Reflection; Practical Necessity or Not?

John Harrison

Abstract

This paper summarizes my talk and the following discussions at the second QED workshop in Warsaw on 20–22 July 1995.

1 Are fully expanded proofs feasible?

Is it acceptable to reduce all mathematical proofs to formal constructions in something like a fairly orthodox Natural Deduction logical system? I believe that for the overwhelming majority of mathematics, the answer is 'yes'. We can split acceptability into two criteria: whether it is feasible to do it at all, and whether the user finds it congenial.

As for the user finding proofs congenial, there are well-established ways of programming higher level derived rules which ultimately decompose to the standard primitives, so only the computer need be concerned with that level of detail. Theorem provers in the LCF family implement this technique very effectively. In a QED-style undertaking, some such mechanism will be required anyway to automate various (domain-specific) inference patterns, and experience with HOL points to its great power and flexibility.

As for feasibility, it is widely accepted that almost all mathematics can *in principle* be reduced in this way. Of course, Gödel's theorems show that any given formal system is incomplete, in that there are true sentences unprovable in it. However such sentences are invariably theoretical pathologies with no obvious relationship to mainstream mathematics. The real question is how big the gap is between 'in principle' and 'in practice'.

We cannot answer this question conclusively, but we can point to some plausible intuitive evidence. Natural deduction systems are so-called because they really correspond closely to how mathematicians actually prove theorems. In practice, they use a few patterns of inference which constantly recur. There seems no reason to suppose that formalization will increase the size of proofs by more than a modest factor. This is supported by practical experience: the Mizar system has been used to formalize a wide variety of mathematics, and the proofs never become unmanageably large. The same experience is derived from the more modest mathematical formalizations undertaken in systems like HOL. Can anyone point to a mainstream mathematics text (for example one of the volumes of Bourbaki) and find an exception?

It is easy to construct true sentences which are provable in a given formal system but only with an unfeasibly long proof. But once again, it seems that like the Gödel sentence these are merely theoretical pathologies. And many proofs are hard to find, but this makes no difference to the business of actually formalizing them.

But even if the above argument is correct, there are accepted theorems whose proofs have never been written out in the conventional way. These are usually theorems which have been established with the aid of extensive computer checking of many cases; the four-colour theorem being the obvious example. (This point was made by Bill McCune.) It may be that here, at least close to the frontiers, it will not be feasible to produce a formal proof in the conventional sense. This situation seems more likely in theorems which are not really mainstream mathematics, but rather arise from verification work.

There are some technical arguments which suggest that any functional program which does some kind of proof can be internalized inside a formal system, and its effect simulated there. This is likely to induce a very substantial slowdown (probably greater if the program used imperative features like arrays and assignments), but only in extreme cases likely to make things completely impossible.

And there is considerable intellectual benefit for a foundational project like QED to insist of fully expanded proofs. It establishes a simple, canonical standard, and facilitates sharing between different systems. Even if this does hobble our ability to encompass some computer-checked proofs, this might be felt to be an acceptable price, especially as such theorems are usually rather rococo and peripheral to the main body of mathematics.

2 What if they aren't?

If fully-expanded proofs are not sufficient, what is to be done? One of the touchstones of the QED project is reliability. We can hardly just accept the results of ad-hoc checking programs written in various different computer languages. It seems that the only principled answer is something which is usually called *reflection*.

The basic idea is to verify the correctness of the computer program which 'proves' the theorem. That is, the semantics of the programming language concerned (C, LISP, FORTRAN or whatever) is formalized inside the system. Then it is verified that, on the basis of this semantics, a particular result of running a program (e.g. that "main" returns 0) indicates that some fact is indeed true (which in practice means 'provable', though as John McCarthy pointed out, one can quite well imagine taking the opportunity to move beyond the constraints of the particular formal system at the same time). This code is then run, or even incorporated into the system itself.

In practice there are difficulties in carrying out such a project. The real semantics of programming languages like C are very complicated. For ease of semantic description, still more actual verification, it's likely that one would have to program in quite a restricted subset. Every simplification and abstraction (e.g. ignoring numeric overflow) makes the result less and less reliable. Then there is the worry whether the intended semantics is correctly implemented by whatever machine/OS/compiler combination is used to run the code. Of course if the same system is used to run the theorem prover, that is already taken for granted. But by relying on reflection, the abstract description of what counts as validity is no longer describable in a couple of pages, but depends intimately on a colossally complex hardware/software system.

3 Logical reflection principles

Without the dangerous jump to verifying code then running it on real machines, one can still use some notion of reflection. In fact, when logicians talk about (logical) reflection they usually mean something like the following. We carry out a Gödel-style formalization of the logic's syntax and proof system inside itself. Then we augment the system with a new rule allowing $\vdash \phi$ to be deduced from $\vdash Provable(n)$ where n is the Gödel number of ϕ. This allows us to use certain kinds of meta-reasoning to establish that something is provable without actually constructing a proof.

This form of reflection is obviously much safer (actually the new reflection rule is a conservative extension provided the original system is 1-consistent, a pretty weak requirement). However it is also less obviously useful. Examples in the literature are for trivialities like using multiset equality to justify associative-commutative rearrangements, completely ignoring the question of whether multiset equality is any easier to prove. And in any case, a decent higher order logic or set theory can generally do all this ostensibly 'syntactic' reasoning without any extension of the logic (this is often referred to, following the work of Howe, as 'partial reflection').

4 Summary

There is little evidence that reflection principles are necessary for the vast body of mathematics. And the leap in conceptual complexity that it entails is best avoided except in the teeth of compulsion. There will be plenty of problems in formalizing mathematics, and the feasibility of expanded proofs does not yet seem one of the more important.

For a more detailed discussion of reflection principles, fuller presentations of the arguments adumbrated here, and an extensive bibliography, see the author's: 'Metatheory and Reflection in Theorem Proving: A Survey and Critique', Technical Report CRC-053, SRI International Cambridge Research Centre, also available on the Web as:

`http://www.cl.cam.ac.uk/users/jrh/papers/reflect.dvi.gz` .