# Towards the Automatic Mathematician

Markus N. Rabe and Christian Szegedy

Google Research
Mountain View, California, USA
{mrabe,szegedy}@google.com

**Abstract.** Over the recent years deep learning has found successful applications in mathematical reasoning. Today, we can predict fine-grained proof steps, relevant premises, and even useful conjectures using neural networks. This extended abstract summarizes recent developments of machine learning in mathematical reasoning and the vision of the N2Formal group at Google Research to create an automatic mathematician. The second part discusses the key challenges on the road ahead.

**Keywords:** Automated reasoning · machine learning · mathematical reasoning · theorem proving · natural language understanding.

## 1 Introduction

The combination of machine learning and mathematical reasoning goes back at least to the 2000s when Stephan Schulz pioneered ideas to use machine learning to control the search process [44], and Josef Urban used machine learning to select relevant axioms [46,47]. With the advent of deep learning, interest in the area surged, as deep learning promises to enable the automatic discovery of new knowledge from data, while requiring minimal engineering. This suddenly offered a flurry of new possibilities also for theorem proving.

One of the most challenging and impactful tasks in automated theorem proving is *premise selection*, that is to find relevant premises from a large body of available theorems/axioms. Many classical reasoning systems do not scale well into thousands of potentially relevant facts, but some pioneering results by Urban et al. [47] proposed fast machine learning techniques using manually engineered features. However, with the inroads of deep learning, it has become clear that large quality improvements are possible by utilizing deep learning techniques. DeepMath [24] demonstrated that premise selection could be tackled with deep learning, directly (i.e., without feature engineering) applying neural networks to the text of the premise and that of the (negated) conjecture.

In DeepMath, both premise and conjecture are embedded into a vector space by a (potentially expensive) neural network and then a second (preferably cheap) neural network compares the embedding of the current state to each available premise to judge whether the premise is useful. Loos et al. [36] for the first time, demonstrated that the same approach as DeepMath yields substantial improvements as an internal guidance method within a first-order automated theorem prover.


A. Platzer and G. Sutcliffe (Eds.): CADE 2021, LNAI 12699, pp. 25–37, 2021.
https://doi.org/10.1007/978-3-030-79876-5_2

*Neural Theorem Provers.* Emboldened by these early works and by breakthroughs in deep learning, several groups extended interactive theorem provers[1] for the use in deep learning research, including Gamepad [23], HOList [5], CoqGym [54], GPT-f [39], and recently TacticZero [51]. A typical tactic application predicted by these systems looks as follows (here in HOL Light syntax):

```
REWRITE_TAC [ PREMISE1 ; PREMISE2 ]
```

*tactic name* — *list of premises*

This specific tactic expects the given premises to be equalities, with which it attempts to rewrite subexpressions in the current proof goal. The hard part about predicting good tactics is to select the right list of premises from all the previously proven theorems. Some tactics also include free-form expressions, which can be a challenge as well.

In contrast to approaches using lightweight machine learning approaches (e.g. [13,25,26,38,31]), neural theorem provers aim to replicate the human approach to proving theorems in ITPs, searching only through a relatively small number (e.g., hundreds) of proof steps that are very promising. To get high-quality proof steps, increasingly large neural networks (currently up to 774M parameters [39]) are trained on human proofs, or with reinforcement learning.

Already, neural theorem provers can prove a significant portion (up to 70% [4]) of test theorems and some have found proofs that are shorter and more elegant than the proofs that human mathematicians have formalized in these systems. For example, for the theorem `CLOSURE_CONVEX_INTER_AFFINE`, proven with over 40 tactic calls in HOL Light [20], HOList/DeepHOL has found a proof with just two tactic calls:

```
let CLOSURE_CONVEX_INTER_AFFINE = prove
  (`!s t:real^N->bool.
      convex s /\ affine t /\ ~(relative_interior s INTER t = {})
      ==> closure(s INTER t) = closure(s) INTER t`,

    SIMP_TAC [INTER_COMM; AFFINE_IMP_CONVEX;
              CLOSURE_INTER_CONVEX; RELATIVE_INTERIOR_AFFINE]

    THEN

    ASM_MESON_TAC [RELATIVE_INTERIOR_EQ_CLOSURE; INTER_COMM;
                   RELATIVE_INTERIOR_UNIV; IS_AFFINE_HULL]);;
```

[1] The focus has been on interactive theorem provers as they are general enough to capture most of mathematics in theory, and several large-scale formalization efforts of the last decades have demonstrated that involved theories can be formalized in practice [28,14,19]. Also ITPs offer relatively short proofs compared to other automated reasoning tools, which allows us to use stronger neural networks for the same computational budget.

Similarly, Polu et al. reported several cases where they found proofs with their neural theorem prover GPT-f that were shorter and more elegant than than those found by humans [39].

*Neural Solvers.* Closely related to neural theorem provers are methods that, instead of predicting proof steps, directly predict the solution to mathematical problems. A first impressive example was proposed by Selsam et al., who showed that graph neural networks can predict satisfying assignments of small Boolean formulas [45]. Lample and Charton have demonstrated that also higher-level representations, such as the integral of a formula, can be predicted directly using a Transformer [29]. They exploited the fact that for some mathematical operations, such as taking the integral, the inverse operation (taking the derivative) is much easier. Hence, they can train on predicting generated formulas from their derivative without needing a tool that can generate the integral in the first place. Recently, Hahn et al. demonstrated that also classical verification problems, such as LTL satisfiability, can be solved directly with Transformers, beating existing tuned algorithms on their own dataset in some cases [18].

## 2 Towards the Automatic Mathematician

We are convinced that the success of neural theorem provers and neural solvers is only the beginning of a larger development in which deep learning will revolutionize automated reasoning, and have set out to build an *automatic mathematician*. Ideally, we could simply talk to an automatic mathematician like a colleague, and it would be able to contribute to mathematical research, for example by publishing papers without human support.

An automatic mathematician would thus go far beyond theorem proving, as it would have to formulate and explore its own theories and conjectures, and be able to communicate in natural language. Yet, we believe that neural theorem provers are an important instrument of our plan, as they allow us to evaluate (generated) conjectures, which grounds the learning process in mathematically correct reasoning steps. And because neural theorem provers build on existing interactive theorem provers, they already come with a nucleus of formalized mathematics that we believe might be necessary to bootstrap the understanding of mathematics. In the following, we review some of the main challenges on the path towards an automatic mathematician and first approaches to address them.

### 2.1 Neural Network Architectures

Naturally, we need neural network architectures that can "understand" formulas, that is, make useful predictions based on formulas. The main question for the design of neural networks appears to be *whether* and, if yes, *how* to exploit the tree structure of formulas.

*Exploiting the Structure of Formulas.* It is tempting to believe that the embeddings of formulas should represent their semantics. Hence, many authors have suggested to process formulas with tree-structured recurrent neural networks (TreeRNNs), which compute embeddings of expressions from the embeddings of their subexpressions, as this resembles the bottom-up way we define their semantics (e.g., [11,1,23,54]). That intuition, however, may be misleading. In our experiments, bottom-up TreeRNNs have performed significantly worse than top-down architectures (followed by a max-pool aggregation) [37]. This suggests that, to make good predictions based on formulas, it is important to consider subformulas in their context, which bottom-up TreeRNNs cannot do easily.

*Sequence Models.* The alternative to representing the formula structure in the neural architecture is to interpret formulas simply as sequences of characters or symbols and apply sequence models. Early works using sequence modeling relied on convolutional networks (simple convolutional networks [24] and wave-nets [36,5]), which compared favorably to gated recurrent architectures like LSTM/GRU. With the recent rise of the Transformer architecture [48] sequence models have caught up to those that exploit the formula structure and yielded excellent performance in various settings [29,41,52,39,18].

Sequence models come with two major advantages: First, it is straightforward to not only read formulas, but also generate formulas, which is surprisingly challenging with TreeRNNs or graph neural networks. This allows us to directly predict proof steps as strings [39,52], and to tackle a wider range of mathematical reasoning tasks, such as predicting the integral of a formula [29], satisfying traces for formulas in linear temporal logics [18], or even more creative tasks, such as missing assumptions and conjectures [41].[2] Second, transformer models have shown a surprising flexibility and promise a uniform way to process not only formulas, but also natural language, and even images [10]. This could prove crucial for processing natural language mathematics, which frequently contains formulas, text, and diagrams, and any model processing papers would need to understand how they relate to each other. Transformers certainly set a high bar for the flexibility, generality, and performance of future neural architectures.

*Large Models.* Scaling up language models to larger and larger numbers of parameters has steadily improved their results [27,22]. Also when we use language models for mathematics, we have observed that larger models tend to improve the quality of predictions [39,41]. GPT-3 has shown that certain abilities, such as basic arithmetic, appear to only materialize in models with at least a certain number of parameters [6]. If this turns out to be true for other abilities, this raises the question how large models have to be to exhibit human-level mathematical reasoning abilities.

[2] Yet, there are still cases where hard-coding some formula structure in transformer architectures can improve the results, as shown, for example, by Wu et al. [21,35,18], which suggests that transformers are not the end of the story regarding formula understanding.

There is also the question of how exactly to scale up models. The mere number of parameters may not be as important as how we use them. More efficient alternatives to simply scaling up the transformer architecture might help with the problem to make large models accessible to more researchers (e.g., [32]).

### 2.2 Training Methodology

Neural networks have shown the ability to learn even advanced reasoning tasks via supervised learning, given the right training data. However, for many interesting tasks, we do not have such data and hence the question is how to train neural networks for tasks for which we have only limited data or no data at all.

*Reinforcement Learning.* Reinforcement learning can be seen as a way to reduce the amount of human-written proof data needed to learn a strong theorem prover. By training on the proofs generated by the system itself, we can improve its abilities to some extent, and the perhaps strongest neural theorem provers often use some form reinforcement learning (e.g., up to 70% of the proofs in HOL Light [4]). But, for an open-ended training methodology, we need a system that can effectively explore new and interesting theories, without getting lost in irrelevant branches of mathematics. Partial progress has been made in training systems without access to human-written proofs [4,51], and to generate conjectures to train on in a reinforcement learning setting [12], but the problem is wide-open.

*Pretraining.* In natural language understanding it is already common practice to pretrain transformers on a large body text before fine-tuning them on the final task, especially when only limited data is available for that task. Even though the pretraining data is only loosely related to the final tasks, transformers benefit a lot from pretraining, as it contains general world knowledge and useful inductive biases [9]. Polu et al. have shown that the same can be observed when pretraining transformers on natural language texts from arXiv [39].

*Self-supervised Training.* The GPT models for natural language have shown that self-supervised language modeling (i.e., only "pre"training without training on any particular task) alone can equip transformers with surprising abilities [42,6]. Mathematical reasoning abilities, including type inference, predicting missing assumptions and conjecturing, can be learned in a very similar way by training transformers to predict missing subexpressions (skip-tree training) [41].

Lample et al. devised several clever approaches to train transformers when data is not directly available. In unsupervised translation training transformers successfully learn to translate between different natural languages starting only with monolingual corpora and without any corresponding pairs of sentences [30]. This approach was even generalized to learn to translate between programming languages without corresponding pairs of programs in different languages [43]. The application of these unsupervised translation ideas to mathematics is tempting, but we experienced that their straight-forward application does not lead to good results. Also Wang et al. [49] report mixed results.

*Learning to Retrieve Relevant Information.* If we apply standard language models to mathematics, e.g., to predict the next proof step, we expect them to store all the information necessary to make good predictions in their parameters. As the large transformer models have shown (see, e.g., GPT [42,6]), this approach actually works pretty well for natural language question answering, and also for mathematical benchmarks it has been surprisingly successful [41,39,53]. However, there may be a limit to this approach in cases where we expect detailed, consistent, and up-to-date predictions. Guu et al. [17] introduced a hybrid of transformer and retrieval model, REALM, which learns to retrieve Wikipedia articles that are relevant to a given question and extract useful information from the article. REALM is trained self-supervised to retrieve multiple articles and try to use each of them individually to make predictions. The article that led to the best prediction is deemed to be the most relevant, and is used to train the retrieval query for future training iterations. This approach has been extended in follow-up work [33,2,34,3] and appears to be a promising approach also to retrieve the relevant context, such as definitions, possible premises, and even related proofs, for mathematical reasoning.

### 2.3 Instant Utilization of New Premises

Theorem proving has a key difference compared to other reinforcement learning settings: whenever we reach one of the goals, i.e., prove a theorem, we can use that goal as a premise for future proof attempts. Any learning method applied in a reinforcement learning setting for theorem proving thus needs the ability to adapt to this growing action space, and ideally does not need to be retrained at all when a new theorem becomes available to be used.

Premise selection approaches that are built on retrieval, such as DeepMath [24,36] and HOList [5,37], offer this ability: When a new theorem is proven, we can add it to the list of premises that can be retrieved and future retrieval queries can return the statement. This appears to work well, even when the provers are applied to a new set of theorems, as demonstrated by the DeepHOL prover when it was applied to the unseen Flyspeck theorem database [5]. We can even exploit this kind of generalization for exploration and bootstrap neural theorem provers without access to human proofs as training data [4].

A new challenge arises from the use of language models for theorem proving. Theorem provers using transformers currently have no dedicated retrieval module, and instead predict the statements or names of premises as part of the tactic string (cf. [39]). In our experience this does not provide the required generalization to unseen premises without retraining. (Though there are experiments that suggest that it might be possible [8].) Future approaches will have to find a way to combine the strong reasoning skills and generative abilities of Transformer models with the ability to use new premises without retraining.

### 2.4 Natural Language

We believe that, perhaps counterintuitively, natural language plays a central role in automated reasoning. The most direct reason is that only a small part of mathematics has been formalized so far, and a pragmatic approach to tap into much more training data is to find a way to learn from natural language mathematics (books and papers on mathematical topics). In this section, however, we want to look beyond the question of feasibility and training data, and discuss the broad advantages of a natural language approach to mathematics.

*Accessibility.* A bridge between natural and formal mathematics could help to make the system much more accessible, by not requiring the users to learn a specific formal language. This might open up mathematics to a much wider audience, enabling advanced mathematical assistants (think WolframAlpha [50]), and tools for education.

Vice versa, an advanced automatic mathematician without the ability to explain their reasoning in natural language might be hard to understand. Even if the system's predictions and theories are correct, sophisticated, and relevant, we might not be able to use them to inform our own understanding if the notions the system comes up with are only available as vast synthetic formal objects.

*Conjecturing, Theory Exploration, and Interestingness.* Various approaches have been suggested to produce new conjectures, including heuristic filters [40], deriving rules from data [7], and learning and sampling from a distribution of theorems using language modeling [41].

A particularly interesting idea is the use of adversarial training to generate conjectures (e.g., [12]). Here, two neural networks compete against each other—one with the aim to prove statements and the other with the aim to suggest hard-to-prove statements, somewhat akin to generative adversarial nets [15]. The idea is that the competition between the two networks generates a curriculum of harder and harder problems to solve and also automatically explores new parts of mathematics (as old parts get easier over time). However, there seems to be a catch: Once the network that suggests problems has figured out how to define a one-way function, it becomes very easy to produce an unlimited number of hard problems, such as to find an input to the SHA256 function that produces a certain output hash. This class of problems is almost impossible to solve, and thus likely leads the process into a dead-end.

Once again, natural language seems to be a possible answer. Using the large body of natural language mathematics could help to equip machine learning models with a notion of what human mathematicians find *interesting*, and focus on these areas.

*Grounding Language Models.* Autoformalization does not only produce formal objects as a desired outcome, it also serves the dual purpose to improve language models. Checking the models' outputs and feeding back their correctness as a training signal would provide valuable grounding for their understanding.

Of course, the gap between formalized and informal mathematics is huge: it will likely require a considerable level of effort to automatically create high quality formalizations. Also, we believe that we will likely need a very high quality theorem prover to bootstrap any autoformalization system. However, recent progress in neural language processing [9,42], unsupervised translation [30,43] and also neural network based symbolic mathematics [29,41,18,39] makes this path seem increasingly feasible and appealing in the long run.

## 3 Conclusion

In this extended abstract, we surveyed recent results in neural theorem proving and our mission to build an artificial mathematician, as well as some of the challenges on this path. While there is no guarantee that we can overcome these challenges, and there might be challenges that we cannot even anticipate yet, mere partial success to our mission could help the formal methods community with tools to simplify the formalization process, and impact adjacent areas, such as verification, program synthesis, and natural language understanding.

In a 2018 survey among AI researchers, the median prediction for when machines "routinely and autonomously prove mathematical theorems that are publishable in top mathematics journals today, including generating the theorems to prove" was in the 2060s [16]. However, over the last years, deep learning has already beaten a lot of expectations (at least ours) as to what is possible in automated reasoning. There are still several challenges to be solved, some of which we laid out in this abstract, but we believe that creating a truly intelligent artificial mathematician is within reach and will happen on a much shorter time frame than many experts expect.

## References

1. Arabshahi, F., Singh, S., Anandkumar, A.: Combining symbolic expressions and black-box function evaluations in neural programs. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=Hksj2WWAW
2. Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., Xiong, C.: Learning to retrieve reasoning paths over wikipedia graph for question answering. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=SJgVHkrYDH
3. Balachandran, V., Vaswani, A., Tsvetkov, Y., Parmar, N.: Simple and efficient ways to improve REALM. CoRR **abs/2104.08710** (2021), https://arxiv.org/abs/2104.08710
4. Bansal, K., Loos, S.M., Rabe, M.N., Szegedy, C.: Learning to reason in large theories without imitation. CoRR **abs/1905.10501** (2019), http://arxiv.org/abs/1905.10501

5. Bansal, K., Loos, S.M., Rabe, M.N., Szegedy, C., Wilcox, S.: Holist: An environment for machine learning of higher order logic theorem proving. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 454–463. PMLR (2019), http://proceedings.mlr.press/v97/bansal19a.html
6. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
7. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **113**(15), 3932–3937 (2016). https://doi.org/10.1073/pnas.1517384113
8. Cao, N.D., Izacard, G., Riedel, S., Petroni, F.: Autoregressive entity retrieval. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
11. Evans, R., Saxton, D., Amos, D., Kohli, P., Grefenstette, E.: Can neural networks understand logical entailment? In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=SkZxCk-0Z
12. Firoiu, V., Aygün, E., Anand, A., Ahmed, Z., Glorot, X., Orseau, L., Zhang, L., Precup, D., Mourad, S.: Training a first-order theorem prover from synthetic data. CoRR **abs/2103.03798** (2021), https://arxiv.org/abs/2103.03798
13. Gauthier, T., Kaliszyk, C., Urban, J.: TacticToe: Learning to reason with HOL4 tactics. In: Eiter, T., Sands, D. (eds.) LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, May 7-12, 2017. EPiC Series in Computing, vol. 46, pp. 125–143. EasyChair (2017), https://easychair.org/publications/volume/LPAR-21
14. Gonthier, G., Asperti, A., Avigad, J., Bertot, Y., Cohen, C., Garillot, F., Roux, S.L., Mahboubi, A., O'Connor, R., Biha, S.O., Pasca, I., Rideau, L., Solovyev, A., Tassi, E., Théry, L.: A machine-checked proof of the odd order theorem. In: Blazy,

S., Paulin-Mohring, C., Pichardie, D. (eds.) Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7998, pp. 163–179. Springer (2013). https://doi.org/10.1007/978-3-642-39634-2_14
15. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2672–2680 (2014)
16. Grace, K., Salvatier, J., Dafoe, A., Zhang, B., Evans, O.: Viewpoint: When will AI exceed human performance? evidence from AI experts. J. Artif. Intell. Res. **62**, 729–754 (2018). https://doi.org/10.1613/jair.1.11222
17. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 3929–3938. PMLR (2020), http://proceedings.mlr.press/v119/guu20a.html
18. Hahn, C., Schmitt, F., Kreber, J.U., Rabe, M.N., Finkbeiner, B.: Teaching temporal logics to neural networks. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
19. Hales, T., Adams, M., Bauer, G., Dang, T.D., Harrison, J., Le Truong, H., Kaliszyk, C., Magron, V., McLaughlin, S., Nguyen, T.T., et al.: A formal proof of the Kepler conjecture. In: Forum of Mathematics, Pi. vol. 5, p. e2. Cambridge University Press (2017)
20. Harrison, J.: HOL Light: A tutorial introduction. In: Srivas, M.K., Camilleri, A.J. (eds.) Formal Methods in Computer-Aided Design, First International Conference, FMCAD '96, Palo Alto, California, USA, November 6-8, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1166, pp. 265–269. Springer (1996)
21. Hellendoorn, V.J., Sutton, C., Singh, R., Maniatis, P., Bieber, D.: Global relational models of source code. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=B1lnbRNtwr
22. Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T.B., Dhariwal, P., Gray, S., Hallacy, C., Mann, B., Radford, A., Ramesh, A., Ryder, N., Ziegler, D.M., Schulman, J., Amodei, D., McCandlish, S.: Scaling laws for autoregressive generative modeling. CoRR **abs/2010.14701** (2020), https://arxiv.org/abs/2010.14701
23. Huang, D., Dhariwal, P., Song, D., Sutskever, I.: Gamepad: A learning environment for theorem proving. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=r1xwKoR9Y7
24. Irving, G., Szegedy, C., Alemi, A.A., Eén, N., Chollet, F., Urban, J.: Deepmath - deep sequence models for premise selection. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 2235–2243 (2016)
25. Jakubuv, J., Urban, J.: ENIGMA: efficient learning-based inference guiding machine. In: Geuvers, H., England, M., Hasan, O., Rabe, F., Teschke, O. (eds.) Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings. Lecture Notes in Computer Science,

vol. 10383, pp. 292–302. Springer (2017). https://doi.org/10.1007/978-3-319-62075-6_20
26. Kaliszyk, C., Urban, J., Michalewski, H., Olsák, M.: Reinforcement learning of theorem proving. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. pp. 8836–8847 (2018)
27. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models (2020), https://arxiv.org/abs/2001.08361
28. Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H., Winwood, S.: seL4: formal verification of an OS kernel. In: Matthews, J.N., Anderson, T.E. (eds.) Proceedings of the 22nd ACM Symposium on Operating Systems Principles 2009, SOSP 2009, Big Sky, Montana, USA, October 11-14, 2009. pp. 207–220. ACM (2009). https://doi.org/10.1145/1629575.1629596
29. Lample, G., Charton, F.: Deep learning for symbolic mathematics. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=S1eZYeHFDS
30. Lample, G., Conneau, A., Denoyer, L., Ranzato, M.: Unsupervised machine translation using monolingual corpora only. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=rkYTTf-AZ
31. Lederman, G., Rabe, M.N., Seshia, S., Lee, E.A.: Learning heuristics for quantified boolean formulas through reinforcement learning. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=BJluxREKDB
32. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Chen, Z.: GShard: Scaling giant models with conditional computation and automatic sharding. In: International Conference on Learning Representations, ICLR. OpenReview.net (2021)
33. Lewis, M., Ghazvininejad, M., Ghosh, G., Aghajanyan, A., Wang, S., Zettlemoyer, L.: Pre-training via paraphrasing. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
34. Lewis, P.S.H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
35. Li, W., Yu, L., Wu, Y., Paulson, L.C.: IsarStep: A benchmark for high-level mathematical reasoning. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
36. Loos, S.M., Irving, G., Szegedy, C., Kaliszyk, C.: Deep network guided proof search. In: Eiter, T., Sands, D. (eds.) LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, May

7-12, 2017. EPiC Series in Computing, vol. 46, pp. 85–105. EasyChair (2017), https://easychair.org/publications/paper/ND13

37. Paliwal, A., Loos, S.M., Rabe, M.N., Bansal, K., Szegedy, C.: Graph representations for higher-order logic and theorem proving. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 2967–2974. AAAI Press (2020), https://aaai.org/ojs/index.php/AAAI/article/view/5689
38. Piotrowski, B., Urban, J.: Atpboost: Learning premise selection in binary setting with ATP feedback. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10900, pp. 566–574. Springer (2018). https://doi.org/10.1007/978-3-319-94205-6_37
39. Polu, S., Sutskever, I.: Generative language modeling for automated theorem proving. CoRR **abs/2009.03393** (2020), https://arxiv.org/abs/2009.03393
40. Puzis, Y., Gao, Y., Sutcliffe, G.: Automated generation of interesting theorems. In: Sutcliffe, G., Goebel, R. (eds.) Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006. pp. 49–54. AAAI Press (2006), http://www.aaai.org/Library/FLAIRS/2006/flairs06-009.php
41. Rabe, M.N., Lee, D., Bansal, K., Szegedy, C.: Mathematical reasoning via self-supervised skip-tree training. In: International Conference on Learning Representations, ICLR. OpenReview.net (2021)
42. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. In: OpenAI Blog (2018)
43. Rozière, B., Lachaux, M., Chanussot, L., Lample, G.: Unsupervised translation of programming languages. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
44. Schulz, S.: Learning search control knowledge for equational theorem proving. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001: Advances in Artificial Intelligence, Joint German/Austrian Conference on AI, Vienna, Austria, September 19-21, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2174, pp. 320–334. Springer (2001). https://doi.org/10.1007/3-540-45422-5_23
45. Selsam, D., Lamm, M., Bünz, B., Liang, P., de Moura, L., Dill, D.L.: Learning a SAT solver from single-bit supervision. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=HJMC_iA5tm
46. Urban, J.: MPTP - motivation, implementation, first experiments. J. Autom. Reason. **33**(3-4), 319–339 (2004). https://doi.org/10.1007/s10817-004-6245-1
47. Urban, J., Sutcliffe, G., Pudlák, P., Vyskocil, J.: Malarea SG1- machine learner for automated reasoning with semantic guidance. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5195, pp. 441–456. Springer (2008). https://doi.org/10.1007/978-3-540-71070-7_37

48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017)
49. Wang, Q., Brown, C.E., Kaliszyk, C., Urban, J.: Exploration of neural machine translation in autoformalization of mathematics in Mizar. In: Blanchette, J., Hritcu, C. (eds.) Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020. pp. 85–98. ACM (2020). https://doi.org/10.1145/3372885.3373827
50. WolframAlpha: WolframAlpha (2016), http://www.wolframalpha.com/
51. Wu, M., Norrish, M., Walder, C., Dezfouli, A.: Tacticzero: Learning to prove theorems from scratch with deep reinforcement learning. CoRR **abs/2102.09756** (2021), https://arxiv.org/abs/2102.09756
52. Wu, Y., Jiang, A., Ba, J., Grosse, R.B.: INT: An inequality benchmark for evaluating generalization in theorem proving. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
53. Wu, Y., Rabe, M., Li, W., Ba, J., Grosse, R., Szegedy, C.: LIME: Learning inductive bias for primitives of mathematical reasoning. In: Proceedings of International Conference on Machine Learning (to appear) (2021)
54. Yang, K., Deng, J.: Learning to prove theorems via interacting with proof assistants. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 6984–6994. PMLR (2019), http://proceedings.mlr.press/v97/yang19a.html