Intrinsic Motivation To Construct Terms In A Dependent Type Theory

Guy Axelrod[1 [0000−0002−1752−8069]], Moa Johansson[1 [0000−0002−1097−8278]], Andrea Silvi[1 [0009−0005−2104−4576]], and Devdatt Dubhashi[1 [0000−0002−9928−2305]]

Chalmers University of Technology, Gothenburg, Sweden
{guya,moa.johansson}@chalmers.se

This extended abstract outlines future work proposed for the first authors PhD thesis. The aim is to receive feedback, advice, ideas, collaboration and/or criticisms from those currently working in similar areas.
In a recent work [5], Poesia et al. introduce *minimo* - an approach to teaching an agent to prove mathematical statements through a process of self-play (comparable to AlphaZero) in the environment of dependent type theory, starting from nothing but a finite collection of axioms. A particularly novel aspect of their approach is its attempt to incorporate concepts from the reinforcement learning literature on *intrinsic motivation* [4] (particularly [3]). That is, we also expect the agent to learn to iteratively pick its own goals (i.e., conjectures) by interacting with the environment. In principle, we wish for these goals to become increasingly "difficult" and "interesting", in the sense that learning to prove them eventually leads to an agent capable of proving conjectures that are extrinsically provided by a human mathematician. Their approach pursues this idea by using a randomly initialized transformer-based language model (LM) to act as both a conjecturer and as the policy and value function for use in a Monte Carlo Tree Search (MCTS) over proof steps. The LM is then jointly trained for conjecturing and proof search by iterating over the following steps:

1. Conditioned on an indication of high "difficulty" in the prompt, the LM is used to generate a batch of conjectures (i.e., terms of type Prop). Constrained Semantic Decoding [7] ensures that only well-typed conjectures are sampled.
2. For each conjecture in the batch, MCTS (where value and policy for a given state are sampled by prompting the LM) is performed with the aim of generating a proof term of the corresponding type.
3. Training examples for both conjecturing and proof search are extracted from the constructed search trees. Paths in the tree for which the leaf corresponds to the desired term are considered to be successful trajectories. The log-likelihood of these trajectories under the policy are taken as a measurement of the "difficulty" of proving the conjecture.
4. The LM is trained using standard cross-entropy loss as the objective.

Aspects of intrinsic motivation arise through the interplay of steps 1 and 3. Specifically, they aim to produce an agent capable of generating challenging but achievable next goals.
Another interesting aspect of their approach is the use of Hindsight Experience

Replay (HER) [1] in step 3, in line with the suggestions of [8]. Trajectories that failed on the generated conjecture can still be considered successful when simply relabeled with goals that were in fact achieved. This allows the extraction of substantially more training data for both conjecturing and proof search.
Through small scale experiments in the theories of propositional logic and natural number arithmetic, the authors claim that a self-improvement loop arises, in which the agent steadily become more successful at proving extrinsically provided conjectures which were not seen during training.

We believe the overall approach does indeed show promise, and provides a good framework for exploring new ideas regarding RL for theorem proving. There are, however, many limitations, and hence possible directions for future work. One direction concerns the choice of proof environment. Currently, proof search is conducted within *Peano* [6], a minimal, experimental proof assistant implemented by the primary author of [5]. They justify this choice by pointing to the fact that *Peano* provides a finite action space for search in a dependent type theory. Nonetheless, action enumeration still becomes prohibitively expensive as one progresses deeper into the search tree. We believe it would be beneficial to extend their approach in order to explore intrinsic motivation in the environment provided by either Lean or Agda, where we may try to adapt [7]'s constrained decoding technique for the task of generating valid proof steps. This further presents us with the opportunity to take advantage of the powerful automation available in these languages. The recently released *Pantograph* [2] provides a rich and convenient interface for interaction with Lean 4's environment. We are currently exploring it's potential use in the *minimo* approach at the time of writing this abstract. On the other hand, we would also be eager to consider Agda as an environment, thus providing an opportunity to possibly collaborate with the authors of [8] and the many Agda developers present at Chalmers.

References

1. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., Zaremba, W.: Hindsight experience replay. Advances in neural information processing systems **30** (2017)
2. Aniva, L., Sun, C., Miranda, B., Barrett, C., Koyejo, S.: Pantograph: A machine-to-machine interaction interface for advanced theorem proving, high level reasoning, and data extraction in lean 4 (2024), https://arxiv.org/abs/2410.16429
3. Campero, A., Raileanu, R., Küttler, H., Tenenbaum, J.B., Rocktäschel, T., Grefenstette, E.: Learning with amigo: Adversarially motivated intrinsic goals. arXiv preprint arXiv:2006.12122 (2020)
4. Chentanez, N., Barto, A., Singh, S.: Intrinsically motivated reinforcement learning. Advances in neural information processing systems **17** (2004)
5. Poesia, G., Broman, D., Haber, N., Goodman, N.D.: Learning Formal Mathematics From Intrinsic Motivation (Nov 2024). https://doi.org/10.48550/arXiv.2407.00695, http://arxiv.org/abs/2407.00695, arXiv:2407.00695 [cs]

6. Poesia, G., Goodman, N.D.: Peano: Learning Formal Mathematical Reasoning. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **381**(2251), 20220044 (Jul 2023). https://doi.org/10.1098/rsta.2022.0044, http://arxiv.org/abs/2211.15864, arXiv:2211.15864 [cs]
7. Poesia, G., Polozov, O., Le, V., Tiwari, A., Soares, G., Meek, C., Gulwani, S.: Synchromesh: Reliable code generation from pre-trained language models (2022), https://arxiv.org/abs/2201.11227
8. Rawson, M., Zombori, Z., Doré, M., Wernhard, C.: Project proposal: Forward reasoning in hindsight