Towards automatic evaluation of expediency of steps in algebraic transformations

Rein Prank
University of Tartu, Estonia
rein.prank@ut.ee

Usually problem solution environments do not check expediency of student's solution steps.

But in some topics this is not so hard to do.

- In this paper we consider solutions of algebraic exercises in propositional logic. For example, the exercise can require expression of formula
  $(A \sim C \vee \neg B) \supset \neg A \& B$
  using only negation and disjunction.

Example

Express $(A \sim C \vee \neg B) \supset \neg A \& B$ using negation and disjunction.
(Priority: $\neg$, &, $\vee$, $\supset$, ~)

$(A \sim C \vee \neg B) \supset \neg A \& B =$

$A \& (C \vee \neg B) \vee \neg A \& \neg (C \vee \neg B) \supset \neg A \& B =$

$\neg(\neg A \vee \neg (C \vee \neg B)) \vee \neg A \& \neg (C \vee \neg B) \supset \neg A \& B =$

$\neg(\neg A \vee \neg (C \vee \neg B)) \vee \neg (A \vee (C \vee \neg B)) \supset \neg A \& B =$

$\neg(\neg(\neg A \vee \neg (C \vee \neg B)) \vee \neg (A \vee (C \vee \neg B))) \vee \neg A \& B =$

$\neg(\neg(\neg A \vee \neg (C \vee \neg B)) \vee \neg (A \vee (C \vee \neg B))) \vee \neg (A \vee \neg B)$

Outline

1. Our exercise environment for algebraic manipulation exercises in propositional (and predicate) logic
2. New situation and new problems
3. Tool for evaluation of expediency of steps
4. Results of analyze
5. Conclusions

1. Exercise environment (1)

In our course Introduction to Mathematical Logic
(autumn term of second year)
most of exercises are solved on computer since 1991:
truth-table exercises,
algebraic manipulation using main equivalencies,
proofs in propositional and predicate calculus,
Turing machines

First version of exercise package was implemented in1988-1991
in MSDOS Turbo Pascal.

Algebraic manipulation environment was rewritten in 2003 (Java)

1. Exercise environment (2)
Solution step dialog

Each conversion step consists of two substeps.

1. **First substep:** the student marks a subformula.
2. **Second substep:** two different modes.

    2a) **INPUT mode:** the student enters a subformula that replaces the marked part,

    2b) **RULE mode:** the student selects a conversion rule from the menu and the program applies it.

Mainly:

Expression through given connectives: in INPUT mode

Normal forms: in RULE mode

Demo

1. Exercise environment (3)
Correctness checking

At first substep (marking):

1) Is the **marked part** a syntactically correct formula? ($\rightarrow$syntax error),
2) Is the **marked part** a proper subformula of whole formula? ($\rightarrow$ order error).

1. Exercise environment (4) Correctness checking

At second substep in RULE mode:

Is the selected rule applicable to marked subformula?

($\rightarrow$conversion error),

At second substep in INPUT mode:

1) Is the entered string a syntactically correct formula ?

($\rightarrow$syntax error),

2) Is it equivalent with marked part?

($\rightarrow$conversion error),

3) missed parentheses

($\rightarrow$order error),

1. Exercise environment (5)
Correctness checking

After pressing the **Answer button**:

Is the solution finished (according to final conditions of actual task type)?
($\rightarrow$answer error),

1. Exercise environment (6)
Summary of checking/nonchecking

The program **checks** that

1) every step preserves equivalence with initial formula,
2) final state is reached (or not).

The program **does not check** whether the **steps are reasonable** (for actual task type).

During many years such design was sufficient.

In 2003 we did not see reasons to change the design.

2. New situation and new problems (1)

After 2003 two changes occurred.

1. The **entrance number of students** of computer science grows year by year.
   The level of weaker students is now lower than earlier.
2. Some years ago we started teach
   the introductory part of propositional logic
   in the **first-term course**
   'Elements of Discrete Mathematics'

The program still be good for indication of direct errors.

But the weaker students need also checking of suitability of solution steps.

2. New situation and new problems (2)

Solution files of many students are

2 times bigger than necessary for expression exercises and
3-4 and even more times bigger for normal form exercises.

Two longest successful solutions

of disjunctive normal form task in test of autumn term 2011
were 263 and 269 steps

2. New situation and new problems (3)

Research question:

What do they do?

After that we can decide:

What monitoring should be added to main program?

What should be added to teaching?

2. New situation and new problems (4)
Tools for teachers?

The main program records and enables to view

1) the table of errors,
2) the solutions (rows with formulas, without indicating marked parts and applied rules)

Demo

It is hard for instructors to decipher and analyze big number of long solutions.

We need a tool that deciphers solution steps and collects statistics.

3. Tool for evaluation of steps (1)

Creating of evaluation tool was started from
normal form exercises
because

1) The problem was sharper: NF algorithm is complex while expression tasks have straightforward solution,
2) NF exercises were solved in rule mode where deciphering of solution steps is easier.

I will speak about NF exercises at TIME 2012 conference (Tartu, July 10-14)

3. Tool for evaluation of steps (2)

After implementation of NF-specific version we tried it on solutions of expression exercises.

Result was surprisingly good. Practically all solution steps of students are really applications of one single conversion rule. Exceptions: the student solved the task on paper and entered the final result.

Analyze tool for express-using-… exercises was received by

1) new evaluation of steps corresponding to conversion rules that express one connective through others,
2) adding the case where the task was solved in one step,
3) in cases where the step does not correspond to any single conversion rule – counting of numbers of concrete connectives

3. Tool for evaluation of steps (3)

When processing a particular solution, the tool displays the following data for **each solution step**:

1) initial formula and resulting formula with highlighted changed/resulting part,
2) the connectives that remain to be eliminated,
3) the rule that was actually applied by the student (if recognized),
4) OK or specification of irrationality.

The tool displays also

- statistics of steps and errors for each solution,
- statistics of errors for all solutions of solution file

Demo

3. Tool for evaluation of steps (4)

The analyze tool creates

1) For each solution file a text file with the same information that was displayed on the screen (analyze of all steps),
2) Summary file with statistics of student group (one line with step and error statistics for each solution)

4. Results of analyze (1)

Two data sets were analyzed:

1. File of one lab session +homework in spring term 2012.
   21 tasks on expression through given connectives.
   31 students.
2. Final test of Elements of Discrete Mathematics of autumn term of 2011.
   One task on expression of formula through
   $\{\&, \neg\}$, $\{\vee, \neg\}$ or $\{\supset, \neg\}$
   185 participants, 162 of them submitted solution files,
   136 solved/26 unsolved.

4. Results of analyze (2)

	Exercise session 2	Final test
Tasks	21	1
Students	31	162
Steps	**3439**	**1022**
Steps taken back	332	99
Suitability errors	**470**	**208**
Elem added (Double neg, parenth, …)	54	12
Duplicating of BAD oper by expression of ~	96	29
GOOD oper to BAD	40	47
BAD oper to BAD	236	81
NF conversions used	42	38

5. Conclusions

- Student's solution steps correspond to one single formal rule much more frequently than expected => diagnose of suitability errors is easier than expected.
- It is reasonable to add to exercise environment error messages that correspond to all suitability errors diagnosed by analyze tool
- Some students confuse propositional operations and types of tasks even at the final test
- It is reasonable to restore the Hint function (existed in our DOS version)
- It is reasonable to speak more about possible unsuitable steps