware verification already lives here. Regulatory and compliance evidence — a claim tree with auditable primitives at the leaves — fits the shape almost exactly. So does dependency-aware refactoring of a legacy mono-

lith, where the hard part has never been writing any single change but knowing which changes are safe and which are already done.

The commercial reading is sharper still. If the marginal cost of the model is falling and the scaffold is the differentiator, then the durable asset in an agentic product is not the prompt library and not the fine-tune. It is the verified state store: the accumulated graph of things your system has established, indexed, and can build on. That asset compounds. Model weights depreciate.

The near-term test is whether anyone ships a general-purpose "proof-of-work graph" as infrastructure. Prove2Me is open source and narrow to mathematics. The equivalent for software, hardware, and audit is un-built.

FIELD NOTES · HOW A PROOF SWARM WORKS

- 1 Decompose.** The target theorem becomes the root of a directed acyclic graph. Every node is a precise statement; every edge says "this proof needs that result first." Nothing is proved yet — the graph is a map of the debt.
- 2 Claim and close.** An agent queries the graph for an open node whose dependencies are already proved, writes a Lean proof, and commits. It never asks another agent what to do. The graph is the schedule, the memory, and the handoff.
- 3 Verify and ratchet.** Lean's kernel checks the commit mechanically. A passing node is permanently true and permanently reusable — so partial progress survives the agent, the session, and even the failed run that produced it.

GLOSSARY

Autoformalization	Automatically translating a proof written for human readers into a formal language a computer can check line by line. Translation, not discovery.
Lean kernel	The small trusted core of the Lean proof assistant that does the actual checking. Everything else in Lean is convenience; only the kernel is believed.
DAG	Directed acyclic graph — nodes with one-way dependency edges and no cycles. Here, a map of which theorems must be proved before which others.
Mathlib	Lean's community-built library of formalized mathematics, assembled by hundreds of volunteers. The Fermat proof is over five times its size.

The bottleneck did not disappear. It *moved* — somewhere worse.

Move 37 was not a good move because it was fast. It was a good move because it valued a region of the board everyone else had written off. Here is the region this week's consensus is writing off.

The celebration says: verification is solved, referees are freed, mathematics accelerates. Look one step downstream and the opposite is true in the short run. Auto-formalization does not remove human review — it *relocates* it, from checking whether a proof is valid to checking whether the formal statement means what the author claims, and whether 13 million machine-written lines are worth maintaining.

Buzzard, who has every incentive to be generous and is, gives the numbers that matter. Mathlib currently carries around 3,000 open pull requests, more than 600 of them active in the review queue. Maintainers will not accept AI review, and are reluctant to review AI-generated code because most of it is poor. The proof compiles — on a 96-core machine it takes nearly twenty times as long as all of Mathlib. Anthropic's own footnote concedes the proof "is likely much longer than it needs to be."

So the artifact is enormous, unidiomatic, and not upstreamable on any near timeline. It settles the theorem. It does not enlarge the shared library that makes the *next* theorem cheaper. The compounding asset this issue has been praising is precisely what did not get built here.

The honest risk list

Trust is conditional. Lean's guarantee is only as strong as its kernel. In summer 2026 a kernel soundness bug briefly let a spurious disproof of Collatz pass. Buzzard checked this repository by hand — inspecting every line Claude flagged as non-mathematical, about a hundred

lines, plus random samples — and found nothing malicious. That is diligence, not a guarantee, and it does not scale.

The result is narrower than the headline. The formalization follows the 1995 exposition, not the modern proof, and its Mazur argument only rules out Frey curves with a point of order $p \geq 17$. Fermat for smaller exponents comes from separate prior work on regular primes. Fine mathematically; not "one run, whole theorem."

Attribution is unresolved. Buzzard was awarded £1M over five years to do this. Anthropic did it in eleven days on top of his project's files, Mathlib, and the Lean FRO's infrastructure. He remains enthusiastic — he still owes the funder a human-explorable document, which the machine did not produce. But funders now face a question with no precedent: how do you keep financing the community substrate that makes the eleven-day run possible, once the eleven-day run is what gets the headline?

Why we still think the read holds

Every objection above is about the *output*. None touches the mechanism. A bloated, unidiomatic proof is still evidence that a coordination substrate turned a five-year human project into a two-week machine one using a model that already existed. Idiomatic output is a tractable engineering goal. Discovering that state, not intelligence, was the binding constraint is not — and that discovery is now free to everyone.

Three developments from elsewhere on the frontier, and what each one is actually telling you.

13.2

MONTHS MEDIAN SURVIVAL

BIOTECH · ONCOLOGY

The FDA approved Revolution Medicines' RASONQUE (daraxonrasib) for metastatic pancreatic adenocarcinoma — the first broad RAS-targeted medicine ever cleared, for a target considered undruggable since its discovery. In the Phase 3 RASolute 302 trial, median overall survival was 13.2 months against 6.7 on chemotherapy, a 60% reduction in risk of death (HR 0.40). The strategic tell is the label, not the efficacy: the drug is approved with or without an identified RAS mutation and needs no companion diagnostic. Precision oncology's core dogma has been mutation selectivity. This is a once-daily pill that deliberately inhibits both mutant and wild-type RAS and wins anyway — trading sequence selectivity for *conformational state* selectivity. The cost shows up honestly in the label: dermatologic toxicity in 86% of patients.

Source · Revolution Medicines, "U.S. FDA Approves RASONQUE," 26 Aug 2026

\$135.1B

EQUIPMENT SALES, +15%

SEMICONDUCTORS

SEMICON Taiwan opened in Taipei with a record 1,300 exhibitors and 100,000 attendees from 65 countries — and an unusually candid framing from SEMI's Terry Tsao: "Data movement in today's AI systems could consume more energy than computation itself, making system architecture, not individual chip performance, the new bottleneck." Co-packaged optics crossed from roadmap to production this year: TSMC's COUPE platform entered mass production and Foxconn expects CPO switch shipments in Q3. SEMI projects 300mm fab equipment spending will pass \$150 billion in a single year for the first time in 2027. The industry has stopped arguing about who has the fastest chip.

Source · Tech Times / SEMI, SEMICON Taiwan 2026, 31 Aug–4 Sep 2026

535.4

OUT OF 600 AT IOI 2026

AI · COMPUTE

NVIDIA researchers ran a 550B-parameter Nemotron variant on the IOI 2026 problem set under the same time and submission limits as human contestants. It scored 535.4 — past the 361.12 gold threshold and past the top human's 498.27, the first AI system to outscore the highest-scoring contestant on an IOI set. The paper is unusually honest about the asterisks: unofficial and unsupervised, a single prospective run, and up to 760 GB300 GPUs in play — "a system-level comparison under the same time and submission limits, rather than an equal-resource comparison." Note the echo of this week's cover story: most of the gain came from GenCorrect, a test-time loop, not a bigger brain.

Source · Ficek, Narenthiran et al., arXiv:2609.02849, 2 Sep 2026

13,000,000

Lines of Lean code Claude wrote — over five times the size of Mathlib, the community library the proof builds on.

11

Days from launch to a proved root node, working largely autonomously. The community project it overtook was funded for five years.

29,500

Intermediate theorems used in the final argument, out of 30,300 machine-verified along the way.

6 billion

Output tokens consumed by a general-purpose internal research model — no bespoke mathematics system.

7%

Share of non-boilerplate lines in the final proof contributed by the earlier attempts that *failed*.

3

Consumer Claude Max subscriptions that formalized Vinogradov's Three Primes Theorem in three days via the same scaffold.

129

Pages in Wiles's 1995 proof — which took months of expert review, and still had a critical gap that took a year to close.

621

Incorrect proofs of Fermat submitted in the first year after a 1908 prize was announced. Verification has always been the hard part.

The cheap thing was scarce all along

Three stories this week rhyme, and the rhyme is worth more than any of them alone.

A theorem was machine-checked in eleven days — not by a smarter model, but by giving an existing one a shared graph to remember in. A coding olympiad was won by a language model whose decisive advantage came from GenCorrect, an iterative generate-evaluate-refine loop applied at inference, not from additional training. And in Taipei, the world's chipmakers spent five days agreeing that the constraint on AI systems is no longer how fast a processor computes but how fast data moves between processors, and pouring capital into optics to fix it.

Three domains, one shape. In each case the thing everyone was optimizing — model intelligence, model scale, transistor speed — turned out not to be binding, and the thing nobody was counting — shared state, test-time iteration, interconnect bandwidth — turned out to be. The pancreatic-cancer drug is the same story in a different key: forty years of chasing selectivity against the

mutation, and the win came from selecting on the protein's *state* instead.

There is a discipline in that. Before the next budget cycle, it is worth asking of any system you own: which resource are we all optimizing because it is legible and measurable, and which resource are we assuming is free? The Fermat run is a clean natural experiment because the model was held constant. Two attempts, identical weights, and the difference between failure and the largest formal proof in history was where the project's memory lived.

Buzzard's own reaction is the right note to end on. He did not conclude that his field was finished. He concluded that if thousands of pages can be formalized end to end in eleven days, then formalization of modern research "on the fly" is coming — and with it, a machine capable of ruthlessly flagging the arguments the literature currently accepts because they are "known to the experts." The interesting consequence of a fast verifier is not the proofs it confirms. It is the ones it declines to.

SOURCES & FURTHER READING

Anthropic, "Formalizing Fermat's Last Theorem," 4 Sep 2026 — <https://www.anthropic.com/research/formalizing-fermats-last-theorem>

Kevin Buzzard, "FLT: Anthropic has beaten me to it," Xena Project, 4 Sep 2026 — <https://xenaproject.wordpress.com/2026/09/04/flt-anthropic-has-beaten-me-to-it/>

Anthropic, fermats-last-theorem repository (Apache 2.0) — <https://github.com/anthropics/fermats-last-theorem>

Chen, Marwaha, Lu, Yuen & Peng, "Prove2Me: An open collaborative platform for scaling math formalization," arXiv:2608.28433 — <https://doi.org/10.48550/arXiv.2608.28433>

Tech Times, "Fermat's Last Theorem Machine-Checked," 5 Sep 2026 — <https://www.techtimes.com/articles/326745/20260905/fermats-last-theorem-machine-checked-claude-completes-11-days-what-took-years-plan.htm>

Revolution Medicines, "U.S. FDA Approves RASONQUE (daraxonrasib)," 26 Aug 2026 — <https://ir.revmed.com/news-releases/news-release-details/us-fda-approves-revolution-medicines-rasonquetm-daraxonrasib>

Tech Times, "SEMICON Taiwan 2026 Kicks Off: AI Chips' Bottleneck Is Wires," 31 Aug 2026 — <https://www.techtimes.com/articles/326056/20260831/semicon-taiwan-2026-kicks-off-ai-chips-bottleneck-wires-connecting-them.htm>

SEMI, 300mm Fab Outlook / SEMICON Taiwan 2026 release — <https://www.semi.org/en/semi-press-release/semi-projects-double-digit-growth-in-global-300mm-fab-equipment-spending-for-2026-and-2027>

Ficek, Narenthiran, Samadi, Majumdar & Ginsburg, "Post-Training Language Models for Gold-Medal Performance in Coding Competitions," arXiv: 2609.02849, 2 Sep 2026 — <https://arxiv.org/abs/2609.02849>

Darmon, Diamond & Taylor, "Fermat's Last Theorem" (1995 exposition) — <https://www.math.mcgill.ca/darmon/pub/Articles/Expository/05.DDT/paper.pdf>

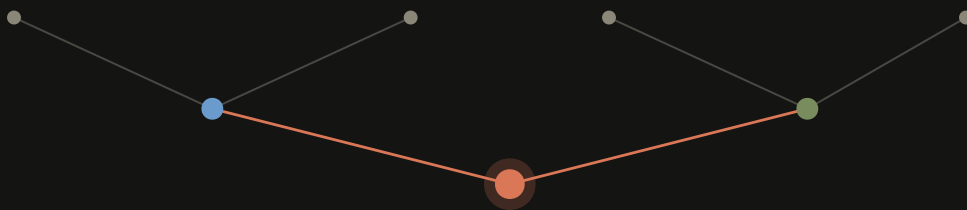
INFLECTION

THE WEEKLY MAGAZINE OF INNOVATION

THE LENS

Which resource is your team optimizing because it is easy to measure — and which one are you assuming is free?

A recurring question. Same question every issue; the answer moves.



NEXT ISSUE • FRIDAY

Researched, written & designed with Claude.
Typeset in Poppins & Lora on the Anthropic palette.
Issue 02 • 11 September 2026