Non-Obviousness

(from Francis Jeffry Pelletier and Piotr Rudnicki)

Some problems that are difficult for automated theorem provers (ATPs) are so merely because of their size, but not because of any logical or conceptual complexity. Examples of this type of difficult problem have been published in the past: see Pelletier [1986: problems 12, 29, 34, 38, 43, 51-53, 62, 66-68, and 71-74]. But there is another type of problem whose statement can be quite simple but whose proofs are nevertheless quite difficult for ATPs (and people) to find. This note gives an example of such a problem.

One use of ATPs has been in the area of proof checking. In this context, one tries to "convince" the system that he has constructed a correct proof of some theorem. He does this by entering a sequence of formulas and justifying each of these by referencing which preceding lines it comes from. But he does not say how the new line is to be generated—just that it can be easily gotten. The reason for not saying precisely how is to allow the mathematician a certain amount of flexibility. After all, if he had to always cite DeMorgan's laws (or other "low-level" rules), the entire point of "machine-aided mathematics" would be lost. Nonetheless, such systems are not to be allowed complete freedom here: we wish to make sure that the student really knows that some new line follows from previous ones. In other words, we want the system to accept any "obvious" inference but require that "non-obvious" inferences be broken down into a sequence of obvious steps.

Of course, this raises the question of just what is to count as "obvious" and as "non-obvious" in the context of a derivation in logic. This seemingly philosophical issue has been discussed in the proof-checking literature. One possible answer was given by Davis [1981]: a new line can be accepted as an obvious step if (a) there is a sequence of "basic, low-level" inferences that would yield that new line (from the lines referenced by the user), and (b) none of these referenced lines had to be "used more than once" (in the sense of being expanded to different members of the Herbrand Universe on the different uses). As simple as this answer is, it goes as long way toward explaining why some "trivial" problems are nonetheless difficult both for ATPs and for people.

In the context of a proof in general, rather than proof checking, we might say that a problem is "non-obvious" if any proof of it requires that one of the clauses be expanded more than once to different individuals. It is sometimes difficult to tell whether a problem is "non-obvious," since this requires that we know of all possible proofs that they require such expansions. Some of the problems in Pelletier [1986] seem quite easy, but we are unable to find an "obvious" proof. However, there seem to be clear cases where the problem *is* "non-obvious," for instance, problems 47 and 54. We present here another case, from Los [1983].

Suppose there are two relations, P and Q. P is transitive, and Q is both transitive and symmetric. Suppose further the "squareness" of P and Q: any two things are either related in the P manner of the Q manner. Prove that either P is total or Q is total.

Natural Form:

$$(\forall x)(\forall y)(\forall z)[Pxy \ \& \ Pyz \rightarrow Pxz]$$

$$(\forall x)(\forall y)(\forall z)[Qxy \ \& \ Qyz \rightarrow Qxz]$$

$$(\forall x)(\forall y)(Qxy \rightarrow Qyx)$$

$$\underline{(\forall x)(\forall y)(Pxy \lor Qxy)}$$

$$[(\forall x)(\forall y)Pxy \lor (\forall x)(\forall y)Qxy]$$

Negated-Conclusion Clause Form:

$$\neg Pxy \lor \neg Pyz \lor Pxz$$

$$\neg Qxy \lor \neg Qyz \lor Qxz$$

$$\neg Qxy \lor Qyx$$

$$Pxy \lor Qxy$$

$$\neg Pab$$

$$\neg Qcd$$

This problem is extremely easy to state, and the conclusion is somewhat startling. If you try to do a proof by hand, you will see that each premiss will have to be expanded to different entities on different occasions, and so the problem is "non-obvious."

THINKER, the natural deduction system described by Pelletier [1982], required 1808 lines in its proof of this problem (THINKER does not discard unused formulas it generates), and used 115.6 seconds of CPU time on an Amdahl 5860.

We believe that a study of the concept of "non-obviousness" as used in proof-checking would provide a source of easy-to-state and interesting problems for ATPs. It would be more interesting for the subject if the difficult problems for ATPs did not rely merely on size.

References

Davis, M. 1981. "Obvious Logical Inferences," IJCAI-81, pp. 530-31.

Los, J. 1983. Personal communication.

Pelletier, F. J. 1982. "Completely Non-Clausal, Completely Heuristically Driven, Automatic Theorem Proving," University of Alberta Technical Report TR82-7.

Pelletier, F. J. 1986. "Seventy-five Graduated Problems for Testing Automatic Theorem Provers," JAR 2, 2.

Rudnicki, P. "Obvious Inferences," unpublished manuscript, University of Alberta.