AI for Mathematical Discovery: From Guessing to Verifying

There's a specific kind of optimism that shows up whenever AI touches mathematics: the dream that a machine will stumble onto a new theorem the way a human does—by taste, by pattern, by obsession. What's actually happening is less romantic and (to me) more interesting: we are learning how to make discovery *operational*.

"Operational" here means a loop you can run: translate a problem into a formal object, generate candidate steps, check them, and feed the verified signal back into the system. Not a single shot "answer", but a process that can grind for hours, days, weeks—without drifting into hallucination.

Why formal systems change the game

If you've used Isabelle/HOL, Lean, Coq, or friends, you already know the key point: a proof assistant is not impressed by eloquence. It accepts only a proof that type-checks and reduces where it should. That strictness is exactly what current language models lack in natural language mode: they can be persuasive while being wrong.

The deep idea in many recent systems is to exploit a verifier as the ultimate critic. The model is allowed to be creative and messy, but every step gets forced through a tiny gate: does the checker accept it?

DeepMind's IMO result as a case study

DeepMind described a system that reached silver-medal level on the 2024 International Mathematical Olympiad by combining two complementary components: AlphaProof (focused on formal reasoning in Lean via reinforcement learning) and AlphaGeometry 2 (a neuro-symbolic geometry solver with a fast symbolic engine). That pairing matters: one part searches proofs in a formal language, the other has domain-specific geometry machinery.

Two details are worth noticing. First, the IMO problems were manually translated into a formal language before the systems could work on them, which highlights that "autoformalization" is still a major bottleneck. Second, once in the formal domain, the systems could spend serious compute time exploring proof space, and the correctness of any found proof is not a vibe—it is mechanically verified.

The real frontier: conjectures, not just solutions

Solving a fixed problem is already hard, but discovery is often about proposing the right intermediate statements. In practice, mathematicians don't move linearly from axioms to theorem; they invent lemmas, strengthen hypotheses, and adjust definitions until the landscape becomes navigable.

AI systems that live inside a formal environment (Lean, Isabelle, etc.) can in principle search not only for proofs but for useful stepping stones—lemmas that shorten proofs, reusable tactics, or alternative formulations that make automation feasible. That's where "math discovery" starts to feel real: not replacing insight, but mass-producing plausible local moves and letting the verifier keep only the ones that survive.

What this suggests for working mathematicians

- **Verification becomes the backbone**: any workflow that keeps the model tethered to a proof checker avoids the most damaging failure mode (confident nonsense).
- **Human time shifts**: less time spent on low-level bookkeeping and more time spent choosing definitions, deciding which subproblems matter, and interpreting what a formal proof is actually saying.
- **Autoformalization is strategic**: the moment “informal formal” becomes reliable, the space of problems accessible to these loops expands dramatically.

None of this guarantees new Fields-level theorems in the short term. But it does point to a plausible medium-term reality: mathematicians using AI the way programmers use compilers and fuzzers—tools that don’t supply meaning, but relentlessly enforce correctness while exploring huge spaces of possibilities.

Contributor: Alessandro Linzi