MiniF2F in Rocq: Automatic Translation Between Proof Assistants — A Case Study

Jules Viennot,[1] Guillaume Baudart,[1]
Emilio Jesús Gallego Arias,[1] Marc Lelarge[2]

[1] IRIF, Université Paris Cité, Inria, CNRS
[2] DI ENS, PSL University, Inria

Recent LLMs have demonstrated impressive ability in proving theorems using interactive theorem provers (ITP) such as Isabelle [1,2], Lean [3,4], or Rocq [5,6]. Unfortunately, fundamental differences between proof systems have resulted in diverse datasets, which makes the comparison between techniques developed for different proof assistant particularly challenging.

However, LLMs are particularly well fit for translating between programming languages that have extensive resources in common [7]. In this work, we explore whether state-of-the-art LLMs can be leveraged to automatically translate a dataset of formal theorems from one proof assistant to another. We focus on MiniF2F [8,9], a dataset of high-school-level problems that has already been formalized in Lean, Isabelle/HOL, and MetaMath. This dataset is a popular benchmark for evaluating ML-based automated proof techniques [10,11,12,13]. Despite previous attempts, this dataset has not yet been formalized in Rocq.[3]

In this work, we conduct an experiment using state-of-the-art LLMs to translate MiniF2F into Rocq. Our approach successfully translated 478 out of 488 theorems.

Methodology The translation task focuses on generating a Rocq theorem based on three sources: a natural language description, the Lean formalization, and the Isabelle formalization. Only theorem statements are considered, proofs, whether described in natural language or formalized in Lean or Isabelle, are deliberately ignored. For each theorem we check that the result of the translation is a valid Rocq statement using Pétanque [14], a machine to machine interactive environment for Rocq.

We conducted our experiment in 3 stages of increasing complexity, from basic one-shot prompting to multi-turn conversations that incorporate feedback from unsuccessful attempts. At each stage, we perform multiple rounds of translation using increasingly advanced models. After each round, we verify that the generated codes matches the informal descriptions and the existing formalizations. To limit the costs, the next round only focuses on theorems that remains untranslated.

– *Stage 1: One-shot prompting.* We prompt the model with the natural language description of the theorem and the existing formalizations (Lean and Isabelle/HOL). In this stage, we used GPT-4o mini, Claude 3.5 Sonnet, o1 mini, and o1.

[3] https://github.com/openai/miniF2F/issues/66

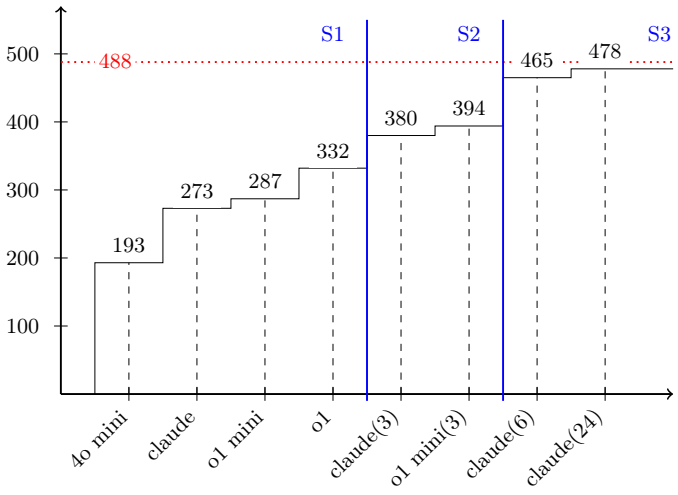

Fig. 1. Translations of MiniF2F to Rocq, experimental results. For stages 2 and 3 we indicate the number of attempts in parenthesis.

- *Stage 2: Multi-turn with errors.* The model can interact up to three times with the theorem provers. Each new try incorporate the previous unsuccessful attempts and the error messages. In this stage we used Claude 3.5 Sonnet and o1 mini.
- *Stage 3: Refined prompt.* We focus on Claude 3.5 Sonnet. The prompt is refined to address common errors related to complex numbers, finite sums or products, prime numbers, the floor function, and typing issues. We increase the number of attempts to 6, and then 24.

For a more technical insight, our work is available at https://github.com/LLM4Rocq/miniF2F-rocq

Results The cumulative results of each round are presented in Figure 1. We observe that with basic one-shot prompting, we translate 68% of the dataset requiring the use of the most advanced model o1. Stages 2 and 3 illustrate that the LLM can leverage successfully previous attempts. At the end of our experiments only 10 theorems (2% of the dataset) remains untranslated.

Audit To validate our results, we asked experts to compare the Rocq translations with the original formalizations for a random sample of 150 theorems. This audit revealed 3 errors, 6 theorems requiring reformulation, and 26 cases where the syntax could be improved.

Discussion We successfully leveraged state-of-the-art LLMs to translate the MiniF2F dataset to Rocq. However, several questions remain. First, regarding model comparison: do models considered superior indeed perform better on this task? Second, concerning optimal translation setup: could performance

improve by excluding certain formalizations from the input? Finally, the resulting formalizations may potentially make proofs more challenging than in other proof assistants. Despite these open questions, we believe our work represents a significant step forward.

References

1. Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. In *NeurIPS*, 2022.
2. Emily First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models. *CoRR*, abs/2303.04910, 2023.
3. Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
4. Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. *CoRR*, abs/2306.15626, 2023.
5. Shizhuo Dylan Zhang, Talia Ringer, and Emily First. Getting more out of large language models for proofs. *CoRR*, abs/2305.04369, 2023.
6. Kyle Thompson, Nuno Saavedra, Pedro Carrott, Kevin Fisher, Alex Sanchez-Stern, Yuriy Brun, João F. Ferreira, Sorin Lerner, and Emily First. Rango: Adaptive retrieval-augmented proving for automated software verification. *CoRR*, abs/2412.14063, 2024.
7. Yichen Xu and Yanqiao Zhu. A survey on pretrained language models for neural code intelligence. *CoRR*, abs/2212.10079, 2022.
8. Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.
9. 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. In *Submitted to The Eleventh International Conference on Learning Representations*, 2022.
10. Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.
11. Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An in-context learning agent for formal theorem-proving. In *First Conference on Language Modeling*, 2024.
12. Maciej Mikuła, Szymon Tworkowski, Szymon Antoniak, Bartosz Piotrowski, Albert Qiaochu Jiang, Jin Peng Zhou, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. Magnushammer: A transformer-based approach to premise selection. *arXiv preprint arXiv:2303.04488*, 2023.
13. Haiming Wang, Huajian Xin, Zhengying Liu, Wenda Li, Yinya Huang, Jianqiao Lu, Zhicheng Yang, Jing Tang, Jian Yin, Zhenguo Li, et al. Proving theorems recursively. *arXiv preprint arXiv:2405.14414*, 2024.
14. Laetitia Teodorescu, Guillaume Baudart, Emilio Jesús Gallego Arias, and Marc Lelarge. Nlir: Natural language intermediate representation for mechanized theorem proving. In *MathAI@NeurIPS*, 2024.