

# Introducing mathematics undergraduate students to the theorem prover lean

Irene Garnelo<sup>1</sup> and Michael Liebendörfer<sup>1</sup>

<sup>1</sup>Paderborn University, Germany; [ireneg@math.uni-paderborn.de](mailto:ireneg@math.uni-paderborn.de)

*Keywords: Mathematical logic, undergraduate studies, computer assisted instruction.*

## Introduction

In the secondary-tertiary transition, many students struggle with proof. Moore (1994) identifies the most prominent problems as understanding the underlying mathematical concepts, the mathematical language and notation, and the strategic knowledge, particularly, finding a proof strategy. While digital tools support learning in many areas, only a few tools are employed to enhance the learning of theorem proving (Sangwin & O’Toole, 2017). These problems have been recently tackled successfully with a theorem prover (Thoma & Iannone, 2022).

Theorem provers are software tools in which a computer-formalized proof can be constructed. The tools provide feedback on the state of the proof interactively making it possible to provide an overview of the mathematical concepts relevant to a proof, and training students to be more careful with their writing by correcting their mistakes interactively. Finally, it gives an overview of proof strategies and shows their effect on the current proof state interactively.

The aims of the research investigation from which we report here are to identify on which level of knowledge of programming (syntactic, conceptual, strategic; Qian & Lehman, 2017) students succeed or face obstacles when working with the learning environment.

## The learning environment

The learning environment (available here: [https://irenegarnelo.github.io/nat\\_zahlen\\_lean/](https://irenegarnelo.github.io/nat_zahlen_lean/)) includes explanatory texts, an online theorem prover, and a set of written exercises. The online tool is a Lean-based website inspired by the “Natural number game” (Buzzard & Pedramfar, 2021), and includes several entry level theorems to be proven in Lean (<https://leanprover-community.github.io/>), as well as mathematical and technical instructions. The written exercises aim to encourage the students to reflect on their usage of the tool and the different representations of proofs.

## Research questions

For this study, our research questions are centred around the usability of the tool and the process of learning Lean as a programming language and read as follows:

1. How do undergraduate mathematics students interact with the learning environment?
2. On which level of knowledge of programming do students succeed or face obstacles when working with the learning environment?

## Study design

The learning environment was used in a module of the lecture “mathematics with a computer” held in the third semester for approximately 30 preservice teachers at the University of Paderborn in two

90-minute units. Students were introduced to computer-assisted proving, Lean, and the learning environment. Then they used the learning environment in pairs and discussed their experiences and their opinions in a common reflection phase, where they were confronted with three questions about theorem provers and their personal experience with them.

During the working and the discussion phases, five students volunteered to be observed, separately from the rest of the students. The work was documented with a screen recording and a camera recording of the working pairs including the audio. The written results of the reflection phase were collected from all students.

## **Preliminary results and discussion**

The navigation in the learning environment worked well and students made use of all available resources, resorting especially to the instructions and previous proofs but not as often to the error messages. An example for a problem on the syntactical level was the misspelling of tactics, on the conceptual level we found that students applied theorems that were invalid given the proof state, while on the strategic level students failed to choose a proof strategy as they would in a pen-and-paper proof. The full identification and categorization of problems can be found on the poster<sup>1</sup>, which also features a visual overview of Lean and the learning environment as well as the conducted study.

The categorized difficulties only partially overlap with the challenges in classical proof environments. Therefore, mastering Lean requires an initial effort. However, at the conceptual and especially at the strategic level, the challenges soon resemble the problems described by Moore (1994). By showing whether mastering challenges with a theorem prover supports learning the analogous challenges in mathematical proof, future research could show whether the initial learning effort is worthwhile.

## **References**

- Buzzard, K., & Pedramfar, M. (2021). *Natural Number Game*. Retrieved April 21, 2023, from [https://www.ma.imperial.ac.uk/~buzzard/xena/natural\\_number\\_game/](https://www.ma.imperial.ac.uk/~buzzard/xena/natural_number_game/)
- Moore, R. C. (1994). Making the transition to formal proof. *Educational Studies in Mathematics*, 27(3), 249–266. <https://doi.org/10.1007/BF01273731>
- Sangwin, C. J., & O’Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), 1133–1152. <https://doi.org/10.1080/0020739X.2017.1315186>
- Thoma, A., & Iannone, P. (2022). Learning about proof with the theorem prover lean: the abundant numbers task. *International Journal of Research in Undergraduate Mathematics Education*, 8(1), 64–93. <https://doi.org/10.1007/s40753-021-00140-1>
- Qian, Y., & Lehman, J. (2017). Students’ misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1–24. <https://doi.org/10.1145/3077618>

---

<sup>1</sup> [https://go.upb.de/lean\\_poster](https://go.upb.de/lean_poster)