Theorem Provers and the Future AI Math Ecosystem

Stephan Schulz

DHBW Stuttgart, Stuttgart, Germany
schulz@eprover.org

Abstract

We propose that the underlying knowledge representation of the computerized mathematical ecosystem must be based on formal logic. Theorem provers can maintain the integrity of mathematical knowledge both by consistency checking and by ensuring sound derivation for new results. Other AI techniques can help e.g. in auto-formalisation of mathematical knowledge, in search control, and in the hypothesis generation part of automatic theory exploration.

1 Mathematics and AI

Going back to Euclid, mathematics deals with definitions and proofs, i.e. certain, reliable knowledge based on explicit (and reasonable, or at least interesting) assumptions. While for many the beauty of mathematics is enough to engage in it, a major impetus for mathematical research comes from its incredible usefulness in describing aspects of reality (see e.g. [19]).

In the last few decades, parts of our collected mathematical knowledge has been translated into formal logics, and, starting a bit later, either checked or reproved with automated or interactive theorem provers, adding another layer of reliability to the corpus. Some results have even been found automatically.

In the field of mathematics, symbolic reasoning has a long history. Automated theorem provers (ATP) and computer algebra systems use sound symbolic inference to perform reasoning steps and provide derivations [14] that are correct by construction (at least in the ideal case) and that can be verified a-posteriori to provide extremely reliable results. However, these systems have weaknesses, some of which can be addressed by modern AI systems.

The field of AI can be broadly split into symbolic and sub-symbolic approaches on the one hand, and into reasoning vs. learning on the other hand. The core distinction between symbolic and sub-symbolic systems is that symbolic systems represent knowledge as collections of specific, discrete, often logic-based objects, while sub-symbolic systems represent knowledge distributed in a large, usually numerical set of parameters. The core of theorem proving falls squarely into the symbolic side of AI.

Both symbolic and sub-symbolic approaches have produced impressive machine learning methods. On the symbolic side we find e.g. approaches based on decision trees, similarity and analogy, and evolutionary methods. On the sub-symbolic side, the most prominent approach is that of artificial neural networks. These typically consist of layers of simple neurons that are interconnected and pass numerical values from one layer to the next to produce an output vector from an input vector. They are trained via error back-propagation. These techniques have been known since the 1970s, but in recent years the combination of much more powerful hardware, much larger datasets, and some changes in network layout and activation functions has inspired and enabled the development of *deep neural networks*. These typically have many more layers than older systems, they have a more complex architecture, interleaving fully connected layers, convolution layers, pooling layers and others. Another big change is that they typically work on much more raw input data and rely less on predefined features. The enormous success of these deep neural networks has fueled the recent successes of artificial intelligence.

On the sub-symbolic side, the combination of deep neural networks with word vector encodings has enabled the creation of large language models (LLMs) - basically neural networks that learn the conditional probability of the next word (more precisely *token*) of a text based on a given context constructed both from an initial prompt and the text generated so far. Such models, trained on internet-sized datasets and applied recursively to their own output, have demonstrated extremely impressive capabilities. They are able to routinely perform language tasks such as summarizing, translation, and reformulation, and they appear as quite intelligent conversation partners. In particular, they apparently can solve mathematical and logical puzzles, which has lead to the idea that such systems might also be useful formal mathematical reasoning.

However, despite their success in conversational settings, I believe that plain large language models will be inherently unable to reliably generate new mathematical results - essentially because they learn models of language and text, not models of real or mathematical structures. As such, they are limited to reproduce existing ideas (even if in many variations) over existing structures. A large part of mathematics, on the other hand, is the discovery of abstract properties of new structures. While LLMs have been able to apparently solve many logical puzzles, the success seems to drastically drop off if such puzzles are reworded with new vocabulary, or if confounding variables are added [7]. Moreover, LLMs tend to *hallucinate*, basically producing intelligent looking but completely counterfactual or nonsensical text streams. Such hallucinations may be unavoidable [1]. Dijkstra's fundamental critique of "natural language programming" [3] applies to mathematics even more strongly:

> It may be illuminating to try to imagine what would have happened if, right from the start our native tongue would have been the only vehicle for the input into and the output from our information processing equipment. My considered guess is that history would, in a sense, have repeated itself, and that computer science would consist mainly of the indeed black art how to bootstrap from there to a sufficiently well-defined formal system

That does not mean that LLMs and other modern AI techniques are useless for formal mathematics. But it needs to be combined with more reliable systems to build trustworthy solutions for mathematical reasoning.

2 Reasoning and Learning for Mathematical AI

I believe that scalable, usable mathematical AI systems must be based on hybrid architectures. The core collection of mathematical knowledge will be encoded in a formal logic, most likely variants of higher-order predicate logic (with large first-order subsets). These languages have a formal syntax and a strictly defined semantics, making it possible to exactly understand and check their reasoning and consistency.

Currently, acquisition of explicit mathematical knowledge is largely based on manual encoding - a process that is expensive and error-prone. LLM-based approaches may be useful as tools to facilitate the formalisation of existing mathematical literature, to help generate interesting conjectures, and to support the generation of human-readable proofs.

The integrity of the mathematical knowledge base will be supported by automated theorem provers such as E [13, 12, 17, 18] and Vampire [6], which will both help to maintain the consistency of this knowledge as new domains are added (as e.g. in [15]), and provide ways to derive new theorems and flesh out new theories. Interactive systems such as e.g. Lean [8] 1or Isabelle [9] will provide the user interface for human mathematicians.

One particular problem of reasoning in formal logic is the control of the reasoning process. The space of possible logical derivations is almost always so large that it is hard to find the interesting reasoning steps, in particular with respect to the task of proving a given conjecture. Various machine learning method and classical optimisation approaches will help ATPs to deal with the complexities of the search space, as already demonstrated by several systems [11, 2, 5, 4, 16, 10].

The integration of various AI techniques, from symbolic reasoning to LLMs, is challenging, but also holds the promise of order-of-magnitude gains in the efficiency of mathematical reasoning.

References

[1] Banerjee, S., Agarwal, A., Singla, S.: LLMs Will Always Hallucinate, and We Need to Live With This (2024), https://arxiv.org/abs/2409.05746

[2] Chvalovský, K., Jakubuv, J., Suda, M., Urban, J.: ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 197–215. No. 11716 in LNAI, Springer (2019)

[3] Dijkstra, E.W.: On the foolishness of "natural language programming" (1978), http://www.cs.utexas.edu/users/EWD/ewd06xx/EWD667.PDF, circulated privately

[4] Goertzel, Z.A., Chvalovský, K., Jakubův, J., Olšák, M., Urban, J.: Fast and Slow Enigmas and Parental Guidance. In: Konev, B., Reger, G. (eds.) 13th International Symposium on Frontiers of Combining Systems. pp. 173–191. Springer (2021)

[5] Jan Jakubuv, K.C., Olsák, M., Piotrowski, B., Suda, M., Urban, J.: ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine (System Description. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Proc. of the 10th IJCAR, Paris (Part II). LNAI, vol. 12167, pp. 448–463. Springer (2020)

[6] Kovács, L., Voronkov, A.: First-order theorem proving and Vampire. In: Sharygina, N., Veith, H. (eds.) Proc. of the 25th CAV, LNCS, vol. 8044, pp. 1–35. Springer (2013)

[7] Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M.: GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models (2024), https://arxiv.org/abs/2410.05229

[8] Moura, L.d., Ullrich, S.: The Lean 4 Theorem Prover and Programming Language. In: Platzer, A., Sutcliffe, G. (eds.) Proc. of the 28th CADE, Pittsburgh. pp. 625–635. Springer (2021)

[9] Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002)

[10] Schäfer, S., Schulz, S.: Breeding theorem proving heuristics with genetic algorithms. In: Gottlob, G., Sutcliffe, G., Voronkov, A. (eds.) Proc. of the Global Conference on Artificial Intelligence, Tibilisi, Georgia. EPiC, vol. 36, pp. 263–274. EasyChair (2015)

[11] Schulz, S.: Learning Search Control Knowledge for Equational Theorem Proving. In: Baader, F., Brewka, G., Eiter, T. (eds.) Proc. of the Joint German/Austrian Conference on Artificial Intelligence (KI-2001). LNAI, vol. 2174, pp. 320–334. Springer (2001)

[12] Schulz, S.: E – A Brainiac Theorem Prover. Journal of AI Communications **15**(2/3), 111–126 (2002)

[13] Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 495–507. No. 11716 in LNAI, Springer (2019)

[14] Schulz, S., Sutcliffe, G.: Proof generation for saturating first-order theorem provers. In: Delahaye, D., Woltzenlogel Paleo, B. (eds.) All about Proofs, Proofs for All, Mathematical Logic and Foundations, vol. 55, pp. 45–61. College Publications, London, UK (January 2015)

[15] Schulz, S., Sutcliffe, G., Urban, J., Pease, A.: Detecting inconsistencies in large first-order knowledge bases. In: de Moura, L. (ed.) Proc. of the 26th CADE, Gothenburg. LNAI, vol. 10395, pp. 310–325. Springer (2017)

[16] Urban, J.: Blistr: The blind strategymaker. In: Gottlob, G., Sutcliffe, G., Voronkov, A. (eds.) Proc. of the Global Conference on Artificial Intelligence, Tibilisi, Georgia. EPiC, vol. 36, pp. 312–319. EasyChair (2015)

[17] Vukmirović, P., Blanchette, J.C., Cruanes, S., Schulz, S.: Extending a Brainiac Prover to Lambda-free Higher-Order Logic. International Journal on Software Tools for Technology Transfer (August 2021). https://doi.org/10.1007/s10009-021-00639-7

[18] Vukmirović, P., Blanchette, J.C., Schulz, S.: Extending a high-performance prover to higher-order logic. In: Sharygina, N., Sankaranarayanan, S. (eds.) Proc. 29th Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'23), Paris, France. pp. 111–132. No. 13994(2) in LNCS, Springer (2023)

[19] Wigner, E.P.: The Unreasonable Effectiveness of Mathematics in the Natural Sciences. Communications in Pure Applied Mathematics **13**(1), 1–14 (Feb 1960). https://doi.org/10.1002/cpa.3160130102