Proving the equivalence of two logical formulae with LogEx

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

[1] Faculty of Science, Open University of the Netherlands, Heerlen, The Netherlands
[2] Dept. of Inf. and Comp. Sciences, Universiteit Utrecht, The Netherlands

Josje.Lodder@ou.nl

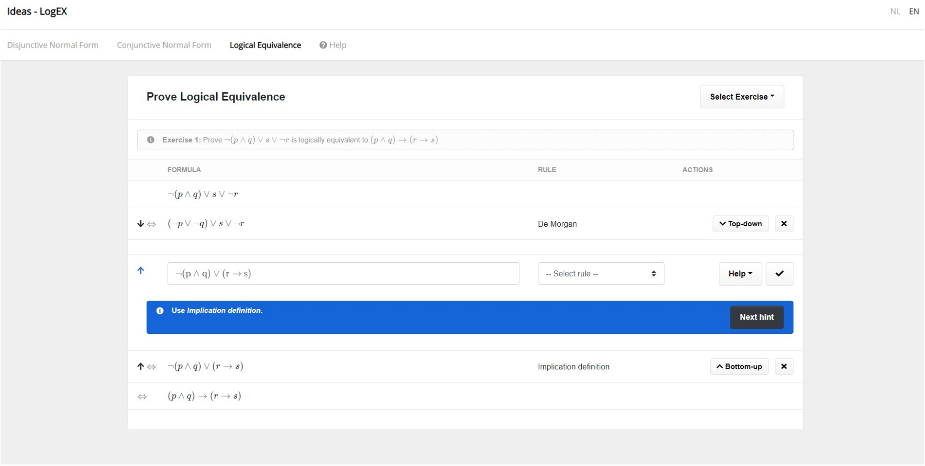
Summary

LogEx [7, 9] is an Intelligent Tutoring System (ITS) in which a student can practice the rewriting of propositional logical formulae using standard equivalences such as De Morgan's law or distribution. Automated feedback is calculated by the LogEx domain reasoner [8] for propositional logic, which provides domain knowledge, such as rewrite steps, common mistakes, and procedures for solving an exercise, for these exercise classes. The system is based on the Ideas framework [4], which supports the development of ITSs that automatically generate feedback and help for step-wise exercises, for a wide range of problem domains.

The LogEx system supports three different types of exercises: rewriting a formula in disjunctive normal form, in conjunctive normal form, and proving the equivalence of two formulae (see Figure 1). Students enter their solution stepwise, providing the next step and the corresponding rule name of the step, and receive feedback after each step. LogEx points out syntax errors, it recognizes a wrongly motivated but correct step, and it provides informative feedback if a common mistake occurs (a so-called buggy rule). An example of the latter might be the distribution of a conjunction over a conjunction. Although the result is an equivalent formula, this application of distribution is not correct.

Students can ask for a hint at any moment during their solution. The hints are layered. A first hint will only mention the rule that can be applied, a second hint will show the result of rewriting with this rule, and finally, LogEx can also perform this next step. These hints are also available when the student makes a detour. To get a worked example or to compare a student solution with the standard solution, LogEx can generate a complete solution. Students can use LogEx to solve their own exercises. LogEx will first verify that the exercise is correct, and can then provide the same feedback and help as for the exercises generated by the system.

Our aim is to let the way of working in LogEx resemble the pen-and-paper exercises as much as possible. Hence, the equivalence proof part of LogEx facilitates working in two directions. For example, a student can start rewriting the second formula, then switch to the first formula, and go back to the second. A first hint in this part of the system could also be to switch direction. The strategy used to solve these exercises is based on rewriting both formulae in normal



Prove Logical Equivalence

Exercise 1: Prove $\neg(p \wedge q) \vee s \vee \neg r$ is logically equivalent to $(p \wedge q) \rightarrow (r \rightarrow s)$

	FORMULA	RULE	ACTIONS
	$\neg(p \wedge q) \vee s \vee \neg r$		
↓ ⇔	$(\neg p \vee \neg q) \vee s \vee \neg r$	De Morgan	Top-down ✕
↑	$\neg(p \wedge q) \vee (r \rightarrow s)$	-- Select rule --	Help ✓
↑ ⇔	$\neg(p \wedge q) \vee (r \rightarrow s)$	Implication definition	Bottom-up ✕
⇔	$(p \wedge q) \rightarrow (r \rightarrow s)$		

Use *Implication definition*.

Fig. 1. LogEx provides a high-level hint for completing the logical equivalence proof

form, but contains several heuristics to provide shorter and smarter solutions, since we also want students to recognize the possibility for a smart solution.

LogEx has been evaluated in several experiments [9]. One of the results of these experiments was that students who practiced with LogEx scored significantly better on the examination, five weeks after the experiment, than students who did not participate, even when correcting for different skills in discrete mathematics.

Lodder et al. [8] provide a detailed comparison between LogEx and other tools for teaching logic, which builds on an overview of such tools collected by Huertas [6]. Related tools for proving equivalences are SetSails [5, 12], Logic Cafe[3], and the FOL equivalence system [3]. SetSails and Logic Cafe also support working in two directions. The distinguishing feature of LogEx, however, is that it can provide feed forward (i.e. next steps and complete solutions) in any situation. In contrast to using domain knowledge for generating feedback in LogEx, the Deep Thought Logic Tutor [2, 1] is data-driven and constructs (deductive) logic proofs and subgoal hints from student solutions.

LogEx is available online[4] as part of a broader tool set for supporting interactive exercises for basic logic courses in higher education. Using similar techniques, we have designed LogAx [10], which is a Hilbert-Style axiomatic proof tutor that can automatically generate hints and feedback, including next steps and reachable subgoals, also when the student diverges from the anticipated solution. LogInd [11] is a tutoring system for practicing the structural induction proof technique. We are currently working on making the logic tool set available for other instructors and courses.

[3] http://thelogiccafe.net/PLI/
[4] http://ideas.science.uu.nl/logex/

References

1. Barnes, T., Stamper, J.: Toward automatic hint generation for logic proof tutoring using historical student data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) Intelligent Tutoring Systems. pp. 373–382. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
2. Eagle, M., Johnson, M.W., Barnes, T.: Interaction networks: Generating high level hints based on network community clusterings. In: EDM. pp. 164–167 (2012)
3. Grivokostopoulou, F., Perikos, I., Hatzilygeroudis, I.: An intelligent tutoring system for teaching FOL equivalence. In: AIED Workshops (2013)
4. Heeren, B., Jeuring, J.: Feedback services for stepwise exercises. Science of Computer Programming, Special Issue on Software Development Concerns in the e-Learning Domain **88**, 110–129 (2014). https://doi.org/10.1016/j.scico.2014.02.021
5. Herding, D., Zimmermann, M., Bescherer, C., Schroeder, U., Ludwigsburg, P.: Entwicklung eines frameworks für semi-automatisches feedback zur unterstützung bei lernprozessen. In: DeLFI. pp. 145–156 (2010)
6. Huertas, A.: Ten years of computer-based tutors for teaching logic 2000-2010: Lessons learned. In: Proceedings of the Third International Congress Conference on Tools for Teaching Logic. pp. 131–140. TICTTL'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2021573.2021589
7. Lodder, J.: The Design and Use of Tools for Teaching Logic. Ph.D. thesis, Open University of the Netherlands (Sep 2020)
8. Lodder, J., Heeren, B., Jeuring, J.: A domain reasoner for propositional logic. Journal of Universal Computer Science **22**(8), 1097–1122 (2016)
9. Lodder, J., Heeren, B., Jeuring, J.: A comparison of elaborated and restricted feedback in LogEx, a tool for teaching rewriting logical formulae. Journal of Computer Assisted Learning **35**(5), 620–632 (2019). https://doi.org/10.1111/jcal.12365
10. Lodder, J., Heeren, B., Jeuring, J., Neijenhuis, W.: Generation and use of hints and feedback in a hilbert-style axiomatic proof tutor. International Journal of Artificial Intelligence in Education **31**, 99–133 (2021). https://doi.org/10.1007/s40593-020-00222-2
11. Lodder, J., Heeren, B., Jeuring, J.: Providing hints, next steps and feedback in a tutoring system for structural induction. In: Quaresma, P., Neuper, W., Marcos, J. (eds.) Proceedings of the 8th International Workshop on Theorem Proving Components for Educational Software (ThEdu'19). p. 17–34 (2020). https://doi.org/10.4204/EPTCS.313.2
12. Zimmermann, M., Herding, D.: Entwicklung einer computergestützten lernumgebung für bidirektionale umformungen in der mengenalgebra. Beiträge zum Mathematikunterricht 2010 (2010)