NOTES ON A PATH TO AI ASSISTANCE IN MATHEMATICAL REASONING

ALEX KONTOROVICH

These informal notes are based on the author's lecture at the National Academies of Science, Engineering, and Mathematics workshop on "AI to Assist Mathematical Reasoning". The goal is to think through a path by which we might arrive at AI that is useful for the research mathematician.

1. Finding a Holy Grail

I find utility in trying to work backwards: What could be our end goal in this game, our "Holy Grail"? One possibility is the following.

Holy Grail 1. *AI solves the Riemann hypothesis.*

Of course this applies more generally to any major, longstanding problem of interest to mathematicians. I see two ways this could go:

(1) AI might give a million-line, dense, incomprehensible proof of RH. In this nightmare scenario, people like me will spend the rest of their lives just trying to understand what it's saying and why.[1]
(2) Alternatively, AI might give a perfectly comprehensible, beautiful proof of RH! (Is this a dream? Or also a nightmare?! Now I'm *really* out of business, and spending my life prompting GPT instead of relishing the thought of solving a hard problem.)

Another potential target, instead of solving *our* problems, may be the following.

Holy Grail 2. *AI starts making its own beautiful definitions, conjectures, and theorems, doing "Alien math" way ahead of what humans can comprehend and utilize.*

The history of mathematics is full of vignettes, such as Monstrous Moonshine, or the Dyson-Montgomery tea conversation, in which patterns across seemingly unrelated fields were serendipitously discovered and exploited to uncover (perhaps still conjectural) underlying structures. Humans already do this on their own, though such discoveries are relatively few and far between. AI has also already proved somewhat effective here; see, e.g., [DVB+21], and the more recent discovery [HLOP22] of

Kontorovich is partially supported by NSF grant DMS-1802119 and BSF grant 2020119.

[1]Perhaps (1) is anyway unreasonable, as some *other* AI will digest the proof and explain it to us in terms we understand?

murmurations in the Fourier coefficients of elliptic curves. (That said, in these examples, "AI" really refers to: statistical inference analysis on large datasets produced via human-written deterministic algorithms. In this sense, it is not all that different from Gauss conjecturing the Prime Number Theorem from large tables of primes.) Can AI discover structures such as perfectoid spaces or infinity categories? If so, might it find novel structures, definitions, conjectures, and theorems, with sufficient rapidity that human beings have no hope of keeping up?

As far as I can tell, both of these Holy Grails are distant science fiction. Here is a more modest, medium-term possibility.

Holy Grail 3. *AI can* assist *mathematicians in theorem proving.*

Certainly if AI can't even do that, then there's no hope of the more ambitious Holy Grails. What might we consider "success" here?

The current workflow of a research mathematician typically involves something like:

- Formulating an idea for a major theorem or a class of theorems.
- Trying to break up these theorems into smaller propositions.
- Deriving each proposition from a set of even smaller lemmas.
- At each scale (lemma/proposition/theorem), employing standard techniques or exploring new, non-standard approaches, perhaps doing literature searches or looking for analogous arguments that have appeared in other settings.[2]
- Iterating through the whole process, modifying the statements, refining approaches, and discovering alternative techniques, etc etc, all hoping for eventual success.

A lot of this does feel like it could be outsourced. In other sciences, PIs have labs of PhD students and postdocs who fight the various local battles in parallel, with the PI serving as a General overseeing the action and making "global" adjustments to the plan of attack. For many reasons, we don't/can't do this in math. But could AI help automate some of these steps, perhaps drastically speeding up the workflow of a mathematician capable of harnessing these tools?

Let's continue working backwards from Holy Grail 3 as our target.

2. Necessity of Interactive Theorem Provers

As experience shows, for "AI" to do anything, it first needs to be trained on extremely **large** data sets. A first question might be: data sets of *what* exactly?

Conjecture 1. *Large language models (LLMs) trained on natural language alone will* not *reason reliably at the level of professional mathematics.*

[2]LLMs like ChatGPT are already capable of *sometimes* giving helpful pointers, but this mostly happens when humans have already found connections and written about them.

Mathematics is communicated in such a way as to appear to be a language amenable to sampling. But at its core, the underlying arguments are always deterministic: either logically sound or not. Language itself exhibits stochastic properties, with various ways to express ideas and infer meaning. Mathematics, on the other hand, relies on an underlying language, such as English, but its essence is entirely deterministic and precise.

For instance, if an LLM generates a 98% believable Bach chorale, we would be delighted. However, if it produces a 98% correct mathematical argument, it may be completely worthless. That missing 2% might be a reduction to a problem as difficult as the original, or it may just be flat out wrong. The processes of "production" and "editing" are distinct functions, even within the human brain. LLMs excel at stochastic generation, producing the next word (or token) by sampling from a statistical distribution, whereas mathematics demands deterministic editing. That is my perspective, for what it's worth.

Counterpoint: OpenAI, Google, and many other groups much smarter than me and with many more resources than I have are actively working to disprove Conjecture 1! They believe that, if only LLMs were given *enough* training data, enough parameters, enough transformers, etc, then they *would* solve professional-level mathematical problems, as they've already succeeded handsomely with a variety of standardized math and reasoning tests. (Their work in this direction is not unrelated to issues of "alignment", which we have neither the time nor expertise to delve into here...)

And even if I'm completely wrong, and LLMs are indeed capable of producing, in natural language, something that reads like a perfect math paper, how can we ever trust it? We would be burdened with the responsibility of refereeing the thousands or millions of papers they generate to discern the correct ones from mere "hallucinations".

Conjecture 2. *The path to AI assisting research mathematicians is through an adversarial process, likely involving Interactive Theorem Provers (e.g., Lean, Isabelle, Coq, etc).*

3. A refined Holy Grail

Here is a suggested mechanism to aim for Holy Grail 3. Imagine the following scenario.

Holy Grail 4. *One asks a ChatGPT-like prompt in natural language about a Lemma or technique idea. The LLM bounces back and forth with Lean,*[3] *and eventually outputs in natural language: "here is what I'm able to prove about your question", together with a formalized certificate that the argument described is valid.*

[3]Here and throughout, we will use "Lean" as a shorthand for any Interactive Theorem Prover, perhaps even one not yet invented.

This possibility seems, to me, much less out of reach, at least in the medium-term. If one accepts this as a desired goal, the question again becomes: how do we train AI for this task? Any of the GPT4, Bard, or other such systems, to work nearly as well as they do, typically require training on trillions of data points, whether these are measured in bytes, lines of text, tokens, etc. In contrast, if we consider the current formalized libraries of mathematics, we might have on the order of 10 million "data points", again, whether that means: raw bytes, lines of code, pairs of goal states and next proof lines, or pairs of natural language lines and formal lines, or something else entirely.

This naturally leads to the following:

Conjecture 3. *The current rate-limiting step for AI proof assistance is: producing orders of magnitude more lines of formalized professional-level mathematics.*

If you believe this conjecture, then the next natural question is: how to produce lots more?

4. Paths to Large Datasets of Formalized Mathematics

Producing high-quality formalized mathematics is difficult! Modern mathematics requires interactions of many different fields in concert. One needs huge libraries that can all interact with one another compatibly (this is one of the main features of Lean's "mathlib" library). I see (at least) three reasons to be somewhat optimistic here.

(1) The pace of production has been steadily increasing. Three vignettes to illustrate this are:
   (a) It took two years from Ellenberg-Gijswijt's 2017 solution [EG17] of the Cap Set Conjecture to its 2019 formalization [DHL19].
   (b) It took one year from Gardam's 2021 counterexample [Gar21] to Kaplansky's unit conjecture for group rings to its 2022 formalization [GT22].
   (c) The formalization [BM22] of Bloom's solution [Blo21] to the Edros-Graham density conjecture on Egyptian unit fractions arrived *before* its referee reports!

   (Of course these results are cherry-picked, and it is still relatively difficult to formalize most research-level mathematics...)

(2) The levels of complexity that Theorem Proving software can handle is advancing. We've gone from:
   (a) Proving difficult statements about simple objects, such the formalization [Avi04] of the Prime Number Theorem: the primes are elementary to define, but the statement on their asymptotic behavior is relatively nontrivial to establish.
   (b) Proving elementary statements about complicated objects, such as the formalization [BCM20] of the mere *definition* of perfectoid spaces.
   (c) Proving difficult statements about complicated objects, such as the success of the Liquid Tensor Experiment [Sch22, Cm22].

```
/-- The quotient by a discontinuous group action of a locally compact t2 space is t2. -/
@[priority 100, to_additive "The quotient by a discontinuous group action of a locally compact t2
space is t2."]
instance t2_space_of_properly_discontinuous_smul_of_t2_space [t2_space T] [locally_compact_space T]
  [has_continuous_const_smul Γ T] [properly_discontinuous_smul Γ T] :
  t2_space (quotient (mul_action.orbit_rel Γ T)) :=
begin
  set Q := quotient (mul_action.orbit_rel Γ T),
  rw t2_space_iff_nhds,
  let f : T → Q := quotient.mk,
  have f_op : is_open_map f := is_open_map_quotient_mk_mul,
  rintros (x₀) (y₀) (hxy : f x₀ ≠ f y₀),
  show ∃ (U ∈ 𝓝 (f x₀)) (V ∈ 𝓝 (f y₀)), _,
  have hx₀y₀ : x₀ ≠ y₀ := ne_of_apply_ne _ hxy,
  have hγx₀y₀ : ∀ γ : Γ, γ • x₀ ≠ y₀ := not_exists.mp (mt quotient.sound hxy.symm : _),
  obtain (K₀, L₀, K₀_in, L₀_in, hK₀, hL₀, hK₀L₀) := t2_separation_compact_nhds hx₀y₀,
  let bad_Γ_set := {γ : Γ | (((•) γ) '' K₀) ∩ L₀ ≠ ∅ },
  have bad_Γ_finite : bad_Γ_set.finite := finite_disjoint_inter_image hK₀ hL₀,
  choose u v hu hv u_v_disjoint using λ γ, t2_separation_nhds (hγx₀y₀ γ),
  let U₀₀ := ⋂ γ ∈ bad_Γ_set, ((•) γ) ⁻¹' (u γ),
  let U₀ := U₀₀ ∩ K₀,
  let V₀₀ := ⋂ γ ∈ bad_Γ_set, v γ,
  let V₀ := V₀₀ ∩ L₀,
  have U_nhds : f '' U₀ ∈ 𝓝 (f x₀),
  { apply f_op.image_mem_nhds (inter_mem ((bInter_mem bad_Γ_finite).mpr $ λ γ hγ, _) K₀_in),
    exact (continuous_const_smul _).continuous_at (hu γ) },
  have V_nhds : f '' V₀ ∈ 𝓝 (f y₀),
    from f_op.image_mem_nhds (inter_mem ((bInter_mem bad_Γ_finite).mpr $ λ γ hγ, hv γ) L₀_in),
  refine (f '' U₀, U_nhds, f '' V₀, V_nhds, mul_action.disjoint_image_image_iff.2 _),
  rintros x (x_in_U₀₀, x_in_K₀) γ,
  by_cases H : γ ∈ bad_Γ_set,
  { exact λ h, (u_v_disjoint γ).le_bot (mem_Inter₂.mp x_in_U₀₀ γ H, mem_Inter₂.mp h.1 γ H) },
  { rintros (-, h'),
    simp only [image_smul, not_not, mem_set_of_eq, ne.def] at H,
    exact eq_empty_iff_forall_not_mem.mp H (γ • x) (mem_image_of_mem _ x_in_K₀, h') },
end
```

FIGURE 1. A formal proof from [KM22]

(3) With enough progress on Application Programming Interface (API), formalized proofs can (sometimes) be made to work almost *exactly* how human ones do. One example of this is the formalization [KM22] of the author and Heather Macbeth of the fact that the quotient X/Γ of a locally compact Hausdorff space X by a discontinuous group action Γ is itself Hausdorff. This statement is certainly nothing earthshattering; it's the kind of thing a 1st year PhD student (or advanced undergrad) should be able to solve, but it's also not entirely trivial. What's more remarkable, to me at least, is that the proof, shown in Figure 1, reads almost verbatim like a human one, with nearly each line of the formalization having a corresponding line in a natural language argument. (Of course one needs to first become familiar with the syntax of Lean, in the same way that one eventually becomes accustomed to reading dollar signs and backslashes in LaTeX and visualizing the outcome of compilation...)

Here is a big reason to be pessimistic.

Theorem 4. *All that said, humans alone will* never *reach a trillion lines of formalized mathematics.*

The proof is obvious. Equally obvious, then, is that humans need automated assistance in formalization! This idea is far from original, and was proposed already by Szegedy in 2020 [Sze20]. Before AI can assist mathematicians with solving *new*

problems, it had better get really good at formalizing known (to humans) solutions! There are many groups already working hard on this from a wide variety of viewpoints, see, e.g., [RLBS20, WJL+22, LLL+22, JWZ+22] for but a sample. Many more are needed, in this author's opinion.[4]

In closing, it seems prudent to direct resources towards developing a positive feedback loop between human and AI formalization, with progress in each driving the other. This can hopefully eventually build up enough of a training database from which other AI systems can effectively learn, and become truly useful and reliable as research assistants. Who knows where things will go from there.

Acknowledgements: The author would like to thank Kevin Buzzard, Drew Sutherland, and Geordie Williamson for many comments and suggestions that improved on an earlier draft.

References

[Avi04] Jeremy Avigad. Notes on a formalization of the prime number theorem, 2004. https://www.andrew.cmu.edu/user/avigad/Papers/pntnotes.pdf. 4

[BCM20] K. Buzzard, J. Commelin, and P. Massot. Formalising perfectoid spaces. *POPL: Principles of Programming Languages, Association for Computing Machinery*, pages 299–312, 2020. 4

[Blo21] Thomas Bloom. On a density conjecture about unit fractions, 2021. https://arxiv.org/abs/2112.03726. 4

[BM22] Thomas Bloom and Bhavik Mehta. Unit fractions, 2022. https://b-mehta.github.io/unit-fractions/. 4

[Cm22] Johan Commelin and mathlib. Completion of the liquid tensor experiment, 2022. https://leanprover-community.github.io/blog/posts/lte-final/. 4

[DHL19] Sander R. Dahmen, Johannes Hölzl, and Robert Y. Lewis. Formalizing the solution to the cap set problem. In *10th International Conference on Interactive Theorem Proving*, volume 141 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 15, 19. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. 4

[DVB+21] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021. 1

[EG17] Jordan S. Ellenberg and Dion Gijswijt. On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression. *Ann. of Math. (2)*, 185(1):339–343, 2017. 4

[Gar21] Giles Gardam. A counterexample to the unit conjecture for group rings. *Ann. of Math. (2)*, 194(3):967–979, 2021. 4

[GT22] Siddhartha Gadgil and Anand Rao Tadipatri. Formalizing Gardam's disproof of Kaplansky's unit conjecture, 2022. https:

[4] Added in print: the authors of [GZA+23] are able to achieve impressive results in the restricted realm of coding, training on "only" billions of tokens, rather than trillions. That said, the path to the former may not be so different from the latter; that is, a mechanism to reach a billion, say, lines of formalized mathematics, seems likely to also reach a trillion.

//siddhartha-gadgil.github.io/automating-mathematics/posts/formalizing-gardam-disproof-kaplansky-conjecture/. 4

[GZA+23] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. https://arxiv.org/abs/2306.11644. 6

[HLOP22] Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of elliptic curves, 2022. https://arxiv.org/abs/2204.10140. 1

[JWZ+22] Albert Q. Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs, 2022. https://arxiv.org/abs/2210.12283. 6

[KM22] Alex Kontorovich and Heather Macbeth. "const_mul_action" in Lean3's mathlib, 2022. https://github.com/leanprover-community/mathlib/blob/f23a09ce6d3f367220dc3cecad6b7eb69eb01690/src/topology/algebra/const_mul_action.lean. 5

[LLL+22] Guillaume Lample, Marie-Anne Lachaux, Thibaut Lavril, Xavier Martinet, Amaury Hayat, Gabriel Ebner, Aurélien Rodriguez, and Timothée Lacroix. Hypertree proof search for neural theorem proving, 2022. https://arxiv.org/abs/2205.11491. 6

[RLBS20] Markus N. Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. Mathematical reasoning via self-supervised skip-tree training, 2020. https://arxiv.org/abs/2006.04757. 6

[Sch22] Peter Scholze. Liquid tensor experiment. *Exp. Math.*, 31(2):349–354, 2022. 4

[Sze20] Christian Szegedy, editor. *A Promising Path Towards Autoformalization and General Artificial Intelligence*, 2020. 5

[WJL+22] Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models, 2022. https://arxiv.org/abs/2205.12615. 6

Email address: alex.kontorovich@rutgers.edu

Department of Mathematics, Rutgers University, New Brunswick, NJ