Gridspace

GRIDSPACE IAP 2024 LECTURE 7
Can LLMs do math?

January 29, 2024

No

TLDR

- LLMs **can't prove interesting theorems** on their own yet
- LLMs can already help us **verify interesting theorems**

PLAN FOR TODAY

- What kind of math? The kind I care about: proofs.
- Different approaches/speculations about having llms do math
  - **FunSearch**
  - **AlphaGeometry**
  - **Formalization of Polynomial Freiman-Rusza conjecture**
- There will be formul**ae**

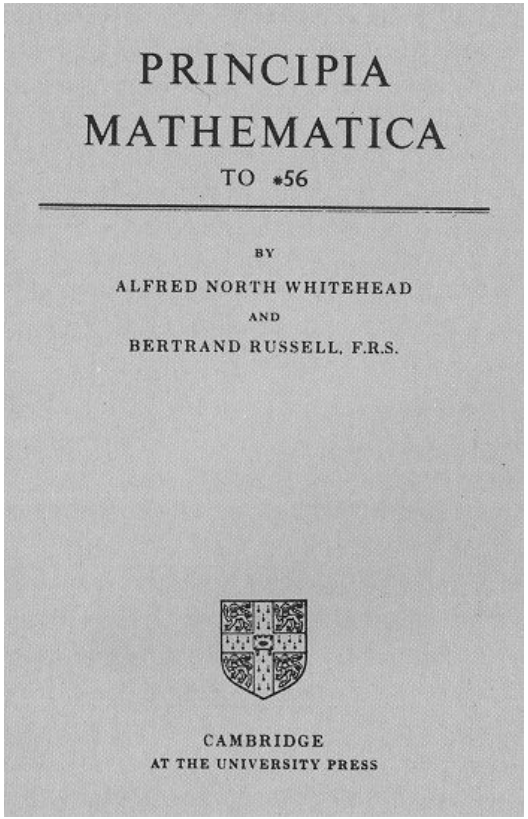
KINDS OF MATH

- Compute **18998*23423** ❌
- Write an **algorithm to sort** a list of numbers ❌
- **Prove** that **2^{½} is irrational** (not a ratio of whole numbers) ✅

PROOFS VS COMPUTATION

- Computation is not treated as something that we can be sure is correct.
- One might provide steps and rationale, but in the end we just measure accuracy.
- **A correctly written mathematical proof cannot be wrong.***
- That's the platonic ideal, but in reality, what counts as a math proof is actually...

PROOFS VS COMPUTATION


PRINCIPIA
MATHEMATICA
TO *56

BY
ALFRED NORTH WHITEHEAD
AND
BERTRAND RUSSELL, F.R.S.

CAMBRIDGE
AT THE UNIVERSITY PRESS

***54·43.** $\vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

[*51·231] $\equiv . \iota`x \cap \iota`y = \Lambda .$

[*13·12] $\equiv . \alpha \cap \beta = \Lambda$ (1)

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda$ (2)

$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash .$ Prop

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2.$

PROOFS VS COMPUTATION

But in reality...

PROOFS VS COMPUTATION

Bryan Birch is credited with once saying that he programmed in a very high-level programming language called "graduate student".

Share Cite Improve this answer

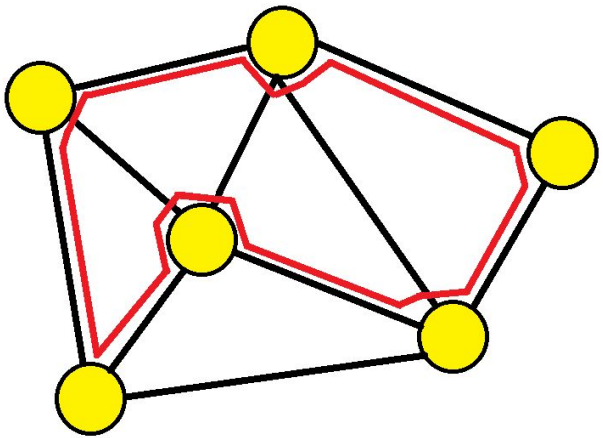
Follow

answered Jan 8, 2010 at 11:26

community wiki
Kevin Buzzard

Math is in NP

- **NP** = set of computational problems where solutions can be checked easily (but not necessarily found easily).



Math proofs are in NP

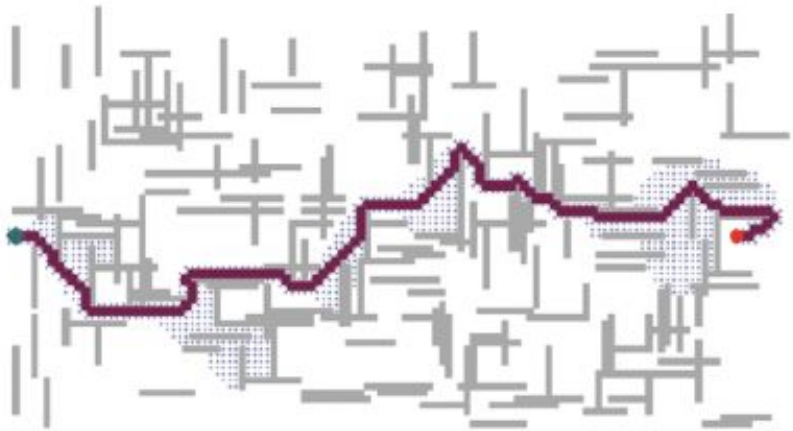
- Lower-level statements are often assumed.
- You can just cram in, ‘by the central limit theorem’...
-

Math is in NP

- **Finding** a proof is a **HARD** (sometimes impossible) search problem, hence Clay millennium prize and so on.
- **Checking** a proof is not (supposed to be) hard.
- **LLMs** could help us with either/both

Search and Verification

- **Planning:** LLMs can help search and score (ToT, Voyager)
- Using LLM in search
  - **Funsearch**
  - **Alphageometry**
- Using LLM in verification
  - **PFR theorem**



The search approach

SEARCH APPROACH

1. Pick a particular type of problem with a crux where **verification is easy**.
2. Use an LLM or other model to **search** for a solution.
3. Reap **incremental** reward

FunSearch

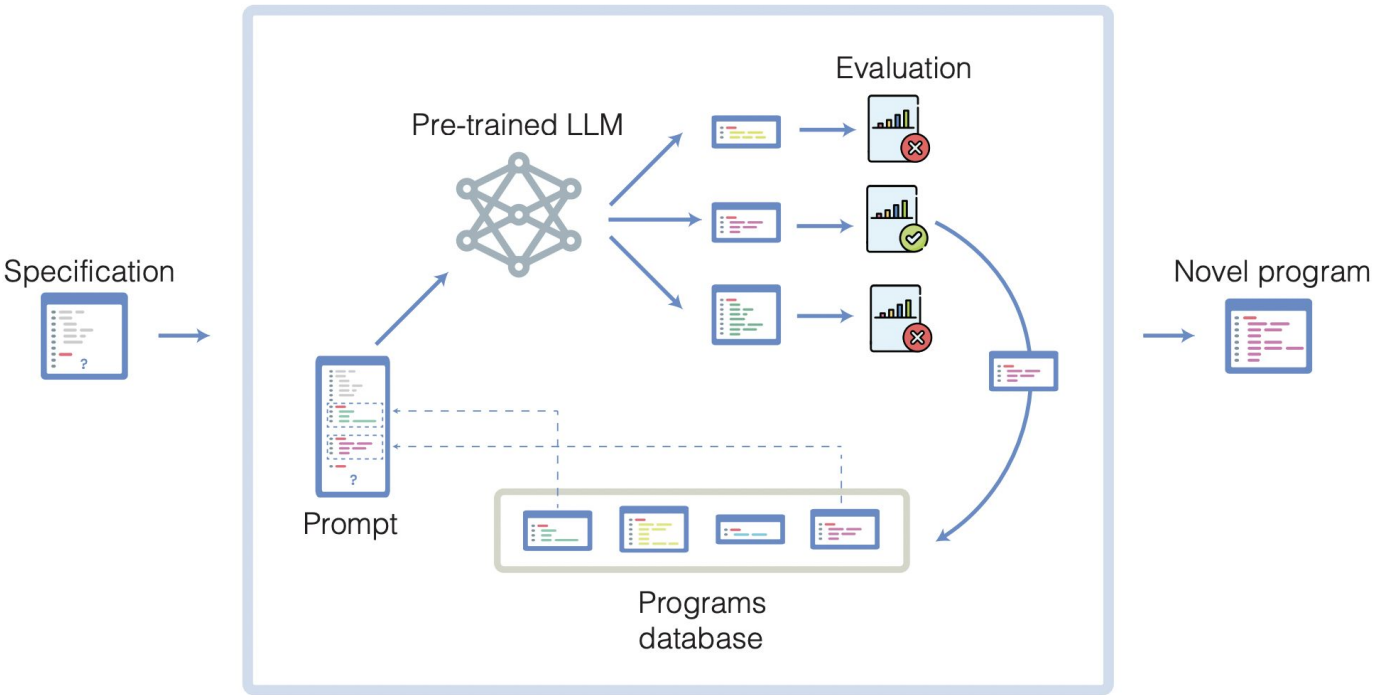
FUNSEARCH

Mathematical discoveries from program search with large language models

Bernardino Romera-Paredes[1*] Mohammadamin Barekatain[1*]
Alexander Novikov[1*] Matej Balog[1*] M. Pawan Kumar[1*]
Emilien Dupont[1*] Francisco J. R. Ruiz[1*] Jordan S. Ellenberg[2]
Pengming Wang[1] Omar Fawzi[3] Pushmeet Kohli[1] Alhussein Fawzi[1*]

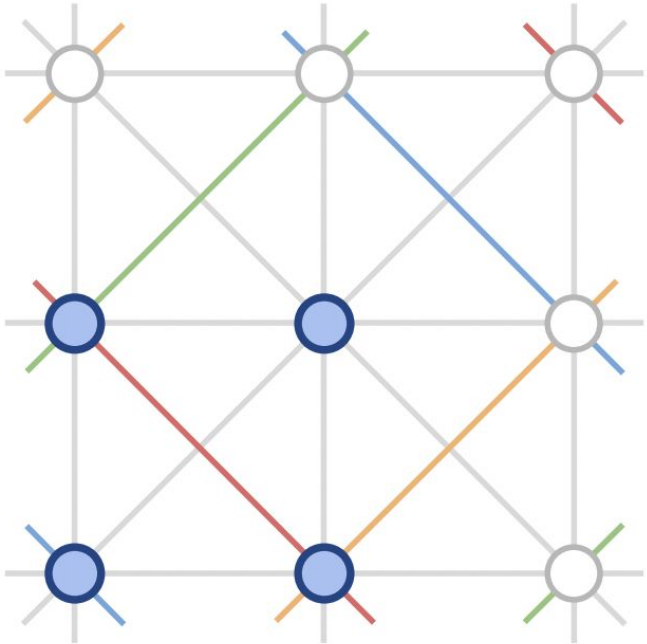
[1]Google DeepMind, London, UK
[2]University of Wisconsin-Madison, Madison, Wisconsin, USA
[3]Université de Lyon (Inria, ENS Lyon, UCBL, LIP), Lyon, France

FUNSEARCH



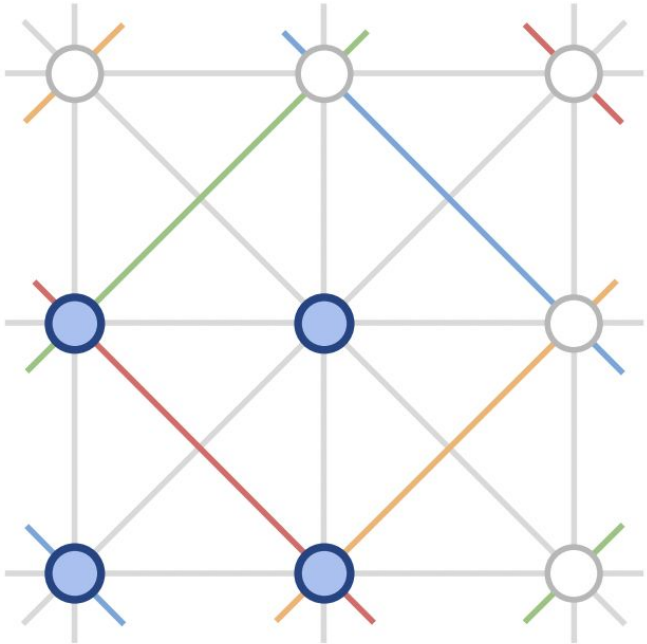
CAPSET PROBLEM

- A **capset** is a set of vectors in **{0, 1, 2}^n** of which **no 3 elements sum to 0** (mod 3)
- **Question:** what's **c_n**, size of largest capset in dimension **n**? (2^n <= c_n <= 3^n)
- **Why care?** Helps with other math problems that actually matter. **Terry Tao likes it.**



CAPSET PROBLEM

- capacity **C := sup_{n -> infty} c_n^{1/n}?**
- There's a special type of capset called **admissible** which can be bootstrapped to higher **n** to lower bound C
- **finite capset** -> **bound C**



CAPSET PROBLEM

- LLM searched for programs which were **constrained** to output admissible sets
- **Eval:** size of capset

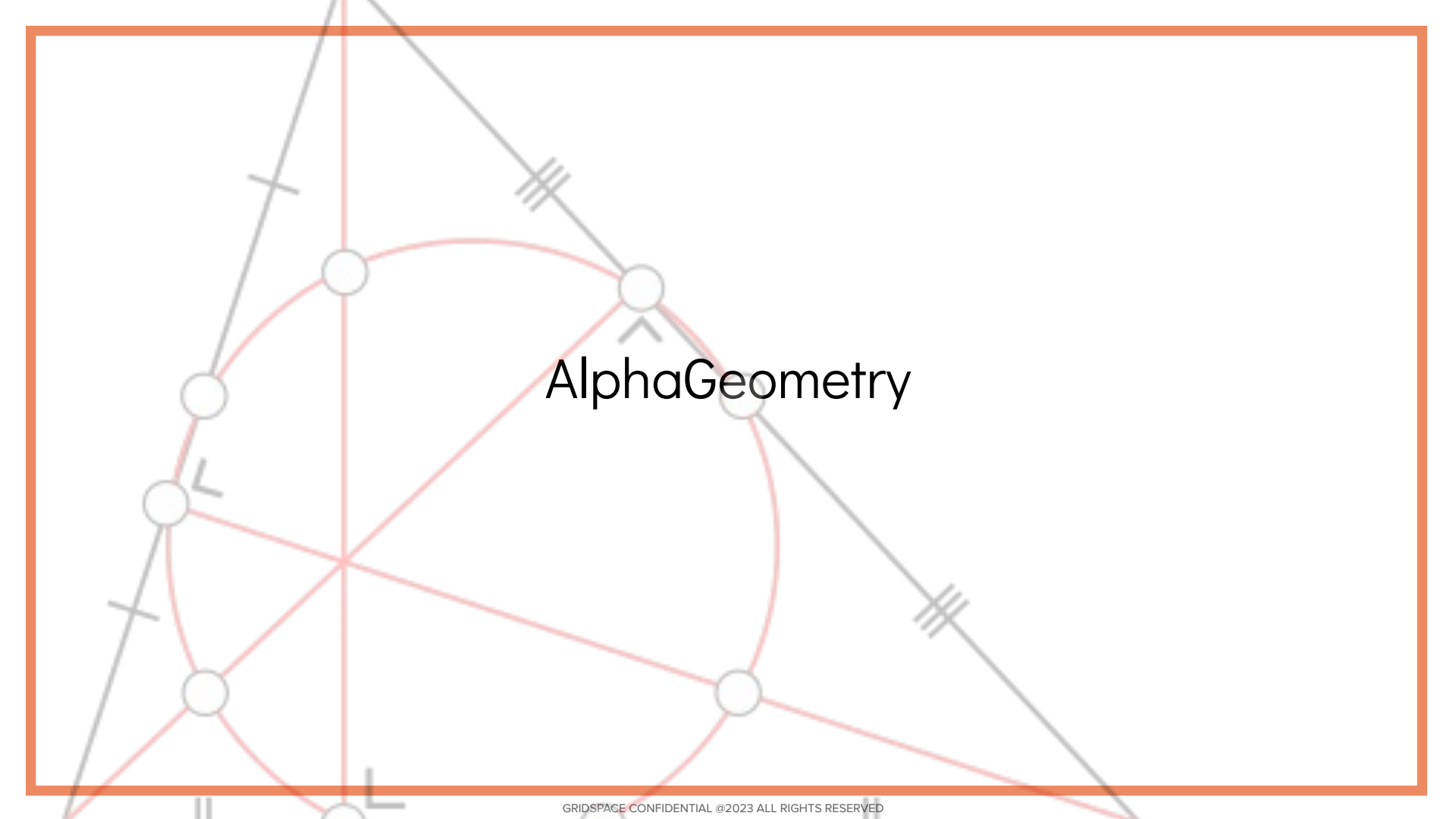
Bound on C	Admissible set ingredient	Source
2.2101	$\mathcal{I}(90, 89)$	(Calderbank and Fishburn, 1994)
2.2173	$\mathcal{I}(10, 5)$	(Edel, 2004)
2.2180	$\mathcal{I}(11, 7)$	(Tyrrell, 2022)
2.2184	$\mathcal{I}(12, 7)$	*FunSearch*
2.2194	$\mathcal{I}(15, 10)$	*FunSearch*
2.2202	$\mathcal{A}(24, 17)$	*FunSearch*

FUNSEARCH TAKEAWAY

- What's neat about this?
- Don't look through all **2^{3^n}** subsets
- search over **low Kolmogorov complexity** elements of the search space.
- After all, you don't learn much from an incompressible set.

FUNSEARCH TAKEAWAY

- After **lots of human effort** to find a way to bootstrap finite solutions into general ones, llms can help us search for these finite solutions
- **LLMs/DNNs** used to provide an **extra 5-10 percent bump** on poorly modeled part of a problem
    - **Deepmind's AlphaTensor** (faster matrix multiplication)
    - **Stockfish** (chess algorithm discussed by anthony)
    - DNN-based '**closure models**' for diff eq solving



AlphaGeometry

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Open access | Published: 17 January 2024

Solving olympiad geometry without human demonstrations

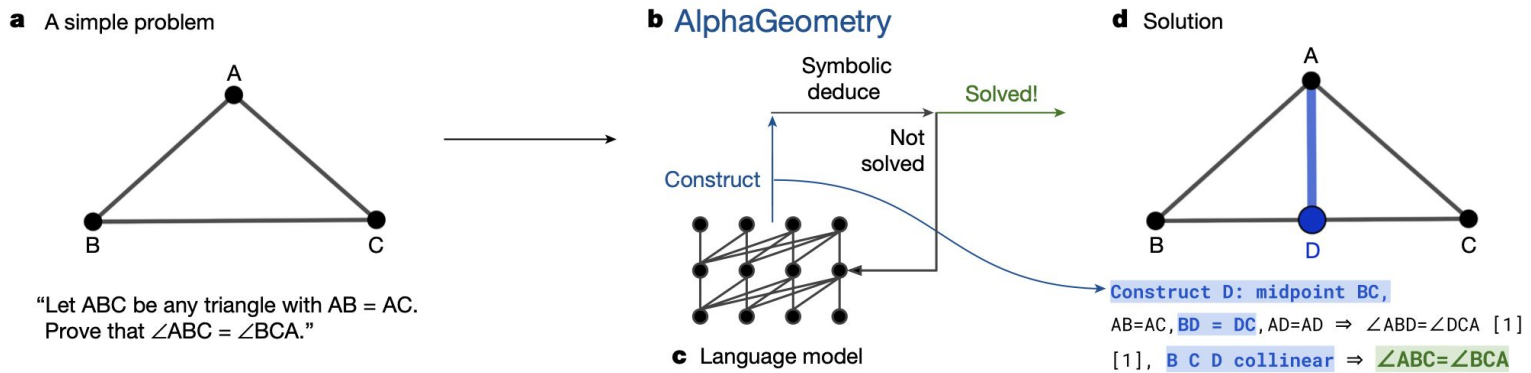
Trieu H. Trinh ✉, Yuhuai Wu, Quoc V. Le, He He & Thang Luong ✉

This paper is awesome

ALPHAGEOMETRY

- **Problems:** IMO geometry proofs
- **Inference:** use an LLM to propose proof steps, symbolic deduction to see if theorem reachable. Iterate.
- **Training:** Heavy use of synthetic data to train LLM

ALPHAGEOMETRY



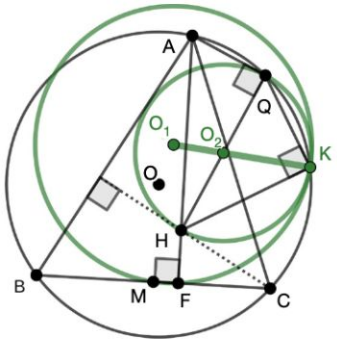
ALPHAGEOMETRY INFERENCE

- LLM uses a specialized formal language to propose **constructions,** considered to be the creative part of geometry proofs
- Symbolic engine (**Deductive Database + Algebraic Reasoning)** interprets construction to check whether the conclusion is shown.
- **Humans must translate** the problem statement into the formal language.

ALPHAGEOMETRY INFERENCE

e IMO 2015 P3

"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that QH ⊥ QA and let K be the point on (O) such that KH ⊥ KQ. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."



→ Alpha-Geometry →

f Solution

...

Construct D: midpoint BH [a]

[a], O_2 midpoint HQ ⇒ BQ // O_2D [20]

...

Construct G: midpoint HC [b] ...

∠GMD = ∠GO_2D ⇒ M O_2 G D cyclic [26]

...

[a],[b] ⇒ BC // DG [30]

...

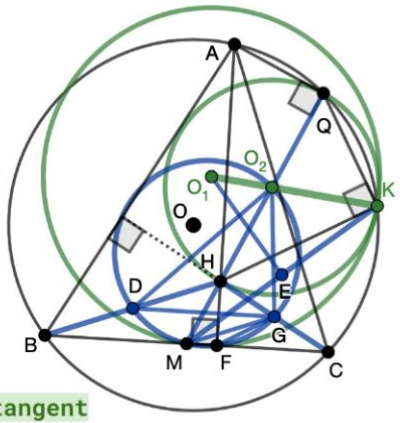
Construct E: midpoint MK [c]

..., **[c]** ⇒ ∠KFC = ∠KO_1E [104]

...

∠FKO_1=∠FKO_2 ⇒ KO_1 // KO_2 [109]

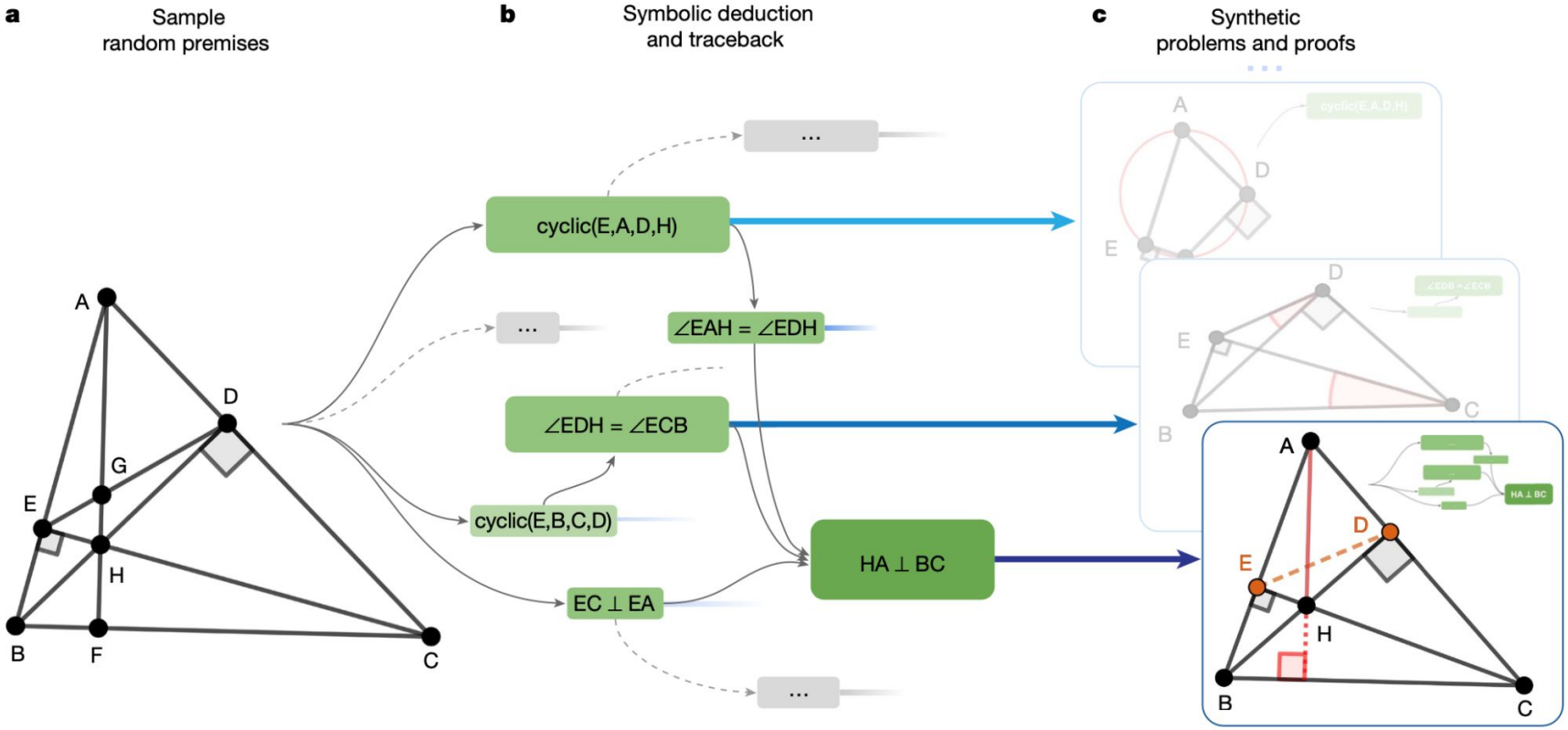
[109] ⇒ O_1O_2K collinear ⇒ $(O_1)(O_2)$ tangent



ALPHAGEOMETRY TRAINING

- Used the formal language to **sample random theorem premises**
- Used the symbolic engine to **prove a bunch of random stuff** from that
- For each conclusion, traceback algorithm extracted the **minimal premises** (the ones needed for the conclusion).
- unused premises where the objects still showed up in the symbolic proofs := **auxiliary constructions.**

ALPHAGEOMETRY TRAINING



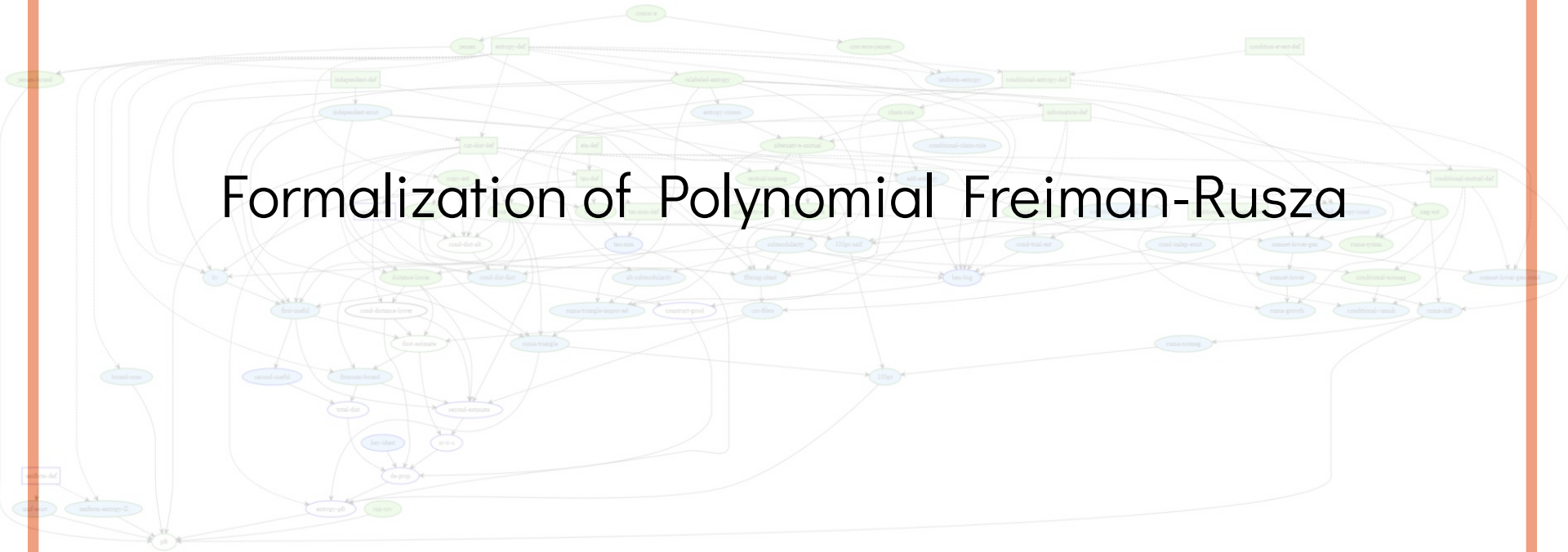
ALPHAGEOMETRY TAKEAWAY

- **High level:** inference llm + checker in a loop.
- **Synthetic data:**
  - ez(er) forward process **premise + constructions -> conclusions**
  - train llm to 'invert' part of that forward process
  - **premise + conclusion -> constructions**
- Still required huge human innovation (**DD + AR + traceback are SOTA on their own**), constrained setting (geometry only)

LESSON FOR NLP

- Synthetic data helps when you want to model the **inverse of a cheap but messy forward process**.
- Generating data from the forward process of humans saying their email address + ASR errors is easier than writing an algorithm to invert it
- **david spelled normally then d as in dog a v i n and then an i at gridspace dot com -> daviddavini@gridspace.com**

Formalization of Polynomial Freiman-Rusza



Formalization

- Is anybody trying to prove **BIG** conjectures with LLMs?
- Well, no. **Incremental/specialized** settings.
- Why jump the gun to letting LLMs have the ideas?
- Just let them **formalize** our ideas.
- The problem: human mathematicians barely formalize important results at all. **Until....**

PFR

ON A CONJECTURE OF MARTON

W. T. GOWERS, BEN GREEN, FREDDIE MANNERS, AND TERENCE TAO

ABSTRACT. We prove a conjecture of K. Marton, widely known as the polynomial Freiman–Ruzsa conjecture, in characteristic 2. The argument extends to odd characteristic, with details to follow in a subsequent paper.

PFR

Conjecture 1.1. *Suppose that* $A \subset \mathbf{F}_2^n$ *is a set with* $|A+A| \leqslant K|A|$. *Then* A *is covered by at most* $2K^C$ *cosets*[1] *of some subgroup* $H \leqslant \mathbf{F}_2^n$ *of size at most* $|A|$.

"A set that's almost closed under addition (mod 2) has to almost be affine (linear plus constant)"

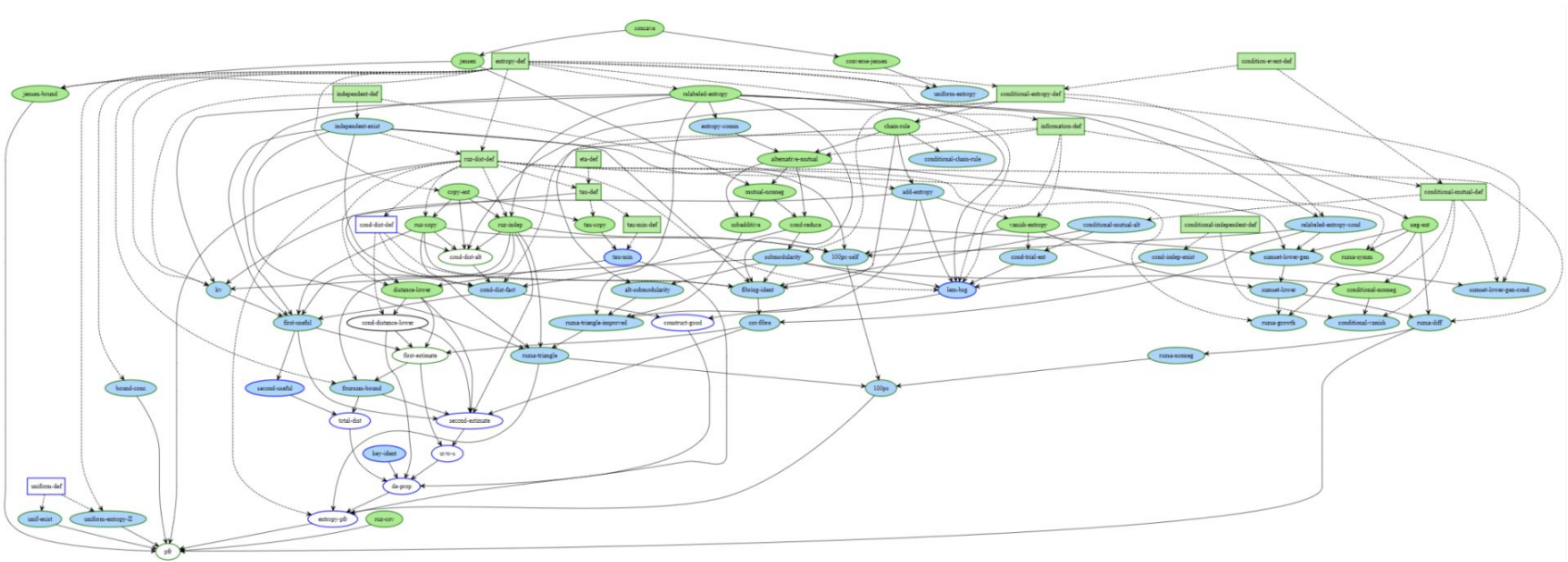
Theorem 1.2. *Conjecture 1.1 is true with* $C = 12$.

1 month later… It's formalized!

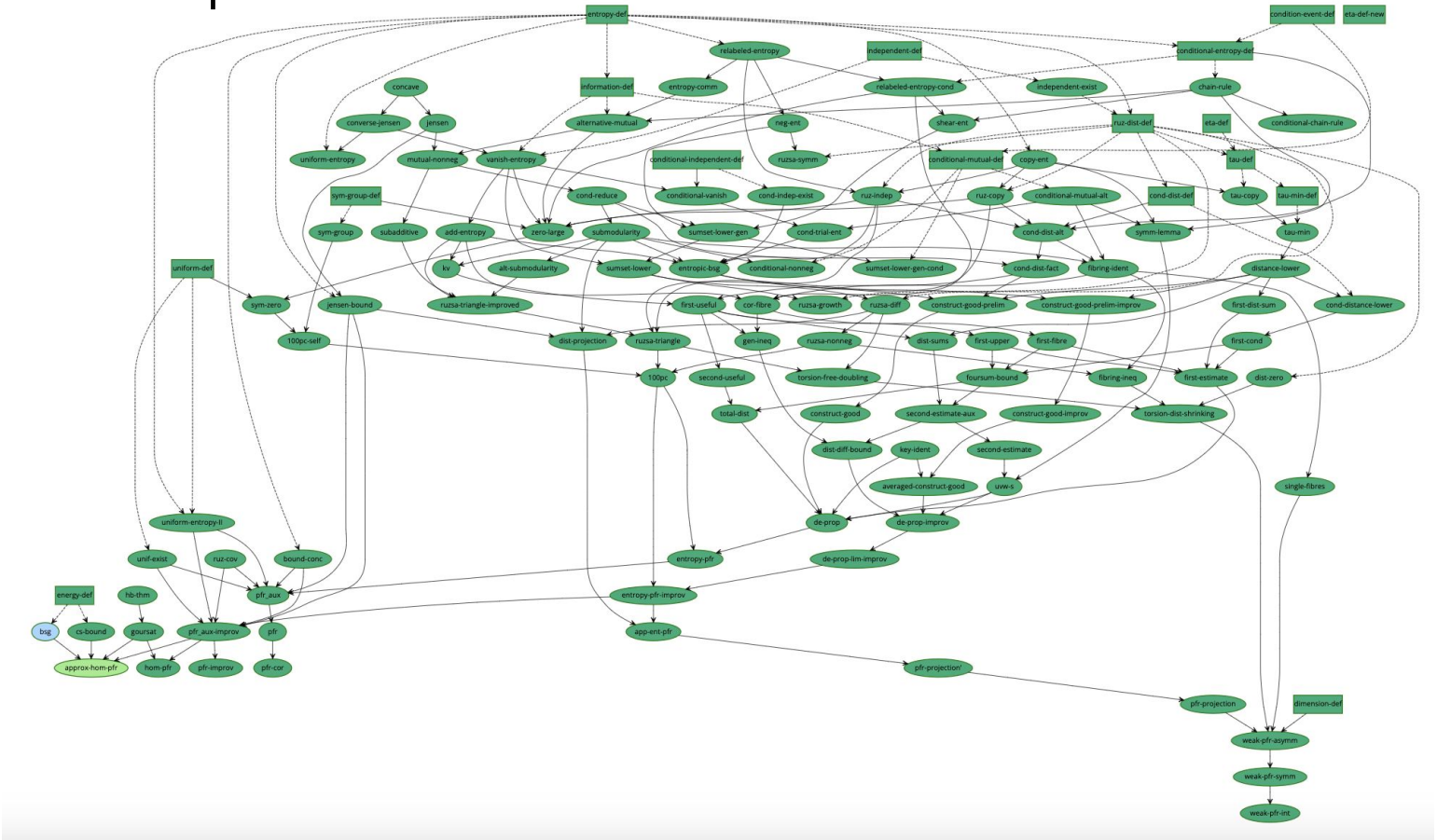
- **Lean** (formal proof language / proof assistant)
- **Blueprint** (DAG of theorems/lemmas in paper + their dependencies)
- **Zulip** chat (basically Discord with a bunch of mathematicians in it)
- Finally… good old **Github copilot**!

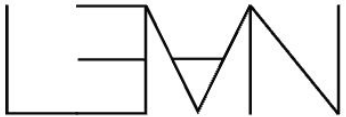
https://terrytao.wordpress.com/2023/11/18/formalizing-the-proof-of-pfr-in-lean4-using-blueprint-a-short-tour/

Blueprint



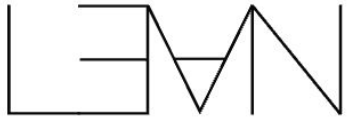
Blueprint





Proving stuff by coding.

```
theorem and_commutative (p q : Prop) : p ∧ q → q ∧ p :=
assume hpq : p ∧ q,
have hp : p, from and.left hpq,
have hq : q, from and.right hpq,
show q ∧ p, from and.intro hq hp
```



```
/--   $$ d[X;Y] \geq 0.$$  -/
lemma rdist_nonneg : 0 ≤ d[ X ; μ # Y ; μ' ] := by
  suffices : 0 ≤ 2 * d[ X ; μ # Y ; μ' ]
  . linarith
  sorry
```

- **Mathlib:** ever expanding repository of theorems proven in lean
- **Tactics:** library of automated tricks you can use to finish off a theorem,

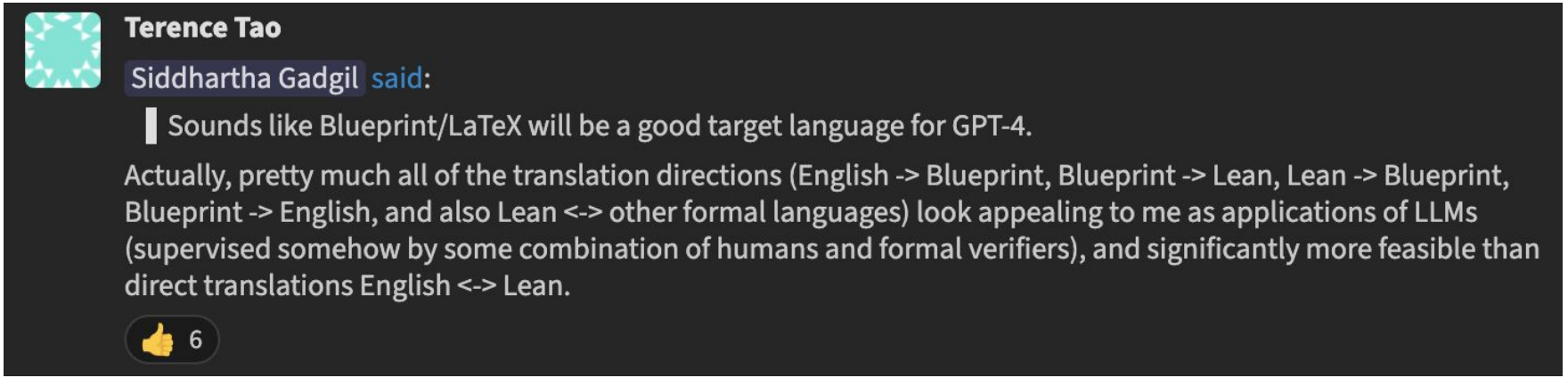
  e.g. *linarith* to deduce **0 < x** from **0 < 2x**
- **Kernel** checks if you are done.

Steps

1. Experts turn paper into **DAG** with Blueprint.
2. Now anybody can pick a node and claim it on Zulip. A node can be:
    a. **statement formalized** (written in lean)
    b. **statement + ancestors statements' formalized**
    c. **formally proved from ancestors**
    d. **proved** (proved from ancestors + ancestors formally proved)
3. **Once 'PFR' node in dag proved, done!**

PFR TAKEAWAY

Terry Tao bullish on LLM



Terence Tao

Siddhartha Gadgil said:

> Sounds like Blueprint/LaTeX will be a good target language for GPT-4.

Actually, pretty much all of the translation directions (English -> Blueprint, Blueprint -> Lean, Lean -> Blueprint, Blueprint -> English, and also Lean <-> other formal languages) look appealing to me as applications of LLMs (supervised somehow by some combination of humans and formal verifiers), and significantly more feasible than direct translations English <-> Lean.

👍 6

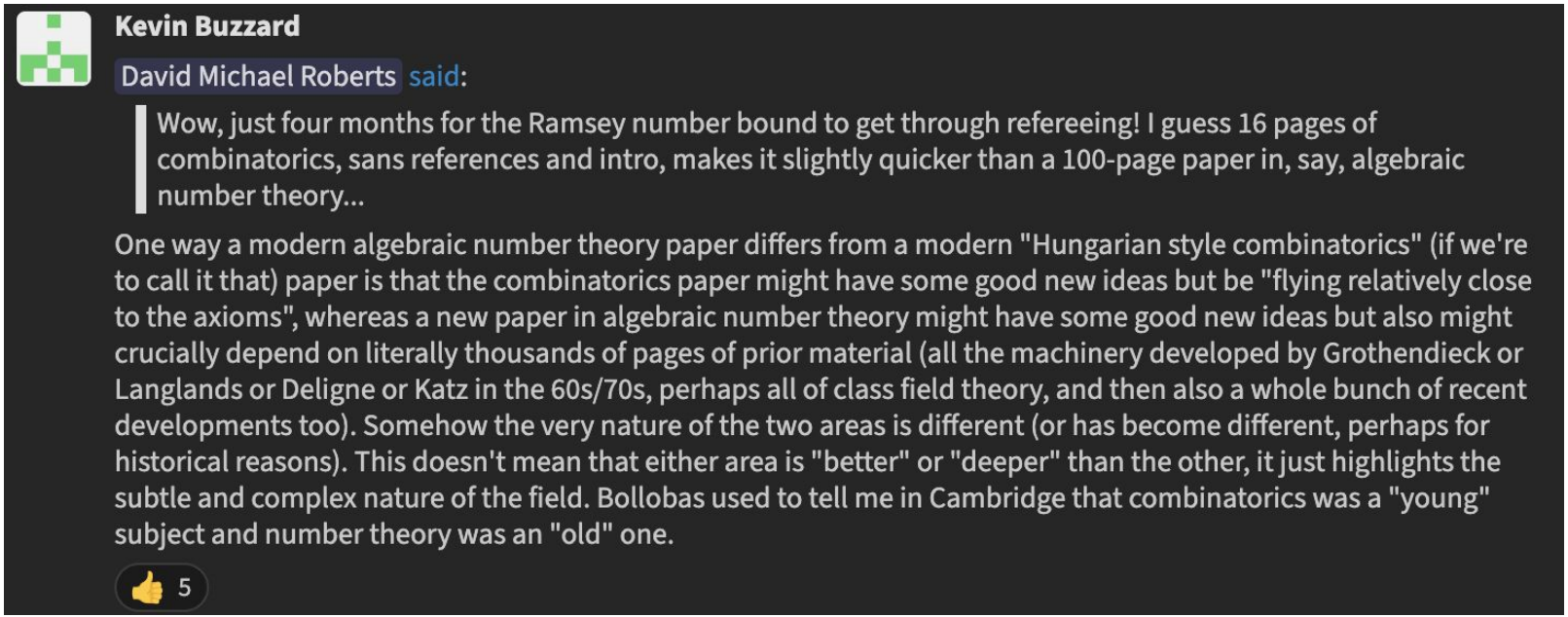
PFR TAKEAWAY

Terence Tao
@tao@mathstodon.xyz

As an experiment, I recently tried consulting #GPT4 on a question I found on #MathOverflow prior to obtaining a solution. The question is at mathoverflow.net/questions/449... and my conversation with GPT-4 is at chat.openai.com/share/53aab67e... . Based on past experience, I knew to not try to ask the #AI to answer the question directly (as this would almost surely lead to nonsense), but instead to have it play the role of a collaborator and offer strategy suggestions. It did end up suggesting eight approaches, one of which (generating functions) being the one that was ultimately successful. In this particular case, I would probably

PFR TAKEAWAY

Additive combinatorics = hard but simple



Kevin Buzzard

David Michael Roberts said:

> Wow, just four months for the Ramsey number bound to get through refereeing! I guess 16 pages of combinatorics, sans references and intro, makes it slightly quicker than a 100-page paper in, say, algebraic number theory...

One way a modern algebraic number theory paper differs from a modern "Hungarian style combinatorics" (if we're to call it that) paper is that the combinatorics paper might have some good new ideas but be "flying relatively close to the axioms", whereas a new paper in algebraic number theory might have some good new ideas but also might crucially depend on literally thousands of pages of prior material (all the machinery developed by Grothendieck or Langlands or Deligne or Katz in the 60s/70s, perhaps all of class field theory, and then also a whole bunch of recent developments too). Somehow the very nature of the two areas is different (or has become different, perhaps for historical reasons). This doesn't mean that either area is "better" or "deeper" than the other, it just highlights the subtle and complex nature of the field. Bollobas used to tell me in Cambridge that combinatorics was a "young" subject and number theory was an "old" one.

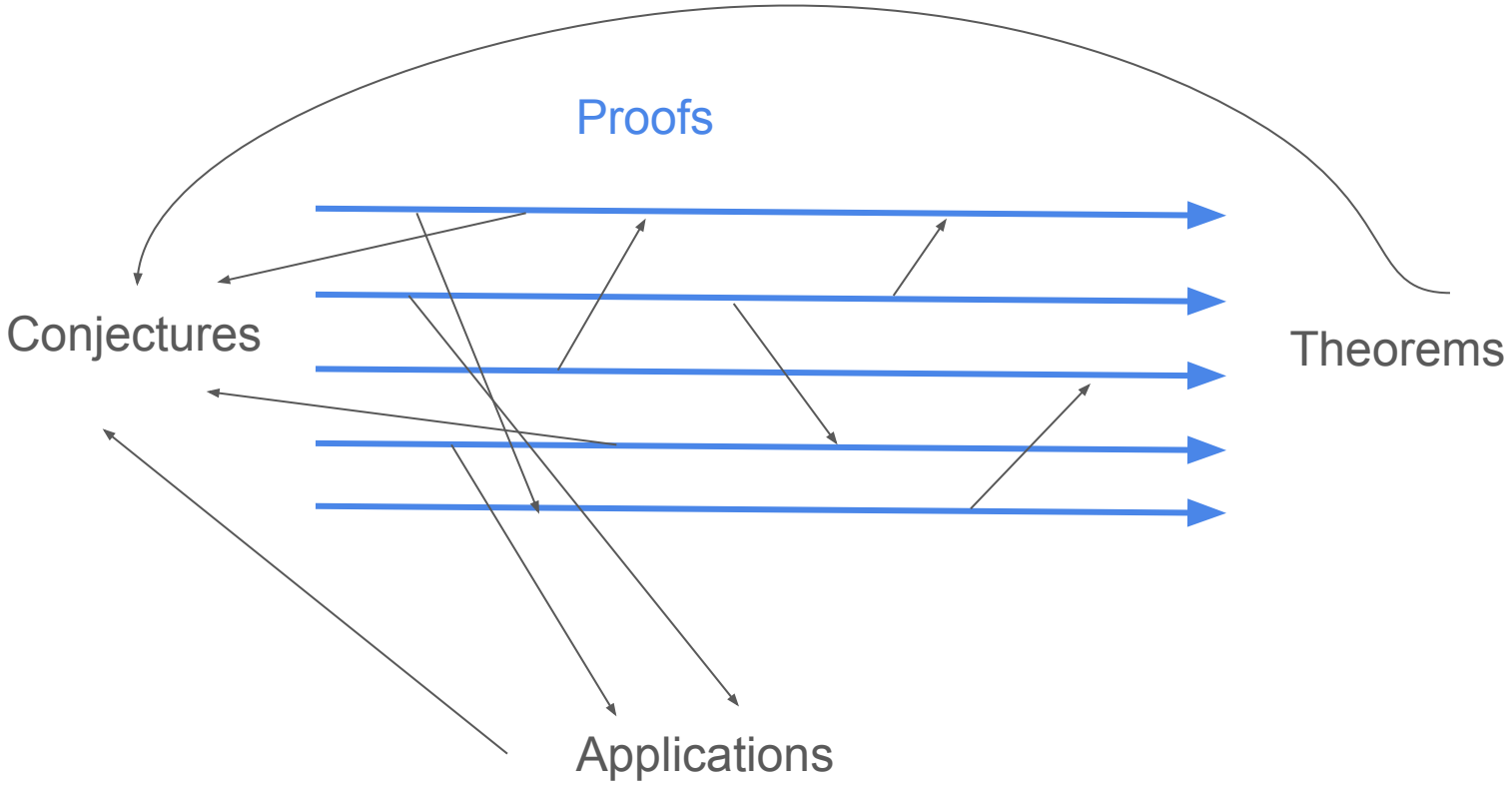
👍 5

What does it all mean?

Where’s it going?

Should mathematicians be having an existential crisis?

PRE LLM MATH



THE POINT OF MATH

- The point of proving theorems is…
  a. Know things for certain
  b. **reach conclusion in a way that builds our understanding of the field/neighboring fields**
- **b is more important** (in my opinion), and letting llms have all the fun is likely to lead to **a without b**.
- Plus, LLMs aren't directly proving real conjectures. **For now, anyway.**

Lecture 7 Exercises

- What is the rule of -f**king- insertion and how do we train an LLM model to correctly insert -f**king- to any word in English?
  - f**king can be added before the primary stress of a word. *MASS-ə-CHOO-sits* -> *MASS-ə-f**king-CHOO-sits*

Lecture 7 Exercises

- How would you define if an LLM is memorizing a dataset? How few parameters would an llm need to have for you to be confident it wasn't memorizing a dataset?
  - It doesn't make sense to say a model is memorizing a single dataset - it could be achieving low loss by actually learning.
  - A model can be capable of memorizing a family of datasets with the same inputs but different outputs.
  - The number of bits in the parameters should be lower than **log_2(# datasets the model can learn perfectly)**

THANKS FOR JOINING GRIDSPACE IAP 2024!

IAP SERIES CONCLUSION

PLANNING

MEMORY

PERCEPTION

LANGUAGE &
SYMBOLIC
REASONING

THAT'S ALL FOLKS, GRIDSPACE X IAP 2024!

- Continue to engage with us iap.gridspace.com & iap@gridspace.com
- Let us know what you liked and perhaps future content you'd like to see
- Subscribe on YouTube: https://www.youtube.com/gridspaceinc
- Follow us on Twitter & Linkedin

JOIN OUR TEAM!

- Recruiting events:
  - MIT xFair - Feb 9, 2024
  - Stanford - April 16-17, 2024
  - LA Office Visits - Spring 2024
- Reach us at hiring@gridspace.com