

# Using IDEAS in teaching logic, lessons learned

Josje Lodder, Harrie Passier, Sylvia Stuurman  
Open University Heerlen, the Netherlands  
Josje.Lodder@ou.nl

## Abstract

*At the Open University we are developing tools to support students in learning procedural skills. The tool for rewriting logical formulas into disjunctive normal form was tested with students in the period 2007-2008. The results of this tests help us to improve our tool, give answers to some questions we had during the development and encourage us to proceed.*

## 1. Introduction

In subjects as mathematics and logic, students have to learn to construct answers to exercises, using rewrite rules and strategies. In the IDEAS project at the Open University we are developing tools to support students in learning these skills by giving interactive and rich feedback.

([http://ideas.cs.uu.nl/wiki.index.php/Main\\_Page](http://ideas.cs.uu.nl/wiki.index.php/Main_Page))

One of these tools concerns the transformation of logical formulae into disjunctive normal form (DNF). In 2007 and 2008 we asked our students to experiment with this tool. We hoped to get answers to the following questions:

- How good is the feedback provided by the tool?
- How do students use the tool?
- Do students learn by using the tool?
- How do students appreciate the tool?

This paper is structured as follows. Section 2 introduces our tool for transforming logical formulas into DNF. Section 3 describes the tests with students, Section 4 describes the results of these tests. In section 5 we conclude and present our plans for the future.

## 2. An interactive tool for manipulating logical formulae

The logical exercise solver helps students to rewrite formulae from propositional logic into disjunctive normal form using standard equivalences. A typical

example of a formula the student can try to transform is:

$$(\neg p \rightarrow \neg q) \rightarrow (r \leftrightarrow \neg s)$$

Using the tool, students can solve this exercise as they would using pen and paper. Provided a student applies a single rule at a time, the tool checks that an expression submitted by a student can be derived from the previous expression by applying one of the standard equivalences.

Figure 1 shows a screenshot of our tool.



FIGURE 1 Screenshot of the tool

At each step the tool supplies feedback. This feedback depends on the kind of mistake the student might have made:

- the formula entered by the student is not well formed: in this case the error-correcting parser suggests a correction
- the formula entered by the student is not equivalent to the previous formula. Using a rewrite analysis and buggy rules (for example a variant of De Morgan rule without the changing of 'and' into 'or'), the tool tries to find a plausible rule the

student intended to use, and gives a correct application of this rule.

- the formula entered by the student is equivalent to the previous formula, but not derivable using a single rule. In this case this message is given to the student.

While designing the tool we had to decide how the tool should react in this last case: do we allow students to go on, or do we force them to apply one rule at a time. This last option was chosen. From a test with a comparable tool for linear algebra [1] it is known that in case of a simple exercise, students prefer to enter the answer without providing the intermediate steps. In our tool easier and more difficult exercises are generated at random, and up to now we don't use a measure for the difficulty of an exercise. However we expected that the weaker students would benefit from the obligation to provide all the steps.

The students not only have to learn to apply the rules correctly, they also have to learn which rules they have to apply to reach an answer (in this case a DNF). To support this, the student can ask for a hint (you can apply the double negation rule) or a step (replace  $\neg\neg p$  by  $p$ ).

Finally, the students have to recognize that the answer is reached. After pushing the Klaar (finished) - button, the tool checks whether the formula is indeed a DNF. If a student continues after reaching a DNF, the message: "you reached a DNF in the previous step" is given. Compared to existing tools teaching rewriting logical formulae (e.g. Organon [6], MLT-PC [7]) the feedback of our tool is much richer, and the way of working for the student is more natural, because students don't have to specify the rules used. A more extensive comparison of our tool with existing tools is given in [2]

### 3. Testing the tool with students

The Open University of the Netherlands is an institution for distance education. Students study at home and they don't have much contact with lecturers. We have used the tool for students computer science in a first course on discrete mathematics. This course contains a module on logic [3]. Four weeks before the exam, we asked registered students if they would participate in a test with our tool. Participating students had to make a pretest, which contained five types of exercises (recognizing a DNF, recognizing applicable rules, rewriting a formula to DNF, rewriting a formula into conjunctive normal form (CNF) and proving the

equivalence of two formulas). After receipt of the pretest we sent the student the url of the tool. Students could practice with the tool as long as they like, meanwhile we logged their use. Part of the students got the tool without the next step button. Afterwards we sent the students a posttest comparable to the pretest and a questionnaire. The objective of the evaluation was to get answers on the following questions:

- How good is the feedback provided by the tool?
- How do students use the tool?
- Do students learn by using the tool?
- How do students appreciate the tool?

## 4. Results of the tests

In total 23 students participated in the evaluation which took place in 2007 and 2008. All students made a pretest (they received the url after sending in this test), 13 students sent the questionnaire back and students made a 10 posttest. On average, the 13 students who returned the questionnaire practiced for 2 hours, making 20 exercises.

### 4.1. Quality of the feedback

We judged the feedback provided by the tool in two ways:

First we asked the students to rank the quality of the feedback on a five-point scale: the score on syntactic feedback (2.5; on the question: is the feedback correct 1 = yes, 5 = no) is somewhat less then on the rule feedback (score 1.9).

Second we analyzed the logfiles of the students and compared the feedback given by the tool with the feedback a teacher would give. In most cases ( $\pm 80\%$ ) the tool gives the expected feedback. Most reasons for wrong feedback can easily be repaired in our next version. For example, we defined a buggy rule concerning the commutativity of the disjunction and conjunction. In practice students never make mistakes using this rule, and so the firing of this rule gives wrong feedback, which can simply be resolved by deleting this rule. However one problem we didn't solve yet; this problem concerns the context used in our rewrite analysis. This analysis is based on the difference between the entered and the previous formula. However, to recognize the mistake this difference is not always sufficient. For example, when a student applies De Morgan and forgets to remove the outer negation, this outer negation is not part of the difference, but is essential for the recognition of the mistake. The tool doesn't recognize the application of De Morgan in this case.

One important observation we made is that wrong feedback can influence the learning of the student negatively: students tend to interpret the message they get as a hint. For example: the student tried to apply De Morgan rule, made a mistake and gets a message "you made a mistake applying distribution". In the next step the student will try to apply distribution even if applying De Morgan was correct according to the strategy. Also wrong suggestions from the error correcting parser (e.g. when a parenthesis is missing) may cause confusion which frustrates the student instead of helping her.

#### 4.2. Use of the tool

We asked some questions concerning the use of the tool to get some insight in the different learning styles. As expected the good students skipped more often a simple exercise, made less use of the hint and next step button, and incidentally used the next button when they knew how to perform the next step, but were too lazy to perform this step by themselves. Weak students didn't skip the simple exercises, but didn't skip the more complicated either, although they didn't complete them always.

#### 4.3. Learning effects

Learning effects were measured in three ways: by comparing pre- and posttest, by asking the students and by analyzing the logfiles.

The effects measured by the pre- and posttest are small, most students performed rather good on both tests. Recognizing a DNF appeared to be the most difficult question, we found no improvement on this skill in the posttest. In rewriting a formula to DNF we found some small improvements in accounting for the used rules, in precision (for example in the use of parentheses) and in using an effective strategy. The scores on rewriting a formula into CNF in pre- and posttest were more or less equal, which means that the tool didn't enforce the strategy on rewriting a formula into DNF too strong. Since both scores on proving equivalence of two formulas were good, we were not able to measure far transfer effects.

From the questions we asked the students, we learn

- using the next step button helps (score 1,7: yes = 1 no = 5)
- using the hint button helps less: (score 2,3 yes = 1 no = 5)

- after using these buttons for a while students can solve the exercises independently (score 1,7 yes = 1 no = 5)
- the feedback helps to recognize their mistakes (score 2 yes = 1 no = 5)
- they make less mistakes after practicing (score 2 yes = 1 no = 5)
- the tool helps to get understanding (score 2 yes = 1 no = 5)
- the tool helps to acquire skills (score 1,4 yes = 1 no = 5)

Analyzing the log files we found:

- The tool forces the student to be very precise in the use of parentheses.
- The next step button is essential for the weaker students. Students who got the version of the tool without next step were not able to complete exercises with more complex occurrences of the distribution rule.
- With this next step button students can finish the exercises. The next button also teaches them to use rules they overlook (e.g. false-true rules to simplify an exercise).
- The obligation to perform one step at a time forces students to recognize mistakes they would overlook otherwise (for example distributing and over and which results in an equivalent formula)
- Learning an efficient strategy is implicit in this version of the tool. Especially students who don't use the hint and next step button can proceed with inefficient strategies without receiving feedback on this aspect.

#### 4.4. Appreciation of the tool

As mentioned in the previous paragraph, students think the tool is helpful in acquiring understanding and skills. They ask for an extended version (for proving equivalences, for predicate logic, for free entering of formulae or for continuing the simplification after a DNF is reached). Especially the good students complain about the obligation to perform one step at a time.

#### 5. Conclusion and future work

With some improvements our tool will be a useful instrument to teach students in rewriting formulae into DNF. Designing a feedback tool should be done very carefully, since wrong feedback can cause confusion or can be misleading. Providing a next step is essential for students insofar that without this type of feedback they are not able to complete more complicated exercises. For beginning students the necessity to supply all the steps helps to recognize mistakes, but for more advanced students this is too time consuming. We will develop a variant of the tool where students can combine several steps. For those students who keep having difficulties in recognizing a DNF, we will develop a small tool for the training of this skill. The teaching of applying efficient strategies is in our tool only implicit. We are also working on tools which teach explicitly strategies to the student. [4, 5] In the future we will combine our tool providing feedback on rules with a tool providing feedback on strategy. Other plans concern an extension of the tool to proving equivalences and a tool for predicate logic. A tool to teach relation algebra will be released fall 2008.

## 6. References

- [1] Corbalan, G., F. Paas, H. Cuypers, "Overview of the results tests 1 and 2", 2008  
[http://ideas.cs.uu.nl/wiki/index.php/Project\\_documents](http://ideas.cs.uu.nl/wiki/index.php/Project_documents)
- [2] Lodder, J., J. Jeuring, H. Passier, "An interactive tool for manipulating logical formulae", *Proceedings of the Second International Congress on Tools for Teaching Logic (SICTTL)*, ed M. Manzano, Salamanca, 2006, pp. 93-99.
- [3] Thijsse, E. , *Logica in de praktijk*, Academic Service, Schoonhoven, 2000.
- [4] Heeren B. J. Jeuring, A van Leeuwen, A Gerdes, "Specifying strategies", 2008  
<http://www.cs.uu.nl/research/techreps/UU-CS-2008-001.html>
- [5] Jeuring J., H. Passier, S. Stuurman, "A generic framework for developing exercise", *Proceedings of the 8th International Conference on Information Technology Based Higher Education and Training (ITHET)*, Kumamoto City, Japan, 2008
- [6] Dostálová L, J. Lang, Organon - the web-tutor for basic logic courses" *Logic Journal of the IGPL 2007 15(4)*, pp. 305-311
- [7] Moreno A., N. Budesca, "Mathematical logic tutor propositional calculus", *Proceedings of the First International Congress on Tools for Teaching Logic (FICTTL)*, ed M. Manzano, Salamanca, 2000, pp. 99-106