Mathematics in the Age of AI

Jose **Brox**

MANOLOFEST!

Universidade de Santiago de Compostela

Instituto de Investigación en Matemáticas

Universidad de Valladolid

25/06/26

A paradigm shift

The future of mathematics?

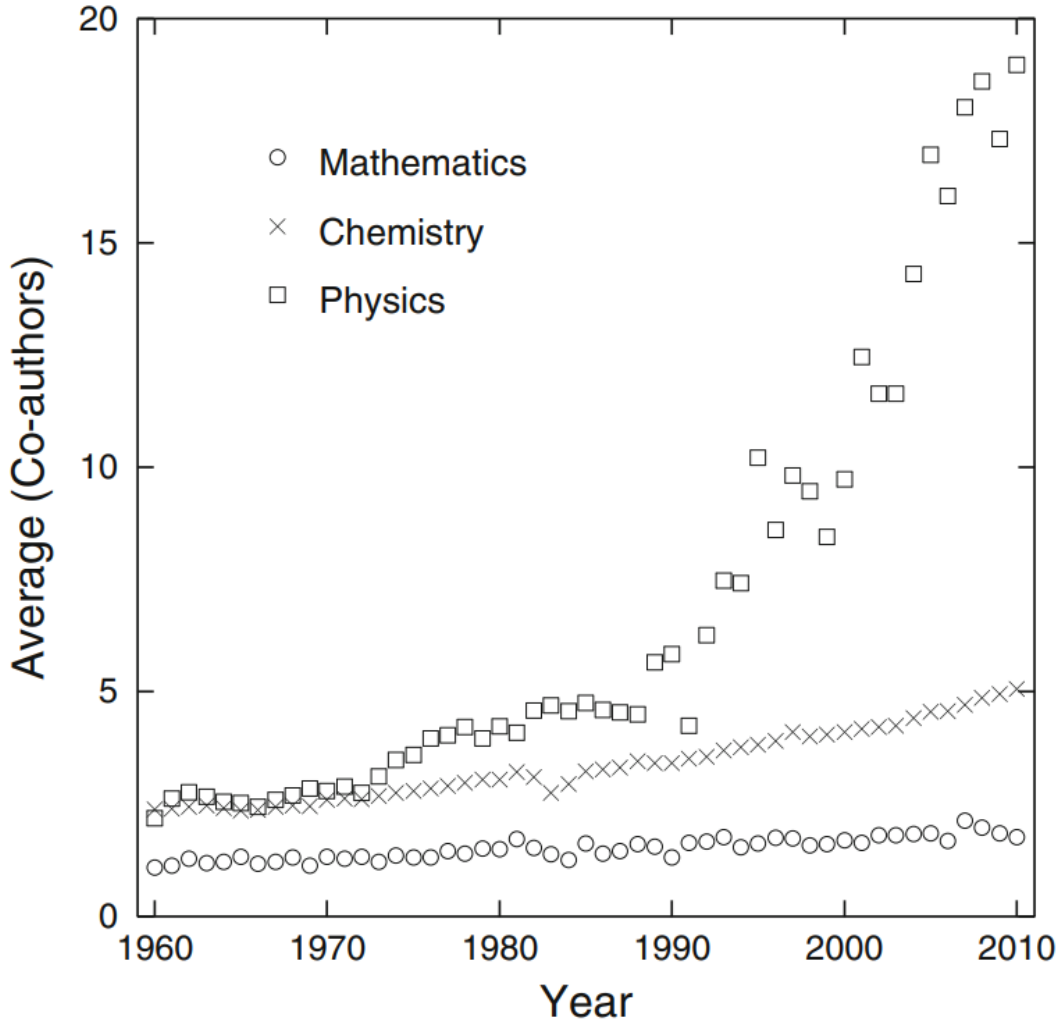
The everyday mathematician will be more like:

1. A **computer programmer** and a **software engineer**.
   - For collaboration, exploration, theorem proving, formalization, and reviewing,
   - by using collaborative platforms, interactive math software, automated provers, and proof assistants.
2. A **principal investigator** and a **thesis supervisor**.
   - For collaboration with generative AI tools
   - in all the previously mentioned tasks.

- Most of you probably already have both feet entering its waters (do you use **Overleaf**? **Mathematica**? Do you ask **ChatGPT**), but haven't entered deep waters yet.

Tao's vision

- How we conduct math research didn't change for centuries



Tao's vision

- **Large-scale and heterogeneous collaboration** are rare in mathematics, due to:
  1. **High entry barriers** (amateurs cannot contribute)
  2. **High specialization among areas** (lack of interdisciplinarity)
  3. **Rigorous proof standards** (AIs cannot contribute)
  4. **Non-scaleable workflows** (1-5 experts in the team)

Having **different contributors** with **different backgrounds and expertise** makes communication and validation difficult, producing a **loss of trust** in both the process and the result

Tao's vision

These handicaps can be surmounted thanks to new tools:

1. **Collaborative platforms with distributed version control**
2. **Formal proof assistants with automation**

Hence allowing:

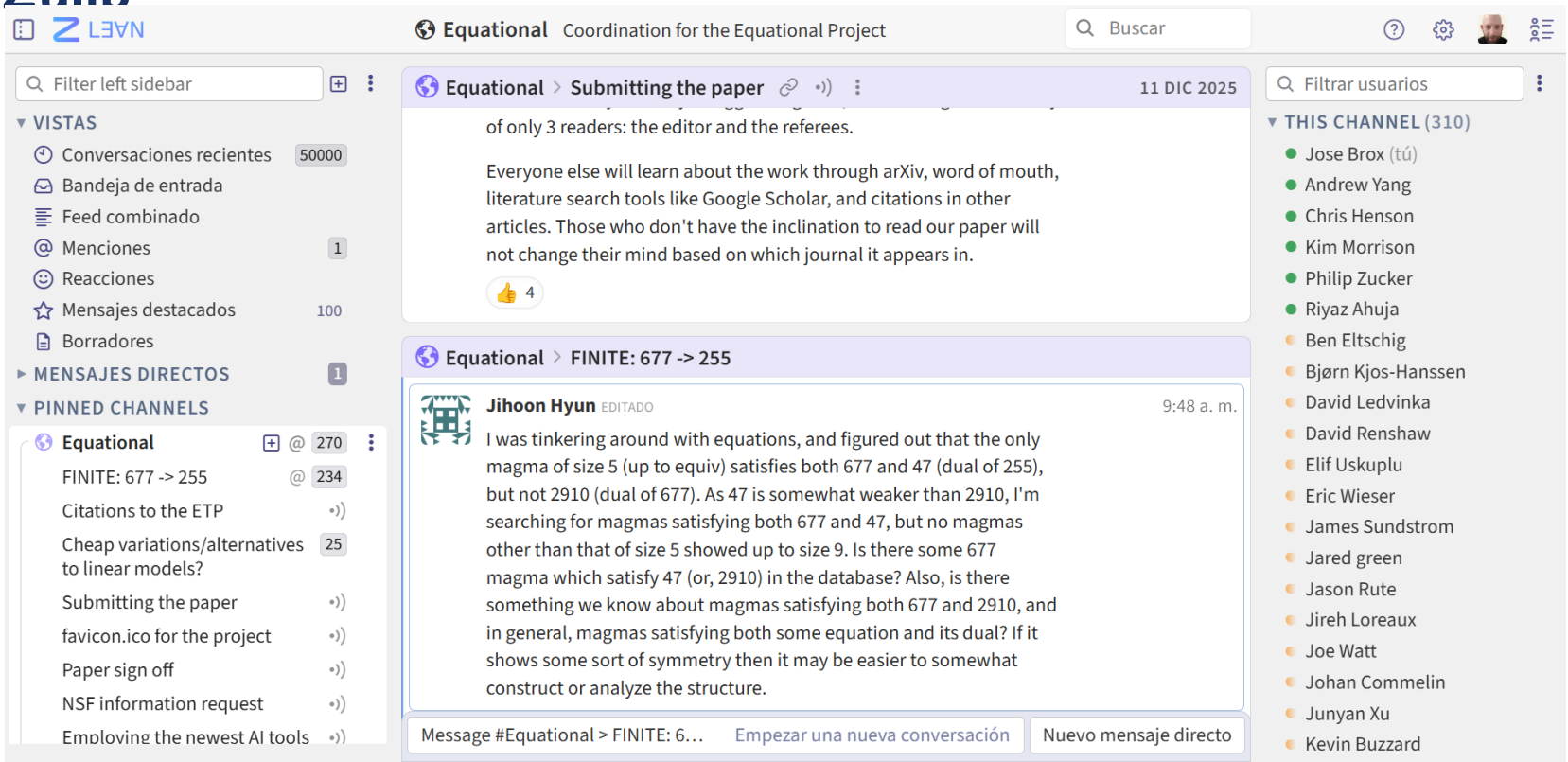
1. **Citizen mathematics**
2. **Participation of AI tools** (ATPs, LLMs)
3. **Trusted interaction between experts** from separate areas
4. **Large-scale** mathematical research projects
5. **Data-driven** mathematics
6. **Transparent dissemination** of both process and results

New tools

Collaboration

Informal communication chats

Zulip



- The core of the **discussion happens here**
- Allow **many threads**
- Allow **writing in LaTeX** and **code snippets**
- Allow **attaching files** and **links to other platforms**

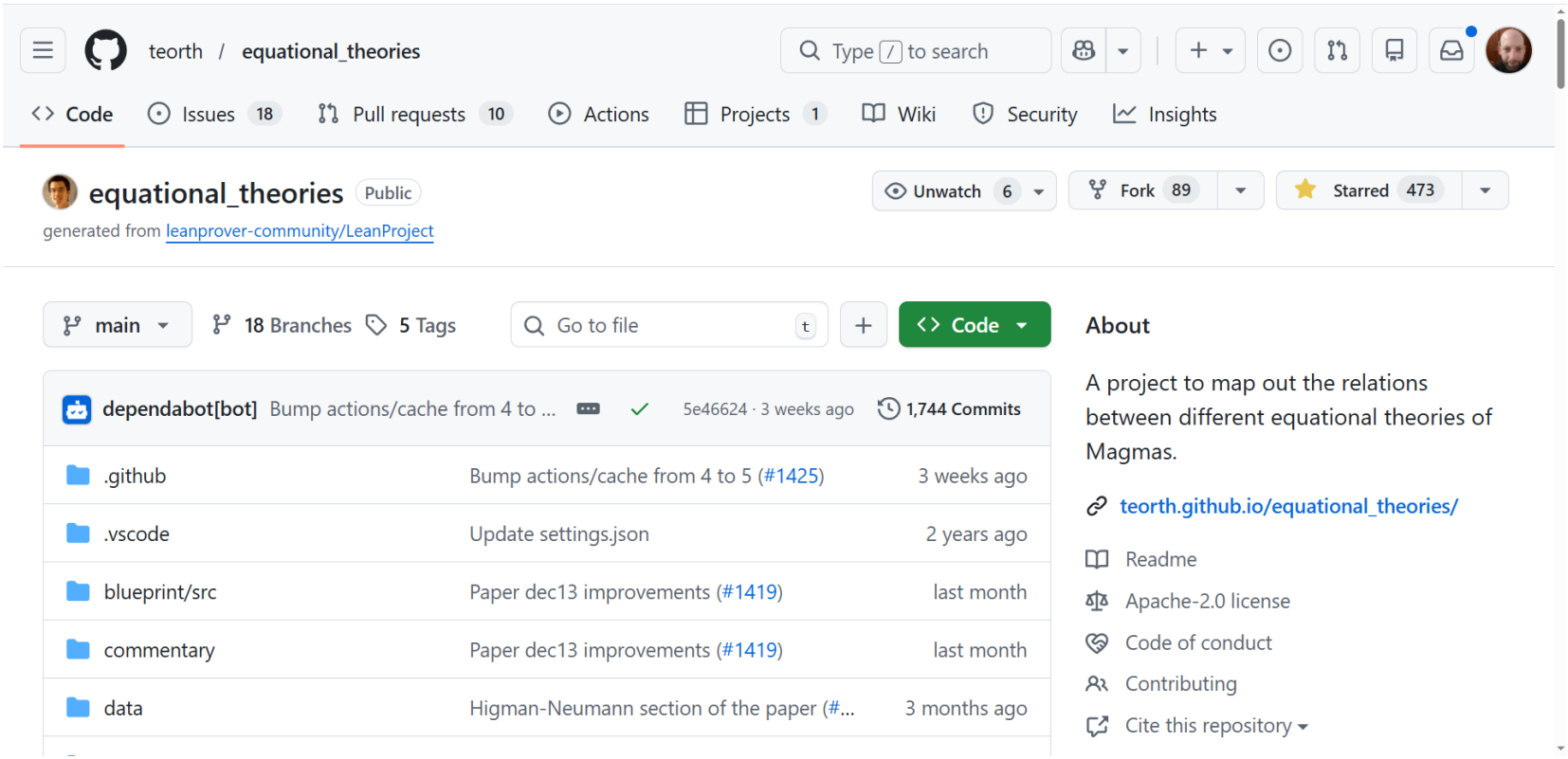
GIT platforms

with distributed version control:

- There is a **master copy** of the project, but
- **each collaborator also owns a full copy** of the project,
- which she **can modify** at will.
- **Changes can be merged** to the master copy,
- **after careful revision**, human or automated.
- Any **changes can be reverted**.
- **Conflicts** due to confluent changes **are detected and fixed**

Work in parallel, propose changes for the stable version,
wait for comments and conflicts, revise, merge

GitHub



- There is a main **repository**
- Each collaborator has its own **fork** repository
- Changes are proposed in **pull requests**
- Tasks to be done are listed as **issues**

New tools

Formalization

Formal proof assistants

- Proofs are **written in a formal language**,
- and then **verified by the assistant**.
- The **soundness** of the proof **depends on previous definitions and results**.
- There are **libraries** with previously formalized definitions and results.
- There are **tactics that can fill routine steps of the proofs**.

Write the proof in the formal language, base it on previous libraries, use tactics, wait for the assistant's veredict, revise

Lean

Powerful automation Mathematics

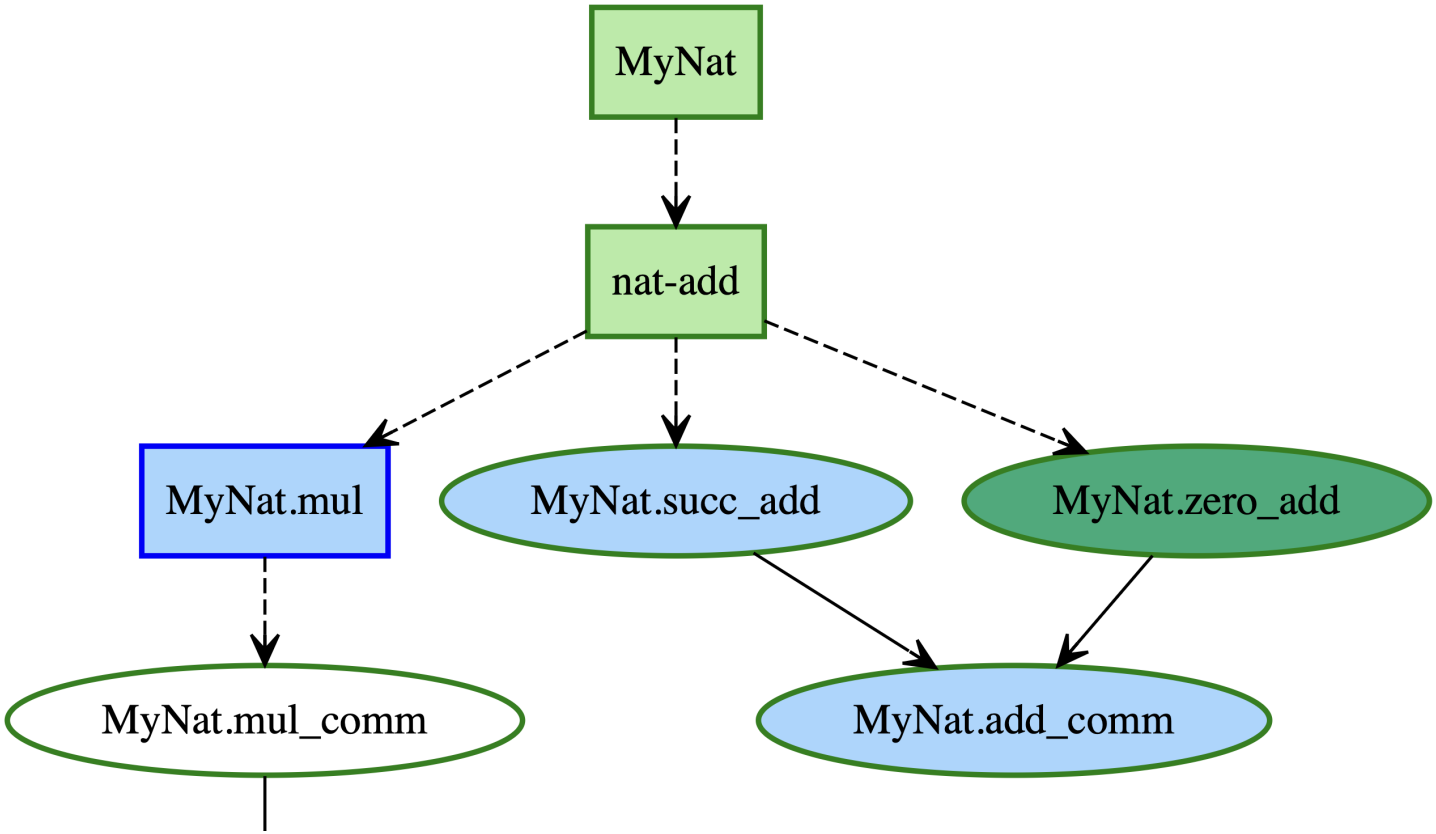
```
/-- A prime is a number larger than 1 with no trivial divisors -/
def IsPrime (n : Nat) := 1 < n ∧ ∀ k, 1 < k → k < n → ¬ k ∣ n

/-- Every number larger than 1 has a prime factor -/
theorem exists_prime_factor :
    ∀ n, 1 < n → ∃ k, IsPrime k ∧ k ∣ n := by
  intro n h1
  -- Either `n` is prime...
  by_cases hprime : IsPrime n
  · grind [Nat.dvd_refl]
  -- ... or it has a non-trivial divisor with a prime factor
  · obtain ⟨k, _⟩ : ∃ k, 1 < k ∧ k < n ∧ k ∣ n := by
      simp_all [IsPrime]
    obtain ⟨p, _, _⟩ := exists_prime_factor k (by grind)
    grind [Nat.dvd_trans]
```

A definition of prime numbers and a proof that there are infinitely many prime numbers.

- There is big **Mathlib** library (which we don't use)
- **Integrates well with GitHub**
- **Beware of external tactics**, as `native_decide` (we can cheat)

Blueprint

Intermediate step between formal and informal proofs:



- Allows to follow **what has yet to be formalized**
- Facilitates **working in parallel**

Benefits of formalization (Jeremy Avigad)

1. Verify theorems
2. Correct errors
3. Gain intuition
4. Build libraries
5. Search for definitions and theorems
6. Refactor proofs
7. Refactor libraries
8. Design concepts
9. Communicate
10. Collaborate
11. Manage complexity
12. Manage the literature
13. Teach
14. Improve access
15. Use mathematical computation
16. Use automated reasoning
17. Use machine learning
18. Support a synthesis of machine learning and symbolic AI

J. Avigad (2024). Mathematics and the formal turn. **Bull. AMS 61**

New tools

Automatization

Dynamic databases

Number	Description	Best lower bound	Best upper
1a	Sidon set autocorrelation constant	1.2802	1.502862
1b	Erdős minimum overlap constant	0.379005	0.380868
2	Crouzeix constant	2	$1+\sqrt{2}\approx 2.4142$
3a	Gyamarti-Hennecart-Ruzsa sum-difference constant	1.1740744	1.33333
3b	Kakeya sums-differences constant	>1.77898	1.83333
3c	4-slope Kakeya-type sum-difference constant	1.67473389	1.75
4a	Cap set constant	2.2202	2.756
4b	Furstenberg–Sárközy square-difference constant	0.733412	1
5a	Sidon set size constant	0	0.97633
5b	Sidon set density inside (4,5) sets	0.5294	0.5714

- Problems are **ordered by logical dependency**
- **A new value** of one constant **updates all** downstream values

Automatization

Symbolic AI

Automated theorem provers (ATPs)

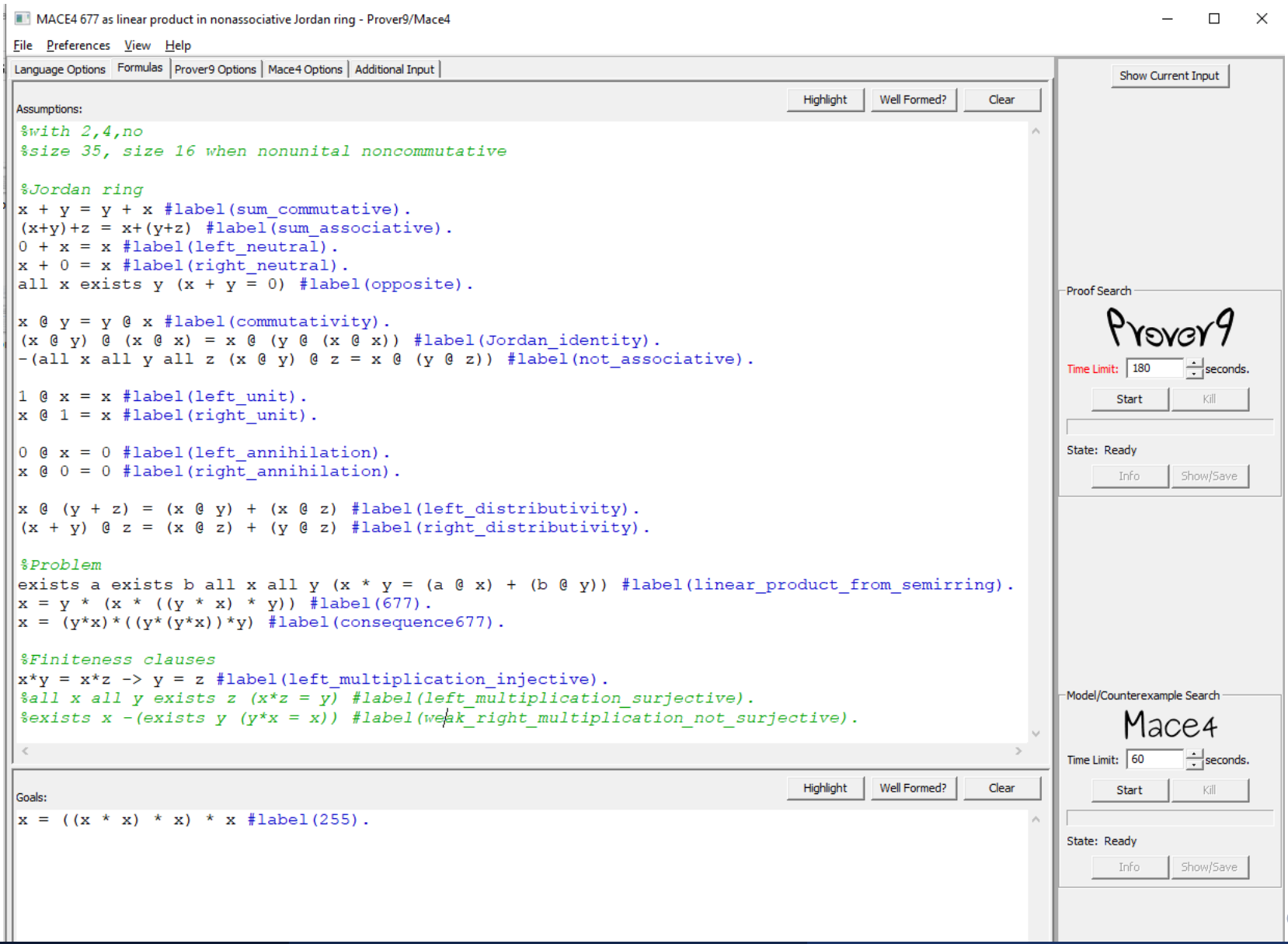
- We enter some axioms and a hypothesis or conjecture
- ATPs **generate proofs**
- They are also able to **find counterexamples**
- They **work with first-order logic**, or perhaps higher-order logic
- They can produce **proofs with many formulas, very long ones**,
- which can be **difficult for humans to understand**

There are interfaces that translate
the output of ATPs into Lean format

Vampire

```
`C(base) josebrox@HISOKA:~./vampire5 -t 10 -sa lrs -t 10 -awr 1:500 -fde none -nwc 0.5 -p tptp
% Running in auto input_syntax mode. Trying TPTP
% Refutation found. Thanks to Tanya!
% SZS status Theorem for TPTP 102744082 implies right mult injective
% SZS output start Proof for TPTP 102744082 implies right mult injective
fof(f488040,plain,(
  $false),
  inference(subsumption_resolution,[],[f487741,f15])).
fof(f15,plain,(
  sK1 != sK2),
  inference(cnf_transformation,[],[f11])).
fof(f11,plain,(
  sK1 != sK2 & m(sK1,sK0) = m(sK2,sK0)),
  inference(skolemisation,[status(esa),new_symbols(skolem,[sK0,sK1,sK2])],[f8,f10])).
fof(f10,plain,(
  ? [X0,X1,X2] : (X1 != X2 & m(X1,X0) = m(X2,X0)) => (sK1 != sK2 & m(sK1,sK0) = m(sK2,sK0))),
  introduced(definition,[],[choice_axiom])).
fof(f8,plain,(
  ? [X0,X1,X2] : (X1 != X2 & m(X1,X0) = m(X2,X0))),
  inference(ennf_transformation,[],[f4])).
fof(f4,negated_conjecture,(
  ~! [X0,X1,X2] : (m(X1,X0) = m(X2,X0) => X1 = X2)),
  inference(negated_conjecture,[status(cth)],[f3])).
fof(f3,conjecture,(
  ! [X0,X1,X2] : (m(X1,X0) = m(X2,X0) => X1 = X2)),
  file('TPTP 102744082 implies right mult injective.p',unknown)).
fof(f487741,plain,(
  sK1 = sK2),
  inference(forward_demodulation,[],[f487109,f481825])).
fof(f481825,plain,(
  ( ! [X3,X0,X4] : (m(m(X3,X4),m(s(X0),X4)) = X3) )),
  inference(forward_demodulation,[],[f481078,f390316])).
```

Prover9/Mace4



Automatization

Generative AI

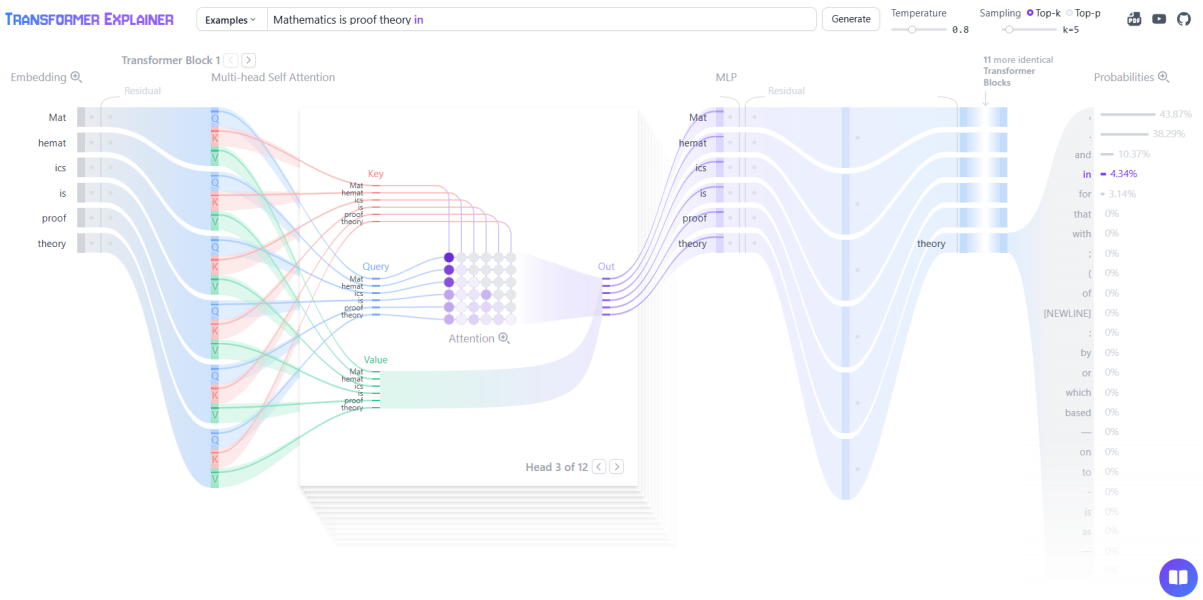
Large language models (LLMs)

- Originally, **words were read sequentially**
- The **processing of a word depended on the immediately preceding one** and on the state of the system

The stochastic-parrots fallacy

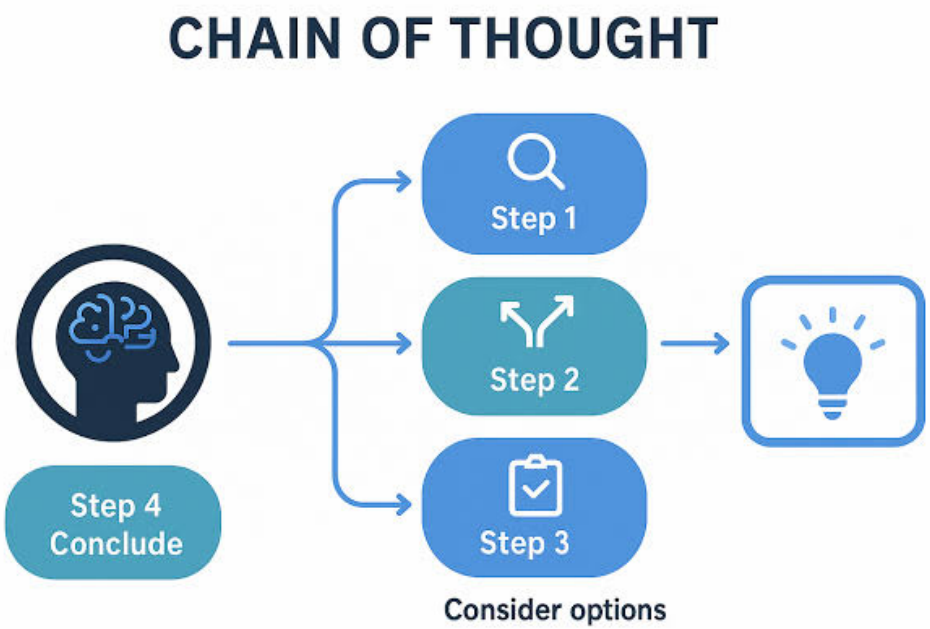
Transformer

- Each word is **compared with the rest** simultaneously



- **Scales with input data, computing capacity, and number of parameters** (neuron weights)

Chain of thought



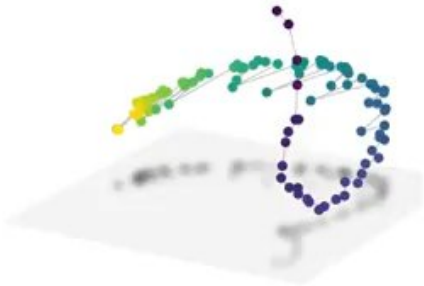
- The LLM **writes out intermediate reasoning steps**
- It **minimizes the error in each of them**
- It **self-checks and backtracks** on its own

Coding AI

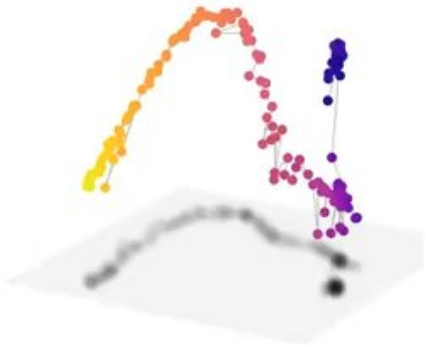
- The LLM is **able to write code**
- It correctly **decides the code given the specification**
- It decides **its own specification**

Agentic AI

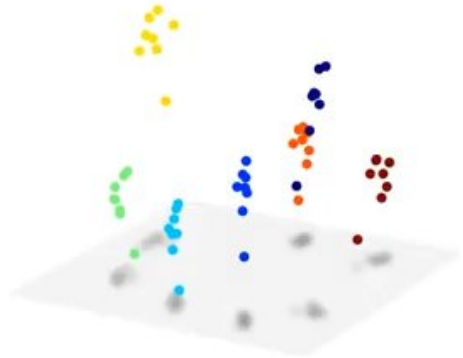
- We give the AI **access to the environment**
- It can receive **feedback from its own actions**
- We can span **multiple agents** and make them **interact**



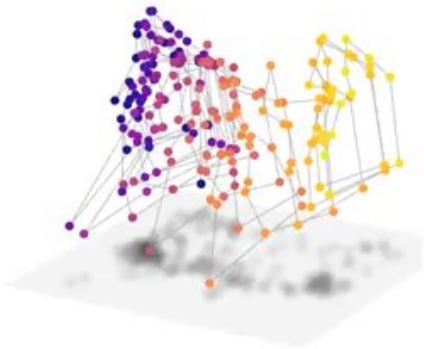
Age



Temperature



Day



Year

Geography

Colors

MP4

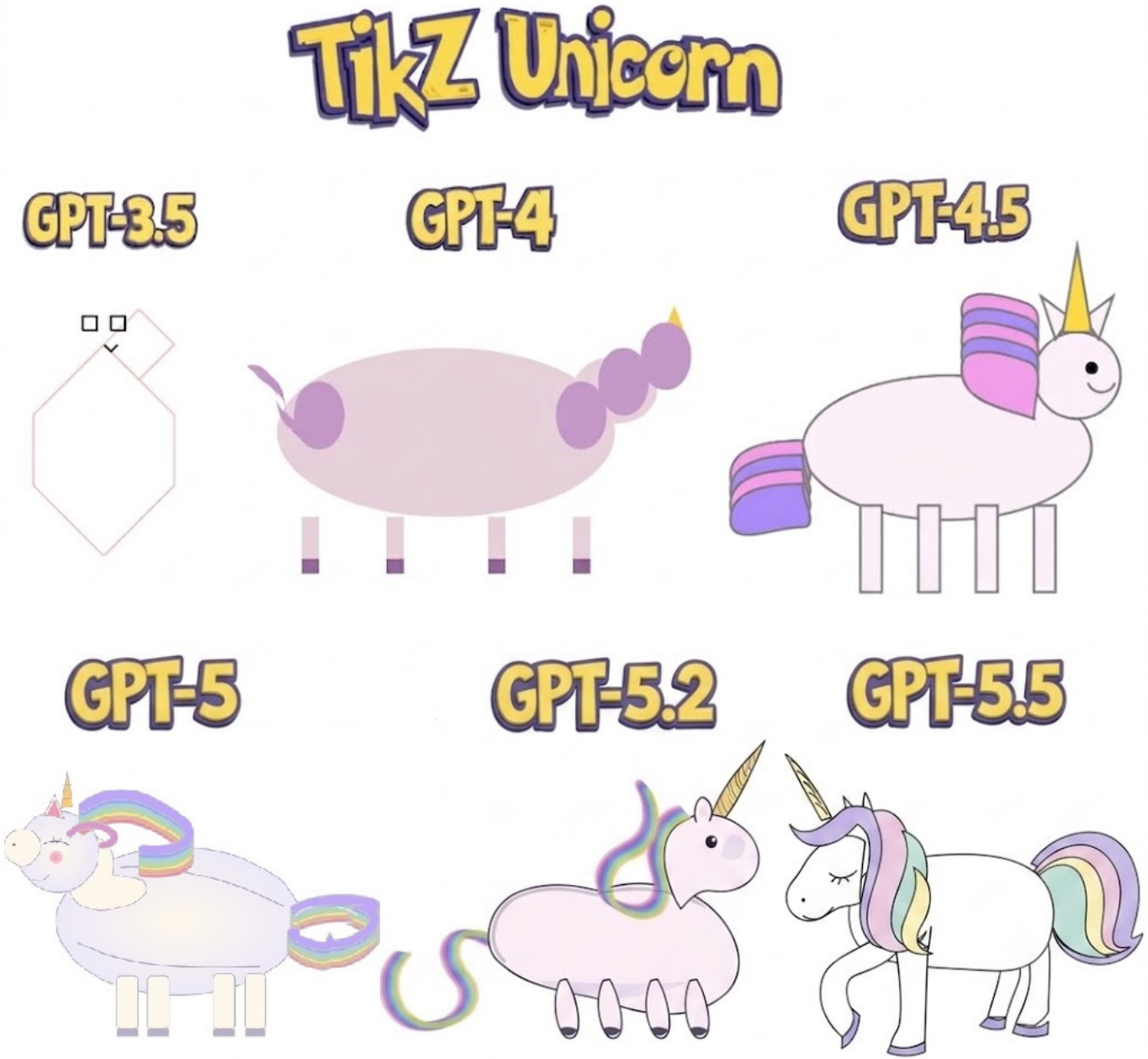
Primes poem

Prompt: Can you write a proof that there are infinitely many primes, with every line that rhymes?

GPT-4:

```
Yes, I think I can,
Though it might take a clever plan.
I'll start by noting Euclid's proof,
Which shows that primes aren't just aloof.
Assume we have a finite list,
Of primes, and that none have been missed.
Multiply them all together,
And add one, just to be clever.
The result will be a number,
That has no prime factors, oh wonder!
But every number has a prime,
That divides it, it's just a matter of time.
So we've found a contradiction,
And our finite list needs eviction.
There must be infinitely many primes,
And that's the end of my rhyming lines.s
```

S. Bubeck et al (2023). Sparks of artificial general intelligence. **arXiv**



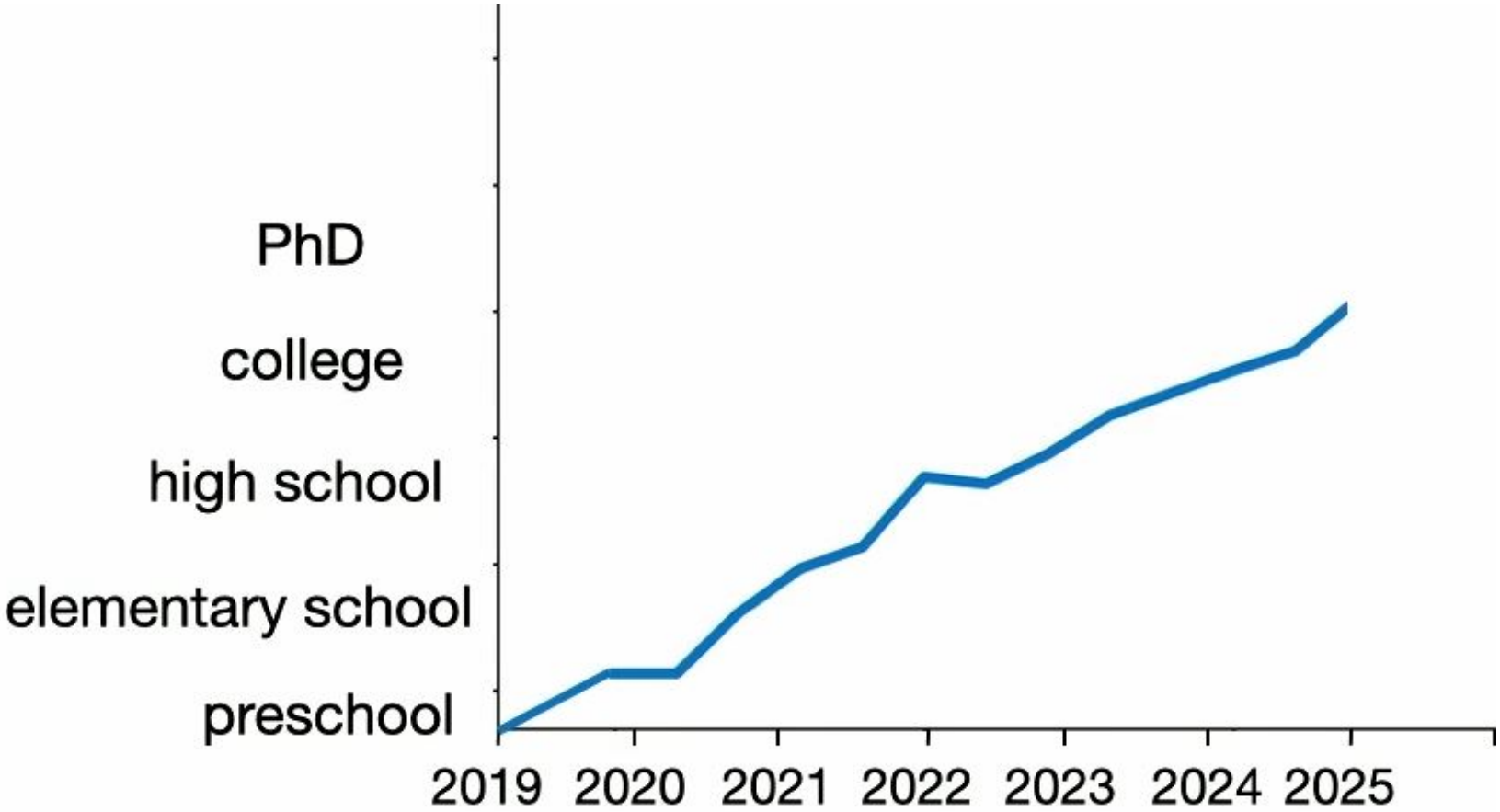
Year	Reasoning & Discovery	Formalization & Verification
2020	GPT-3: reliable **2-digit arithmetic (emergent)**	**Liquid Tensor Experiment** launched — Clausen–Scholze condensed math (done 2022)
2021	—	miniF2F: first cross-system formal benchmark
2022	Chain-of-thought + Minerva (∼50% MATH); AlphaTensor matmul records	**First LLM autoformalization**: ∼25% of competition problems to Isabelle
2023	FunSearch: first genuinely new construction (cap-set bound)	**Polynomial Freiman–Ruzsa conjecture** formalized in Lean (Tao et al., ∼3 weeks)
2024	AlphaProof + AlphaGeometry 2: **IMO silver** (28/42); FrontierMath; o1/o3	AlphaProof: ∼100M autoformalized theorems

	Reasoning & Discovery	Formalization & Verification
2025		
May	AlphaEvolve: 4×4 matmul in 48 mults (**improves Strassen**)	**Equational Theories Project** (Tao): 4694 laws
Jul	**IMO gold in natural language** (Gemini, OpenAI): 35/42	**Harmonic's Aristotle: IMO gold in formal Lean**
Sept-Dec	**LLMs saturate Putnam**	Math Inc.'s Gauss: **Prime Number Theorem** formalized
2026		
Jan	Erdős "long tail" solved autonomously (GPT-5.2), Tao-checked	
Feb	**First Proof** (Hairer et al): 6-8/10 on fresh research lemmas	Gauss + Viazovska: **sphere packing** formalized, dims 8 & 24
Mar	AlphaEvolve: lower bounds for 9 classical **Ramsey numbers**; Claude solves **Knuth problem**; Aletheia solves a **Kirby list problem**	
Apr	**Erdős #1196** (GPT-5.4 Pro): good mathematicians tried 7 years	
May	**Gower's PhD problem solved**; **Unit-distance conjecture** disproved	Aleph Prover: unit-distance disproof formalized in Lean 4

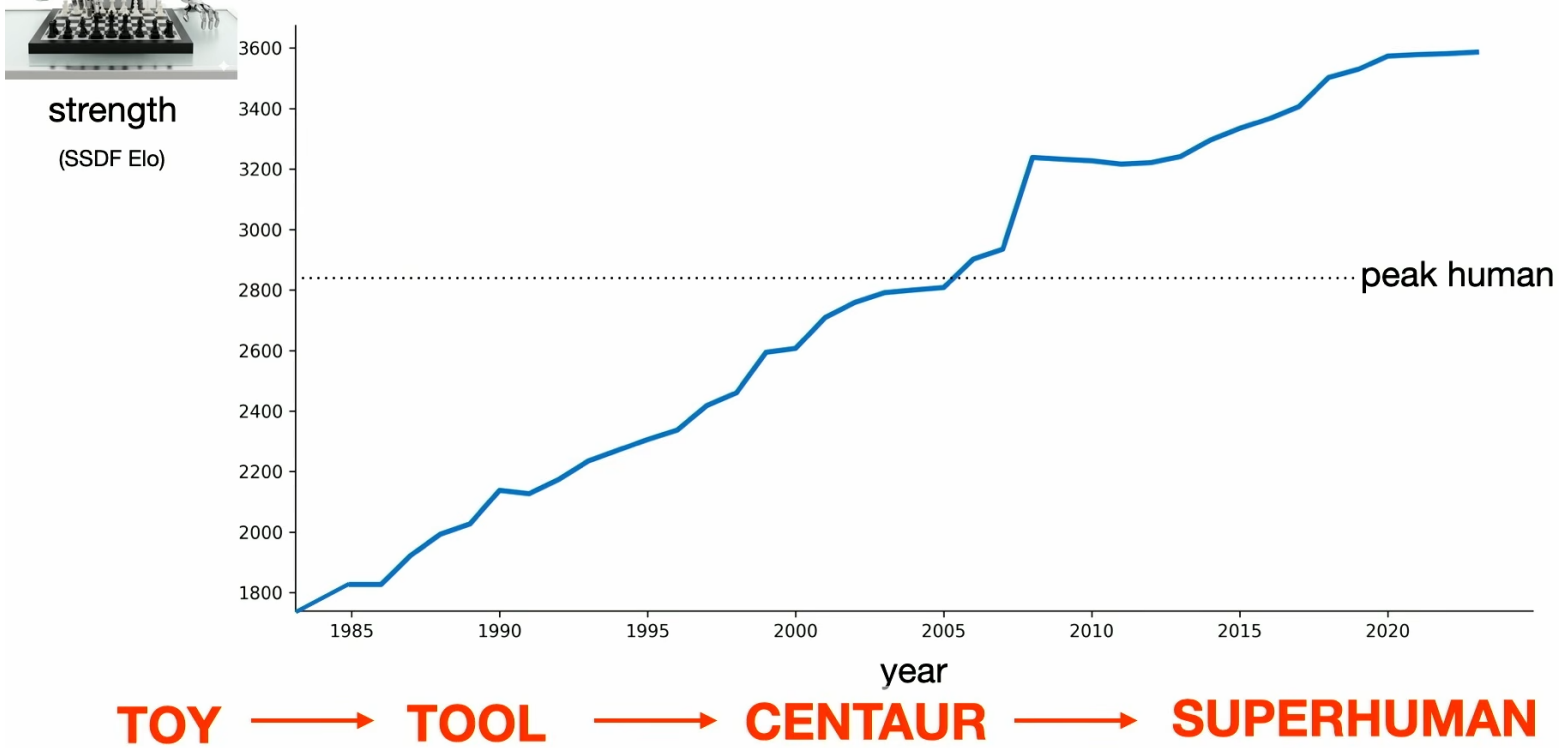
Sociology of the future

Consequences of AI for research

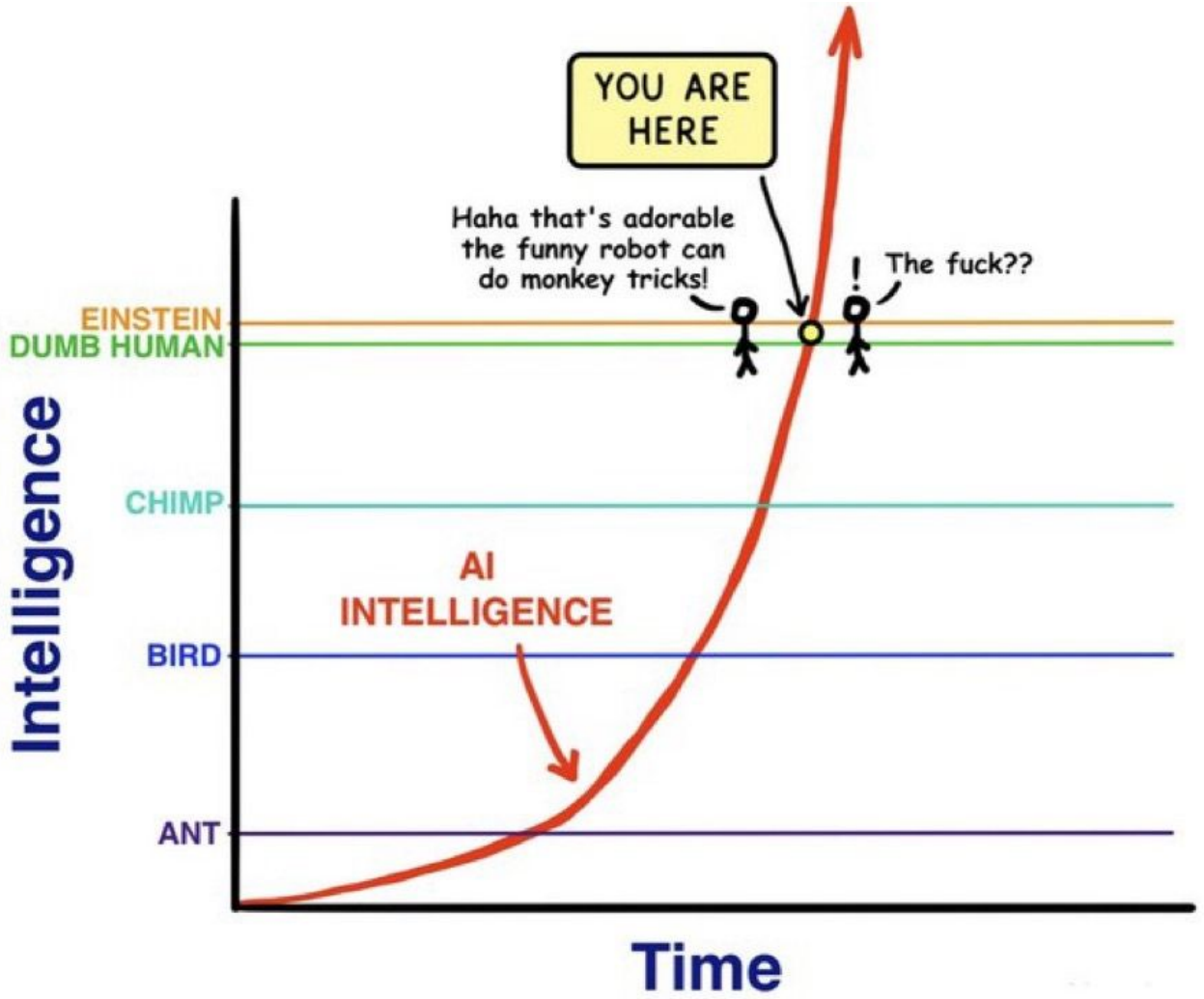
LLMs math skills



LLMs chess skills

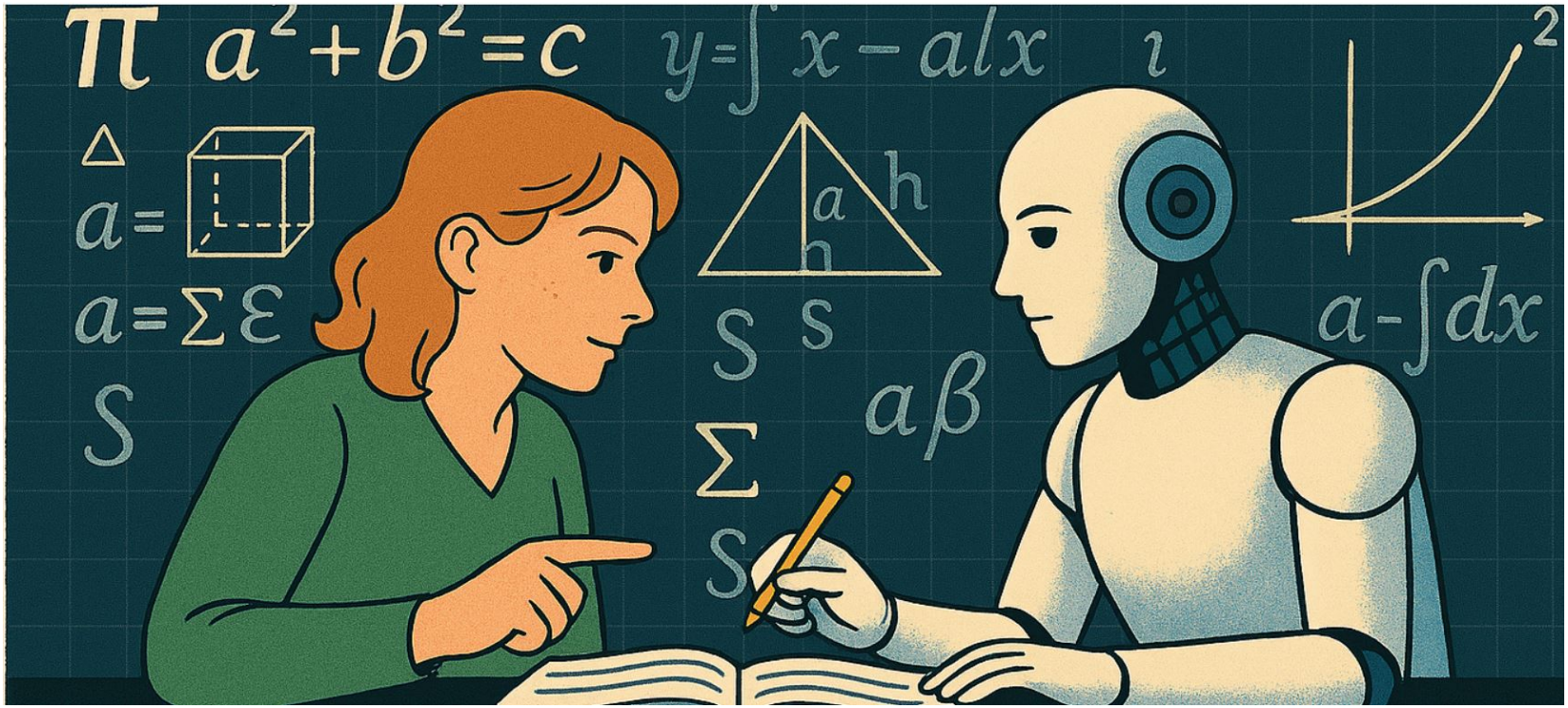


Stochastic parrots, monkey tricks?



Consequences for research

Option 1:

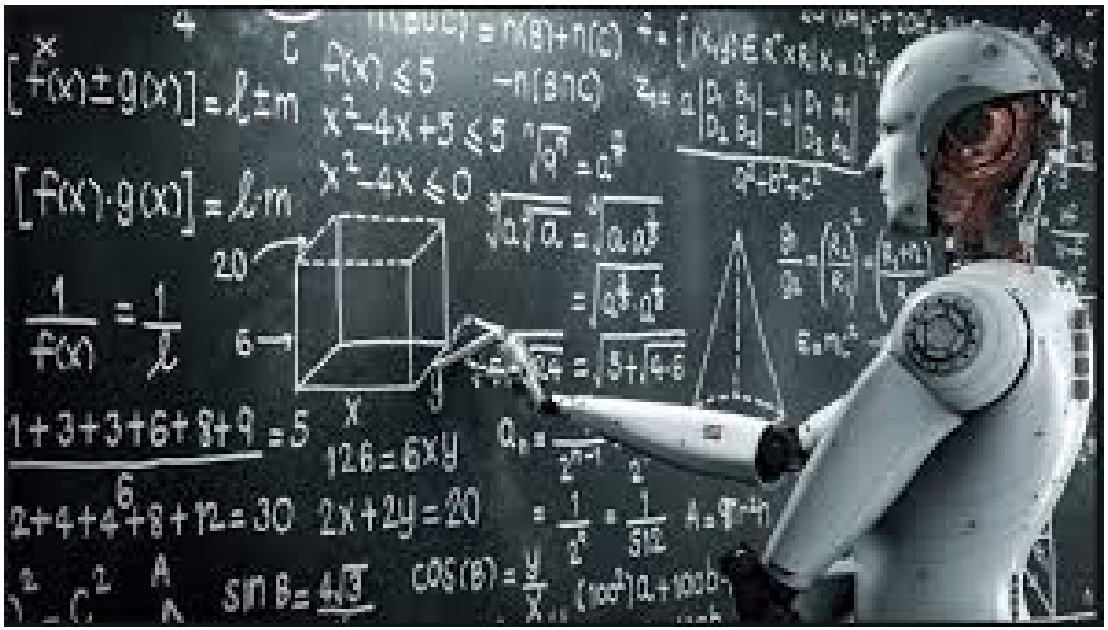


Consequences for research. Option 1

- **Losing funding** sources
- **Journals requiring formalized proofs**
- Repositories cannot keep up
- **AI-generated preprint servers**
- **AI reviewers**
- Do **papers and journals make sense?**
- Does **attribution make sense?**
- **Learning very quickly** from AI
- **Mathematicians working on more areas**
- Will **high-level LLMs be available to everyone**, reserved for a few privileged universities, or only for the company and its clients?
- Do **PhDs have a future?**

Consequences for research

Option 2:



- Let us not be naive about our chances in this case...

Play with sound

CONGRATULATIONS, MANOLO!