A Method for Teaching Rigorous Mathematical Reasoning

Ralph-Johan Back and Joakim von Wright
Åbo Akademi University and Turku Centre for Computer Science
Email: `{ralph.back,jockum.wright}@abo.fi`

1. Introduction

Mathematics rests on the combination of abstract concepts and formal notation. Today, nobody questions the use of symbols for numbers and arithmetic operations. The symbols allow us to express ideas concisely. The rules associated with the symbols allow us to reason efficiently, combining formal reasoning (using the rules) with intuitive reasoning about the real-world concepts that the symbols represent.

One of the cornerstones of formal reasoning is mathematical proof. However, proof is often considered difficult and consequently today's High School curricula typically mention proof only in connection with geometry. Strong arguments have been presented in favour of more training in rigorous reasoning [HanJah93, Hoyles97]. Furthermore, logical notation is used very little, and it is not taught in a systematic way. Instead, the notion of proof is informal and not uniform. Where logic is taught, it is seen as a separate object of study, rather than a tool to be used when solving mathematical problems. Instead, the solutions to mathematics problems are written in an unstructured and informal way. This makes it hard for students to know when a problem has been acceptably solved, and it also makes it difficult to look at solutions afterwards and discuss them. Only in connection with a few specific problem areas (typically algebraic simplification and equation solving) is a more uniform format for writing solutions sometimes used. However, even then it is often not clear to students, e.g., why deriving 0=1 from an equation means that the equation has no solutions.

Within the Computer Science community, a *calculational* paradigm for manipulating mathematical expressions has emerged. The following is an example of such a derivation, deriving a distributivity property (namely, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$) of set operators from distributivity properties of logical operators:

$$
\begin{array}{ll}
 & x \in A \cap (B \cup C) \\
\Leftrightarrow & \{ \text{ set comprehension (intersection) } \} \\
 & x \in \mathrm{A} \ \& \ (\mathrm{x} \in B \cup C) \\
\Leftrightarrow & \{ \text{ set comprehension (union) } \} \\
 & x \in A \ \& \ (x \in B \vee x \in C) \\
\Leftrightarrow & \{ \text{ distributivity of logical operators } \}
\end{array}
$$

$$
\begin{array}{ll}
 & (x \in \mathrm{A} \ \& \ x \in B) \vee (x \in A \ \& \ x \in C) \\
\Leftrightarrow & \{ \text{ set comprehension (intersection, union) } \} \\
 & x \in (A \cap B) \cup (A \cap C)
\end{array}
$$

The two main features of this format are: (a) the relationship (in this case logical equivalence) between the different expressions is made explicit, and (b) the justification for each step is stated explicitly. This format has been successfully used in Discrete Mathematics courses for University-level Computer Science students [GriSch95].

An obvious example of an area of school mathematics where this kind of format can be used is equation solving:

$$
\begin{array}{ll}
 & 2x + 3 = 5 \\
\Leftrightarrow & \{ \text{ subtract 3 from both sides } \} \\
 & 2x = 2 \\
\Leftrightarrow & \{\text{divide both sides by 2}\} \\
 & x = 1
\end{array}
$$

By transitivity of equivalence, this derivation shows that the original equation is equivalent to x=1, i.e., $2x + 3 = 5$ is true exactly when x=1 is true. The fact that x=1 is such a simple expression (it gives us immediate and full information about x) makes us accept it as an ”answer”.

It is not uncommon that students are taught to use an explicit equivalence symbol to separate equations, in the same way as we have used in the calculational solution above. However, we feel that the real meaning of this equivalence symbol is usually not made explicit enough. Logical equivalence is a relation and it relates equations (which denote truth values), exactly like equality and ”less-than” relate arithmetic expressions (which denote numbers). For this, students need some familiarity with propositional logic: truth values, connectives, truth tables, and algebraic manipulation of boolean expressions. When these basic facts are taught, we feel it is essential that truth values are not confused with numbers. Furthermore, implication (as a connective) can be omitted at first, in order to avoid the confusion that often arises in association with its truth table.

2. Structured derivations

The calculational format has been generalised to allow hierarchical nesting of subderivations inside a derivation [BaGrWr98]. We have shown how this format is suited for writing solutions to various kinds of Mathematics problems [BaWr97]. Below we show how an equation pair is solved using a structured derivation (to the left). When we *focus* on the right equation, we start a subderivation in which that equation is transformed, and the left equation can be used as an assumption (assumptions are written in square brackets). Note how the ”solution” is written as a logical expression ($x = 2$ *and* y = 3). Explicit logical notation can help avoid the confusion that often arises when comparing this with how the solution to a second-degree equation is written ($x = 2$ *or* $x = 3$).

$$
\begin{array}{lllll}
 & x = 2 \ \& \ y = x + 1 & & & \\
\Leftrightarrow & \{ \text{ rewrite right conjunct } \} & & & \\
 & \quad\quad [x = 2] & & & \\
 & \quad\quad y = x + 1 & & & \\
 & \quad \Leftrightarrow \{\text{use assumption } x = 2\} & & & \\
 & \quad\quad y = 3 & & & \\
\cdot & x = 2 \ \& \ y = 3 & & &
\end{array}
$$

$$
\begin{array}{ll}
 & x = 2 \ \& \ y = x + 1 \\
\Leftrightarrow & \{ \ \underline{\text{rewrite right conjunct}} \ \} \\
\cdot & x = 2 \ \& \ y = 3
\end{array}
$$

To the right we have written the derivation with the subderivation hidden, indicated by the underlined justification. This shows how structured derivations support varying levels of detail (using computer support, this can be done interactively; more is said about this in Section 4 below).

In addition to the more obvious application areas (algebraic simplification and equation solving), we have investigated using the method for solving an unbiased collection of exercises: the Finnish high-school matriculation exam. Our experience was that more than half the exercises could be written concisely and clearly as structured derivations. In fact, solutions to more complex problems of number theory and real analysis can also be written using this format. For further examples and for a detailed description of the logical foundations of structured derivation as an inference system, see [BaWr97].

3. Learning rigorous reasoning

We claim that it is not only possible to write solutions to problems of high-school mathematics as structured derivations, but we also feel it worthwhile to investigate whether this solution format can provide a foundation for a successful method for teaching and learning mathematics. When the format is used in a demonstration, the solution grows step by step, and it also remains to be viewed as a whole afterwards. The use of derivations forces students to make the individual steps in their solution strategies explicit. Furthermore, the whole solution (a structured derivation) becomes an object that can be reread, communicated, discussed, and modified later on. Furthermore, the fact that structured derivations can be published on the Internet and browsed (see Section 4 below) gives an additional inspiration, and it also can make Mathematics less dependent on the physical presence of an instructor.

As a small experiment to see whether our claim is reasonable, we gave three High-School students (just graduated) a short crash course on how to write structured derivations and then their task was to solve the problems of the Advanced Mathematics Course in the Finnish matriculation exam, from the years 1995 to 1997. (The matriculation exam is taken after 12 years of school, and the Advanced Course in

Mathematics is generally required of those who continue studying Mathematical Sciences or Engineering.)

Our general conclusion is that it seems realistic for students to learn using the format, and that it seems to make a difference to the way in which they think about their solutions. Our main critical observations (based on analysis of the solutions and discussions with the students) can be summarised as follows:

- The format requires familiarity with the notation of basic propositional logic (truth values and connectives), and learning this takes time.
- Manipulating expressions that involve both arithmetic and logic is a new challenge.
- It is hard to break the habit of writing solutions as a collection of disconnected ``scribbles" spread out over a paper.

4. Computer Support

It is possible to use computer support for doing structured derivations. Such support can either be in the form of an editor (which supports the writing of the the derivations) or of a theorem prover (which verifies that only valid derivation steps are included). An example of the latter kind of support is provided by the *Refinement Calculator* [BGLRW97]. It was originally developed for stepwise derivation of provably correct programs, but it can also be used for general logical and mathematical derivations, provided corresponding supporting theories are developed for these application areas..

In addition to tools that aid doing derivations, it is of interest to have ways of publising them. A prototype system that allows structured derivations to be published as browsable web pages has also been developed by Jim Grundy [Grundy96]. This allows the reader to choose whether to see the details of a subderivation or not, as in the example in Section 2 (solutions to exercises from the 1995 Finnish mathematics matriculation exam can be viewed at `http://www.abo.fi/~jwright/schoolmath`). This prototype was reworked by the three students involved in the experiment mentioned in Section 3, using Java programming. Furthermore, we are currently investigating how MathML could be used, to make the web-based mathematics presentation more flexible than allowed by traditional character-based HTML. Furthermore, we also intend to investigate the possible use of OpenMath to interface the solution format to other systems (for example, Computer Algebra systems).

5. Conclusions

Structured derivations are a generalisation and a formalisation of the principles used in standard equation solving, where a new (equivalent) version of an equation is written immediately below the previous version. However, we want to stress the use of the logical symbol that shows the relationship between the two equations and the justification for the step. In traditional equation solving, the relationship between equations (which can be equivalence, implication, or reverse implication) is usually left implicit. This easily leads students to draw erroneous conclusions, in particular in situations when the equation in question has either no solutions or more than one solution. In such cases, rigorous use of logical connectives also helps avoid mistakes in the final steps of the solution.

We argue that this method gives a number of advantages over traditional informal ways of writing solutions but also over other structured formats, such as calculational proofs without subderivations:

- It forces the student to be explicit about what strategies and rules are used, and in what order.
- Solutions to problems can be inspected and discussed afterwards, and they are easy to check.
- It is possible to refine an existing solution, making it more accurate and more detailed.
- The method supports self-study and distributed cooperation.
- There is computer-support for browsing solutions written in this format; thus the person looking at a solution can decide how much detail to look at.

Successful use of the method requires the basic notions of propositional logic (truth values and connectives) to be included in the basic mathematical notation. However, the amount of logical notation is small, and it is shown to be very useful in practice, so the extra effort spent on learning logic is a good investment for the student. Our experiment involving high-school students was encouraging and we feel we can argue for more extensive experiments where logic (in the form of structured derivations) is used as a tool for mathematics, starting already from the first year of high school. In fact, we feel that the elements of the method should be included already before that, in the context where it comes natural (including but not limited to algebraic simplification and equation solving). For this, tools should be developed that support the method and are suitable for classroom use

Bibliography

[BaGrWr98] R. J. Back, J. Grundy, and J. von Wright. Structured calculational proof. *Formal Aspects of Computing*, 9:469–483, 1997.

[BaWr97] R. J. Back and J. von Wright. Doing High School mathematics carefully. Tech. Rpt. 140, Turku Centre for Computer Science, November 1997.

[BGLRW97] M. Butler, J. Grundy, T. Långbacka, R. Ruksenas and J. von Wright. The Refinement Calculator: Proof Support for Program Refinement. Proc. FMP'97 – Formal Methods Pacific, Wellington, New Zealand. Discrete Mathematics and Theoretical Computer Science. Springer-Verlag 1997.

[GriSch95] D. Gries and F. Schneider. Teaching math more effectively through calculational proofs. *Am. Math. Monthl*y, pp 691–697, October 1995.

[Grundy96] J. Grundy. A browsable format for proof presentation. *Mathesis universalis* (electronic journal: http://www.pip.com.pl/MathUniversalis), 1(2), 1996.

[HanJah93] G. Hanna and H. N. Jahnke. Proof and application. *Educational Studies in Mathematics*, 24:421–438, 1993.

[Hoyles97] C. Hoyles. The curricular shaping of students' approaches to proof. *For the Learning of Mathematics*, 17(1):7–16, February 1997.