# Harmonic Aristotle: Architecture and Performance - 2026-09-23 v4

*Editor's note: As we go to press (23 September 2026), OpenAI claimed its unreleased internal model has solved more than 100 open mathematical problems (cf. [my Substack post]) but has not published a list or their proofs, pending review, verification, and guidance by a new Mathematics and AI Advisory Group at the Institute for Advanced Study.*

*Kris Carlson prompting Gemine 3.1 Pro and then Gemini 3.8 Flash because Pro kept fouling things up.*

*Highlights:*

> *The primary structural bottleneck limiting modern systems is global compositional synthesis: While an AI agent can decompose a problem into short lemma chains and verify them without error, problems requiring intricate, multi-layered global invariants still exceed the reach of unsupervised search.*
>
> *The Erdős Problem 124 episode highlights an important failure mode in automated mathematics: An AI can produce a flawlessly verified Lean proof, but if the formal statement fails to fully capture the intended mathematical depth, the result proves a weaker theorem rather than resolving the core open conjecture.*

Harmonic introduced Aristotle as an automated theorem-proving architecture engineered for mathematical verification [1]. Large language models frequently hallucinate intermediate logical steps when tasked with extended deductive chains; Aristotle prevents invalid reasoning by executing its entire deductive verification loop directly within Lean 4, a functional programming language and interactive theorem prover [1]. By coupling informal natural-language reasoning with rigorous formal checking, every generated theorem is guaranteed to be mathematically sound and free of hallucinations [1].

## System Architecture

### Lean Proof Search Algorithm

Aristotle utilizes a highly parallel Monte Carlo Graph Search (MCGS) algorithm operating directly on formal Lean proof states [1]. A 200B+ parameter transformer functions jointly as a policy network and value estimator, scoring proof states and proposing subsequent Lean tactics [1]. Because multiple tactics can introduce subgoals, Aristotle organizes search traces into a directed hypergraph where states represent Lean goals and edges represent tactical steps [1]. For instance, when decomposing a proof by case analysis using the *cases* tactic, the system generates separate branches for each discrete case; Aristotle explores these branches in parallel and confirms completion only when all individual branches evaluate to zero remaining goals [1].

### Lemma-Based Informal Reasoning

To avoid getting trapped in myopic syntactic searches, Aristotle employs a natural-language planning module that drafts high-level proof sketches in natural language [1]. This engine systematically fragments complex theorems into sequences of manageable intermediate lemmas [1]. The system autoformalizes each lemma into Lean, attempts verification via the proof engine, and parses compiler error traces back into the reasoning prompt to revise statements iteratively [1]. For example, in competitive algebra, human solvers often invent auxiliary functions or definitions not present in the prompt; Aristotle autonomously formulates these auxiliary objects, establishes their properties as formal lemmas, and integrates them back into the top-level theorem [1].

### Dedicated Geometry Solver

Geometric theorems typically present combinatorial bottlenecks when encoded purely through symbolic Lean axioms [1]. Aristotle routes Euclidean plane geometry tasks to Yuclid, a specialized C++ solver relying on deductive databases and algebraic elimination methods [1]. For example, when evaluating diagrams containing perpendicular bisectors, Yuclid automatically detects implicit geometric invariants—such as deriving relations of the form $AC^2 + BD^2 = AD^2 + BC^2$ from perpendicularity—and transmits these derived identities into the Lean environment as proven auxiliary facts, entirely avoiding manual axiomatic coordinate verification [1].

## International Mathematical Olympiad 2025 Performance

At the 2025 International Mathematical Olympiad (IMO), Aristotle achieved a gold-medal-equivalent result by successfully producing full, gapless Lean 4 proofs for five of the six contest problems [1]. Human involvement was strictly bounded: Human mathematicians formalized the initial problem statements into Lean 4 theorem declarations without providing hints, guidance, or partial proofs [1]. Beyond this initial problem translation, the entire proof search, lemma generation, and Lean code synthesis were completed fully autonomously by Aristotle [1]. Because the final solutions compiled successfully without placeholders (*sorry*), no human judging was required to evaluate proof correctness [1].

### Problem 1: Sunny Lines

*Formal Statement:* A line in the Cartesian plane is called sunny if it is not parallel to the x-axis, the y-axis, or the line $x + y = 0$. Let $n \geq 3$ be an integer. The problem requires determining all non-negative integers k such that there exist n distinct lines in the plane satisfying two conditions: First, for all positive integers a and b with $a + b \leq n + 1$, the point (a, b) lies on at least one line. Second, exactly k of the n lines are sunny. The admissible values are $k \in \{0, 1, 3\}$.

*Resolution Method and Human Contribution:* While standard human arguments rely on geometric convexity—such as noting that a line intersects the boundary of a convex polygon at most twice—this spatial fact is notoriously difficult and verbose to encode formally in Lean [1]. Aristotle bypassed this topological complexity by discovering an algebraic reformulation based on permutations and derangements, constructing an explicit discrete counterexample configuration up to $n = 5$ that ruled out valid configurations for all larger integers [1]. Human involvement was strictly restricted to supplying the initial Lean theorem definition.

### Problem 2: Geometric Configurations

*Formal Statement:* Let Ω and Γ be circles with centers M and N, respectively, where the radius of Ω is strictly less than the radius of Γ, intersecting at two distinct points A and B. Line MN intersects Ω at C and Γ at D such that C, M, N, D lie collinearly in that order. Let P be the circumcenter of triangle ACD. Line AP intersects Ω again at E ≠ A and Γ again at F ≠ A. Let H be the orthocenter of triangle PMN. The problem asks to prove that the line passing through H parallel to AP is tangent to the circumcircle of triangle BEF.

*Resolution Method and Human Contribution:* Aristotle routed the geometric diagram to the Yuclid engine, which systematically deduced synthetic facts, angle congruence relations, and coordinate invariants, supplying the necessary geometric lemmas directly into Lean for formal closure [1]. Human contribution was limited to transcribing the geometric configuration into formal theorem syntax.

### Problem 3: Bonza Functions

*Formal Statement:* Let ℕ denote the set of positive integers. A function f : ℕ → ℕ is called bonza if f(a) divides $b^a - f(b)^{f(a)}$ for all positive integers a and b. The task is to determine the smallest real constant c such that f(n) ≤ cn for all bonza functions f and all n ∈ ℕ.

*Resolution Method and Human Contribution:* Solving the problem required mathematical intuition to define intermediate structural objects [1]. Novelty emerged when Aristotle autonomously conceived an auxiliary set S(f) = {n ∈ ℕ : f(n) ≠ n}, an entity completely absent from the problem statement [1]. By analyzing the topological behavior, fixed points, and cardinality bounds of this custom set, Aristotle proved the bound [1]. Humans provided only the Lean definition of a bonza function.

### Problem 4: Sums of Three Divisors

*Formal Statement:* A proper divisor of an integer N is a positive divisor other than N itself. An infinite sequence a1, a2, ... consists of positive integers, each having at least three proper divisors. For each n ≥ 1, an+1 is defined as the sum of the three largest proper divisors of $a_n$. The objective is to determine all possible values of the initial term a1.

*Resolution Method and Human Contribution:* During initial proof sketches, Aristotle produced informal arguments containing subtle gaps in modular arithmetic [1]. However, when attempting to formalize these steps in Lean 4, the compiler rejected the invalid transitions [1]. Aristotle parsed the compiler errors, revised its informal strategy, eliminated the gaps, and synthesized a valid proof [1]. Human involvement was strictly bounded to setting up the recurrence in Lean.

### Problem 5: Inekoalaty Game

*Formal Statement:* Alice and Bazza play a two-player game governed by a parameter λ > 0. On turn n (starting at n = 1): If n is odd, Alice picks a real number $x_n \geq 0$ such that x1

$+ \ldots + x_n \leq \lambda n$. If n is even, Bazza picks a real number $x_n \geq 0$ such that $x_1^2 + \ldots + x_n^2 \leq n$. If a player cannot move, the other player wins; if the game continues indefinitely, neither wins. The goal is to determine all values of λ for which Alice or Bazza possesses a winning strategy.

*Resolution Method and Human Contribution:* Aristotle tackled the continuous game dynamics by utilizing continuous approximations and calculus to derive candidate bounds, then discharged the discrete formal inequalities inside Lean by invoking automated nonlinear arithmetic tactics (*nlinarith*) alongside formal filter theory (*filter_upwards*) [1]. Humans provided only the formalized game rules.

### Problem 6: The Combinatorial Failure Mode

*Formal Statement:* Consider a 2025 × 2025 grid of unit squares. Matilda places rectangular tiles, possibly of differing dimensions, such that each tile boundary aligns with grid lines and every unit square is covered by at most one tile. The problem asks for the minimum number of tiles Matilda must place so that each row and each column of the grid contains exactly one unit square that remains uncovered.

*Resolution Method and Cause of Failure:* Aristotle failed to solve Problem 6 [1]. While Aristotle's lemma generator successfully formalized and proved several local properties, it was unable to maintain the global combinatorial bookkeeping required across multi-step structural exchanges [1, 2]. As demonstrated in formal case studies of similar IMO Problem 6 challenges (such as the Grasshopper problem), current theorem provers frequently succeed at verifying localized lemmas (such as invariance under adjacent transpositions) but fail to bridge the global counting contradictions needed to close the primary theorem [2].

### Generalization of Competition Results

The IMO 2025 results demonstrate that automated theorem provers have largely solved the challenge of localized deductive search and formal verification [1]. The primary structural bottleneck limiting modern systems is global combinatorial synthesis: While an AI agent can decompose a problem into short lemma chains and verify them without error, problems requiring intricate, multi-layered global invariants still exceed the reach of unsupervised search [1, 2].

## Erdős Problem 124

*Formal Statement and Context:* Erdős Problem 124 represents a longstanding question posed by Paul Erdős concerning complete sequences of integer powers and digit distributions [3]. The core formulation asks: Given k natural numbers $d_i \geq 2$ satisfying $\sum 1/(d_i - 1) \geq 1$, does there exist for every natural number n a choice of coefficients $a_i$ such that $n = \sum a_i$, where each $a_i$ in base $d_i$ uses only digits from {0, 1}? The original 1995/1996 asymptotic formulation requires proving infinite divisibility and terminal-zero digit properties across arbitrarily large parameters, a conjecture that cannot be established via finite computation [3].

*Resolution Method and Human Contribution:* Harmonic mathematician Boris Alexeev provided human direction by selecting the problem statement and submitting it to the Aristotle API [3]. Aristotle subsequently generated a machine-checked Lean 4 proof within 6 hours, verifying fully in under 1 minute [3]. However, mathematical scrutiny revealed that Aristotle proved a weakened variant of the conjecture: The statement compiled by Aristotle omitted a crucial asymptotic constraint present in Erdős's intended question, solving an easier version whose core criterion was already known in prior literature [3]. This episode highlights an important failure mode in automated mathematics: An AI can produce a flawlessly verified Lean proof, but if the formal statement fails to fully capture the intended mathematical depth, the result proves a weaker theorem rather than resolving the core open conjecture [3].

## References

[1] Aristotle: IMO-level Automated Theorem Proving. Tudor Achim, Alex Best, Alberto Bietti, Kevin Der, Mathïs Fédérico, Sergei Gukov, Daniel Halpern-Leistner, Kirsten Henningsgard, Yury Kudryashov, Alexander Meiburg, Martin Michelsen, Riley Patterson, Eric Rodriguez, Laura Scharff, Vikram Shanker, Vladmir Sicca, Hari Sowrirajan, Aidan Swope, Matyas Tamas, Vlad Tenev, et al., arXiv:2510.01346 [cs.AI], 2025. License: Creative Commons Attribution 4.0 International (CC BY 4.0).

[2] Using Aristotle API for AI-Assisted Theorem Proving in Lean 4: A Formalisation Case Study of the Grasshopper Problem. Gabriel Rongyang Lau, arXiv:2605.20120 [cs.AI], 2026. License: Creative Commons Attribution 4.0 International (CC BY 4.0); Pagination: 11 pages (arXiv preprint).

[3] Harmonic's AI Aristotle Claims Solution to Historic Math Puzzle. Mindplex Magazine, 2025. News and analysis report detailing Boris Alexeev's deployment of Aristotle on Erdős Problem 124 and the distinction between the solved variant and the original conjecture. License: Proprietary / Editorial web publication.

[4] Aristotle Learns to Code, Achieving New State-of-the-Art of 96.8% on Code Verification Benchmark. Harmonic Research Announcements, 2026. Technical overview of Aristotle's verification performance on the VERINA benchmark. License: Proprietary / Corporate technical release.