# Automated Scaffolding and Feedback for Proof Construction: A Case Study

**Korbinian Staudacher, Sebastian Maderand François Bry**
**Institute for Informatics, Ludwig Maximilian University of Munich, Germany**
korbinian.staudacher@gmx.de
sebastian.mader@ifi.lmu.de
bry@lmu.de
DOI: 10.34190/EEL.19.114

**Abstract:** Beginners are often unaccustomed to the abstract and formal thinking required in tertiary computer science education. This can be alleviated through close support provided by an experienced person, like a teacher. Nowadays, such support is hardly possible because, in computer science, instructors face classes of a few ten to a few hundred students. This article shows how scaffolding and feedback needed by computer science beginners can be provided by software. The particularly difficult case of building logical proofs is considered so as to demonstrate the effectiveness of the approach. The approach to computer-provided scaffolding and feedback presented in this article is based on specifically designed proof editors that relieve learners from some choices and tasks – scaffolding – and provide immediate feedback on those tasks left to the learners. Building logical proofs is especially challenging – not only for beginners – because it includes various challenges: Understanding definitions, building a syntactically correct proof, and correctly applying complex proof building rules. This article introduces two specialized editors that support students in learning the proof methods Resolution and Natural Deduction. The editors make it possible for learners to focus on the correct order of rule application – the most challenging part of proof building –, by relieving them of the other aspects of proof building, and providing immediate feedback on the correctness of proof construction tasks performed by the learners. The contributions of this article are twofold: First, the conception and implementation of two original graphical proof editors, and second, a report on an evaluation of the editors and the feedback and scaffolding they provide pointing to the educational approach's effectiveness.

**Keywords**: scaffolding, interactive learning environment, problem-solving learning environments, STEM education

## 1. Introduction

Coming from secondary education, beginners are often unaccustomed to the abstract and formal thinking required in tertiary math education (Gueduet, 2008) which is a part of computer science education. This is traditionally best alleviated through close support provided by an experienced person, like a teacher, but is nowadays hardly possibly with teachers facing classes of a few ten to a few hundred students.

Computer-provided feedback and scaffolding both lessens the workload of instructors (allowing them to focus on students that require their personal help) and empowers learners to engage with the subject matter on their own terms independently from lecturers or fixed hours. One way to implement computer-provided feedback and scaffolding are specialized editors that support users while they learn a particular topic. Such editors relieve learners from some choices and tasks – scaffolding – and provide immediate feedback on those tasks left to the learners. Both scaffolding and immediate feedback can already be found in many computer-based tools, such as in linters for programming languages which provide users with error messages and propose solutions which can be seen as a form of scaffolding. A tool for building automatons, JFLAP (http://www.jflap.org/) does not allow syntactically incorrect automatons to be built at all. Another area are logical proofs, the building of which is especially challenging – not only for beginners – because of various challenges: Understanding definitions, building a syntactically correct proof, and correctly applying complex proof building rules.

This article introduces two specialized editors that support students in learning the proof methods *Resolution* and *Natural Deduction*. The editors make it possible for learners to focus on the correct order of rule application – the most challenging part of proof building –, by relieving them of the other aspects of proof building, and providing immediate feedback on the correctness of proof construction tasks performed by the learners, especially on the proof's correctness.

The contributions of this article are twofold: First, the conception and implementation of two original graphical proof editors; second, a report on an evaluation of the editors and the feedback and scaffolding they provide pointing to the educational approach's effectiveness.

This article is structured as follows: This section is the introduction. Section 2 introduces related work. In Section 3 both editors and the ways in which they provide feedback and scaffolding are described. Section 4 reports on the evaluation of the editors and Section 5 summarizes the article and gives perspectives for future work.

## 2. Related work

The editors described in this article relate to feedback, scaffolding, and the use of interactive editors in learning and teaching.

### 2.1 Feedback

Feedback has been subject of much research regarding its effectiveness in various situations especially in the context of education. Feedback can be defined as "actions taken by (an) external agent(s) to provide information regarding some aspect(s) of one's task performance" (Kluger and DeNisi, 1996, p. 255). Thus, feedback can be seen as a "consequence of performance" (Hattie and Timperley, 2007, p. 81). If students are completely unfamiliar with the subject they learn, feedback has no effect, but otherwise it can help students to recognize mistakes and enhance their learning behavior (Hattie and Timperley, 2007). In a study analyzing more than hundred factors influencing students' achievement, feedback ranked among the top ten factors (Hattie and Timperley, 2007).

Hattie discriminates between four levels of feedback (Hattie and Timperley, 2007, p. 90):

- *Task-level Feedback* gives information about the worked on task.
- *Process-level Feedback* gives information on how to solve tasks related to the worked on task.
- *Self-regulation Feedback* aims to improve the learner's abilities regarding self-monitoring, self-evaluation, and directing actions.
- *Self-level Feedback* is a task-unrelated evaluation of the student.

According to Hattie, the most effective levels of feedback are process and self-regulation, as the former improves understanding of the underlying task scheme rather than that of a single task, while the latter guides students towards independent learning. Feedback at task level is most effective when combined with feedback at process level. When feedback should be given depends on the level: Feedback at task level is more effective when errors are corrected immediately, as it can lead to faster acquisition rates, whereas feedback at process level is best given delayed, as "immediate error correction [...] can detract from the learning of automaticity" (Hattie and Timperley, 2007, p. 98). The effectiveness of self-regulated feedback is further backed by Orsmond and Merry (2013) who suggest that the ability of self-regulated learning corresponds to being a "high achieving" student (Orsmond and Merry, 2013). Furthermore, Orsmond and Merry (2013) suggest that self-regulation is essential to process feedback at all.

Nicol and Macfarlane-Dick (2006) propose principles that good feedback should adhere to as well as various ways to meet this requirements. According to them, feedback should "help clarify what good performance is" (Nicol and Macfarlane-Dick, 2006, p. 205). This can be met by the provision of exemplary solutions. Another principle is that feedback should "encourage motivation and self esteem" (Nicol and Macfarlane-Dick, 2006, p. 205) which can be met by many "easy-to-solve" tasks instead of a single difficult task, as well as providing feedback comments instead of scores or grades.

### 2.2 Scaffolding and fading

Scaffolding is a teaching method in which students are supported while working on tasks the solving of which would be otherwise beyond their abilities. Originally used in context of Vygotsky's Zone of Proximal Development to describe the potential development of a child under guidance of a more capable person, more recent research has transferred the theory to general education with scaffolding now being used for guiding and helping learners during their learning process (Gibbons, 2002). The support given throughout the learning process should be gradually removed ("faded") until it is no longer necessary (Jackson, 1996). While scaffolding, as provided by human tutors, has been well-established as an effective means of supportive learning (Jackson, 1996, p. 1), Azevedo and Hadwin (2005) conclude that scaffolding must not necessarily be provided by a human but can be provided by technology as well. In studies, computer-provided scaffolding has been shown to be effective in "moving students towards more sophisticated models" as well as increasing the students ability to self-regulate (Azevedo and Hadwin, 2005, p. 371). A study by Sao, Gobert and Baker (2014) examining the effects of computer-provided feedback observed that students who received automated scaffolding by software during various exercises performed significantly better afterwards than students who did not.

According to Van Merriënboer, Kirschner and Kester (2003), the aim of scaffolding is to reduce cognitive load during learning which allows learners to focus on fewer important concepts at a time. Scaffolding can either reduce *intrinsic* cognitive load, which can be done by gradually moving from simple to more complex tasks, or *extraneous* cognitive load, which could, e.g., be done by beginning with worked-out examples, to filling out gaps, and finally moving towards conventional tasks (Van Merriënboer, Kirschner and Kester, 2003). Providing whole tasks instead of parts of tasks supports students in gaining an overview of the structure and relations between the various parts of a task (Van Merriënboer, Kirschner and Kester, 2003). For optimal scaffolding, the correct information should be given at the correct time: *Supportive information*, such as cognitive strategies and mental models should be presented before the task is worked on, while *procedural information*, such as parts that stay the same across similar tasks should be explained while the task is worked on (Van Merriënboer, Kirschner and Kester, 2003). However, tasks can be also be simplified up the point that students do not have to mind certain aspects of the tasks at all: A concept closely related to scaffolding and fading is “Didactic Reduction”, a term coined by Grüner (1967), and describes breaking down a concept to its most basic parts, explaining those, and adding more advanced concepts step by step afterwards to the aim of reducing cognitive load on learners. Grüner differentiates between vertical and horizontal didactic reduction: While horizontal reduction aims to simplify tasks by helping students with examples or similar, vertical reduction simplifies the task by omitting some details of the task (Grüner, 1967). In the process of fading, these details can be reintroduced later during the learning process.

### 2.3 Interactive editors in STEM

In the past decades, various interactive editors for learning purposes were designed. Generally, these editors aim to assist students in performing exercises for a certain subject or in a certain task category. Research has shown that students usually perform better solving tasks with the help of such editors than without: A study examining the interactive geometry editor GeoGebra (https://www.geogebra.org) has shown that the usage of this editor has a significant positive influence on student learning (Zengin, Furkan and Kutluca, 2012). eChem, an interactive editor for visualizing chemical representations of molecular models has shown positive results on students’ performance as well (Wu, Krajcik and Soloway, 2001). Regarding the field of logical reasoning, there already exist a variety of interactive tools and editors. P-Logic Tutor (Lukins, Levicki and Burg, 2002) supports students in understanding the concepts of propositional logic and theorem proving. While the effect on students’ performance has not been measured for this editor, a study examining ITA (Yacef, 2005), a set of editors similar to P-Logic Tutor, has shown that using the editors has a strong impact on students’ performance. ILTIS (Geck et al., 2018) is another recently developed system of logic editors including an editor for resolution in propositional logic, where authors concluded that the feedback provided by their system supports students in understanding the subject matter as well. However, although there are already various editors for logical reasoning, the authors are not aware of any editor for Resolution for first order logic. Furthermore, already existing editors for Natural Deduction, such as Alfa (Hallgren and Ranta, 2000), require a certain familiarity with the topic and can therefore considered less suitable to help beginners. The editors in this paper were designed for both propositional and first order logic with the specific aim to help beginners getting familiar in the field of logical proofs using Resolution and Natural Deduction.

## 3. Logic editors

This section introduces the editors for the proof techniques Resolution and Natural Deduction used for the evaluation described in Section 4. A more detailed explanation of the proof techniques and editors can be found in (Staudacher, 2018).

### 3.1 Resolution

Resolution is a proof technique developed by Robinson with which the unsatisfiability of formulas can be proven in both propositional and first order logic (Robinson, 1965). The editor can be seen in Figure 1: The left side shows the clauses of the formula of which the unsatisfiability is to be shown, and the right side shows the current proof tree. At each step (i.e., one line on the right side), the student chooses clauses from the right which contain complementary literals (M and ¬M in Figure 1), selects them in the tree, which are then eliminated in the subsequent step (line 2 on the right side of Figure 1). The steps of adding a clause from the left and eliminating complementary literals continue until the resulting clause is the empty set what proves unsatisfiability. For proofs in first order logic, the editor supports variable substitutions (in form of a most general unifier) and factorization rules.

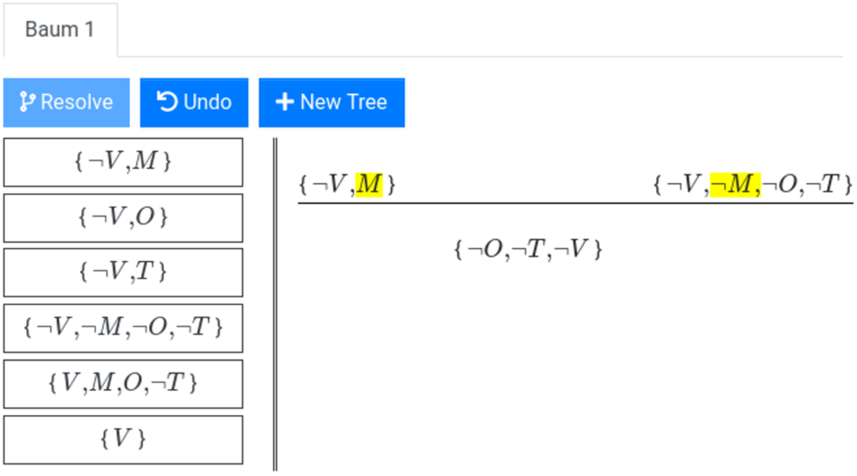


**Figure 1**: Editor for resolution by example of an exercise about propositional logic

The editor provides users with immediate feedback on the correctness of a step, as incorrect steps are met with an error message and are not performed on the proof tree. Error messages serve both as feedback at task level and at process level, as students not only learn why their current step was incorrect, but also about the general effect of a resolution step. While working on a proof, users can focus on the correct application of the rules and do not have to think about the creation of a syntactically correct proof tree, which eliminates one potential source of errors and acts as didactic reduction. As of scaffolding, the editor provides the aforementioned feedback during the creation of the proof and can as well be pre-filled with a partly constructed proof tree (c.f., Section 2.2).

## 3.2 Natural Deduction

Natural Deduction is a proof technique developed by Gentzen (1935) that allows to prove the satisfiability of a formula in both propositional and first order logic. The editor for Natural Deduction can be seen in Figure 2 and is split into three parts: The left side shows the rules of Natural Deduction, the right side shows assumptions (which are required for or created by the use of some rules), and the middle shows the actual proof tree. Users select one or more formulas in the proof tree and select a rule, which will subsequently be applied to the selected formulas, resulting in a new level of the proof tree.

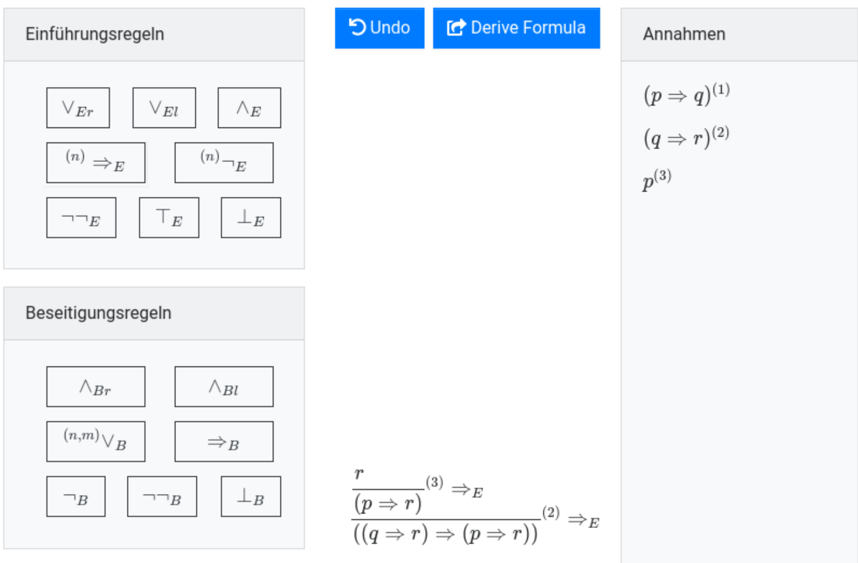


**Figure 2:** Editor for Natural Deduction by example of an exercise about propositional logic

Proofs using Natural Deduction can be either built beginning with the formula to be proven or with the assumptions. Depending on the actual exercise, one way or the other or a mix of both makes sense. Therefore, the editor generally works bottom-up, but allows users to derive parts of the proof in a separate window top-down and add those parts of the proof afterwards to the main tree.

The editor for Natural Deduction provides feedback on the overall correctness of exercises as well, which is especially important for beginners since it is often not obvious when a proof is correctly finished. Feedback is provided when applying rules as well: If a rule was applied incorrectly, users are provided with an error message describing what error has been made. If an applicable rule was selected, users can immediately check if the application of the rule matches their expected result and thereby gain better understanding on the application of the rules. Feedback on rule application works both on task and process level, as the rules are independent from the exercise and applicable for all proofs using Natural Deduction. Similar to the editor for Resolution, scaffolding is achieved by the automatic application of rules and the possibility to provide partly constructed proofs. Following the concept of didactic reduction, the editor keeps track of the used and generated assumptions so that users do not have to manage assumptions at all, removing another potential roadblock on the way to a correct proof.

## 4. Evaluation

For the evaluation, both editors were integrated into the Backstage learning platform (https://backstage2.pms.ifi.lmu.de:8080). Additionally to the features of the editors described in Section 3, Backstage provided the students with exemplary solutions after they attempted an exercise and an overview on how their peers performed in the exercise.

A total of 13 exercises were provided to students in the week before the examination in July 2018. Five of the exercises were on Resolution and consisted of three exercises on propositional logic and two exercises on first order logic. Within each group and across those groups, the difficulty of the exercises increased.

The remaining eight exercises were exercises on Natural Deduction. The first four exercises focused on the usage of the implication rules, which is generally one of the more intuitive rules. Therefore, these exercises were considered easy. These exercises were followed by two exercises requiring a more extensive set of rules, therefore being considered as medium difficulty. The last block contained two difficult exercises which required large proof trees and use of *counter-intuitive* rules.

### 4.1 Methods

Two data sources were polled so as to evaluate the editors: A survey conducted after the course's examination and data collected from the Backstage system. The survey consisted of three parts:

- A block of six questions referring to the editor for Resolution.
- A block of six questions referring to the editor for Natural Deduction.
- Two questions to be answered with free text asking about positive and negative aspects of both editors.

For (1) and (2) a six-point Likert scale ranging from strongly agree (assigned value of 6) to strongly disagree (assigned value of 1) was utilized.

Data collected directly from the system included all attempts to all exercises and for each attempt the start and the end time. The time a student spent working on an exercise was calculated from those values. Attempts that took less than 30 seconds and empty submissions were not considered in the results below, as well as attempts that took more than 13.5 minutes. For the former, it can be assumed that those attempts were no serious tries, for the latter, that the student interrupted for a while and resumed the solving of the exercise at a later point in time.

### 4.2 Results

A total of 603 students were registered to the course on Backstage, of whom 193 attempted to solve at least one of the exercises. For the evaluation of the collected data, four performance measures were considered for each exercise:

- Number of students attempting the exercises
- Number of students solving the exercise correctly
- Number of students solving the exercise correctly in their first try
- Average time students spent working on an exercise until their first correct submission

The results for (1) and (2) can be seen in Table 1. For both Resolution and Natural Deduction, the number of students attempting subsequent exercises decreases steadily. For Natural Deduction, the number of students attempting subsequent exercises drops by 10 students per exercise, for Resolution by 20. The number of students submitting correct answers decreases as well: For Resolution exercises, a clear difference between correct answers for propositional logic (R1, R2, R3) and first order logic (R4, R5) can be observed, with the number of correct attempts decreasing by more than 50%. For Natural Deduction exercises, the number of students that submitted a correct answer remains more constant compared to the Resolution exercises, settling on around 66 with the beginning of the second and until the sixth exercise. 31 students attempted every Resolution exercise, with 20 of them solving each exercise correctly. All of the Natural Deduction exercises were attempted by 27 students, 9 of them solving each exercise correctly.

**Table 1**: Number of unique students and number of students that submitted a correct attempt per exercise

| Exercise | ND1 | ND2 | ND3 | ND4 | ND5 | ND6 | ND7 | ND8 | R1 | R2 | R3 | R4 | R5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Unique students | 133 | 109 | 92 | 86 | 78 | 85 | 96 | 80 | 135 | 137 | 119 | 94 | 54 |
| Students who submitted correct answer | 85 | 66 | 69 | 66 | 64 | 66 | 44 | 19 | 108 | 91 | 92 | 45 | 37 |

Figure 3 shows the number of students correctly solving Natural Deduction exercises on their first tries: Except for the last two exercises, there were always more than 40 students with an average of 55 students (Median: 57.5) submitting correct answers on their first tries. The results for Resolution, which are not displayed here due to space limitations, again show a cut happening between propositional and first order logic. Up to 80 students with an average of 64 (Median: 73) submitted a correct answer on their first try for propositional logic, while the first order logic exercises were only solved by less than 30 students on their first try.

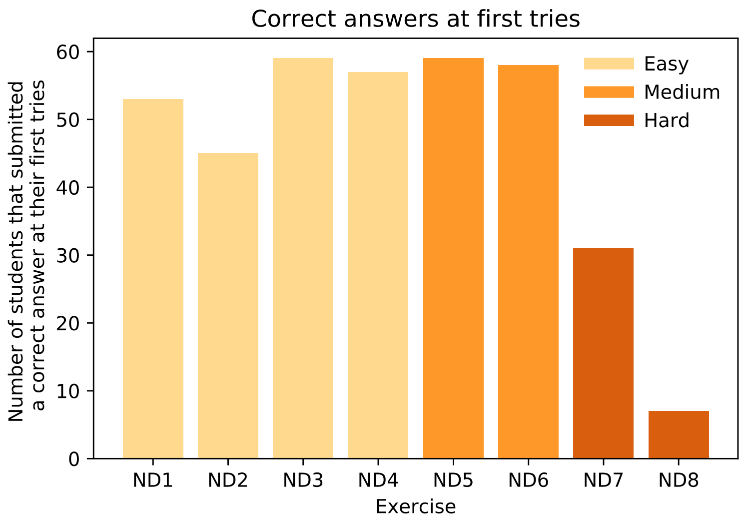


**Figure 3**: Number of students that submitted correct answers at their first tries for exercises on Natural Deduction

Figure 4 and Figure 5 show the average working time for Natural Deduction and Resolution exercises respectively. For Resolution exercises, students generally required twice as much time to solve exercises on propositional logic compared to exercises on first order logic. For Natural Deduction, the average working time decreases after the first two exercises until the fifth with an average working time of two minutes. After that, working time increases again with increasing exercise difficulty.

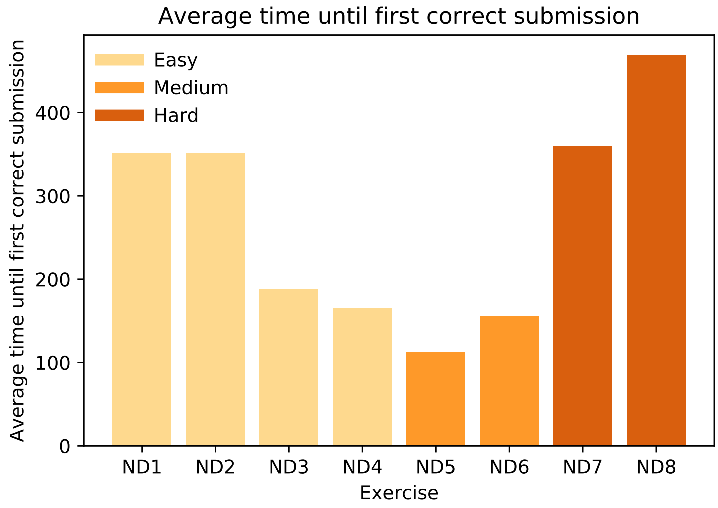


**Figure 4**: Average work time until first correct submission for exercises on Natural Deduction

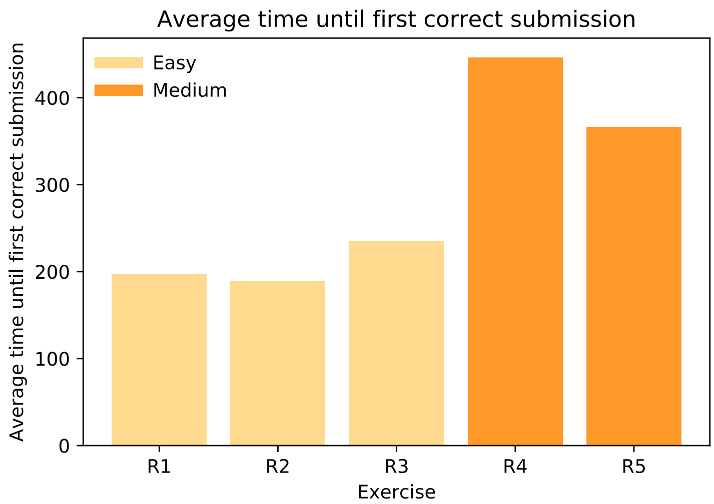


**Figure 5:** Average work time until first correct submission for exercises on Resolution

The questionnaire was answered by 17 students, but as four of those did not answer the questions on the editors the results refer to 13 students. Table 2 shows the results of the survey: Results are fairly consistent between the editors. Both editors and the feedback they provide helped students clearing up misconceptions, with feedback being the more important facilitator in both cases. Students were in disagreement whether the editor made doing the exercises easier than on paper.

**Table 2**: Results from the survey about editors for Resolution and Natural Deduction (n = 13). Statements were shortened due to lack of space

| Statement | Resolution | | Natural Deduction | |
|---|---|---|---|---|
| | Avg | SD | Avg | SD |
| Proof technique understood before using editor | 4.14 | 1.51 | 3.85 | 1.72 |
| Exercises using editor easier than doing exercises on paper | 3.46 | 1.76 | 3.46 | 1.80 |
| After familiarizing with the editor able to solve exercises | 4.77 | 1.48 | 4.61 | 1.76 |
| Feedback of the editor helped learning the proof technique | 4.23 | 1.59 | 4.23 | 1.54 |
| Editor helped me to clear up misconceptions about the proof technique | 4.31 | 1.65 | 4.23 | 1.54 |
| Feedback helped to clear up misconceptions about the proof technique | 4.46 | 1.56 | 4.31 | 1.55 |

## 4.3 Discussion

When looking at the first six Natural Deduction exercises, results show a decline in average working time, with the number of students that submitted a correct solution and the number of students that submitted a correct solution at their first tries remaining constant. This result suggests that students attempting the exercises were getting better while working through the exercises. However, it is not possible to determine if students were only getting more familiar using the editors or if students were getting better in understanding the principles of Natural Deduction. Furthermore, no small number of students dropped out of the exercises at some point, which could mean that the editors were not able to give them the scaffolding and feedback required for them to successfully solve the exercises. Indeed, the decline in average working time could be explained if only students with prior knowledge continued with the next exercises, as the scaffolding and feedback required students to have a basic understanding of the proof techniques, which the dropouts may have been lacking. While the number of attempts for the last two exercises in Natural Deduction stay consistent with the number of attempts of the previous exercises, the number of students successfully solving the exercises drop sharply. This may be explained that the two last exercises where extremely difficult and required the use of more "counter-intuitive" rules. Nevertheless, authors were positively surprised by the high numbers of students that were able to solve those two exercises.

The results for Resolution are not in accordance with those for Natural Deduction: None of the performance measures shows students improving while working through the exercises. The number of students attempting an exercise as well as the number of students submitting a correct solution drop sharply from propositional logic to first order logic, something that could not be observed in such magnitude for Natural Deduction exercises. This drop could be explained by the editor for Resolution requiring more *manual work* (calculating a most general unifier without support of the editor and entering it by hand) from students.

Students felt that they were supported by the editors and the feedback provided by them throughout the working process which indicates that the feedback and scaffolding provided by the editors worked as intended. However, as only about two percent of the students of the course took part in the survey, conclusions stemming from the survey have to be viewed critically. The survey items referring to feedback and scaffolding were rated worse for Natural Deduction than for Resolution, which could be explained by the difference in previous understanding of the proof techniques – less students agreed with having understood Natural Deduction than Resolution what could have led to those students not being able to process with the feedback and scaffolding provided by the editor. The results on the question asking whether the editors made solving the exercises easier than on paper show an unclear picture for both editors: Some students found working with the editors easier than working on paper, some not. As the answers to that question do not correlate with previous understanding of the proof techniques, it seems that other factors determine whether users prefer interactive editors or paper.

## 5. Conclusion and perspectives

This article introduced interactive editors helping students to work on exercises requiring the use of the proof techniques Resolution and Natural Deduction. Throughout the working process, the editors support students with feedback and scaffolding. As proof trees are generated by the editors, students are prevented from making formal mistakes and can completely focus on the steps of the proof. The correctness of each step is checked immediately, and if the step is valid, the proof tree is updated. Otherwise, an error message with information about the mistake is shown.

For evaluation, both editors were provided to students for examination preparation where students expressed that the editors and the feedback provided by them supported them in clearing up misconceptions and in learning the proof techniques. For Natural Deduction, results suggest that a learning process took part while students worked on exercises using the editor. This result could not be reproduced for Resolution, which could be explained by the editor for Resolution requiring more work outside of the editor from students, what could have deterred them from sufficiently engaging with the editor. However, although the evaluation indicates that the editors help students in learning logical proof techniques, the data collected is not sufficient to prove this assumption. However, this case study is intended as a foundation for further research. The authors are currently planning another study to determine the impact of the editors on students' performance.

The evaluation suggested potential improvements: The dropout across both types of exercises was fairly big, which could be addressed in various ways, e.g., by providing inexperienced students with a partly constructed proof tree, or adaptively determining which exercise to solve next to provide each student with an exercise they most likely can solve.

Generally, interactive editors could be deployed for most STEM activities involving formal languages, as the correctness of single steps or at least complete solutions can easily be checked for correctness by a machine, e.g., chemical equations or electrical circuits. Going further, even in cases were automatically generated feedback is not possible, peer reviews opens up ways for generating feedback: Technology enables a form of peer review where students can immediately start reviewing other students' submissions providing instant feedback.

## References


Azevedo, R. and Hadwin, A.F. (2005) "Scaffolding self-regulated learning and metacognition–Implications for the design of computer-based scaffolds", *Instructional Science*, *33*(5), pp.367-379.

Geck, G., Ljulin, A., Peter, S., Schmidt, J., Vehlken, F and Zeume, T. (2018) "Introduction to ILTIS: an interactive, web-based system for teaching logic", *arXiv preprint arXiv:1804.03579*.

Gentzen, G. (1935) "Untersuchungen über das logische Schließen", I. *Mathematische Zeitschrift*, *39*(1), pp.176-210.

Gibbons, P. (2002) "Scaffolding language, scaffolding learning", Portsmouth, NH: Heinemann.

Grüner, G. (1967) "Die didaktische Reduktion als Kernstück der Didaktik", *Die Deutsche Schule*, *59*(7/8), pp.414-430.

Gueudet, G. (2008) "Investigating the secondary–tertiary transition", *Educational studies in mathematics*, *67*(3), pp.237-254.

Hallgren, T. and Ranta, A. (2000) "An extensible proof text editor", *International Conference on Logic for Programming Artificial Intelligence and Reasoning,* pp.70-84.

Hattie, J. and Timperley, H. (2007) "The power of feedback", *Review of educational research*, *77*(1), pp.81-112.

Jackson, S.L. (1996) "The design of guided learner-adaptable scaffolding in interactive learning environments", *Proceedings of the 1996 international conference on Learning sciences* (pp. 575-576). International Society of the Learning Sciences.

Kluger, A.N. and DeNisi, A. (1996) "The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory", *Psychological bulletin*, *119*(2), p.254.

Lukins, S., Levicki A., and Burg J. (2002) "A tutorial program for propositional logic with human/computer interactive learning", *ACM SIGCSE Bulletin,* 34(1), pp. 381-385.

Nicol, D.J. and Macfarlane-Dick, D. (2006) "Formative assessment and self-regulated learning: A model and seven principles of good feedback practice", *Studies in higher education*, *31*(2), pp.199-218.

Orsmond, P. and Merry, S. (2013) "The importance of self-assessment in students' use of tutors' feedback: A qualitative study of high and non-high achieving biology undergraduates", *Assessment & Evaluation in Higher Education*, *38*(6), pp.737-753.

Robinson, J.A. (1965) "A machine-oriented logic based on the resolution principle", *Journal of the ACM*, *12*(1), pp.23-41.

Sao Pedro, M.A., Gobert, J.D. and Baker, R.S. (2014) "The impacts of automatic scaffolding on students' acquisition of data collection inquiry skills", *Roundtable presentation at American Educational Research Association*.

Staudacher, K. (2018) "Conception, Implementation and Evaluation of Proof Editors for Learning", *Bachelor's Thesis*, Institute for Informatics, Ludwig Maximilian University of Munich.

Van Merriënboer, J.J., Kirschner, P.A. and Kester, L. (2003) "Taking the load off a learner's mind: Instructional design for complex learning", *Educational psychologist*, *38*(1), pp.5-13.

Wu, H.K., Krajcik, J.S. and Soloway, E. (2001) "Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom", *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, *38*(7), pp.821-842.

Yacef, K. (2005) "The Logic-ITA in the classroom: a medium scale experiment", *International Journal of Artificial Intelligence in Education*, *15*(1), 41-62.

Zengin, Y., Furkan, H. and Kutluca, T. (2012) "The effect of dynamic mathematics software geogebra on student achievement in teaching of trigonometry. *Procedia-Social and Behavioral Sciences*, *31*, pp.183-187.