Received: 12 July 2018 | Revised: 15 March 2019 | Accepted: 24 March 2019
DOI: 10.1111/jcal.12365

WILEY Journal of Computer Assisted Learning

ARTICLE

A comparison of elaborated and restricted feedback in LogEx, a tool for teaching rewriting logical formulae

Josje Lodder[1] | Bastiaan Heeren[1] | Johan Jeuring[1,2]

[1]Faculty of Management, Science and Technology, Open University of the Netherlands, Heerlen, The Netherlands
[2]Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands

Correspondence
Josje Lodder, Faculty of Management, Science and Technology, Open University of the Netherlands, PO Box 2960, Heerlen 6401 DL, The Netherlands.
Email: josje.lodder@ou.nl

Abstract

This article describes an experiment with LogEx, an e-learning environment that supports students in learning how to prove the equivalence between two logical formulae, using standard equivalences such as DeMorgan. In the experiment, we compare two groups of students. The first group uses the complete learning environment, including hints, next steps, worked solutions, and informative timely feedback. The second group uses a version of the environment without hints or next steps, but with worked solutions, and delayed flag feedback. We use pretest and posttest to measure the performance of both groups with respect to error rate and completion of the exercises. We analyse the loggings of the student activities in the learning environment to compare its use by the different groups. Both groups score significantly better on the posttest than on the pretest. We did not find significant differences between the groups in the posttest, although the group using the full learning environment performed slightly better than the other group. In the examination, which took place 5 weeks after the experiment, the group of students who used the complete learning environment scored significantly better than a control group of students who did not participate in the experiment.

1 | INTRODUCTION

Students learning propositional logic practice by solving different kinds of exercises. Many of these exercises are solved stepwise. To support a student solving such an exercise, an intelligent tutoring system can be very effective (VanLehn, 2011). These systems offer several kinds of assistance, for example, step by step feedback, instructions to repair common errors, hints or next steps, or even complete solutions. The timing of this assistance varies: directly after the performance of a step or only after the completion of an exercise. Based on a review of the literature, Koedinger and Aleven (2007) state that offering assistance can make learning more efficient, but misuse of help can cause shallow learning. On the other hand, withholding information forces students to construct their own solution, which may benefit attention, but might waste time and result in confusion. Koedinger and Aleven introduce the term "assistance dilemma" and review several experiments that compare different strategies for giving and withholding feedback. The conditions immediate versus delayed yes/no feedback were studied in an experiment with a Lisp tutor (Anderson, Corbett, Koedinger, & Pelletier, 1995) and an Excel tutor (Mathan & Koedinger, 2005). The first study concludes that immediate feedback causes students to learn faster and better than with feedback after completion of the exercise, but in the experiment with an Excel tutor, where repairing mistakes was one of the learning goals, allowing initial errors resulted in better performance not only on a posttest but also on long-term retention and transfer. An experiment with the Geometry Proof Tutor (Koedinger & Aleven, 2007) comparing explanatory feedback with yes/no feedback resulted in a significantly lower posterror rate in the explanatory feedback condition. The question whether a hint containing conceptual information is more effective than providing a next step is partially answered by a study that compares explanatory error messages with correcting next steps, where the former strategy turns out to be more effective. These experiments support the approach of balancing giving and withholding information taken in cognitive tutors. However, Koedinger and Aleven claim that the question of how to decide which information should be given at what moment is a fundamental open problem.

Studies on the assistance dilemma often address a particular subproblem, such as whether or not supplying worked examples results in more efficient learning. The outcomes of studies related to worked

examples vary. Although a comparison of untutored learning versus worked examples shows that worked examples are superior (Sweller & Cooper, 1985), the results of comparing tutored learning with worked examples are less clear and may depend on the level of a student, exercise difficulty, or content (procedural vs. conceptual; Kim, Weitz, Heffernan, & Krach, 2009; Razzaq & Heffernan, 2009; Shrestha et al., 2009). Strategies where (untutored) problems are alternated with worked examples are superior when a worked example is followed by a problem instead of a problem followed by a worked example (van Gog, Kester, & Paas, 2011). Offering a worked solution can be seen as a special case of providing a worked example. Compared with a situation where exercises are scaffolded by giving students hints, good students perform better when receiving a worked solution, but for average students, this is the other way around (Razzaq, Heffernan, & Lindeman, 2007). As far as we know, the question whether adding the possibility to ask for hints and next steps supports learning in a situation where a student can ask for a worked solution has not yet been studied.

Several models try to explain the effects of different assistance strategies. Chi (2009) introduces a framework to differentiate the terms "active, constructive, and interactive" in terms of observable activities and underlying cognitive processes. She classifies physical activities as active, the production of output beyond the presented information as constructive, and performing a dialogue taking the partner's contributions into account as interactive. The involved cognitive processes are attending processes, creating processes, and creating processes that incorporate a partner's contributions, respectively. She uses this classification to hypothesize that constructive processes have better learning results than active processes, and interactive processes have better results than constructive processes. Cognitive load theory is also used to explain differences between assistance strategies. According to Salden, Koedinger, Renkl, Aleven, and McLaren (2010), worked examples reduce extraneous cognitive load and save time. The interactive tutoring feedback model (Narciss, 2013) introduces a framework that distinguishes an internal learner's feedback loop and an external feedback loop. The model suggests that learning not only depends on external factors such as content and timing of feedback but also depends on learner characteristics. Narciss et al. (2014) study the influence of learner characteristics in an experiment with sixth and seventh graders working on fractions. One of the outcomes is that male students profit less from feedback than females.

A second subquestion of the assistance dilemma concerns the timing and amount of feedback when a student makes an error. Based on a review of the literature, Shute (2008) lists several guidelines to enhance formative feedback, but she does not give definitive answers. According to these guidelines, immediate feedback should be used for retention of procedural knowledge and delayed feedback for transfer of learning. The question remains which approach is best for a particular domain of study.

In this paper, we describe an experiment with LogEx[1], a learning environment (LE) that supports students in rewriting propositional logical formulae using standard equivalences. The learning goals addressed by LogEx are as follows: After practicing with LogEx, a student can

- correctly apply rewriting rules for propositional logic
- prove the equivalence of two formulae using standard equivalences
- demonstrate strategic insight in how to efficiently prove an equivalence

Here, an efficient proof is a solution that uses a minimal number of steps.

The main research question we investigate in this paper is: do students reach the above learning goals by practicing with LogEx? We also want to contribute to the assistance dilemma by investigating whether or not hints and immediate feedback have an effect on student learning. Do students who receive hints and feedback while practicing perform better than students who practice with a version of LogEx with just delayed feedback and worked solutions? We hypothesize that students who receive immediate feedback and who can use hints make fewer errors and can complete more exercises.

This paper is organized as follows. The next section reviews several evaluation studies with other LEs for rewriting logical formulae or proving logical consequences. We continue with describing LogEx in more detail, together with a short review of previous studies performed with LogEx. The experiment is described in Section 4. Section 5 presents and discusses the results of the assessment tests and loggings. Section 6 summarizes our conclusions and proposes future research.

2 | EVALUATION RESULTS FROM OTHER LEs

This section discusses related work in educational experiments with logic tutors.

In a previous paper (Lodder, Heeren, & Jeuring, 2016), we reviewed six e-LEs comparable with LogEx. Only one of these environments has been used in an experiment with students. In FOL (Grivokostopoulou, Perikos, & Hatzilygeroudis, 2013), students rewrite first-order logical formulae using standard equivalences. Feedback is presented in stages: first, a student chooses a rule that can be applied, and only after the system approves, the student can continue with the rewriting step. The designers of FOL compared a group of students who practice with the LE for one week 20 min a day with a control group of students who solve homework using pen and paper, discussed by the teacher afterwards. The results show a statistically significant better performance on a posttest by the group who practiced with FOL.

We have found a number of evaluation studies using LEs for teaching logic focusing on different kinds of exercises.

Logic Tutor (Yacef, 2005) supports learning how to prove a consequence using rewriting rules (such as DeMorgan) in combination with inference rules. It presents proofs in a linear form and only allows rewriting in one direction. It provides feedback, for instance, about a missing reference to a previous proof line, at each step, but offers no hints or next steps. Student interactions are logged and can be analysed by teachers, for example, to improve their teaching. Several experiments with the Logic Tutor were performed in 2000–2003. Answers to exam questions show improvement from year to year, partly because of the use of the tutor by students but also because of teachers analysing the loggings of the tool.

Deep Thought (Mostafavi & Barnes, 2017; Stamper, Eagle, Barnes, & Croy, 2011b) offers exercises comparable with Logic Tutor but presents proofs as trees and allows students to construct a proof by adding forward and backward steps. An evaluation study of Deep Thought addressed the question of whether the use of data-driven methods in problem selection and feedback in the development of Deep Thought influences student dropout and the time needed to complete the exercises in the tutor (Mostafavi & Barnes, 2017). A comparison of four versions of Deep Thought showed that in each new version, student dropout and time to complete the exercises in the tutor decreased significantly. An experiment where students either solved a set of three or four problems or watched the worked solution of one or two of these problems and solved another two showed that worked examples reduced hint dependency for high proficiency students. Students who received two worked solutions constructed shorter solutions but also made more mistakes. On the other hand, low proficiency students in the worked example condition made more mistakes and produced longer solutions than low proficiency students who did not receive worked examples (Liu, Mostafavi, & Barnes, 2016). An earlier paper showed that students who could ask for hints performed significantly better than students who did not receive hints (Stamper et al., 2011b).

Miwa, Terai, Kanzaki, and Nakaike (2014) describe an intelligent tutor to help a student with solving natural deduction problems. It contains a complete problem solver, which provides various kinds of support, for example, which rule can be applied to which formula or which set of rules is applicable. An experiment with the LE showed that students who used the LE performed significantly better on easy posttest exercises than a control group that received traditional classroom instruction. There was no significant difference in performance on the more difficult exercises.

3 | LogEx

LogEx is an LE in which a student practices rewriting propositional logical formulae. LogEx contains three kinds of exercises: rewriting a formula in DNF, in CNF, and proving the equivalence of two formulae. A student enters her solution stepwise. To illustrate the functionality of the LE, we give an example of how a student might solve an exercise in LogEx.

Suppose the student has to prove that

$$p \wedge (q \vee s) \Longleftrightarrow (q \wedge \neg s \wedge p) \vee (p \wedge s).$$

In LogEx, the left-hand side of this equivalence is shown at the top of the screen and the right-hand side at the bottom. A student might recognize that after swapping p and q in the bottom line, it is possible to take the variable p out of the conjunctions by applying distribution in reverse. LogEx allows to rewrite the bottom formula, and after applying commutativity and distribution, the partial proof is of the form given in Figure 1. The student can continue by rewriting the formula in the edit field, using shortcuts or a small keyboard to enter the logical connectives, and motivating the step by choosing the name of the rule applied from a drop-down list. The student can also change the direction in which she is working at any moment. For example, she could proceed by rewriting the line $p \wedge (q \vee s)$ at the top of the proof into $(q \vee s) \wedge p$.

In the complete version of LogEx, a student receives feedback after each step. Feedback concerns syntax errors, such as missing parentheses, or rule feedback. After a student enters a formula, LogEx tries to recognize the rule that is used. If it detects a rule, it compares this rule with the rule specified by the student and gives an error message giving the correct rule name if the wrong rule name is specified. LogEx uses a set of common mistakes, also called buggy rules, to try to give informative feedback. For example, if a student rewrites $\neg(p \vee q) \vee (\neg\neg p \wedge \neg q) \vee \neg q$ into $(\neg p \vee \neg q) \vee (\neg\neg p \wedge \neg q) \vee \neg q$, then LogEx reports that this step is incorrect and mentions that when applying DeMorgan's rule, a disjunction is transformed into a conjunction. If no rule or buggy rule is detected, LogEx checks whether or not the new and old formulae are semantically equivalent. If they are not equivalent, LogEx mentions that an error is made, otherwise the student receives a message that she either combined two or more steps in one or made a mistake. In the version of LogEx discussed in this paper, a student can only proceed after correcting a mistake.

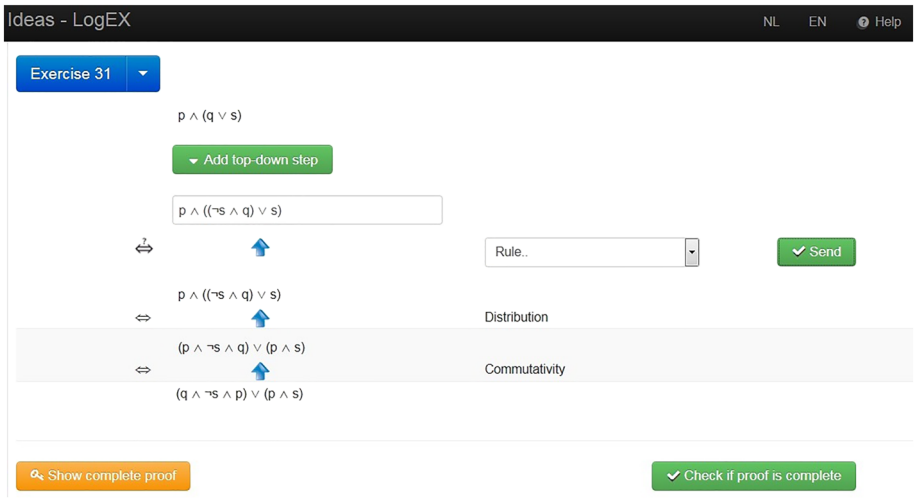

FIGURE 1 Screenshot of LogEx [Colour figure can be viewed at wileyonlinelibrary.com]

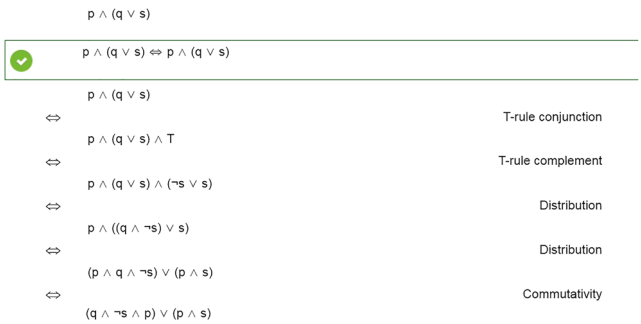

FIGURE 2 The complete solution of the exercise in Figure 1, generated by LogEx [Colour figure can be viewed at wileyonlinelibrary.com]

In LogEx, a student can ask for

- a hint, for example, in the situation of Figure 1, LogEx first hints to rewrite the boxed formula, continuing in the same direction of the proof, and then to apply Distribution
- a next step, for example, LogEx rewrites $p \wedge ((q \wedge \neg s) \vee s)$ into $p \wedge ((q \vee s) \wedge (\neg s \vee s))$
- or a complete worked solution as shown in Figure 2

at any moment. The LE uses solution strategies to calculate this feed forward. This strategy can be restarted after each rewriting, so that hints and next steps can also be given when a student diverges from the solution of the problem that is calculated by the LE. A student can choose between exercises of different difficulty levels or enter her own exercise. Feedback and feed forward are available for all exercises, including user-defined problems. LogEx integrates improved versions of earlier tools to rewrite formulas in disjunctive normal form (Lodder, Jeuring, & Passier, 2006; Lodder, Passier, & Stuurman, 2008) and to prove equivalences (Lodder & Heeren, 2011).

3.1 | Pilot studies

We have evaluated various aspects of LogEx in several pilot studies (Lodder, Heeren, & Jeuring, 2015; 2016; Lodder et al., 2008). We have used these pilot studies to evaluate the usability of LogEx and to prepare for a large scale experiment (Shute & Regian, 1993). In our first experiments, we compared the complete version of LogEx, in which hints and next steps are available and a user gets feedback directly after performing a step, with a version without hints or next steps and a user receives postponed feedback. The number of participating students was too low to draw firm conclusions, but the loggings of the use of LogEx in these experiments indicated that (Lodder et al., 2008)

- the possibility to ask for a next step is essential for weaker students. Students who used a version of LogEx without the availability of next steps could not complete more complicated exercises.
- the availability of next steps teaches students to use rules they overlook (e.g., false–true rules to simplify an expression).
- the requirement to perform one step at a time forces students to recognize mistakes they would overlook otherwise. An example of such a mistake is applying distributivity on equal connectives, which results in an equivalent formula, but is not a correct application of distributivity.
- because learning an efficient strategy is implicit in LogEx, students who do not use the hint and next step button can proceed with inefficient strategies without receiving feedback on this aspect.

In a second experiment (Lodder et al., 2015), analysis of the loggings showed that during the experiment, students gradually need less time to complete an exercise, and feedback helps students to recognize and correct their mistakes.

4 | METHOD

In September 2017, we performed an experiment with LogEx at a university of applied sciences. The participants were second year computer science students taking a course in discrete mathematics, which has propositional logic as one of its topics. Students have to learn to simplify formulae using standard equivalences and to prove the equivalence of formulae.

4.1 | Pilot

To prepare for the experiment, we performed a pilot with 13 part-time students in May 2017. This experiment took place directly after class-based instruction on equivalences. The pilot consisted of

- a short introduction about the purpose of the experiment and instruction on how to use LogEx.
- a 20-min pretest consisting of three exercises comparable with the LogEx exercises.
- working with LogEx for 50 min.
- a 20-min posttest consisting of three exercises comparable with the pretest.

Students were divided into two groups. One group used the complete version of LogEx and the other group could not use hints or next steps and only received check marks for correct steps after completing an exercise. The latter group could also ask for a worked solution and compare it with their own solution. All students could use a formula sheet so that they did not have to memorize the logic rules.

The main outcome of this experiment was that students scored very low on the pretest. On average, students completed only half of the first exercise, 5% of the second, and nothing of the third. This implies that the pretest cannot be used to differentiate between student levels.

Because these results are not very encouraging for students, and not very useful for teachers and researchers, we changed our experiment in two ways. First, we planned the experiment the week after class-based instruction of standard equivalences. This way, students could review the topic before the experiment and already practice a bit. Second, we replaced the first exercise of the pretest with an exercise that was slightly easier.

4.2 | Experiment

Three classes with a total of 74 students participated in the experiment. The participants were males between 19 and 31 years. We

TABLE 1 Functionalities of the full and restricted versions of LogEx

Functionality	Full LogEx	Restricted LogEx
Hints	✓	×
Next step	✓	×
Complete solution	✓	✓
Immediate feedback	✓	×
Informed feedback	✓	×
Delayed check mark feedback	×	✓

TABLE 2 Pretest and posttest

Test 1		Test 2
Pretest		
1.	$q \rightarrow \neg(p \vee q) \Longleftrightarrow \neg q$	$q \rightarrow \neg(p \vee q) \Longleftrightarrow \neg q$
2.	$(p \vee q \vee r) \wedge (r \vee \neg p) \Longleftrightarrow (q \wedge \neg p) \vee r$	$(p \wedge q) \rightarrow (q \wedge r) \Longleftrightarrow q \rightarrow (p \rightarrow r)$
3.	$((\neg p \vee q) \wedge p) \vee (\neg(\neg p \vee q) \wedge \neg p) \Longleftrightarrow q \wedge p$	$((p \wedge q) \vee (\neg p \wedge \neg q)) \rightarrow p \Longleftrightarrow p \vee q$
Posttest		
1.	$((\neg p \vee q) \wedge \neg p) \rightarrow p \Longleftrightarrow p$	$((\neg p \vee q) \wedge \neg p) \rightarrow p \Longleftrightarrow p$
2.	$(p \wedge q) \rightarrow (q \wedge r) \Longleftrightarrow q \rightarrow (p \rightarrow r)$	$(p \vee q \vee r) \wedge (r \vee \neg p) \Longleftrightarrow (q \wedge \neg p) \vee r$
3.	$((p \wedge q) \vee (\neg p \wedge \neg q)) \rightarrow p \Longleftrightarrow p \vee q$	$((\neg p \vee q) \wedge p) \vee (\neg(\neg p \vee q) \wedge \neg p) \Longleftrightarrow q \wedge p$

compared two conditions: the use of the complete version of LogEx with elaborated feedback (Narciss, 2008), versus the version without hints and next steps, and with delayed check mark feedback, see Table 1.

To validate the pretest and posttest, we divided both groups into two subgroups, for which we used the two variants of the pretest and posttest given in Table 2. Both tests consist of three exercises. We used Exercise 1 of the pretest to measure the difference in rewriting skills between the groups before the start of the experiment and hence offered this exercise to both groups. The first exercise in the posttest is a slightly more complicated variant of the first exercise in the pretest. Exercises 2 and 3 of pretest version 1 are the same as Exercises 2 and 3 in the posttest in version 2, and vice versa. We used these exercises to measure learning gains, as described, for example, by Bartsch, Bittner, and Moreno (2008). We use the following abbreviations for the subgroups of students:

- F1: students using the full version of LogEx and Test 1
- F2: students using the full version of LogEx and Test 2
- R1: students using the restricted version of LogEx and Test 1
- R2: students using the restricted version of LogEx and Test 2
- F: F1 + F2, all students using the full version of LogEx
- R: R1 + R2, all students using the restricted version of LogEx
- 1: F1 + R1, all students taking Test 1
- 2: F2 + R2, all students taking Test 2

TABLE 3 Number of students in different groups

Group	Test 1		Test 2		Total
Full LogEx F	F1	21	F2	9	30
Restricted LogEx R	R1	29	R2	15	44
Total		50		24	74

Table 3 shows the number of students in each group.

The organization of the experiment was comparable with the pilot: a short introduction, a 20-min pretest, followed by 50 min practicing with LogEx, and, after a short break, a 20-min posttest. Students were allowed to use a formula sheet during pretest, practicing, and posttest. We logged the use of LogEx. The list of 12 exercises used in LogEx can be found in the Appendix. All raw data are available via data.mendeley.com.[2]

5 | RESULTS AND DISCUSSION

5.1 | Results of pretest and posttest

The first exercise in the pretest, which was the same for all students, was used to test whether the prior knowledge of all groups was comparable. We scored the exercise in two ways. The first score is the completion rate of the exercise: the number of completed steps divided by the total number of a completed version of the student's solution. The second score is the relative number of incorrect lines (the number of incorrect steps divided by the total number of steps). The first score is used to measure the learning goal "being able to prove equivalence," and the second to measure the learning goal "applying the rules correctly." Some students make a mistake in the first or second line but continue without mistakes, which may result in a shorter solution. Because we do not know whether students are able to finish the exercise had they not made the mistake, we grade these cases as follows: the grade consists of the number of completed steps divided by the total number of steps in the standard solution. The descriptive statistics of both measures are shown in Table 4. The statistics indicate that differences between the four groups F1, F2, R1, and R2 are small, although group R seems to make some more mistakes.

We use nonparametric tests to compare the distribution of the variables completion rate and relative number of incorrect lines in the four different Groups F1, F2, R1, and R2, because the Kolmogorov–Smirnov

TABLE 4 Descriptive statistics of completion and relative number of incorrect lines in the first exercise of the pretest

	Group 1		Group 2		Total	
	Mean	Std. dev	Mean	Std. dev	Mean	Std. dev
Completion						
Full LogEx	0.86	0.27	0.81	0.29	0.84	0.27
Restricted LogEx	0.83	0.30	0.82	0.18	0.83	0.26
Total	0.84	0.29	0.82	0.22	0.84	0.27
Relative number of incorrect lines						
Full LogEx	0.12	0.21	0.15	0.34	0.13	0.25
Restricted LogEx	0.20	0.33	0.12	0.22	0.17	0.29
Total	0.17	0.28	0.13	0.26	0.15	0.27

TABLE 5 Results of the Kruskal–Wallis test on differences between Groups F1, F2, R1, and R2 in performance in com pletion of, and relative number of incorrect lines in, pretest Exercise 1

	Completion	Relative number of incorrect lines
Chi square	1.7	1.1
df	3	3
p	.65	.79

test on normality of the variables completion rate and relative numbers of incorrect lines fails. The results can be found in Table 5. The outcome of the Kruskal–Wallis test indicates that there is no difference in the distribution of these variables between the different groups. A comparison of Group F versus Group R, and of Group 1 versus Group 2, also shows no significant difference. We use a Mann–Whitney *U* test with threshold $p = .05$ (Nachar, 2008; Hayes, 1988) and find significance levels of .58 for completion and .50 for relative numbers of incorrect lines when we compare Group F with Group R, and significance levels of .26 for completion and .48 for relative number of incorrect lines when we compare Group 1 with Group 2. We conclude that prior knowledge was evenly distributed between the four groups F1, F2, R1, and R2, and that the difference in the number of mistakes between the groups F and R is not significant. Because we only look at differences between pretest and posttest, a small variation between the groups does not influence our conclusions.

Our first research question is: do students learn by using LogEx, or, more precisely, do students learn to apply rules correctly, prove the equivalence of two formulae using standard equivalences, and solve these exercises efficiently. We use the second and third exercise of the pretest and posttest to answer the first and second subquestion. To correct for a possible difference in the level of difficulty between the exercises in the pretest and posttest, we divided the students into four groups, F1, F2, R1, and R2, as described in Table 3.

First, we looked at the overall knowledge gain, independent of the version of LogEx, which means that we take Groups F1 and R1 (Group 1) and F2 and R2 (Group 2) together. For Groups 1 and 2, we compared completion of Exercises 2 and 3 in the pretest with completion in the

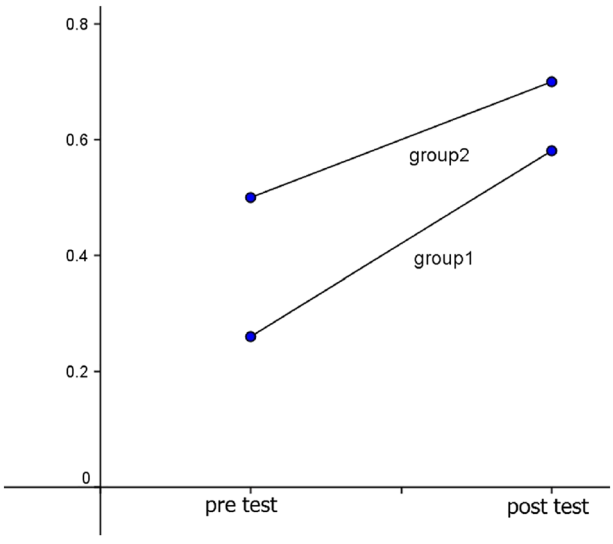

FIGURE 3 Pretest and posttest completion rates on Exercises 2 and 3 [Colour figure can be viewed at wileyonlinelibrary.com]

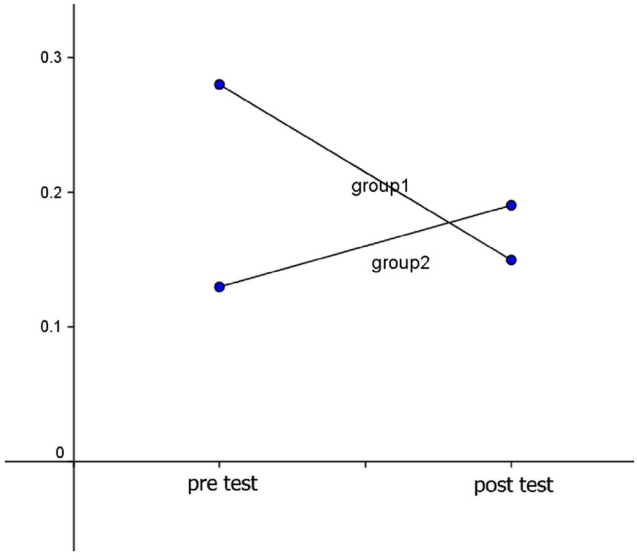

FIGURE 4 Relative numbers of incorrect lines in pretest and posttest Exercises 2 and 3 [Colour figure can be viewed at wileyonlinelibrary.com]

posttest. Figure 3 shows the results. The graph indicates that pretest 1 (= posttest 2) might be more difficult than pretest 2 (= posttest 1), but both groups complete more of the exercises in the posttest than in the pretest. This is confirmed by tests on effect size: Cohen's *d* for Group 1 equals 0.72 (confidence interval [0.59, 0.82]) and for Group 2 equals 0.42 ([0.20, 0.58]), so despite the possibly more difficult posttest in Group 2, the effect can be classified as medium–high. The results for the relative number of incorrect lines were less conclusive: students in Group 1 made fewer mistakes in the posttest than in the pretest (Cohen's $d = -0.54$ [−0.60, −0.44]), but in the second group, this was the other way around (Cohen's $d = 0.25$ [0.15, 0.36]), see Figure 4. Note that in this case, a negative number means fewer mistakes. Although the second group made more mistakes in the posttest, the decrease in mistakes in Group 1 was larger than the increase in Group 2.

To interpret the numbers on effect size, we compare them with Hattie's list of effects ranks. He mentions "Computer aided instruction" with an effect size of 0.37, and an effect size of 0.6 is reached, for example, by teaching strategies or problem-solving teaching (Hattie, 2012). Compared with other interventions, the effect of practicing with LogEx on completion is indeed substantial. We conclude that working with LogEx helps students to learn how to prove equivalence between formulas. Although our measure for errors already takes into account the total number of steps in a solution, the inconclusive results on errors might be explained by the fact that because students complete more of the exercises, they will also have to rewrite more complicated formulae. Because students could use a formula sheet, errors are not caused by incorrectly remembered rules. There are several other sources of errors, such as sloppiness, misunderstanding, overgeneralization, or just creative rule interpretation to finish a proof. We looked more closely at the errors made but concluded that without asking students, it is hard to categorize the errors. For example, a student who forgets to change a disjunction into a conjunction while applying DeMorgan may misunderstand the rule but may also be sloppy. In the same way, distributing a conjunction over a conjunction may be caused by sloppiness but also by overgeneralization. We think that a large part of the mistakes are slips and that practicing with LogEx for 50 min is too short to address this. Fatigue might also have influenced these results, as may have the fact that the students knew they were not going to be graded based on their results.

To answer the question whether giving feedback and hints has an effect on student learning, we compare the results of the group using the full version of LogEx with the restricted version. We compare the normalized knowledge gain between Groups F1 and R1 and between Groups F2 and R2. Normalized knowledge gain is defined by: normalized gain = $(post - pre)/(100 - pre)$ where *post* and *pre* can reach values between 0 and 100 (Hake, 1998).

We use the completion rates of Exercises 2 and 3 as results of the pretest and posttest. Because the maximum score for the completion rate of the exercises is 2, we use the following variant of normalized gain:

$$\frac{post2 + post3 - pre2 - pre3}{2 - pre2 - pre3}.$$

We also compared the relative error gain:

$$relative\ error\ gain = \frac{errorpost2 + errorpost3}{\#post2 + \#post3} - \frac{errorpre2 + errorpre3}{\#pre2 + \#pre3},$$

where *errorpre*2 is the number of lines containing one or more errors in pretest Exercise 2, and *#pre*2 is the number of lines in the student submission of Exercise 2 in the pretest. The definition of the other variables is similar. Because quite a number of students did not fill out any line in Exercise 2 or 3 in the pretest, we also compared the absolute number of errors made in pretest and posttest. The results can be found in Table 6.

We used a Mann–Whitney *U* test to examine whether the users of the full version of LogEx performed significantly better than the users of the restricted version. According to the test, this result is not statistically significant, see Table 7.

TABLE 6 Descriptive statistics of normalized gain in completion Exercises 2 and 3, relative error gain, and error gain

	Norm completion gain			Relative error gain			Error gain		
	n	Median	Mean	*n*	Median	Mean	*n*	Median	Mean
Group 1									
Full LogEx	21	0.15	0.21	15	−0.21	−0.26	21	0	0.33
Restricted LogEx	29	0.11	0.13	16	−0.04	−0.07	29	0	0.38
Group 2									
Full LogEx	9	0.07	0.17	8	0	0.03	9	0	0.33
Restricted LogEx	14	0.04	0.10	14	−0.02	0.03	15	1	0.33

TABLE 7 Results of the Mann–Whitney *U* test on differences between users of the full version of LogEx versus the restricted version, for Group 1 and Group 2

	Normalized gain	Relative error gain	Absolute error gain
Group 1			
Mann–Whitney *U*	257.5	86	278
Z	−0.93	−1.35	−0.55
p	.18	.092	.298
Group 2			
Mann–Whitney *U*	54	53	64
Z	−0.57	−0.21	−0.22
p	.29	.43	.44

There are several reasons why the differences between Group F and Group R are small. Although Group R could not use hints or next steps, they could ask for a complete solution and use this as a worked example. Learning with worked examples can be very effective (Sweller, Ayres, & Kalyuga, 2011), and in paragraph 5.3, we will show that Group R indeed used the complete solution to get a hint. Where most of the studies described by Koedinger and Aleven (2007) showed better results for immediate and informed feedback, in our experiments, the effect on the number of errors is not significantly different between the two groups. A possible reason might be that our students could use a formula sheet, which makes informed feedback partly superfluous. Because male students profit less from feedback than female students Narciss et al. (2014), our 100% male population might be another explanation for the nonsignificant effects. Students worked individually on the pretest and posttest but could help each other while working with LogEx, and we actually observed this. As argued by Chi (2009), helping each other might make more difference than the presence or absence of feedback. Another reason could be that the experiment was too short to yield significantly different results between both versions of LogEx. The opposite results for high and low proficient students in the study by Liu et al. (2016) suggest that a separate analysis for these groups might yield significant results. However, the number of students in our experiment was too low to perform such an analysis.

We also wanted to find out whether students learn to solve exercises efficiently, by which we mean that students construct short solutions. We measure efficiency by dividing the total number of steps a student takes to solve an exercise by the number of steps of a worked solution generated by LogEx. Hence, a low score means an efficient solution. When a student finds a shorter solution than LogEx, this score is less than 1, which actually happened in a few cases (for three exercises, with respectively two, five, and one student). Efficiency is only measured when a student finishes an exercise correctly. In the pretest and posttest, the number of correct solutions for Exercises 2 and 3 was too low to draw conclusions. Students who finished the first exercise in the pretest found an efficient solution (efficiency = 1 in Group F and 1.2 in Group R). The solutions of the slightly more difficult Exercise 1 in the posttest were less efficient (1.6 for both groups).

TABLE 8 Exam results for the exercises on rewriting logical formulae and the total exam score

		Logic exercise		Total	
Group	*n*	Mean	Std. dev	Mean	Std. dev
N	43	2.5	2.5	46.5	15.8
F	30	4.1	2.1	50.8	17.2
R	38	3.5	2.4	55.8	19.5

TABLE 9 Results of the Mann–Whitney *U* test on differences in the results on the logic exer cises in the exam between users of the full version of LogEx versus the restricted version

	Logic exercises
Mann–Whitney *U*	502.5
Z	−0.88
p	.2

We conclude that the pretest and posttest do not provide enough information to decide whether students develop strategic insight.

5.2 | Exam results

To measure the medium-term effect of the use of LogEx, we analysed the results of two exam questions. The exam took place 5 weeks after the experiment and contained two questions on rewriting propositional formulae, besides other questions in discrete mathematics. In the first question, students had to simplify a propositional formula using rewrite rules, in the second question, they had to prove an equivalence. One hundred and eleven students took the exam, 43 of which did not participate in the experiment. Of the remaining 68 students, 30 practiced with the full version of LogEx and 38 with the restricted version. Most of the students who did not participate in the experiment were taking a resit. The scores of this group are much lower than those of the other students. In the following, we denote these students by Group N. The maximum score for the exam was 100 points, six of which could be earned by correct answers to the questions on rewriting logical formulae. The results of the students on the questions on rewriting logical formulae can be found in Table 8.

On average, the students using the full version of LogEx performed better on the rewriting logical formulae questions than the users of the restricted version. They performed slightly worse on the overall results of the exam. The difference in performance when working with LogEx was not statistically significant, see Table 9.

Because we did not have results of a pretest of the students who did not participate in the experiment, we cannot compare their results with the students who did participate. However, because we have their exam results, we can use these as a measure of the general level and compare the difference of this general level with the results on the rewriting items. Therefore, we normalize the results by dividing the total score by 10 and multiplying the score for the logic questions by 10/6, and subsequently, we look at the difference between these normalized scores. For example, a student with 60 points in total (normalized 6) and 5 points for the logic questions (normalized 8.3) scores 2.3 better on the logic question than expected. This logic score versus total score is normally distributed, and hence, we can use a one way analysis of variance test and post hoc tests to compare the

TABLE 10 Descriptive statistics of the difference between the results of the logic exercises and overall exam performance

Group	*n*	Mean	Std. dev	95% Confidence interval
N	43	−0.54	3.53	[−1.63 , 0.54]
F	30	1.75	2.92	[0.66, 2.84]
R	38	0.29	3.37	[−0.81, 1.41]
Total	111	0.36	3.42	[−0.27, 1.01]

TABLE 11 Post hoc comparison using Tukey HSD test of the difference of normalized scores from the logic exercises and the exam results for the three groups

Groups	Mean difference	Std. error	Sig.
N versus F	−2.30	0.79	.012
N versus R	−0.84	0.74	.49
F versus R	1.46	0.81	.18

differences. Again, Group F performs better than Group R, and Group R performs better than the students who did not participate. Table 10 shows the descriptives.

The effect of practicing with LogEx on the difference is significant, $F(2, 108) = 4.23$, $p = .017$. Post hoc comparisons using the Tukey HSD test indicate that Group F performs significantly better than Group N. The other comparisons do not show a significant difference, see Table 11.

This is an interesting result. It seems to indicate that in general, students have more problems with the logic questions than with the other questions of the exam but that after practicing with LogEx, this is the other way around.

5.3 | Results of the loggings

We analysed the loggings of LogEx to answer the question whether students learn while working with LogEx and to detect possible differences between the groups using the full and restricted version. The logging data consist of all the steps students take, all the hints, next steps, or worked solutions they ask for, together with time stamps. We analyse the loggings in various ways. We determine

- the number of mistakes students make over time, and whether or not this number decreases.
- what kind of mistakes students make.
- how many of the exercises students complete.
- the time students need to take a step, and whether or not this decreases the longer they work in LogEx.
- how long student solutions are compared with the solutions generated by LogEx.
- at what point students in Group F use hints and next steps.

In the rest of this section, we describe each of these aspects in detail.

Students may show progress by making fewer mistakes after practicing with LogEx for some time. However, this progress may not be present in the data, because the first exercises in LogEx are rather

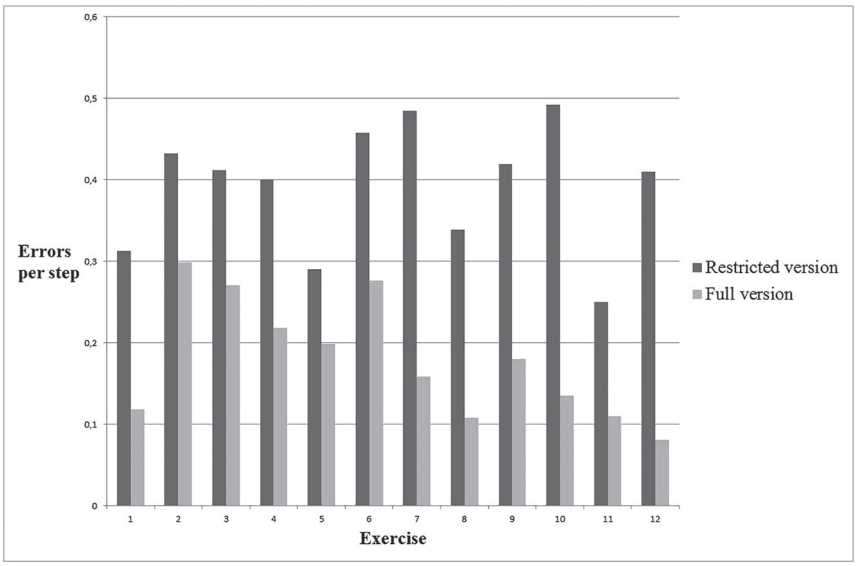

FIGURE 5 Errors per step for each exercise

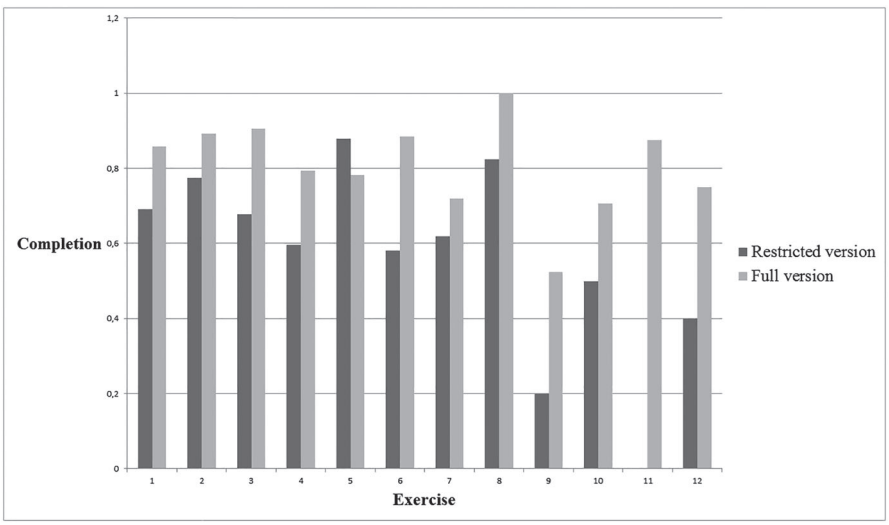

FIGURE 6 Completion per exercise

simple whereas the last exercises are more complicated and present a student with longer formulae. Figure 5 shows the number of erroneous steps per total number of steps for each exercise. Both groups make many more mistakes in the second exercise than in the first. Group F gradually makes fewer mistakes except for Exercises 6, 9, and 10 (see the Appendix for the list of exercises). The last two exercises require more complicated steps, which leads to more mistakes. In Exercise 6, students tend to perform more than one step at a time, which is not allowed. Group R does not make fewer mistakes while working with LogEx.

Further inspection of the loggings shows that students in Group R perform multiple steps at a time also in the other exercises, and they do this much more often than the students in Group F, probably because a student in Group F cannot proceed with an exercise after performing several steps simultaneously. The difference in the number of mistakes per step between the two groups is mainly due to these multiple steps error, but when we correct for these errors, the number of errors made by students in Group R still does not decrease while working with LogEx. Because students in Group F could not continue an exercise before correcting an error, these students might have been more careful when taking steps after some practice with LogEx.

We also examined the completion rate of exercises in our loggings. Here, we measure the percentage of students that complete an exercise from the number of students that started the exercise and took at least one step. The results are shown in Figure 6. In general, students from Group F complete more of the exercises than students from Group R, and this difference is larger in the more difficult exercises. This is in line with our findings in the pilot studies: students need hints and next steps to complete an exercise.

Another way to examine whether students learn to solve exercises while working with LogEx is by measuring the time it takes to perform a step. Obviously, practice makes perfect, and we expect that with practice, the time to perform a step decreases. In the first exercises, students familiarize themselves with LogEx, but we expect that while working with LogEx, they decide faster which rule to apply. Figure 7 shows the average step time per exercise. This varies per exercise

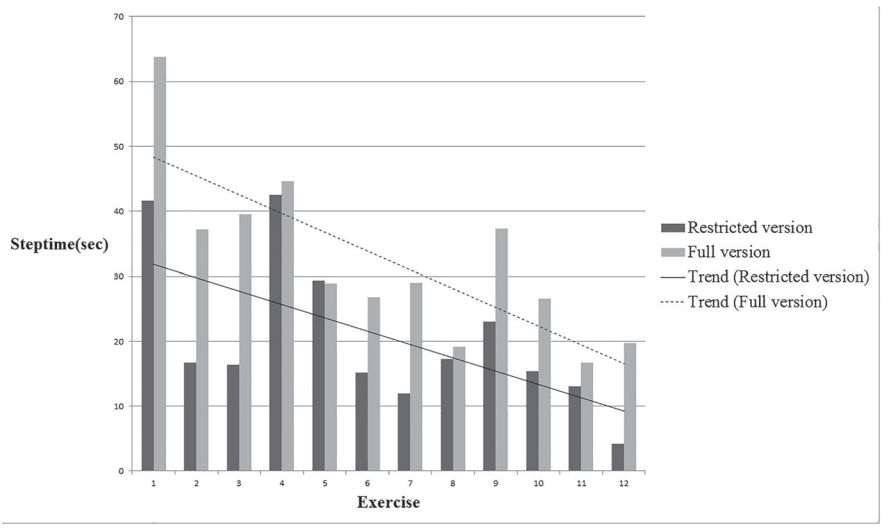

FIGURE 7 Average step time per exercise

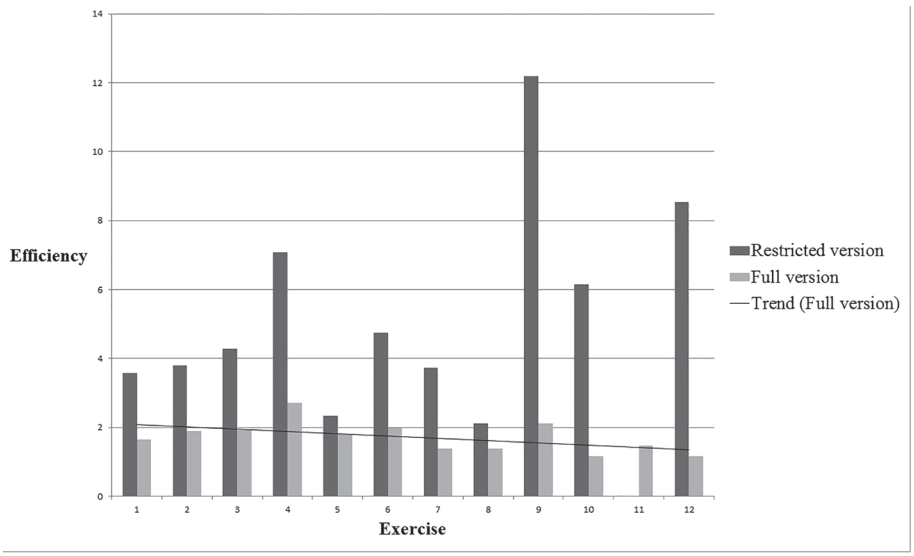

FIGURE 8 Efficiency measured by the number of performed steps as a fraction of the number of steps in a worked solution per exercise

(with outliers for the more complicated exercises), but the trend line suggests that students in both groups gradually solve exercises faster. Figure 7 suggests that after the first exercise, students have learned to use LogEx, and they complete the rather easy second and third exercises much faster. They need more time to solve the more complicated fourth exercise, after which the step time gradually decreases. This is in line with our pilot experiments.

We compared the efficiency of working with LogEx for both groups. Figure 8 shows the efficiency per exercise. The linear regression trend line indicates that over time, Group F learns to solve the exercises slightly more efficient. Because the use of hints or worked solutions can influence the efficiency, we also show the use of hints and next steps for Group F and worked solutions for Group R in Figures 10 and 9. These figures suggest that the apparent progress in efficiency is in fact a direct result of the increased use of hints or worked solutions. These results are consistent with our findings in the pilot studies. We hypothesize that practicing with LogEx for 50 min is too short to learn an efficient solving procedure, in particular because LogEx does not provide explicit strategic information. We expect that more and longer practice will help with the construction of efficient solution strategies. This is in line with findings from other studies, see Section 2. In the experiment with FOL, students practiced for 1 week, and

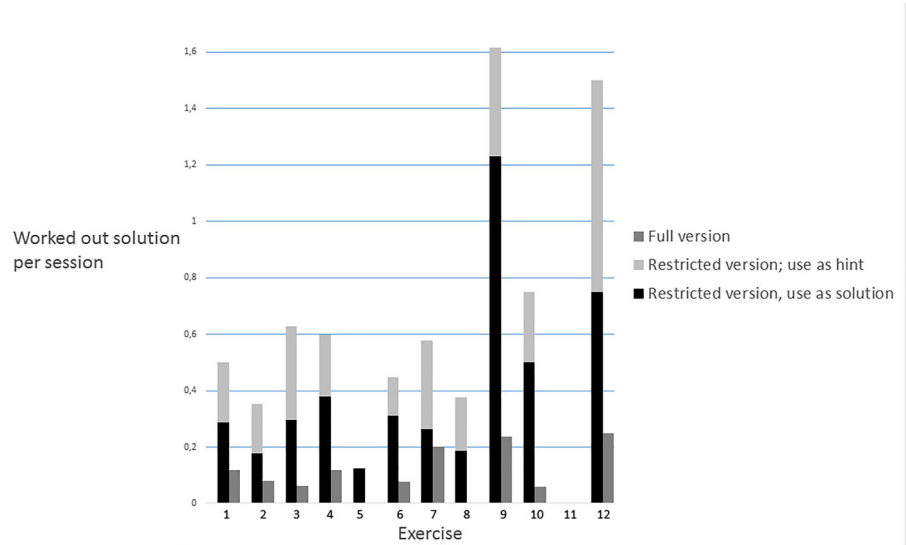

FIGURE 9 Use of worked examples in Groups R and F, Group R divided in hint use and solution use [Colour figure can be viewed at wileyonlinelibrary.com]

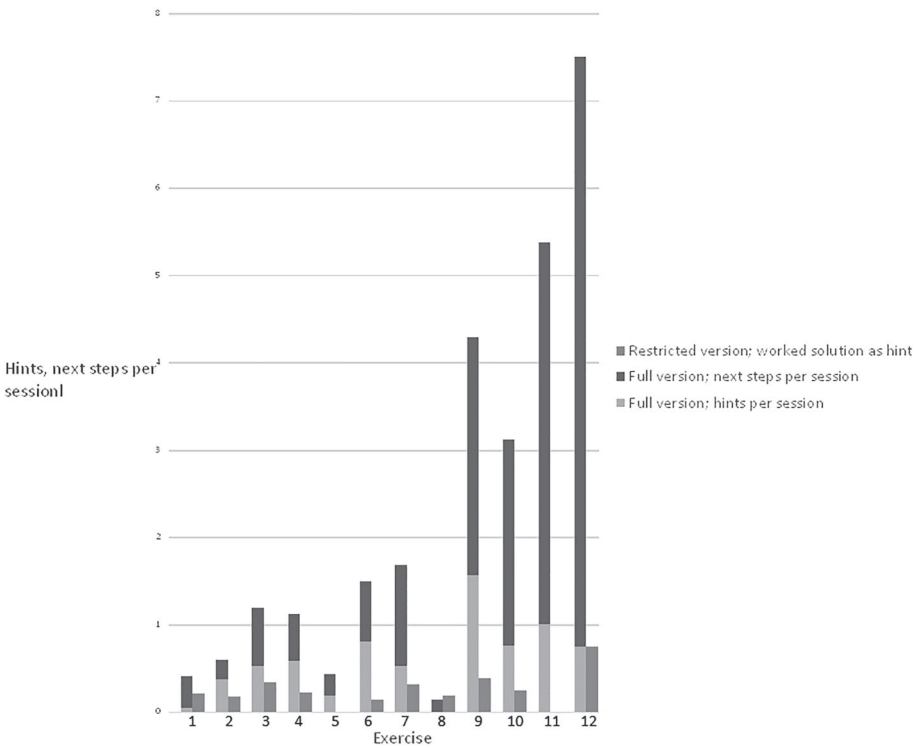

FIGURE 10 Use of hints and next steps in Group F compared with the hint use of worked examples in Group R

the evaluation studies of Logic Tutor and Deep Thought took several weeks. All these studies found significant results. The experiment with the natural deduction LE took only one session of 80 min, and only showed a significant difference on the easy exercises (Miwa et al., 2014). Students in Group R could ask for a complete solution at any moment. By counting the number of times that a worked out solution was directly followed by a student step, we found that students use this solution as a hint on how to proceed in about one third of the cases, see Figure 9. Figure 10 shows that although students in Group R often use the complete solution to obtain a hint, they still ask much less help than the students in Group F. Worked solutions also contain more information than a hint or a next step. Not only does a student receive all steps but possibly also a clue about the usefulness of a next step. Further research is necessary to determine whether a student can indeed extract this kind of information from a solution. Further inspection of the loggings shows that some students in Group F use the possibility to ask for a next step to obtain a worked solution. More than half of the next steps belong to sequences of next steps in which part of a worked solution is constructed. Another possible explanation for why students in Group F ask for more help is that they cannot proceed after a wrong step and ask the system for help in these situations. Interviews with students could show whether this is indeed the case, but the loggings already give an indication: a hint is asked twice as much after an incorrect step than after a correct step.

6 | CONCLUSION AND FUTURE WORK

We have performed an experiment in which we study the effect on student learning of using LogEx, an LE for proving the equivalence between two logical formulae using standard equivalences. Furthermore, we compare different ways to give support in LogEx. The experiment indicates that LogEx can be a helpful LE for students who practice rewriting logical formulae. We conclude that students do learn to prove the equivalence of two formulae. The number of mistakes they make while working with LogEx decreases, but they do not exhibit improved strategic insight. The exam results (4.1 points out of 6 on average for students using the full LE and 3.5 out of 6 for students using the restricted version) provide additional support for the conclusion that students reach the first two learning goals. Further research is needed to find out whether practicing with LogEx for a longer time improves strategic insight. Another way to improve strategic insight could be to provide strategic feedback when a student solution is longer than necessary. The loggings show that especially students who practice with the full LE hardly use the possibility to ask for a worked solution. When students in this group finish an exercise successfully, they do not compare their solution with a possibly shorter example solution. Although in general students learn more when they have to ask for help themselves (VanLehn, 2006), in this case, it might be necessary to let the system give help without being asked. Yet another way to improve LogEx could be by providing explicit strategic hints. LogEx recognizes when a student solution diverges from one of the possible paths determined by LogEx. In a next version, we might give a warning in such a case. In this way, LogEx would exploit the fact that it generates proofs from a strategy, in contrast with data-driven or example-based tutors such as Deep Thought (Stamper, Barnes, & Croy, 2011a; Mostafavi & Barnes, 2017).

The extra features of the full version of LogEx, namely, providing hints, next steps, and informative feedback after each step, do not have a significant effect on the exam results of students. Students using the full version perform slightly better, and on the exam, this group performed significantly better than a control group of students who did not practice with the tool. In a next experiment, we could measure the effects of informative timely feedback versus delayed feedback, and the effects of providing hints and next steps versus worked solutions separately. Because in both conditions in our experiment students could ask for a worked solution, they could use this solution as a hint. Therefore, the distinction between the two groups was less clear, with possibly negative effects on the significance of our results. The number of students in our experiment was too small to analyse whether there was a difference in effects on weak students or good students. This is also a question we would like to address in a follow-up study.

ACKNOWLEDGEMENTS

We thank students and teachers (in particular Hieke Keuning) for their cooperation in the experiment and permission to use their data for scientific purposes, and the Open University students Marco Huijben and WouterTromp for building an analysis tool, Maarten Hemker and Peter Dol for developing the first version of a user interface for our learning environment for proving equivalences, and René Dohmen and Renaud Vande Langerijt for the extension which incorporates exercises on rewriting to a normal form.

CONFLICT OF INTERESTS

Because one of the editors of JCAL is affiliated with the Open Univerisity Netherlands and the other with the Utrecht University, and the authors are affiliated with these institutions, there might be a conflict of interest.

ENDNOTES

1 http://ideas.cs.uu.nl/logex/
2 https://doi.org.10.17632/4wdj3b2t5g.1

ORCID

Josje Lodder https://orcid.org/0000-0003-0568-7844

REFERENCES

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, *4*, 167–207.

Bartsch, R. A., Bittner, W. M. E., & Moreno, J. E. Jr. (2008). A design to improve internal validity of assessments of teaching demonstrations. *Teaching of Psychology*, *35*(4), 357–359.

Chi, M. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, *1*(1), 73–105.

Grivokostopoulou, F., Perikos, I., & Hatzilygeroudis, I. (2013). An intelligent tutoring system for teaching fol equivalence. In *AIED Workshops*, pp. 20–29.

Hake, R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, *66*(1), 64–74.

Hattie, J. (2012). *Visible learning for teachers: Maximizing impact on learning*, Education (Routledge). London ; New York: Routledge.

Hayes, W. L. (1988). *Statistics* (4th ed.). Holt, Rinehart and Winston, Inc.

Kim, R., Weitz, R., Heffernan, N., & Krach, N. (2009). Tutored problem solving vs."pure" worked examples. In N. A. Taatgen, & H. van Rijn (Eds.), *Proceedings of the 31st annual conference of the cognitive science society*. Austin, TX: Cognitive Science Society, pp. 3121–3126.

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, *19*(3), 239–264.

Liu, Z., Mostafavi, B., & Barnes, T. (2016). Combining worked examples and problem solving in a data-driven logic tutor. In *Proceedings of the 13th International Conference on Intelligent Tutoring Systems*, Zagreb, Croatia, pp. 347–353.

Lodder, J., & Heeren, B. (2011). A teaching tool for proving equivalences between logical formulae. In Blackburn, P., Ditmarsch, H., Manzano, M., & Soler-Toscano, F. (Eds.), *Tools for teaching logic*, LNCS (Vol. *6680*, pp. 154–161). Springer-Verlag.

Lodder, J., Heeren, B., & Jeuring, J. (2015). A pilot study of the use of logex, lessons learned. CoRR, abs/1507.03671.

Lodder, J., Heeren, B., & Jeuring, J. (2016). A domain reasoner for propositional logic. *Journal of Universal Computer Science*, *22*(8), 1097–1122.

Lodder, J., Jeuring, J., & Passier, H. (2006). An interactive tool for manipulating logical formulae. In Manzano, M., Pérez Lancho, B., & Gil, A. (Eds.), *Proceedings of the Second International Congress on Tools for Teaching Logic*.

Lodder, J., Passier, H., & Stuurman, S. (2008). Using ideas in teaching logic, lessons learned. *Computer Science and Software Engineering, International Conference on*, *5*, 553–556.

Mathan, S. A., & Koedinger, K. R. (2005). Fostering the intelligent novice: Learning from errors with metacognitive tutoring. *Educational Psychologist*, *40*(4), 257–265.

Miwa, K., Terai, H., Kanzaki, N., & Nakaike, R. (2014). An intelligent tutoring system with variable levels of instructional support for instructing natural deduction. *Transactions of the Japanese Society for Artificial Intelligence*, *29*(1), 148–156.

Mostafavi, B., & Barnes, T. (2017). Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education*, *27*(1), 5–36.

Nachar, N. (2008). The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution. *Tutorials in Quantitative Methods for Psychology*, *4*, 13–20.

Narciss, S. (2008). Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*, *3*, 125–144.

Narciss, S. (2013). Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model. *Digital Education Review*, *23*, 7–26.

Narciss, S., Sosnovsky, S. A., Schnaubert, L., Andres, E., Eichelmann, A., Goguadze, G., & Melis, E. (2014). Exploring feedback and student characteristics relevant for personalizing feedback strategies. *Computers & Education*, *71*, 56–76.

Razzaq, L., & Heffernan, N. (2009). To tutor or not to tutor: That is the question. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED 2009*, pp. 457–464.

Razzaq, L., Heffernan, N. T., & Lindeman, R. W. (2007). What level of tutor interaction is best? In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts that Work*, IOS Press, Amsterdam, The Netherlands, The Netherlands, pp. 222–229.

Salden, R. J. C. M., Koedinger, K. R., Renkl, A., Aleven, V., & McLaren, B. M. (2010). Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review*, *22*(4), 379–392.

Shrestha, P., Maharjan, A., Wei, X., Razzaq, L., Heffernan, N. T., & Heffernan, C. (2009). Are worked examples an effective feedback mechanism during problem solving? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1876–1881.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, *78*(1), 153–189.

Shute, V. J., & Regian, J. W. (1993). Principles for evaluating intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, *4*(2-3), 245–271.

Stamper, J. C., Barnes, T., & Croy, M. (2011a). Enhancing the automatic generation of hints with expert seeding. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 153–167.

Stamper, J. C., Eagle, M., Barnes, T., & Croy, M. (2011b). Experimental evaluation of automatic hint generation for a logic tutor. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, AIED'11, Springer-Verlag, Berlin, Heidelberg, pp. 345–352.

Sweller, J., Ayres, P., & Kalyuga, S. (2011). The worked example and problem completion effects, *Cognitive load theory*. New York, NY: Springer, pp. 99–109.

Sweller, J., & Cooper, G. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*, 59–89.

van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology*, *36*(3), 212–218.

VanLehn, K. (2006). The behavior of tutoring systems. *Journal of Artificial Intelligence in Education*, *16*(3), 227–265.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221.

Yacef, K. (2005). The logic-ita in the classroom: A medium scale experiment. *International Journal of Artificial Intelligence in Education*, *15*(1), 41–62.

How to cite this article: Lodder J, Heeren B, Jeuring J. A comparison of elaborated and restricted feedback in LogEx, a tool for teaching rewriting logical formulae. *J Comput Assist Learn*. 2019;35:620–632. https://doi.org/10.1111/jcal.12365

APPENDIX

List of the exercises used in the experiment:

1. $\neg(p \wedge q) \vee s \vee \neg r \iff (p \wedge q) \rightarrow (r \rightarrow s)$
2. $p \wedge q \iff \neg(p \rightarrow \neg q)$
3. $(p \wedge q) \rightarrow p \iff \top$
4. $\neg(p \vee (\neg p \wedge q)) \iff \neg(p \vee q)$
5. $\neg(p \wedge (q \vee r)) \iff \neg p \vee (\neg q \wedge \neg r)$
6. $(p \rightarrow q) \vee (q \rightarrow p) \iff \top$
7. $\neg((p \rightarrow q) \rightarrow (p \wedge q)) \iff (p \rightarrow q) \wedge (\neg p \vee \neg q)$
8. $\neg(\neg p \wedge \neg(q \vee r)) \iff p \vee q \vee r$
9. $p \wedge (q \vee s) \iff (q \wedge \neg s \wedge p) \vee (p \wedge s)$
10. $(p \rightarrow q) \wedge (r \rightarrow q) \iff (p \vee r) \rightarrow q$
11. $(p \rightarrow \neg q) \rightarrow q \iff (s \vee (s \rightarrow (q \vee p))) \wedge q$
12. $p \rightarrow (q \rightarrow r) \iff (p \rightarrow q) \rightarrow (p \rightarrow r)$