OnlineProver: First Experience with Teaching Formal Proofs*

Joachim Tilsted Kristensen[1][0000−0002−1619−5944], Ján Perháč[2][0000−0001−6347−2409], Lars Tveito[1], Lars-Bo Husted Vadgaard[1][0009−0007−9056−7260], Michael Kirkedal Thomsen[1,3][0000−0003−0922−3609], Oleks Shturmov[1,3][0000−0001−5963−8509], Samuel Novotný[2][0009−0005−1675−2965], and Sergej Chodarev[2][0000−0002−9293−0859]

[1] Department of Informatics, University of Oslo, Oslo, Norway
{joachkr,larstvei,larsvad,michakt,olekss}@ifi.uio.no
[2] Department of Computers and Informatics, Faculty o Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovakia
{jan.perhac,samuel.novotny,sergej.chodarev}@tuke.sk
[3] Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

Abstract. OnlineProver is an interactive proof checker, tailored for the educational setting. The main features are a user-friendly interface for editing and checking proofs, and a DSL for specifying deduction systems and exercises about them. The user interface provides feedback directly in the derivation, based on error messages provided by a proof checking web-service. A basic philosophy of the tool, is that it should aid the student, while still maintain that the students construct the proofs as if it was on paper. We gathered feedback on the tool through a questionnaire, and we provide an intervention of its effectiveness for students in a class setting, along with technical aspects. The first intervention showed that the students were satisfied with using OnlineProver as part of the teaching and gave a first conformation of the learning approach behind. This gives clear directions for future developments with the potential to find and evaluate OnlineProver can improve teaching of natural deduction.

1 Introduction

When teaching theoretic computer science courses (e.g. introduction to first order logic), the focus is often not on programming; instead we try to teach an understanding of discrete abstractions and their properties as proofs. For this reason, spending several weeks of the course on introducing a theorem prover

* The authors thanks EEA and Norway Grants for support of this work under initiative no. FBR-PDI-025.
"*Working together for a green, competitive and inclusive Europe.*"
https://www.eeagrants.sk/

can be counter productive, as it will divert the attention of the students. In our project[1], we propose a tool, OnlineProver[2], for providing exercises for the Gentzen style derivation proofs over user definable deduction systems.

The focus is to build an educational tool that incorporates some of the capabilities of computer aided theorem proving, without introducing the students to a new syntax, or spending several lectures on introducing tools. Our initial setup, is a tool for which the teacher must learn a simple DSL for specifying derivation systems and exercises. However, the student is limited to solving exercises, handed out by the teacher, through a web interface designed for the purpose. The interface is capable of checking students' derivations, and bringing feedback directly in the derivation where it is needed.

Computer science students typically follow several programming courses before delving into theoretical computer science, during which they have to gain extensive experience for a trial-and-error approach [20]. On the other side, theoretical university course all necessitate mathematical prerequisites, which computer science students frequently struggle with. First research [14] and personal experiences [15] suggest that the pedagogical approaches effective for teaching CS students differ from those tailored to mathematics students; analogous findings can be seen within physics [17].

Building upon this foundation, we have with OnlineProver built a tool that aid CS students by facilitating the exercise of mathematical rigour, while providing a taste of trial-and-error exploration. Students must specify all derivations and type all rule constructs of the proofs; there is no interactive rule selection. We want the tool to closely resemble the class room teaching with pen and paper. The learning objectives that OnlineProver could support in a course will thus be:

– Skills in deciding and proving formal properties of logical formulas (e.g. satisfiability, validity, implication and equivalence) by natural deduction arguments.

Following the tradition in CS education [5], we therefore aim to foster reflection through meticulous tool design, rather than solely promoting problem-solving through trial-and-error. Drawing from experiences in introductory programming, we acknowledge that students appreciate the simplicity of block-based frameworks (e.g., Scratch), yet may find them lacking in authenticity [20]. Research on tool for automated assessment (auto-graders) [2, 4, 10], often used for giving quick and direct test feedback on programming tasks, have also shown that this can remove the students focus from problem solving. Instead students tends to optimise towards making specific tests pass.

As an initial story, we have introduced the tool in a exercise class at the Technical University of Košice, Slovakia in the master-level course *Logic for informaticians*. The following is a description of our first experience use of mechanical reasoning systems for teaching. First, in Section 3 we outline the design

[1] https://onlineprover.github.io/
[2] http://onlineprover.com/

of OnlineProver, after which Section 4 will exemplify the experience of the students. In Section 5 we present the intervention and the results. These will be discussed in Section 6 and finally in Section 7 we conclude. The online version of OnlineProver tool along with all exercises that is used in this experiment, can be found at http://onlineprover.com/.

2 Related work

Although "traditional" proof assistants such as Coq, Agda, and Isabelle use programming style, we do not find them well-suited for introductory courses. It takes a significant amount of time to learn the underlying system's language(s), to be able to start working on formal proofs.

Several other tools have been developed with a focus on a domain-specific framework and implements trial-and-error approach that gives students less room for reflection. Such tools are dedicated to various frameworks e.g. proof trees for Hoare logic [11], natural deduction [3, 13, 16], or sequent calculus [6] in the Gentzen style [8], Tableau [18, 19], and λ-calculus [7]. However, these are designed to be limiting for the students and can exhibit the same problems as block-based programming frameworks. Closer to our approach is the work by [12], that in a more general and flexible framework implements support for Fitch style proofs.

The authors we not able to find research where these or similar tool have been evaluated with respect to improvements in learning outcome of the students.

3 Tool Design

OnlineProver consist of a pair of domain-specific programming languages for specifying formal languages and exercises, and a web-editor for interactively solving the exercises guided by the a language and exercise program.

The teacher provides these programs, and generally, several exercise programs can be provided about a single language program. The language program describes the syntax and semantics in sections of an assignment in a markdown-flavoured DSL as exemplified below.

```
# syntax **Syntax for terms and values**

t : Term
  | ‘(‘ t0 ‘,‘ t1 ‘)‘
  | ...

v : Value
  | ...

# judgement **Evaluation** [ gamma ‘|-‘ t ‘->‘ v ]

gamma |- t1 ~> v1
```

```
gamma |- t2 ~> v2
---------------------------------------(pair-rule)
gamma |- (t1, t2) ~> (v1, v2)
```

Together with an exercise description program, this compiles into a text/tree format that can be manipulated and checked through the web-interface.

The formal-language description language acts as a lightweight Lex/Yacc for specifying small toy programming languages. In the tool the teacher and the students will then be able to focus on those toy languages without dealing with the overhead of figuring out how to use a fully fledged theorem prover like Coq or Agda.

Furthermore, because the proof checker is designed specifically for teaching, the proof checker can potentially know what the student is trying to show. For instance, error messages can be guided by suggested solutions as part of the teacher's program.

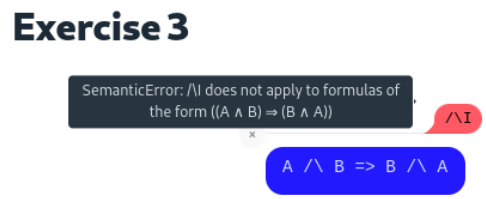
4 Teaching with OnlineProver: Initial Exercise Set on Natural Deduction

This section will exemplify the experience with OnlineProver from a student perspective.

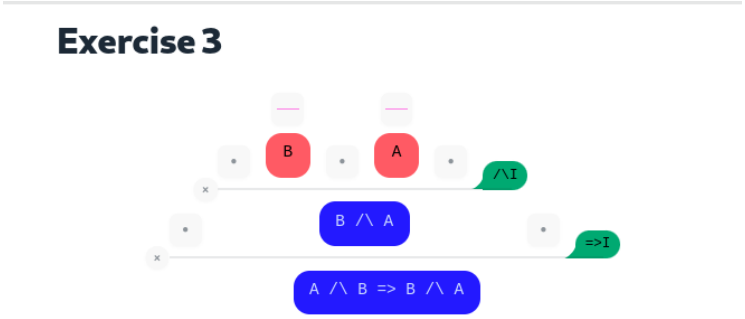
We have developed for this experiment an initial version of the OnlineProver system for simple Gentzen style derivations. The choice of Gentzen style derivations is somewhat arbitrary and reasoned by the use on the target courses and related material.

In this version, an exercise is a list of incomplete derivations, that the student must complete in an empty context. For our first experiment, we implemented exercises about natural deduction in the style found in *The Logic Manual* by Volker Halbach [9]. In the graphical user interface an exercise, in this logic, is a formula, on top of which one must provide the corresponding derivation. For example here is Exercise 3.

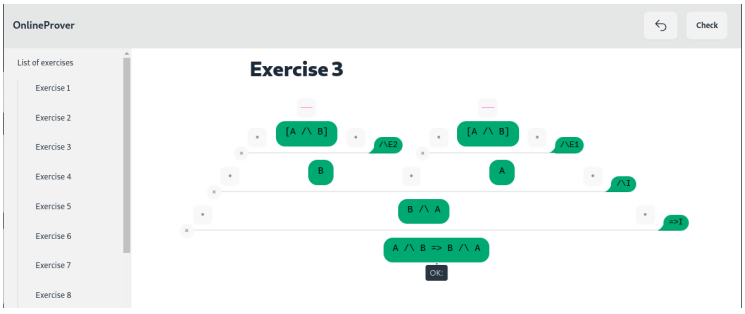
The button with a line symbol, adds a line on top of the formula, and a corresponding text box for inputting the rule name. If the rule applies, then the rule box becomes green, and otherwise it becomes red, and a hover text provides the error message like so



Currently, nodes are also coloured red when the proof is not closed. In this example, the student may for instance (correctly), input premises `B` and `A` like so



Finally, the proof is closed when all parts of the formula are green, and the student can select a new exercise from a pane in the left hand side of the webpage.



The first version of OnlineProver supports only derivation style proofs, but we plan to support proof by induction, where the student must provide a derivation for each rule in a deduction system, in order to show that an equation holds.

5 Intervention of First Teaching Experiments

As an initial story, the formal intervention of OnlineProver is based on one semester - Spring 2024, in the course *Logic for informaticians* with 59 students [3], at the Technical University of Košice.

Since the questionnaire collected open questions, we have categorized and quantified all the answers by the method explained in Section 5.3. In total 30 students filled out the questionnaire. We were inspired by the rubric [1], therefore our goal was to collect feedback on functionality, technical details, teaching, and cognitive presence.

5.1 Experiment

During the 3rd week of the semester, the students completed 7 exercises dedicated to proving formulæ of propositional logic in natural deduction using the traditional pen and paper approach. Additionally, homework consisted of 5 formulæ, and finally students were required to submit an assignment with proof of a complex formula.

In the 9th week of the semester, students were introduced to the OnlineProver tool, with the implemented natural deduction in propositional logic utilizing the same syntax as in the pen and paper exercises. We have implemented 16 exercises, including 7 that were already solved during a previous class by pen and paper.

During the exercise, we presented the students with a short demonstration of the tool's functionalities and a sample proof of a formula (exercise 3). We have emphasized the tool's feedback in case of incorrect approach, use of a wrong rule, number of premises, etc.

We sent students an email with instructions and a short manual for using the tool, and asked them to solve the presented exercises. After that, they were asked to fill out a questionnaire.

5.2 Research questions

Based on the mentioned rubric [1], we have formulated the following research questions:

RQ1: Did our tool help students understand the principles of natural deduction?
RQ2: Is the tool's feedback helpful?
RQ3: How do students evaluate an interactive tool in terms of learning approach, compared to other tools presented to them during the course?
RQ4: How do students overall evaluate the tool, what are the pros and cons, additional comments on the tool or user interface?

[3] additional 4 master students were asked to fill out the questionnaire, all passed the subject last year

Based on the research questions, we have formulated 8 questions for the questionnaire that is included in Appendix A. The first three questions answer the RQ1-3. Then, we asked the fourth *control* question, to gain an overview of students' experience in the topic, as one of the main purposes of the tool is to help beginners. The questions 5-8 are dedicated to the RQ4.

5.3 Results

As mentioned in the previous section, 30 students filled out the questionnaire. According to the control question, we can conclude that all of the students are beginners in the area of proofs. 28 students (93.3%) declared that this course was their first touch with formal proofs. Another 2 students (6.7%) mentioned little experience from the previous university courses.

To quantify the results, we use the following method: Each answer is assigned 1 point if it fully agrees with the research question (RQ), 0.5 points if it partially agrees, and 0 points if it disagrees. Since not every question was answered by each student, we standardized the scores into a 100-point grade scale. At the end, we aggregate the gained points, and then convert the ratio of points to the number of answers into a percentage, resulting in a score on a scale from 0 to 100.

Research Question 1 *RQ1: Did our tool help students understand the principles of natural deduction?*

30 students answered the question. Quantified results according to the method introduced in Section 5.3 are summarized in Table 1. We gain an overall score of 83.33.

Agreement	No. of students
0	2
0.5	6
1	22

Table 1. Agreement with the RQ1.

Through RQ1, we aimed to get feedback on the overall understanding of the given topic. Even though our tool is still a pre-alpha version, based on the positive feedback we can conclude that it fulfils its main goal as a teaching tool. 73% of students reported that the tool helped them to better understand the given topic, while an additional 20% claimed that it helped them partially (e.g. applying deduction rules, checking the correctness of their solution). 7% students reported that it did not help them in any way.

Research Question 2 *RQ2: Is the tool's feedback helpful?*

The feedback is one of the main features, that we are focusing on. Because our students are studying the computer science field, we want to develop a teaching tool that simulates pen and paper proof but with additional feedback. Our students are not very experienced with proving mathematical theorems but were exposed to several programming classes. We hypothesize that such an approach following the programming trial and error style will be beneficial.

30 students answered the question. Quantified results according to the method introduced in Section 5.3 are summarized in Table 2. We gain an overall score of 85.

Agreement	No. of students
0	1
0.5	7
1	22

Table 2. Agreement with the RQ2.

Based on the results, we can conclude that the feedback feature follows our hypothesis. 73.3% of students mentioned that it helped them to solve the exercise, while 7.3% of students positively commented on it with suggestions for improvement. Only 1 student (3.3%) mentioned that they found the feedback to be confusing and it did not help at all.

Research Question 3 *RQ3: How do students evaluate an interactive tool in terms of learning approach, compared to other tools presented to them during the course?*

During the course, students were presented and tested several teaching tools, that visualize different formal methods[4] [5]. Compared to OnlineProver, these tools function similarly to a calculator, with very limited interaction features. The main difference is, that OnlineProver does not compute a solution or apply a deduction rule automatically. Therefore, the RQ3 aims to collect the students' opinions on OnlineProver in terms of the learning approach.

28 students answered the question. Quantified results according to the method introduced in Section 5.3 are summarized in Table 3. We gain an overall score of 82.1.

Based on the gained score, we can conclude that students appreciate the learning approach used by OnlineProver. Most of the answers directly commented that in terms of learning, this approach is better than using "tools that do everything for the user". The simulation of pen and paper proofs together with the provided feedback was especially appreciated. On the other hand, several

[4] Such as https://malstan.github.io/interactive-proof-system/

[5] https://benjenbekiaris.github.io/FolSolverApp/

Agreement	No. of students
0	2
0.5	6
1	20

Table 3. Agreement with the RQ3.

answers mentioned situations, that if a student is stuck with an exercise and even provided feedback does not help, then OnlineProver would not help them any more. Overall 66.6% of students agreed that this approach is better than other tools presented to them, and another 20% claimed that they find this approach better, but they also miss the feature of automated solving. While 6.6% of the students claimed that they consider the approach used in other tools to be better.

Research question 4 *RQ4 How do students overall evaluate the tool, what are the pros and cons, additional comments on the tool or user interface?*

The last RQ4 is dedicated to overall satisfaction with our tool. We have asked 4 questions in the questionnaire Appendix A to answer it. Question no. 5 asks directly about overall satisfaction, therefore we use it to compute the score.

Questions no. 6-8 are dedicated to collecting feedback on what students find positive, or negative about the tool, and they can add any additional comments/suggestions. We will discuss answers to these questions in Section 6.

30 students answered the question. Quantified results according to the method introduced are summarized in Table 4. We gain an overall score of 88.33.

Agreement	No. of students
0	0
0.5	7
1	23

Table 4. Agreement with the RQ4.

The overall satisfaction with the tool is very high. No student replied, that they consider this tool to be unhelpful. While 23.3% of students added comments for a possible improvement, more than 76% of students were satisfied. Here, we would like to point out one positive answer: "*I realized that I didn't understand this topic well enough, but by solving a few examples I improved my knowledge in this topic.*".

6 Discussion

In this section, we discuss open questions 6-8, which are aimed at gathering feedback from students regarding what they perceive as both positive and negative aspects of the tool. Additionally, students have the opportunity to include any additional comments or suggestions they may have.

6.1 Positive aspects

The answers contained positive comments on the tool are summarized in Table 5.

Comment	No. of answers
Feedback	14
Possibility to check solution correctness	8
UI, easy to use	5
Visualization of proof trees	4
Faster/more convenient than paper proof	4
Paperless approach	7

Table 5. Positive aspects of the tool. Note; the number of answers is higher than the number of respondents because some answers commented more than 1 thing.

As we can see in Table 5 above, the most often commented aspect is the feedback, and the possibility to check the correctness of the solution at any time. We also gained positive feedback on the UI and visualization of proof trees, and the students also appreciated the possibility of doing proofs without paper, while some of them consider this approach to be even faster than doing paper proof.

6.2 Negative aspects

The answers contained negative comments on the tool are summarized in Table 6.

The most commented aspect is a method of inputting special symbols into proof trees. Currently, we are using ASCII notation e.g. / \for conjunction, etc. We agree, that it is not a very convenient way. In the future versions of OnlineProver, we plan to implement more possibilities of inputting special symbols, such as using LaTeX notation (\wedge), or natural language (and), etc.

The second relevant issue is the missing list of assumptions that are created by an application of specific rules of natural deduction. During class, we were writing them on the side of a proof tree, therefore it is understandable that the students are missing the list. We also plan to implement this in the future iteration of OnlineProver.

Comment	No. of answers
Method of inputting special symbols	10
Missing list of assumptions	6
It is faster to do paper proofs	8
Complicated deleting of nodes	2
No possibility of splitting a proof tree	1
No possibility to enter own example	1

Table 6. Negative aspects of the tool. Note; The number of answers is lower than the number of respondents because some answers commented that they found nothing negative about the tool.

Some students also claim that it is faster to do proofs on paper, while in the previous question, some students mentioned that they consider writing proofs to be faster in OnlineProver. The solution to this problem will be the implementation of more convenient ways to input special symbols (as mentioned in the previous paragraph).

Another negative comment is about a way of deleting nodes in proof trees. We also plan to improve this in the future iterations of OnlineProver.

One comment is dedicated to missing the possibility of creating subtrees. A student in their comment correctly anticipated a possible inconvenience, if the tree gets too big to fit into the screen. But in such a case, we already have a way of splitting a tree into subtrees within the teacher's DSL.

The last comment mentioned missing the possibility to enter the user's own example. We do not plan to implement this functionality, as currently the tool is developed in a way that a teacher specifies exercises.

6.3 Additional comments on the tool

The last question was dedicated to the additional comments on the tool. Here the answers mostly repeated from previous questions. 4 students mentioned the missing list assumptions, and 2 students commented on a method of inputting special symbols.

To conclude, 8 students mentioned missing documentation or that they would appreciate having a list of deduction rules somewhere on the side of the proof tree. We also plan to implement this in the future iteration of OnlineProver.

7 Conclusion and Future Work

This paper presents OnlineProver, our teaching tool designed for educational use. We have briefly discussed its design and main ideas behind implementation, and how it differs from other teaching tools. We presented a brief, motivating example of teaching with OnlineProver, demonstrating a solution to an exercise.

The main asset of this paper is its intervention, based on the rubrics of [1], of the feedback collected from the first experimental use of OnlineProver in a class.

Based on this intervention, we conclude the following. The students consider this tool to be beneficial from learning aspect, and teaching aspect, as well. We have gained very positive feedback, not only education aspects, but on technical solution too. It has a potential to became useful teaching aid for computer science students. But also there is a space for improvement. It is worth mentioning that during testing by students, not a single bug was found. It showed that the students' perception is mostly positive regarding to the learning approach by OnlineProver; namely to model the environment from the traditional pen and paper proofs.

As an initial story, on the future, we plan to use OnlineProver in courses dedicated to theoretical computer science at the Technical University of Košice, Slovakia and the University of Oslo, Norway. We are currently also conducting an experiment with the IT University of Copenhagen, Denmark, and are open to provide the facility to other university courses.

The authors are aware that this work is based on self-reporting and much further research is needed to understand if students actually benefit from using OnlineProver over pen-and-paper proofs or even other tools. At first, the authors were interested in understanding how students receive at such a tool, which then became the focus of this study at the Technical University of Košice. In a future study conducted at IT University of Copenhagen we want to better understand of how students actually work with OnlineProver. Pen-and-paper proofs have been integrated into teaching on these courses over many years, and we need to understand how receive and use OnlineProver to have a natural integration in teaching. In a future study, we would like to conduct a randomised study to investigate if we can see an improvement of students understanding of natural deduction using OnlineProver.

Currently, our tool only supports derivation proofs. The next step is to extend our tool by support of proof by induction. The more challenging plan is to use OnlineProver for conducting research in the area of data-driven teaching methods that use theorem provers to check students' proofs. This could lead to exploration of the potential use adaptive feedback and perhaps even generative AI to provide a formative feedback.

References

1. Anstey, L., Watson, G.: A rubric for evaluating e-learning tools in higher education. Educause Review **10**(09) (2018)
2. Baniassad, E., Zamprogno, L., Hall, B., Holmes, R.: Stop the (autograder) insanity: Regression penalties to deter autograder overreliance. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. pp. 1062–1068. SIGCSE '21, Association for Computing Machinery (2021). https://doi.org/10.1145/3408877.3432430
3. Broda, K., Ma, J., Sinnadurai, G., Summers, A.: Pandora: A reasoning toolbox using natural deduction style. Logic Journal of the IGPL **15**(4), 293–304 (2007). https://doi.org/10.1093/jigpal/jzm020

4. Deeb, F.A., Hickey, T.: Impact of reflection in auto-graders: an empirical study of novice coders. Computer Science Education **0**(0), 1–22 (2023). https://doi.org/10.1080/08993408.2023.2262877
5. Edwards, S.H.: Using software testing to move students from trial-and-error to reflection-in-action. In: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education. p. 26–30. SIGCSE '04, Association for Computing Machinery (2004). https://doi.org/10.1145/971300.971312
6. Ehle, A., Hundeshagen, N., Lange, M.: The sequent calculus trainer with automated reasoning - helping students to find proofs. In: Quaresma, P., Neuper, W. (eds.) Proceedings 6th International Workshop on Theorem proving components for Educational software, Gothenburg, Sweden, 6 Aug 2017. Electronic Proceedings in Theoretical Computer Science, vol. 267, pp. 19–37. Open Publishing Association (2018). https://doi.org/10.4204/EPTCS.267.2
7. Frank, M., Kreitz, C.: A theorem prover for scientific and educational purposes. In: Quaresma, P., Neuper, W. (eds.) Proceedings 6th International Workshop on Theorem proving components for Educational software, Gothenburg, Sweden, 6 Aug 2017. Electronic Proceedings in Theoretical Computer Science, vol. 267, pp. 59–69. Open Publishing Association (2018). https://doi.org/10.4204/EPTCS.267.4
8. Gasquet, O., Schwarzentruber, F., Strecker, M.: Panda: A proof assistant in natural deduction for all. a gentzen style proof assistant for undergraduate students. In: Blackburn, P., van Ditmarsch, H., Manzano, M., Soler-Toscano, F. (eds.) Tools for Teaching Logic. pp. 85–92. Springer Berlin Heidelberg (2011)
9. Halbach, V.: The Logic Manual. Oxford University Press (2010)
10. Karavirta, V., Korhonen, A., Malmi, L.: On the use of resubmissions in automatic assessment systems. Computer Science Education **16**(3), 229–240 (2006). https://doi.org/10.1080/08993400600912426
11. Korkut, J.: A proof tree builder for sequent calculus and hoare logic. In: Quaresma, P., Marcos, J., Neuper, W. (eds.) Proceedings 11th International Workshop on Theorem Proving Components for Educational Software, Haifa, Israel, 11 August 2022. Electronic Proceedings in Theoretical Computer Science, vol. 375, pp. 54–62. Open Publishing Association (2023). https://doi.org/10.4204/EPTCS.375.5
12. Leach-Krouse, G.: Carnap: An open framework for formal reasoning in the browser. In: Quaresma, P., Neuper, W. (eds.) Proceedings 6th International Workshop on Theorem proving components for Educational software, Gothenburg, Sweden, 6 Aug 2017. Electronic Proceedings in Theoretical Computer Science, vol. 267, pp. 70–88. Open Publishing Association (2018). https://doi.org/10.4204/EPTCS.267.5
13. Machín, B., Sierra, L.: Yoda: a simple tool for natural deduction. In: Proceedings of the Third International Congress on Tools for Teaching Logic (TICTTL'11). vol. 6680 (2011)
14. Morgan, C.: (in-)formal methods: The lost art. In: Liu, Z., Zhang, Z. (eds.) Engineering Trustworthy Software Systems. pp. 1–79. Springer International Publishing, Cham (2016)
15. Newsome Crossley, J.: What is mathematical logic? an australian odyssey. Logic Journal of the IGPL **31**(6), 1010–1022 (2023)
16. Seligman, J., Thompson, D.: Teaching natural deduction in the right order with natural deduction planner (2015)
17. Sherin, B.L.: A comparison of programming languages and algebraic notation as expressive languages for physics. International Journal of Computers for Mathematical Learning **6**(1), 1–61 (2001). https://doi.org/10.1023/A:1011434026437, https://doi.org/10.1023/A:1011434026437

18. del Vado Vírseda, R., Orna, E.P., Berbis, E., de León Guerrero, S.: A logic teaching tool based on tableaux for verification and debugging of algorithms. In: Blackburn, P., van Ditmarsch, H., Manzano, M., Soler-Toscano, F. (eds.) Tools for Teaching Logic. pp. 239–248. Springer Berlin Heidelberg (2011)
19. Vasconcelos, D.R.: Anita: Analytic tableau proof assistant. In: Quaresma, P., Marcos, J., Neuper, W. (eds.) Proceedings 11th International Workshop on Theorem Proving Components for Educational Software, Haifa, Israel, 11 August 2022. Electronic Proceedings in Theoretical Computer Science, vol. 375, pp. 38–53. Open Publishing Association (2023). https://doi.org/10.4204/EPTCS.375.4
20. Weintrop, D., Wilensky, U.: To block or not to block, that is the question: students' perceptions of blocks-based programming. In: Proceedings of the 14th International Conference on Interaction Design and Children. p. 199–208. IDC '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2771839.2771860

A Questionnaire

The following is the questionnaire that was given to the students following the use.

1. Did our tool help you better understand the principles of natural deduction?
2. How do you evaluate the feedback provided by the tool?
3. How do you compare the OnlineProver and other tools presented in the course, considering the learning approach?
4. How much practice do you have with making proof exercises on paper?
5. What is your overall impression after making proof exercises in OnlineProver?
6. What do you find positive when doing proofs in OnlineProver compared to paper proofs?
7. What do you find negative when doing proofs in OnlineProver compared to paper proofs?
8. Any other comments on a tool, UI, etc.?