



HAL
open science

Use of two proof assistants in an introduction to proof course: an experiment

Frédéric Tran Minh

► **To cite this version:**

Frédéric Tran Minh. Use of two proof assistants in an introduction to proof course: an experiment. LCIS, Grenoble-INP. 2023. hal-04705617

HAL Id: hal-04705617

<https://hal.science/hal-04705617v1>

Submitted on 23 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Research Report

Use of two proof assistants in an introduction to proof course: an experiment

Frédéric Tran Minh 

UGA - Grenoble INP - Esisar

Abstract

The use of proof assistants to support the teaching of mathematics at the undergraduate level is gaining popularity but faces obstacles such as the appropriation of these tools by students, or their difficulty in transferring their acquired computer skills into an ability to write pen/pencil proofs. Various types of interface (declarative or imperative, graphical or textual) have been developed to overcome them. In this article, we describe an experience where students interact with two different pieces of software (Edukera, Lean) and we attempt to analyze their subjective impressions, in order to contribute to a reflection on possible software or pedagogical adaptations favoring the impact of such tools on proof learning.

1 Use of proof assistants in teaching: current context

Learning proof in mathematics is a major challenge at the high school-university interface, particularly its logical and formal aspects [25, 15].

Since AutoMath in 1967 [5], numerous interactive proof assistance software has been developed [28] including the proof assistants Mizar (1972, [14]), Isabelle (1988, [17, 27]), Coq (1989, [7]), and more recently Lean (2015, [8]).

As an early experience, Krzysztof and Zalewska [20] report the use of Mizar as a tool for teaching propositional logic in 1975. Since then, many courses in theoretical computer science or in formal logics use proof assistants (Delahaye et al. [9], Nipkow [16], Pierce [18], Avigad [1]). Several e-Learning environments have also emerged to support geometry proof learning in secondary education (Geometrix [10] ; AgentGeom [6] ; Baghera [24] ; Turing [21]; QEDTutrix [13]). However, this practice seems to be largely less widespread regarding standard undergraduates mathematics: in 2021, Hanna and Yan stated that [11] “A few universities have offered courses on the use of digital proof assistants, but undergraduate mathematics curricula have neglected them”.

Some reports nevertheless attest to their use in undergraduate mathematics education, particularly in algebra and analysis [19, 12, 4]. A review of some proof assistants and classification criteria with regards to their potential impact for teaching are given in [2].

The available tools seem to allow a better appropriation of mathematical proof mechanisms by students [23] but still pose problems, such as the transfer to pen and paper proofs and the adaptation to the syntax of the chosen proof language. To try to resolve these difficulties, various tools have been developed on top of Coq or Lean: Edukera (a graphical interface based on Coq, closed source, Rognier [22]); CoqWeb (Blanc et al., [3]) ; Lean Verbose (Lean tactics language imitating natural language style ("controlled natural language"), Patrick Massot, [12]); Deaduction (graphical interface for Lean, Frédéric Leroux, [12]); Coq Waterproof (similar to Lean Verbose, includes also a graphical interface to assist students, Jim Portegies *et. al.* [26]).

In addition, the didactic question of the impact of proof assistants on proof learning would still need further research, since, according to Hanna and Yan, “there does not appear to be any published systematic research that has explored their potential in any educational context” [11].

2 Experiment Objectives

We aim to introduce the use of proof assistants in first-year mathematics education at Grenoble-INP-Esisar engineering school, to support an introduction to deductive reasoning and proof learning, with the following pedagogical objectives compared to traditional teaching:

- Increased ability for students to work independently;
- Possibility for students to receive rapid feedback on the correctness ¹ of their work;
- Increased and more repetitive training in the rigor of formalism, and appropriation of mathematical notation and syntax;

¹From the proof assistant point of view, a correct proof is a proof-term that type-checks

- Practical elaboration of “proof objects” leading to better understanding of its essential nature and mechanism ;
- Practical manipulation of the concept of object type (in the computer science sense) leading to a better identification of the nature of mathematical objects.

We approach the experiment described here with the following questions:

- To what extent is the direct use of a proof assistant language, without prior preparation, a hindrance or a driving force for learning the mechanism of proof, for first-year undergraduate students?
- To what extent could these potential obstacles be overcome by appropriate educational support?
- To what extent could these potential obstacles be overcome by a technical adaptation of the tools?
- Does the relative ease of using a graphical interface encapsulating and hiding a proof assistant solve these difficulties or introduce other biases?

To this end, we proposed to a group of students to use, over the same period, the Lean proof assistant language on the one hand, and the Edukera graphical tool on the other.

- Edukera, whose interface is depicted in Figure 2 is a didactic graphical environment dedicated to the learning of formalization and deductive reasoning, created in 2015 by Benoit Rognier and Guillaume Duhamel [22]. It includes a database of exercises (not modifiable), a tutorial - it is thus designed for self-use and self-training - and its integration into Moodle makes it easy for the teacher to collect evaluation results. It is designed for online use (with internet access, in the browser). Its engine is the Coq proof assistant, but the Coq language is never exposed and proofs are not checked by Coq. The code is closed and proprietary.

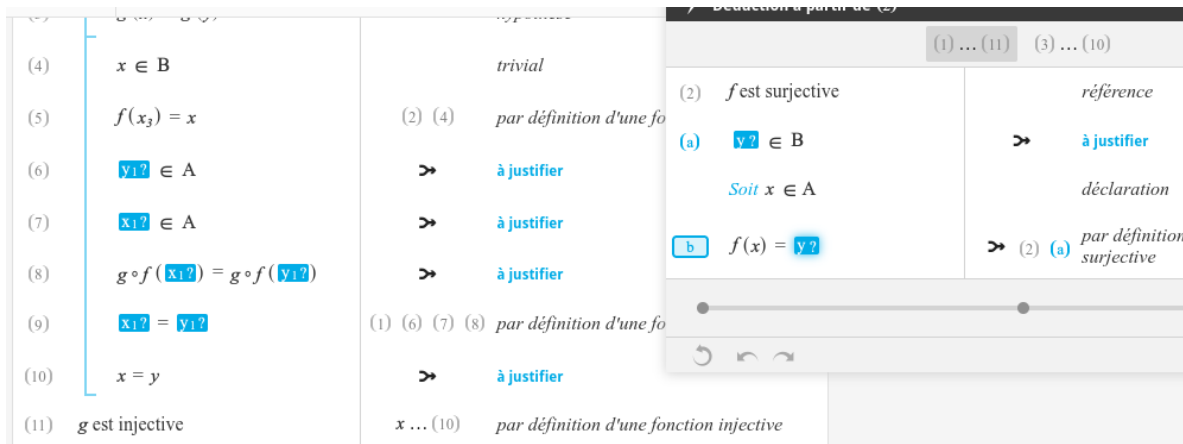


Figure 1: Edukera graphical environment

- Lean ² (Figure 2) is a proof assistant developed since 2013 mostly by Leonardo de Moura at Microsoft Research and Carnegie Mellon University ([8]). Its source code is open. It now has a large mathematical library developed by the community, covering at least the knowledge of a mathematics degree, and much more in certain specific areas. It can be used either on the command line, with a development environment (VsCode), or online in a browser (Javascript version).

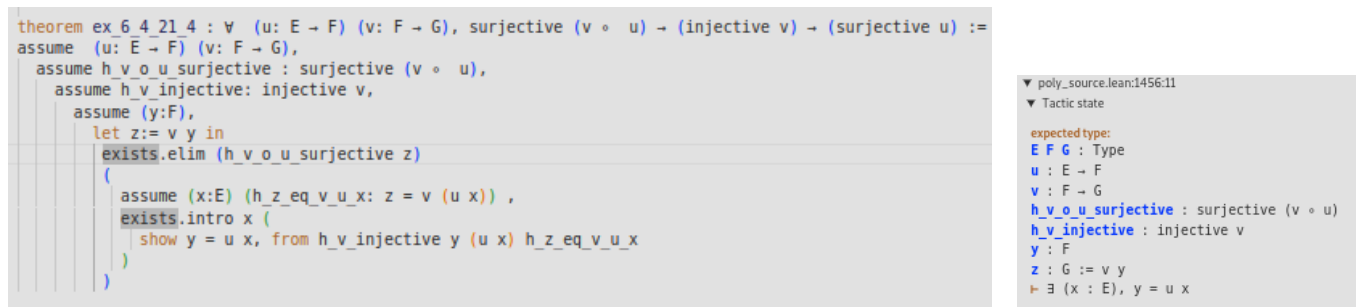


Figure 2: Lean: source code excerpt (proof-term style) (left) and tactic state display (right)

²<https://leanprover.github.io/about/>

3 Context of the experiment

3.1 Human and material context

The experiment was carried out with first-year undergraduate students of a 5-year engineering school, training computer scientists, electronic engineers, and automation engineers. It takes place over a semester, in the context of a small module of 6 sessions of an hour and a half, in parallel with a classical course in algebra and analysis taught by another teacher (96 hours of lectures and tutorials on the same semester). The class size is 64 students, and the sessions take place in a computer room in half groups of 32 students; enrollment and attendance are mandatory. During the same period, these students also take a class in algorithmics and programming (30 hours of lectures, tutorials, and practical work), and their curriculum includes a course in functional programming in Haskell during their second year.

3.2 Pedagogical organization

After an introduction to the Edukera software during about one session, most of the following sessions were devoted to the use of the Lean proof assistant. At the same time, a choice of 39 exercises from the Edukera database was left to be completed autonomously during the semester, with occasional help during the sessions. Edukera is used in its integrated version in Moodle to facilitate monitoring of progress. The use of Lean was quite directed for about two sessions (lectures, correction of exercises at the board) and then autonomous for 3 sessions (occasional interventions by the teacher based on individual questions). Lean is used in its browser version (Javascript front-end).

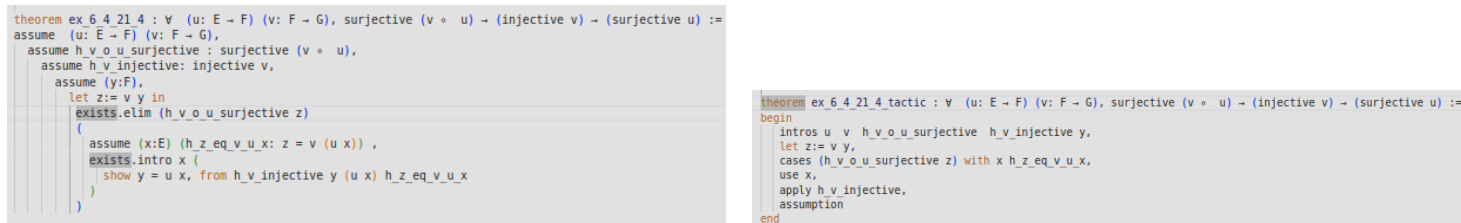
Student evaluation on the module divides into three parts:

- Progress in autonomous work on Edukera
- A paper test consisting of a multiple-choice questionnaire on the introduction/elimination concepts in Lean, and an exercise converting Lean source code into natural language mathematical text
- A machine test where students were required to state definitions and theorems in Lean, and prove theorems in Lean.

All course materials as well as the assessment subject are available on

https://github.com/ftranminh/Esisar_MA121_HA_lean_2023.

3.3 Term mode and tactic mode



```
theorem ex_6_4_21_4 : ∀ (u: E → F) (v: F → G), surjective (v ∘ u) → (injective v) → (surjective u) :=
assume (u: E → F) (v: F → G),
assume h_v_o_u_surjective : surjective (v ∘ u),
assume h_v_injective: injective v,
assume (y:F),
let z:= v y in
exists.elim (h_v_o_u_surjective z)
(
  assume (x:E) (h_z_eq_v_u_x: z = v (u x)) ,
  exists.intro x (
    show y = u x, from h_v_injective y (u x) h_z_eq_v_u_x
  )
)

theorem ex_6_4_21_4_tactic : ∀ (u: E → F) (v: F → G), surjective (v ∘ u) → (injective v) → (surjective u) :=
begin
  intros u v h_v_o_u_surjective h_v_injective y,
  let z:= v y,
  cases (h_v_o_u_surjective z) with x h_z_eq_v_u_x,
  use x,
  apply h_v_injective,
  assumption
end
```

Figure 3: Comparison between term-mode et and tactic-mode

In Lean, two modes of proof writing are available. Figure 3 illustrates these modes. In "proof-term" mode, a theorem statement is a type τ ; and a proof of it is a λ -term that Lean checks to be of that type. Following the correct syntax consists in applying a function to arguments (that may themselves result from nested function calls). In "tactic" mode, the writing of the proof term is hidden by the use of higher-level instructions called tactics, which sequentially and imperatively construct a proof term by manipulating the hypotheses and/or the goal at each step.

In practice, the use of one or the other of these modes may appear very similar. As far as we know, in most research or teaching situations, the tactic mode is used. However, in the context of this experiment, I chose to present the proof-term mode, firstly to highlight the rigorously structured, nested nature of a mathematical proof, similar to a functional programming term (reminder: functional programming is part of our students' future curriculum); and secondly to demystify the opacity of the application of tactics.

3.4 Covered topics

Three types of exercises are addressed with Edukera:

1. Formalization of sentences in French (e.g. : the sentence "not everybody has read *Les Misérables*" has to be formalized as $\exists z, H(z) \wedge \neg R(z, LM)$, where $H(\cdot)$ means " \cdot is human" and $R(\cdot, \cdot)$ means " \cdot has read \cdot ").

2. Formalization of propositions or mathematical definitions expressed in French (e.g. the proposition “ f is increasing on D ” should be formalized as $\forall x, x \in D \implies \forall y, ((y \in D \wedge x \leq y) \implies f(x) \leq f(y))$).
3. Some elementary proofs on the theme of sets/functions (little addressed)

Four themes are addressed with Lean:

1. Elements of propositional logic
2. Some elementary arithmetic proofs on integers (first-order logic)
3. Some properties of functions (about injectivity, surjectivity, image and preimage...)
4. Basics of analysis (definition of limit of a numerical sequence, some proofs) (little discussed)

4 Results

The experience received a fairly positive qualitative and subjective feedback. A questionnaire has been submitted to the students during the last session. The questions and the numerical results are presented in appendix 8)

Three open questions were also proposed to obtain broader impressions:

- The use of the computer allows for rapid feedback on your mathematical proofs. In your opinion, what is the impact of this contribution on your progress?
- Can you propose improvements (technical, pedagogical) aimed at improving the efficiency of this type of practical work?
- Free comments.

Regarding Edukera, the ease of use seemed very easy to easy (87%), but the exercises seemed difficult or of heterogeneous difficulty (72%). They felt that the use of Edukera helped them to understand the syntax and role of quantifiers (93% each), but 25% think that it was of no help in understanding the role of proof or in learning how to construct proofs.

Regarding Lean, the handling seemed difficult to very difficult (100%) and the exercises seemed difficult or of heterogeneous difficulty (100%). The use of Lean would have allowed 80% to improve their syntax skills and understand the role of proof, but 33% consider this experience to have no effect on identifying the nature of objects or on learning how to construct proofs.

The sessions were experienced as very fun by 54% of the students (and at least a little by 95%).

Finally, 77% of respondents believe that difficulties encountered with Lean are related to the syntax of the language, but only 26% attribute them to mathematical shortcomings. The teacher’s opinion however, based on interactions during the sessions, is more nuanced... For example, considering that the definition of injectivity translates almost literally in Lean, the students writing the definition of surjectivity instead, but relating their failure to syntactical hindrances are not credible... A pedagogical challenge of such a module may rely on helping the students relate their difficulties with the syntax of the proof assistant to logical and mathematical shortcomings to be overcome. This hypothesis would require further research.

From answers to open questions, aspects that participants specifically wanted to highlight are collected. 44% explicitly describe a positive impact of the sessions on their mathematical learning (compared to 3% explicitly stating no impact). 41% highlight the interest of rapid and live feedback, and 11% explicitly welcome the autonomy allowed and the self-correction capacity offered by this feedback; however, 18% assert that the feedback is binary (whether the proof is correct or not) but does little to help them find the source or cause of the error. Several students praised the individualization of the pace of work, noted that their attention have been attracted to rigor and details, or appreciated the experimental and playful nature of this kind of mathematical activity.

It should be noted that 5% express a misunderstanding in the case of a correct proof: indeed, the proof is correct as soon as Lean does not give any feedback (does not highlight any error), a tacit behavior that can be surprising and apparently not sufficiently explained.

16% deplore the difficulty of appropriating the Lean language and in particular its syntax, and 33% have asked for more lecturing on the syntax of the language; however, 15% express a marked opposition between the syntactic difficulty of the language and difficulties in mathematical logic, an opposition that I partly contest as mentioned above.

Finally, 13% have requested additional sessions and/or exercises to consolidate or deepen their experience with Lean.

5 Interpretation

While the Edukera interface seems attractive and easy to use (and even self-sufficient thanks to the tutorials), it is worth questioning whether it is possible to answer correctly through trial and error without developing a deep understanding or an ability to construct a written proof. Furthermore, as definitions are provided and even presented in a form where the main connector has already been introduced, the effort of memorization is taken off the learner, which may be deplored

in certain situations. However, Edukera natively allows for the verification of the correctness of the formalization of a statement described in natural language.

Conversely, access to programming in Lean is initially more austere, as its rigorous syntax requires a longer learning process (however, once this barrier is overcome, the activity seems playful). Worse, the rigor of this syntax may appear, to those who practice it, falsely disconnected from the demand for mathematical rigor, while it is precisely to convince of the necessity of mathematical formalization that one would like to rely on a proof assistant for.

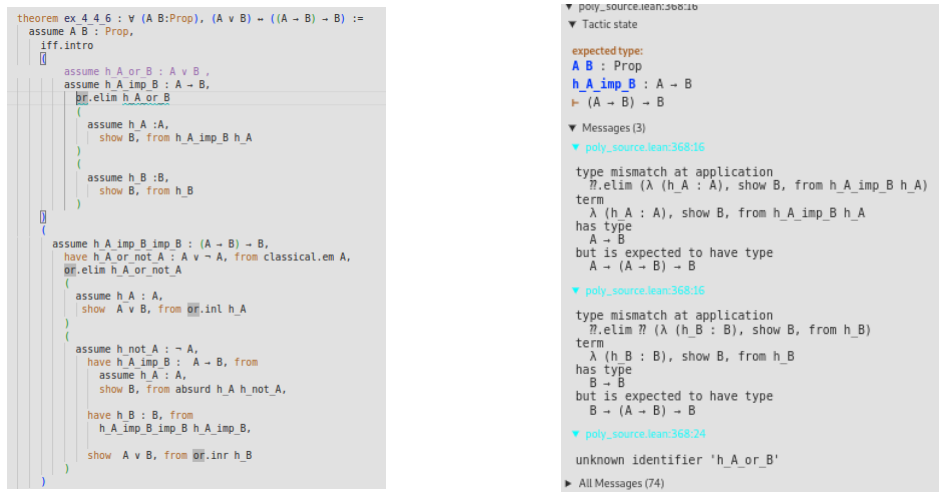


Figure 4: Error message in Lean, resulting from forgetting the commented line (introduction of `h_A_or_B`)

As a third observation, one of the main remarks from the students is the opacity, to their eyes, of the Lean error messages (Figure 4). Their feeling is that instant feedback is given on their work, but it is binary. On the one hand, a correct interpretation of these returns requires a deeper knowledge of the language than is necessary to translate elementary proofs. On the other hand, the error message may be signaled in a place that is not exactly the real source of the error (notably: problem of matching parentheses, etc.). Finally, and most importantly, the errors reported are at the microscopic level of syntax or typing error, while the students most often expect a more global indication about their strategy. These weaknesses result from the fact that the proof assistants making it possible to formalize undergraduate mathematics (notably through the availability of a sufficiently large mathematical library of definitions and lemmas) are intended for professional use rather than for pedagogical use.

6 Perspectives

From the previous observations, we formulate the following research questions:

- Would a hybrid tool efficiently bridge the gap between the ease of handling a graphical tool and the demand of mental elaboration permitted by the full initiative of constructing by hand a proof script in the target language (Coq, Lean...)?
- How could the student usability of proof assistants error messages be improved?
- Do overlays of proof assistants enabling the use of a “controlled natural language” (like Lean Verbose developed by Massot [12], or Coq/Waterproof developed by Portegies [26]) convince students of the correlation between learning the proof assistant syntax and improving their pen and paper proof skills?
- In the context of regular use of a proof assistant in mathematics classes during a whole degree, for students whose curriculum explicitly includes computer science training (particularly in functional languages), would the investment necessary for learning the specific syntax of a proof assistant be profitable? In this precise type of context, does a tool that requires the student to make explicit the hierarchical structure of the proof facilitate the appropriation of a common abstraction to mathematics and computer science?

7 Acknowledgements

Many thanks to Laure Gonnord and Julien Narboux for their valuable advice.

References

- [1] Jeremy Avigad. Learning Logic and Proof with an Interactive Theorem Prover. In Gila Hanna, David A. Reid, and Michael de Villiers, editors, *Proof Technology in Mathematics Research and Teaching*, volume 14, pages 277–290. Springer International Publishing, Cham, 2019. Series Title: Mathematics Education in the Digital Era.
- [2] Evmorfia Bartzia, Antoine Meyer, and Julien Narboux. Proof assistants for undergraduate mathematics and computer science education: elements of a priori analysis. In María Trigueros, editor, *INDRUM 2022: Fourth conference of the International Network for Didactic Research in University Mathematics*, Hanovre, Germany, October 2022. Reinhard Hochmuth.
- [3] Jérémy Blanc, J.P. Giacometti, André Hirschowitz, and Loïc Pottier. Proofs for freshmen with Coqweb. In *Proceedings PATE07*, pages 93–107, 2007.
- [4] Jasmin Blanchette, Jeremy Avigad, Julien Narboux, Heather Macbeth, Gihan Marasingha, and Patrick Massot. Panel: Teaching with proof assistants. *Lean Together*, 2021.
- [5] N.G. Bruijn, de. *Automath : a language for mathematics*. EUT report. WSK, Dept. of Mathematics and Computing Science. Technische Hogeschool Eindhoven, 1968.
- [6] Pedro Cobo, Josep Fortuny, Eloi Puertas, and Philippe Richard. AgentGeom: A multiagent system for pedagogical support in geometric proof problems. *International Journal of Computers for Mathematical Learning*, 12:57–79, Apr 2007.
- [7] Thierry Coquand. *Une Théorie des Constructions*. PhD thesis, Université Paris 7, January 1985.
- [8] Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean Theorem Prover (System Description). In Amy P. Felty and Aart Middeldorp, editors, *Automated Deduction - CADE-25*, pages 378–388, Cham, 2015. Springer International Publishing.
- [9] David Delahaye, Mathieu Jaume, and Virgile Prevosto. Coq, un outil pour l’enseignement. Une expérience avec les étudiants du DESS Développement de logiciels srs. *Technique et Science Informatiques*, 24:1139–1160, 11 2005.
- [10] Jacques Gressier. Géométrie. <http://revue.sesamath.net/spip.php?article42>, 2006. <http://geometrix.free.fr/site/>.
- [11] Gila Hanna and Xiaoheng Yan. Opening a discussion on teaching proof with automated theorem provers. *For the Learning of Mathematics*, Nov 2021.
- [12] Marie Kerjean, Frédéric Leroux, Patrick Massot, Michaela Mayero, Zoé Mesnil, Simon Modeste, Julien Narboux, and Pierre Rousselin. Utilisation des assistants de preuves pour l’enseignement en L1 - Retours d’expériences. *Gazette SMF*, August 2022.
- [13] Nicolas Leduc. *QED-Tutrix : système tutoriel intelligent pour l’accompagnement des élèves en situation de résolution de problèmes de démonstration en géométrie plane*. PhD thesis, École Polytechnique de Montréal, Dec 2016.
- [14] Roman Matuszewski and Piotr Rudnicki. MIZAR: the first 30 years. *Mechanized Mathematics and Its Applications*, 4:3–24, Mar 2005.
- [15] Robert C. Moore. Making the Transition to Formal Proof. *Educational Studies in Mathematics*, 27(3):249–266, 1994.
- [16] Tobias Nipkow. Teaching Semantics with a Proof Assistant: No More LSD Trip Proofs. pages 24–38, Jan 2012.
- [17] Lawrence C Paulson. The foundation of a generic theorem prover. Technical Report UCAM-CL-TR-130, University of Cambridge, Computer Laboratory, March 1988.
- [18] Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hrițcu, Wilhelm Sjöberg, and Brent Yorgey. *Software Foundations*. Electronic textbook, 2017. Version 5.0.
- [19] C. Raffalli and R. David. Computer assisted teaching in mathematics. *Workshop on 35 years of Automath*, 2002.
- [20] Krzysztof Retel and Anna Zalewska. Mizar as a Tool for Teaching Mathematics. *Mechanized Mathematics and Its Applications 4 (1)*, page 35–42, Jan 2005.
- [21] P.R. Richard and J.M. Fortuny. Amélioration des compétences argumentatives à l’aide d’un système tutoriel en classe de mathématique au secondaire. *Annales de didactique et de sciences cognitives ISSN 0987-7576 (imprimé) - Revue internationale de didactique des mathématiques*, 12:83–216, January 2007.
- [22] Benoit Rognier and Guillaume Duhamel. Présentation de la plateforme Edukera. In Julien Signoles, editor, *Vingt-septièmes Journées Francophones des Langages Applicatifs (JFLA 2016)*, Saint-Malo, France, January 2016.
- [23] Athina Thoma and Paola Iannone. Learning about Proof with the Theorem Prover LEAN: the Abundant Numbers Task. *International Journal of Research in Undergraduate Mathematics Education*, 8, April 2022.
- [24] C. Webber, L. Bergia, S. Pesty, and N. Balacheff. The Baghera project: a multi-agent architecture for human learning. *Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments*, pages 12–17, 2001. Laboratoire Leibniz UJF Grenoble, AIED2001, San Antonio, TX, USA.
- [25] Keith Weber. Student difficulty in constructing proofs: The need for strategic knowledge. *Educational Studies in Mathematics*, 48(1):101–119, Oct 2001.
- [26] Jelle Wemmenhove, Thijs Beurskens, Sean McCarren, Jan Moraal, David Tuin, and Jim Portegies. Waterproof: educational software for learning how to write mathematical proofs, 2022.
- [27] Makarius Wenzel, Lawrence C. Paulson, and Tobias Nipkow. The Isabelle Framework. In Ait Mohamed, Munoz, and Tahar, editors, *Theorem Proving in Higher Order Logics (TPHOLs 2008)*, volume 5170 of LNCS, pages 33–38. Springer, 2008.
- [28] Freek Wiedijk. *The Seventeen Provers of the World*. Springer, Berlin, Heidelberg, 2006.

8 Appendix: results of the questionnaire

