Author(s): Valmari, Antti

Title: Automated Checking of Flexible Mathematical Reasoning in the Case of Systems of (In)Equations and the Absolute Value Operator

Year: 2021

Version: Accepted version (Final draft)

Please cite the original version:

Valmari, A. (2021). Automated Checking of Flexible Mathematical Reasoning in the Case of Systems of (In)Equations and the Absolute Value Operator. In B. Csapó, & J. Uhomoibhi (Eds.), CSEDU 2021 : Proceedings of the 13th International Conference on Computer Supported Education. Vol. 2 (pp. 324-331). SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0010493103240331

Automated Checking of Flexible Mathematical Reasoning in the Case of Systems of (In)Equations and the Absolute Value Operator

Antti Valmari[a]
University of Jyväskylä, Faculty of Information Technology, PO Box 35, FI-40014 University of Jyväskylä, FINLAND
antti.valmari@jyu.fi

Keywords: High school mathematics, Elementary university mathematics, Automatic feedback, eLearning Software, Non-trivial reasoning patterns, Higher order thinking skills.

Abstract: We present an approach and a tool for automatically providing feedback on solutions that involve complicated reasoning patterns. Currently the tool supports linear systems of equations and inequations that may also contain the absolute value operator and a restricted form of rational functions. This suffices for designing problems that are laborious to solve with standard mechanical procedures, but much easier using short-cuts that students may find by creative thinking. Earlier research has found that struggling with important mathematics promotes conceptual development. Our goal is to encourage students to such struggling. A crucial feature is to give them great freedom to choose the paths via which they solve problems, and at any time ask the tool to check the work done so far, no matter what path was chosen. This was implemented by adopting standard notation from mathematical logic, and developing some new logical notation. The tool has been used in a course on elementary university-level mathematics. It has worked reliably, but there is not yet any statistics on the pedagogical merits. The tool is expected to also support quadratic (in)equations in the near future.

1 INTRODUCTION

The book "Adding It Up: Helping Children Learn Mathematics" by the National Research Council characterizes mathematical proficiency as consisting of five strands: *conceptual understanding*, *procedural fluency*, *strategic competence*, *adaptive reasoning* and *productive disposition* (Kilpatrick et al., 2001, p. 5). The extent to which the development of each of these can be supported by modern computer tools varies. Automatic answer checkers are widely used for developing procedural fluency. Conceptual understanding benefits from playing with tools that illustrate and facilitate experimenting with mathematical concepts.

The present study addresses the use of our computer tool to develop adaptive reasoning, defined by (Kilpatrick et al., 2001) as *capacity for logical thought, reflection, explanation, and justification.* (Anthony and Walshaw, 2009) observed that "Questions that have a variety of solutions or can be solved in more than one way have the potential to provide valuable insight into student thinking and reasoning." We believe that such questions also have the potential to develop the students' reasoning skills. To illustrate our idea, consider the following problem. It is from the famous book "How to Solve It" (Pólya, 1945).

Solve the following system of equations.

$$\left.\begin{array}{rcr} x+7y+3v+5u &=& 16 \\ 8x+4y+6v+2u &=& -16 \\ 2x+6y+4v+8u &=& 16 \\ 5x+3y+7v+\ \ u &=& -16 \end{array}\right\} \qquad (1)$$

Of course, this system can be solved by following blindly a mechanical procedure. For instance, one could eliminate variables one at a time by solving a variable from one equation and assigning the solution to the remaining equations. Doing this for x and the first equation produces the following:

$$\begin{array}{l} x \ = \ 16-7y-3v-5u \\ 8(16-7y-3v-5u)+4y+6v+2u \ = \ -16 \\ 2(16-7y-3v-5u)+6y+4v+8u \ = \ \ \ 16 \\ 5(16-7y-3v-5u)+3y+7v+\ \ u \ = \ -16 \end{array}$$

Carrying the solution through with this approach would consist of lots of tedious, error-prone, unmotivating work.

Pólya wanted the student *not* to take that approach. His book is about developing general mathematical problem solving skills. His intention was that the student should find clever short-cuts via which the system could be solved with much less work.

[a] https://orcid.org/0000-0002-5022-1624

For instance, one could start by adding the second original equation to the third, resulting in

$$10x+10y+10v+10u \;=\; 0 \tag{2}$$

Adding the first and last original equation yields

$$6x+10y+10v+6u \;=\; 0 \tag{3}$$

By subtracting (3) from (2) we get

$$4x+4u \;=\; 0$$

from which obviously $u=-x$. Assigning $-x$ in the place of u in (2) or (3) yields $10y+10v=0$, that is, $v=-y$. These facilitate immediate simplification of the first two original equations to

$$\begin{aligned} -4x+4y &= 16 \\ 6x-2y &= -16 \end{aligned}$$

Adding them gives $2x+2y=0$, implying $y=-x$. This together with $6x-2y=-16$ yields $8x=-16$. We conclude $x=-2$, $y=2$, $v=-2$ and $u=2$.

The idea is thus to give students problems that can be solved with little work if they use some creativity, but are laborious or impossible to solve by blindly following mechanical procedures. The present paper is about how to support this idea with automated feedback or automated checking of the student's solution.

Of course, traditional automatic checking of final answers can be employed to check that, for instance, in the above example, the student ended up with the correct values $x=-2$, $y=2$, $v=-2$ and $u=2$. However, as such, it does not suffice. Assume that the student made a sign error and, instead of (3), got

$$6x+10y+10v+6u \;=\; 32 \tag{4}$$

Continuing like above results in $x=-8$, $y=0$, $v=8$ and $u=0$. Traditional automatic checking can only tell that it is incorrect and reveal the correct answer. This does not give the student much indication of where the error took place. Furthermore, the student has spent a lot of effort in progressing from (4) to the incorrect final answer. This work is now wasted, which is frustrating.

We have developed a tool with which the student can check each step immediately after writing it (Kaarakka et al., 2019; Valmari and Rantala, 2019). If the student asks the tool to check (4), the tool tells that it is incorrect. So the student becomes immediately aware of the problem and can focus on fixing it, instead of first doing a lot of futile work in developing an incorrect final answer. We are aware of no other tool that supports such step-by-step checking of both arithmetic and logical reasoning.

The students have full control. They choose the paths via which to solve the problems: which variable to solve first and from which equation; or not to start like that at all and instead add equations to each other like was done above; or do something else. They choose what intermediate steps to write and what to perform only in their heads. They choose when to ask the tool to check what they have written.

A student that is fluent with a problem may proceed in big steps. If the tool accepts the step, great; and if not, then the student may add sub-steps to make the tool tell the location of the error more precisely. A newcomer may proceed in very small steps, pressing the check button after each step.

The tool is not restricted to systems of linear equations. Currently it also supports $\leq$, $<$, $\geq$, $>$, $\neq$, absolute values, a restricted form of rational functions, and a range of logical symbols including quantifiers. Univariate quadratic rational inequations with absolute values are being implemented. The tool works by using advanced algorithms from mathematical logic to check whether each step written by the student is logically correct or not. In many cases this means that the tool need not know in advance anything about the problem that the student is solving. As a consequence, a teacher may give students new problems without feeding in any model solution or similar. This is the case, for instance, with the problem (1) by Pólya.

Alternatively, the teacher may give a mathematical formulation of a problem to the tool, and present the problem to the students in natural language. Then it is a part of the task of the students to translate the problem from natural language to a mathematical representation. The formulation need not be precisely the same as the teacher's; it suffices that it is logically equivalent.

Of course, for this to work, there must be a notation in which the students express their reasonings. The tool uses symbols from mathematical logic and a small number of additional symbols. The latter are needed to express non-trivial patterns of reasoning in a practical way.

The features described in this paper have been implemented only recently. They have been technically tested and many of them have been used in a real-life course. There is, however, not yet any systematic statistics from the pedagogical point of view.

In the next section we introduce the basic concepts of our approach, only using a straightforward reasoning pattern. Section 3 discusses various kinds of feedback that our tool gives. The topic of Section 4 is how to express complicated claims such as systems of equations. Reasoning patterns that are not straightforward are illustrated in Section 5. The section contains the solution to (1) developed above, in a form accepted by our tool. Beyond the most elementary

```
|2x-1| = 5
<=> 2x-1 = -5 \/ 2x-1 = 5
<=> x = -2 \/ x = 3
```

Figure 1: An example input by the student.

$$|2x - 1| = 5$$
$$\Leftrightarrow\ 2x - 1 = -5 \vee 2x - 1 = 5$$
$$\Leftrightarrow\ x = -2 \vee x = 3$$

Correct!

Figure 2: The output by the tool for the input in Figure 1.

notions, the notation system for logic has not become standardized. In Section 6 we compare our notation to those found in textbooks. Our concluding remarks are in Section 7.

2 BASIC CONCEPTS

The key technical concepts in this paper are *formulas* and *reasoning steps*. Figure 1 shows a reasoning that consists of three formulas that have been organized into two reasoning steps. It has been written in the input language of our tool. Figure 2 shows the response by our tool to the input in Figure 1. It consists of a copy of the input translated to standard mathematical notation, followed by the verdict that the reasoning was correct.

The first formula in Figure 1 (or Figure 2) expresses the equation $|2x-1| = 5$. We call it the *original problem*, because it is what the student is expected to solve. The *roots* of an equation are the values of the unknown that make the equation hold. For instance, the roots of $|2x-1| = 5$ are -2 and 3.

The last formula of a reasoning is called the *final answer*. In this case it is $x = -2 \vee x = 3$. It says that x is -2 or x is 3 ($\vee$ denotes the logical or). Because it lists the roots as the possible values of x, it is correct.

The second formula $2x-1 = -5 \vee 2x-1 = 5$ is an *intermediate result*. A reasoning that solves a difficult problem may contain many intermediate results.

In general, a formula expresses a claim about the values of the variables in question. As was illustrated above, the original problem is represented as a formula, the final answer is represented as a formula, and each intermediate result is represented as a formula.

In this section, we say that a *formula is correct* if and only if it is true exactly when the original problem is true. (This becomes more complicated in Section 5.) For instance, every formula in Figure 2 is true when x is -2 or 3, and everyone is false when x has any other value, so they are all correct. This notion is different from "tautology" and "valid formula" in mathematical logic (they mean, roughly speaking, that the formula is always true). The original problem is correct by its nature, but intermediate results and the final answer may be incorrect. It is the task of our tool to check whether they are correct.

A reasoning step consists of two formulas and a *reasoning operator* that relates them. Until Section 5, we only use one reasoning operator $\Leftrightarrow$, meaning that the formula on its left hand side is true exactly when the formula on its right hand side is true. If this holds, we call the reasoning step correct, and otherwise incorrect. For instance, if $x = -2$, then both $|2x-1| = 5$ and $x = -2 \vee x = 3$ are true; if $x = 3$, then both are true; and if x has any other value, then both are false (and thus not true). Therefore, $|2x-1| = 5 \Leftrightarrow x = -2 \vee x = 3$ is a correct reasoning step.

A *counter-example* to an incorrect $\Leftrightarrow$-step is values of the unknowns such that one side of the step is true and the opposite side is not true. Every incorrect and no correct step has a counter-example. For instance, $|2x-1| = 5 \Leftrightarrow x = 3$ has the counter-example $x = -2$, because it makes $|2x-1| = 5$ true (indeed, $|2 \cdot (-2) - 1| = 5$) but $x = 3$ false (because $-2 \neq 3$).

Successive reasoning steps may share a formula. Figure 1 (and 2) shows two reasoning steps that share the formula $2x-1 = -5 \vee 2x-1 = 5$.

If every reasoning step in a reasoning of the form "original problem $\Leftrightarrow$ formula $\Leftrightarrow$ ... $\Leftrightarrow$ formula" is correct (in the sense of correctness of $\Leftrightarrow$-steps just introduced), then every formula in the reasoning is correct (in the sense of correctness of formulas introduced earlier on). This property is known as *transitivity*.

More reasoning operators will be introduced in Section 5. Each of them has its own notion of correctness.

3 FEEDBACK BY THE TOOL

Figure 3 shows an example of the feedback by the tool to an incorrect input. It says that if $a = -3$, then $|a-3| = 2a$ is false (F) but $a-3 = -2a \vee a-3 = 2a$ is true (T). Indeed, if $a = -3$, then $|a-3| = |-6| = 6$ but $2a = -6$, so $|a-3| = 2a$ is false. Furthermore, $a-3 = 2a$ is true, making $a-3 = -2a \vee a-3 = 2a$ true. That is, the tool gave a counter-example to the incorrect reasoning step. We see that $a-3 = 2a$ may hold such that $2a$ is negative, and then $|a-3| = 2a$ does not hold because |anything| is non-negative. We will discuss a correct solution in Section 4.

According to (Hiebert and Grouws, 2007, p. 383, 387), struggling with important mathematics is a key

$$|a-3| = 2a$$

$$\Leftrightarrow\ a-3 = -2a \vee a-3 = 2a$$

Relation does not hold when $a = -3$

left $=$ F
right $=$ T

Figure 3: The output by the tool for an incorrect input.

feature that promotes conceptual development. They emphasize that struggling does not mean "needless frustration" in front of an impossible task. It means that "students expend effort to make sense of mathematics, to figure something out that is not immediately apparent." We believe that in many cases, a counter-example may be a good starting point for such struggling. For instance, the counter-example in Figure 3 may lead the student to realize that when $|\ldots|$ is replaced by $\ldots$ or $-(\ldots)$, it may be necessary to ensure that the substitute is non-negative.

Sometimes every existing counter-example is a root of the original problem. This can be intentionally caused by giving F as an intermediate result. Then the counter-example reveals the correct final answer or a part of it. This is not a big issue, if the intention of the student is not to cheat but to learn. Furthermore, it is possible to switch off the printing of counter-examples.

It may also happen that incorrect thinking leads to a correct result by accident. Consider again $|a-3| = 2a$. A student may plan to write $a-3 = -2a \vee a-3 = 2a$, write $a-3 = -2a$, and be interrupted by a phone call. After the call (s)he no longer remembers what (s)he was doing. There is $a-3 = -2a$ on the screen, so (s)he may continue by solving it. Altogether (s)he wrote $|a-3| = 2a \Leftrightarrow a-3 = -2a \Leftrightarrow a = 1$.

The tool accepts this as correct, because the same solution could have arisen from correct thinking: An experienced student sees quickly that the root of $a-3 = 2a$ is $a = -3$. However, it cannot be a root of the original equation, because it makes $2a$ negative. Therefore, there is no need to write a branch for the case where $a-3 = 2a$.

Of course, the tool rejects $|a-3| = 2a \Leftrightarrow a-3 = 2a$. If the student solves many and diverse enough exercises, the odds are that most misunderstandings are caught sooner or later, although not necessarily at the first time when the student makes the error.

Figure 4 shows the reply by the tool when the student has not developed the answer to an *explicit* form. The formula $x = -2 \vee 2x = 6$ is correct, that is, it is true precisely when the original formula $|2x-1| = 5$ is true. However, the root $x = 3$ is not directly shown in the formula, but only indirectly via $2x = 6$. Because the answer is mathematically correct, and the problem is only that the student has not simplified it far enough, the tool shows the remark in magenta instead of red.

$$|2x-1| = 5$$

$$\Leftrightarrow\ 2x-1 = -5 \vee 2x-1 = 5$$

$$\Leftrightarrow\ x = -2 \vee 2x = 6$$

x has not been solved.

Figure 4: An incompletely solved example.

In the most typical case, an explicit form is $x = c_1 \vee \cdots \vee x = c_n$, where x is the unknown and the c_i are constants. Also the opposite direction $c_i = x$ is allowed for the reason discussed soon. If the equation has no roots (e.g., $x = x+1$), then the explicit form is F, and if every real number is a root (e.g., $x+1 = x+1$), then it is T.

The set of roots may also contain intervals. This is obvious in the case of inequations (e.g., Figure 5), but may also occur with equations using the absolute value operator (e.g., $|x| = x \Leftrightarrow x \geq 0$). Therefore, also $x \geq c_i$, $c_i < x \leq c_i'$, etc., are treated as being in the explicit form. Alternatively, $c_i < x \leq c_i'$ can be written using the logical and as $c_i < x \wedge x \leq c_i'$.

Figure 5 was obtained with the input

```
(2x+1)/(5x-2) >= 0 <=> x <= -1/2 \/ x >= 2/5
```

If the latter >= in the input is replaced by >, the tool accepts the input as correct.

Figure 5 also illustrates that our tool uses a third truth value "undefined" (U). This is a long story, but let us quickly mention some most important aspects. When $x = \frac{2}{5}$, $\frac{2x+1}{5x-2}$ is undefined, because the divisor $5x-2$ is 0. (In)equalities whose one or both sides is undefined must not be treated as true, because otherwise the corresponding value of the unknown would be a root. Therefore, with only F and T available, both $\frac{2x+1}{5x-2} \geq 0$ and $\frac{2x+1}{5x-2} < 0$ should be F when $x = \frac{2}{5}$. On the other hand, we would expect $\neg(\frac{2x+1}{5x-2} \geq 0)$ to be $\frac{2x+1}{5x-2} < 0$, and $\neg$F is T ($\neg$ is the logical not). So $\frac{2x+1}{5x-2} < 0$ should be both F and T when $x = \frac{2}{5}$, which is impossible.

$$\frac{2x+1}{5x-2} \geq 0\ \Leftrightarrow\ x \leq -\frac{1}{2} \vee x \geq \frac{2}{5}$$

Relation does not hold when $x = \frac{2}{5} \approx 0.4$

left $=$ U
right $=$ T

Figure 5: Intervals of roots, and division by zero.

$$|2x-1|=5 \;\Leftrightarrow\; x=3 \lor x=\;-2 \lor x=3$$

The final expression must be in a more simplified form.

Figure 6: An unnecessarily complicated final answer.

This obstacle is removed by treating $\frac{x}{0} \geq 0$ neither as true nor as false, but as undefined. The negation of undefined is undefined as well. The law $\neg(x \geq y) \Leftrightarrow x < y$ remains valid even if x or y is undefined. "Formula1 $\Leftrightarrow$ formula2" still means that formula1 is true precisely when formula2 is true. This works even if formula1 yields U and formula2 yields F, because neither U nor F is T. Because $\Leftrightarrow$ resides not in but between formulas, it cannot be within the scope of $\neg$.

Currently the tool makes some attempts to check that the final answer is not more complicated than necessary. An example of this is shown in Figure 6. However, the tool could do much more towards this end than it currently does. This is work in progress. If the teacher is willing to touch the tool, (s)he can set an upper bound to the length of the final answer. This has proven a very efficient method of catching too complicated final answers.

4 USE OF LOGICAL AND

The equations in a system of equations are intended to hold simultaneously. This can be obtained by combining them with the logical and $\land$. Similarly, the subformulas that tell the simultaneous values of different unknowns are combined with $\land$. Figure 7 shows an example of both of these. The figure also illustrates that if a system with two unknowns has more than one root, the alternative roots are separated with $\lor$.

Let us return to the example in Figure 3. By definition, $|x| = x$ if x is positive or zero, and $|x|$ is $-x$ if x is negative. Therefore, the (or at least a) correct method to solve $|a-3| = 2a$ is to split it to two cases. In one case $a-3$ is positive or zero, and $a-3$ is used in the place of $|a-3|$. It can be written as $a-3 \geq 0 \land a-3 = 2a$. Solving $a-3 = 2a$ yields $a = -3$. However, if $a = -3$, then $a-3 \geq 0$ does not hold, so -3 is not a root in the end. In the other case $a-3$ is negative, and $-(a-3)$ is used in the place of $|a-3|$. That is, $a-3 < 0 \land -(a-3) = 2a$. We have $-(a-3) = 2a \Leftrightarrow a = 1$. If $a = 1$ then $a-3 < 0$ holds, so 1 indeed is a root. In conclusion, $|a-3| = 2a \Leftrightarrow a = 1$.

$$|x|+y=6 \land 3x+5y=22$$

$$\Leftrightarrow\; x=4 \land y=2 \lor x=\;-1 \land y=5$$

Correct!

Figure 7: Two examples of the use of $\land$.

$$|a-3|=2a$$

$$\Leftrightarrow\; a-3 \geq 0 \land a-3=2a \lor a-3<0 \land\; -(a-3)=2a$$

$$\Leftrightarrow\; a-3 \geq 0 \land\; -3=a \lor a-3<0 \land 3=3a$$

$$\Leftrightarrow\; a-3 \geq 0 \land a=\;-3 \lor a-3<0 \land a=1$$

$$\Leftrightarrow\; a=1$$

Correct!

Figure 8: A clumsy solution to $|a-3| = 2a$.

This reasoning can be expressed with $\land$, $\lor$ and $\Leftrightarrow$, but only in a clumsy fashion. Figure 8 shows it. On the second, third and fourth line, the two cases are combined with $\lor$. The two equations $a-3 = 2a$ and $-(a-3) = 2a$ are solved in lockstep, although it would be more convenient to solve them one at a time. The problem is that with no other reasoning operator available than $\Leftrightarrow$, each intermediate result must reflect the original problem in full; that is, it is not possible to say that for a while, we focus on one aspect of the original problem and ignore the rest. In the next section, we introduce notation that removes this restriction and facilitates handy representation of many reasoning patterns.

Our tool also knows the symbols $\neg$ (not), $\rightarrow$ (if ... then) and $\leftrightarrow$ (if and only if). For instance, the tool accepts the reasoning step $\neg(x < 0) \Leftrightarrow x \geq 0$. The second line in Figure 8 can also be represented as $(a-3 \geq 0 \rightarrow a-3 = 2a) \land (a-3 < 0 \rightarrow -(a-3) = 2a)$, and some people find it more natural.

5 MORE REASONING PATTERNS

In the previous section we only used a straightforward type of reasoning, where the formulas are separated with $\Leftrightarrow$, and each formula must be true precisely when the original problem is true. In this section we discuss more general patterns of reasoning. They are important for giving the student the freedom to choose the path via which to solve the problem.

The key concept is logical implication, expressed with $\Rightarrow$ and in reverse direction with $\Leftarrow$. The rigorous definition is that "formula1 $\Rightarrow$ formula2" is a correct reasoning step if and only if all those value combinations of the unknowns that make formula1 true, also make formula2 true. For instance, Figure 9 tells that $|x| + y = 6 \land 3x + 5y = 22 \Rightarrow x = 4 \land y = 2$ is incorrect because of the counter-example $x = -1 \land y = 5$, but the opposite direction is correct.

There is evidence that the asymmetry of $\Rightarrow$ is very difficult for many students. For instance, the ACM & IEEE curriculum recommendation specifies the corresponding topic with exceptional detail: "Notions

$|x| + y = 6 \land 3x + 5y = 22 \;\Rightarrow\; x = 4 \land y = 2$

Relation does not hold when $x = -1$ and $y = 5$

left $=$ T
right $=$ F

$|x| + y = 6 \land 3x + 5y = 22 \;\Leftarrow\; x = 4 \land y = 2$

Correct!

Figure 9: An incorrect implication whose converse is correct.

of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction" (ACM and IEEE., 2013, p. 78). A hopefully helpful way to think of it is that if and only if "formula1 $\Rightarrow$ formula2", then formula2 contains the same or strictly less information about the values of the unknowns than formula1. For instance, $3 < x \leq 5$ contains less information about the value of x than $2 \leq x \leq 4$, because the latter specifies the value of x more precisely.

Consider again (1), with $\land$ added to the end of its first three lines. Intuitively, picking two equations from among the four amounts to throwing some information away and keeping the rest. Therefore, we should have "all four in (1) $\Rightarrow$ the 2nd and 3rd in (1)". The first two lines of Figure 10 show that our tool is of the same opinion. (The figure also illustrates that the student may write comments in a solution, and the tool copies them to the output in brown colour.)

As a matter of fact, "formula1 $\Rightarrow$ formula2" applies to any reasoning that does not use other situation-specific information than what is in formula1. (General laws such as $x + 0 = x$ are not situation-specific information.) Adding two equations only uses information in those equations (in addition to general laws), so also the third line in Figure 10 is correct.

The fourth line uses the keyword `original` to handily refer to the original problem. It derives the sum of the first and fourth original equation without showing any intermediate results. Because the formulas on the third and fourth line follow from the original problem, also the result of subtracting one from the other does. This is shown on the fifth line, followed by the solving of u in terms of x.

The solution described in Section 1 could be carried all the way through only using `original`, $\Rightarrow$ and $\Leftrightarrow$, but we chose to use also other features, to illustrate them. An arbitrary reasoning (that is within the capacity of the tool) may be enclosed between `subproof` and `subend`. The first such instance in Figure 10 takes the results from the third and fifth line, and reasons $v = -y$ from them.

The second instance starts by lifting $u = -x \land v = -y$ to the status of a general law that is valid throughout the instance. This makes it possible to compare $x + 7y + 3v + 5u = 16 \land 8x + 4y + 6v + 2u = -16$ to $-4x + 4y = 16 \land 6x - 2y = -16$ with $\Leftrightarrow$ instead of $\Rightarrow$, although only the former uses u and v. As a consequence, the tool checks fully the correctness of the elimination of u and v, while $\Rightarrow$ would have allowed, for instance, $-4x + 4y = 16$ as such (that is, without $6x - 2y = -16$) on the right hand side.

This is because $\Rightarrow$ allows throwing *any* information away. It is used to intentionally throw *some* information away. However, the logic and the tool do not know which pieces of information were thrown away intentionally and which accidentally, so $\Rightarrow$ must allow throwing any information away.

This issue does not make the tool accept incorrect final answers. If the reasoning as a whole fails to establish the $\Leftrightarrow$-relation between the original problem and the final answer (for instance, only establishes the $\Rightarrow$-relation), then the tool says in magenta that "Implication was used without returning to the original". However, it can cause the error message occur much later than where the actual error was, potentially confusing students.

The second subproof in Figure 10 continues in an already seen fashion to the value of y in terms of x, and to the numeric value of x. Here "original" refers to the first formula of the subproof after the assumption. Having derived $u = -x$, $v = -y$, $y = -x$ and $x = -2$, the final answer is easy to write.

We have now presented, in a form that our tool was able to check, the reasoning in Section 1 that solved Pólya's problem via short-cuts. The reasoning could be made shorter still by replacing the subproofs with `original` $\Rightarrow$ -structures, but we chose to use this opportunity to illustrate subproofs and assumptions.

Let us discuss some further examples that illustrate other aspects of solving equations. Figure 11 shows an example of using assumptions to express the conditions of the cases of an absolute value operation. In the first branch, thanks to the assumption $x \geq 0$, we have $|x| = x$ and thus $|x| + y = 6 \Leftrightarrow x + y = 6 \Leftrightarrow x = 6 - y$. Similarly in the second branch $|x| + y = 6 \Leftrightarrow -x + y = 6 \Leftrightarrow x = y - 6$. The advantage of this solution is that it avoids $\Rightarrow$ altogether. It thus avoids the risk of postponed error messages mentioned above.

A solution by cases can be written without subproofs by starting the first case with $\Leftarrow$ and the remaining cases with `original` $\Leftarrow$. This works because each case contains *at least* the same information as the original problem: all (in)equations in the original problem apply, and also the (in)equation that specifies the case applies. Analogously to $\Rightarrow$, $\Leftarrow$ runs the risk of accidentally putting too much information to a case, causing an error that the tool detects later than

$x + 7y + 3v + 5u = 16 \wedge 8x + 4y + 6v + 2u = -16 \wedge 2x + 6y + 4v + 8u = 16 \wedge 5x + 3y + 7v + u = -16$

$\Rightarrow 8x + 4y + 6v + 2u = -16 \wedge 2x + 6y + 4v + 8u = 16$ Just copied the 2nd and 3rd equation.

$\Rightarrow 10x + 10y + 10v + 10u = 0$ Added the previous two equations.

Original $\Rightarrow 6x + 10y + 10v + 6u = 0$ Added 1st and 4th original equations.

Original $\Rightarrow 4x + 4u = 0 \Leftrightarrow u = -x$ Subtracted the previous from its previous and solved u.

Subproof Solve v from the above results.

$10x + 10y + 10v + 10u = 0 \wedge u = -x \Rightarrow 10y + 10v = 0 \Leftrightarrow v = -y$

Subend

Subproof Eliminate u and v from the 1st and 2nd original. Solve y then x.

Assume $u = -x \wedge v = -y$

$x + 7y + 3v + 5u = 16 \wedge 8x + 4y + 6v + 2u = -16$

$\Leftrightarrow -4x + 4y = 16 \wedge 6x - 2y = -16 \Rightarrow 2x + 2y = 0 \Leftrightarrow y = -x$

Original $\Rightarrow 6x - 2(-x) = -16 \Leftrightarrow x = -2$

Subend

Original $\Leftrightarrow x = -2 \wedge y = 2 \wedge v = -2 \wedge u = 2$ Final answer picked from the intermediate results.

Correct!

Figure 10: Solving Pólya's problem (1).

where it actually occurred.

With the next version of the tool, $\frac{x^2-9x+18}{x^2+2x-15} = 0$ can be solved by solving $x^2 - 9x + 18 = 0$, and then rejecting those roots that make $x^2 + 2x - 15$ zero. The roots are 3 and 6, of which 3 must be rejected. The advantage is that $x^2 + 2x - 15 = 0$ need not be solved. This solution can be presented with $\Rightarrow$ or a subproof. By assuming $x^2 + 2x - 15 \neq 0$ in the latter, $\Rightarrow$ can be avoided. (All this already works on a prototype.)

6 ON ALTERNATIVE NOTATIONS

Although our use of $\Leftrightarrow$, $\Rightarrow$ and $\Leftarrow$ agrees with fairly common informal practice, the reader is warned that textbooks on mathematical logic tend to disagree with both our use and each other. We make a distinction between $\rightarrow$ that resides *in* a formula and produces a truth value; and $\Rightarrow$ that resides *between* two formulas and expresses a correct or incorrect reasoning step (and similarly with $\leftrightarrow$ and $\Leftrightarrow$). Most if not all authors introduce the former notion. Many denote it with $\rightarrow$, but many others use $\Rightarrow$ instead. Most authors avoid the latter notion altogether, use ad-hoc notation, or confuse it with the former notion.

$|x| + y = 6 \wedge 3x + 5y = 22$

Subproof

Assume $x \geq 0$

Parent's original $\Leftrightarrow x = 6 - y \wedge 18 - 3y + 5y = 22 \Leftrightarrow y = 2 \wedge x = 4$

Subend

Subproof

Assume $x < 0$

Parent's original $\Leftrightarrow x = y - 6 \wedge 3y - 18 + 5y = 22 \Leftrightarrow y = 5 \wedge x = -1$

Subend

Original $\Leftrightarrow x = 4 \wedge y = 2 \vee x = -1 \wedge y = 5$

Correct!

Figure 11: Exploiting `assume` in a solution by cases.

For instance, (Hammack, 2018) uses $\Rightarrow$, $\Leftrightarrow$ and $=$ in the roles of our $\rightarrow$, $\leftrightarrow$ and $\Leftrightarrow$ (e.g., p. 51: $P \Leftrightarrow Q = (P \wedge Q) \vee (\sim P \wedge \sim Q)$). It apparently lacks our $\Rightarrow$. (Hein, 1995) uses $\rightarrow$ similarly to us, $\equiv$ similarly to our $\Leftrightarrow$, and lacks our $\leftrightarrow$ and $\Rightarrow$. (Huth and Ryan, 2004) use $\rightarrow$ similarly to us, do not use $\leftrightarrow$, and use $\vdash$ and $\dashv\vdash$ somewhat similarly to our $\Rightarrow$ and $\Leftrightarrow$.

Our $\Rightarrow$ and $\Leftrightarrow$ are *relational*: $P \Rightarrow Q \Rightarrow R$ is correct if and only if $P \Rightarrow Q$ is correct and $Q \Rightarrow R$ is correct. This is crucial for handy writing of reasonings, which in turn is central to our goal of supporting the development of adaptive reasoning.

Unfortunately, the existence of different conventions can lead to dramatically different meanings for the very same expression. For instance, (Stanford University, 2021, Section 2.2) uses a right-associative $\Rightarrow$ in the role of our $\rightarrow$. Under that convention, $P \Rightarrow Q \Rightarrow R$ means the same as $P \Rightarrow (Q \Rightarrow R)$, that is, $Q \Rightarrow R$ is evaluated first, and its result is then used as the X in the evaluation of $P \Rightarrow X$. Consider $x = 2 \Rightarrow x = 1 \Rightarrow x = 0$. When $x \neq 2$, it is T because it is then of the form $\mathsf{F} \Rightarrow$ something. When $x = 2$, it is T because $x = 1 \Rightarrow x = 0$ is then of the form $\mathsf{F} \Rightarrow \mathsf{F}$ which is T, so the expression as a whole is of the form $\mathsf{T} \Rightarrow \mathsf{T}$, and thus T. That is, although in our convention $x = 2 \Rightarrow x = 1 \Rightarrow x = 0$ is just plain wrong, under the Stanford convention it is T for every value of x!

In our convention, $P \Rightarrow (Q \Rightarrow R)$ is a syntax error.

The notations `original`, `subproof`, `subend` and `assume` are by us, but of course the ideas are next to trivial. Together with `assume`, `original` $\Rightarrow$ is essentially the same as $\Gamma \models$ in hard-core formal logic.

The undefined truth value U is almost never mentioned in textbooks. Its behaviour in the case of $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ is fairly unproblematic; the tool follows (Kleene, 1952). Beyond that, in particular regarding $\Rightarrow$ and $\Leftrightarrow$, the issue becomes complicated indeed. A number of different approaches were discussed by (Schieder and Broy, 1999). The tool uses our own approach (Valmari and Hella, 2017).

7 CONCLUDING REMARKS

We illustrated how our tool can automatically check various patterns of reasoning in a restricted domain of problems, and give feedback that the student can use as a starting point for improving the solution. Although the domain is restricted, it can meaningfully accommodate many non-straightforward patterns of reasoning. The example by (Pólya, 1945) is certainly challenging enough to bring forward the benefits of creative thinking over blindly following mechanical procedures. It was successfully dealt with by the tool.

It is worth emphasizing that our tool facilitates verbal problems that the students have to translate to (in)equations themselves. The teacher gives the (in)equation (or anything logically equivalent, such as the roots) to the tool as the original problem and asks the tool to keep it hidden from the students.

An earlier feature of our tool has been found to statistically significantly improve performance in an examination (Kaarakka et al., 2019). The implementation of the features discussed in this paper started in spring 2020. Therefore, there is not yet much pedagogical experience. In a course, students had little problems with the tool itself, but had big problems with mathematics that they were supposed to already master: absolute values, pairs of equations, and so on. They also had big problems in modelling verbally expressed problems mathematically. Their earlier studies had not developed these skills to the promised level.

Until now it has not been possible to make students solve this kind of problems in great numbers, because of the lack of teachers to check the answers and provide feedback. Now we have a tool for this.

REFERENCES

ACM and IEEE. (2013). *Joint Task Force on Computing Curricula, Association for Computing Machinery and IEEE Computer Society, Computer Science Curricula 2013 : Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA.

Anthony, G. and Walshaw, M. (2009). *Effective pedagogy in mathematics*. UNESCO/IBE.

Hammack, R. (2018). *Book of Proof*. Open access.

Hein, J. L. (1995). *Discrete Structures, Logic, and Computability*. Jones and Bartlett Publishers, Inc., USA.

Hiebert, J. and Grouws, D. A. (2007). The effects of classroom mathematics teaching on students' learning. In Lester, F. K., editor, *Second Handbook of Research on Mathematics Teaching and Learning*, pages 371–404. Information Age Pub.

Huth, M. and Ryan, M. D. (2004). *Logic in Computer Science - Modelling and Reasoning About Systems (2. ed.)*. Cambridge University Press.

Kaarakka, T., Helkala, K., Valmari, A., and Joutsenlahti, M. (2019). Pedagogical experiments with MathCheck in university teaching. *LUMAT: International Journal on Math, Science and Technology Education*, 7(3):84–112.

Kilpatrick, J., Swafford, J., and Findell, B., editors (2001). *Adding It Up: Helping Children Learn Mathematics*. The National Academies Press, Washington, DC.

Kleene, S. C. (1952). *Introduction to metamathematics*. Van Nostrand, New York.

Pólya, G. (1945). *How to Solve It*. Princeton University Press.

Schieder, B. and Broy, M. (1999). Adapting calculational logic to the undefined. *The Computer Journal*, 42(2):73–81.

Stanford University (2021). Stanford introduction to logic: An online course on symbolic logic. http://intrologic.stanford.edu/; accessed 2021-01-23.

Valmari, A. and Hella, L. (2017). The logics taught and used at high schools are not the same. In Karhumäki, J., Matiyasevich, Y., and Saarela, A., editors, *Proceedings of the Fourth Russian Finnish Symposium on Discrete Mathematics*, volume 26 of *TUCS Lecture Notes*, pages 172–186, Turku, Finland. Turku Centre for Computer Science.

Valmari, A. and Rantala, J. (2019). Arithmetic, logic, syntax and MathCheck. In Lane, H., Zvacek, S., and Uhomoibhi, J., editors, *Proceedings of the 11th International Conference on Computer Supported Education, CSEDU 2019, Heraklion, Crete, Greece, May 2-4, 2019, Volume 2.*, pages 292–299, Setúbal, Portugal. SciTePress.