AAR NEWSLETTER

No. 6 September 1986

From the President...

This year's CADE 8 conference can certainly be recorded as a success. Not only was the attendance excellent, but I was delighted at the outstanding quality of the papers.

Reflecting this growing interest in automated reasoning, we have received several interesting articles from AAR members. As a result, we will be issuing two AAR newsletters rather closely together. This newsletter features both challenges to users of automated theorem provers (see the article on non-obviousness by Pelletier and Rudnicki) and a success story about the use of resolution (see the article on set theory by McCune). Additionally, we include announcements about forthcoming conferences related to automated reasoning.

The next issue, anticipated sometime next month, will feature two articles on Smullyan's book *To Mock a Mockingbird.* Also included will be information about a new program for translating first-order predicate calculus to clausal form, and announcements about new books and journals related to automated reasoning.

Conferences

Symposium on Logic Programming

The Third Symposium on Logic Programming, sponsored by the IEEE Computer Society, will be held on September 21-25, 1986, in Salt Lake City, Utah.

The program will feature papers on all areas of logic programming, including

- Applications of logic programming
- Computer architectures for logic programming
- New language features
- Parallel logic programming models

For further information, write to the conference chairperson Gary Lindstrom, Department of Computer Science, University of Utah, Salt Lake City, Utah 84112.

2nd Intn'l Conference on Rewriting Techniques and Applications

The Second International Conference on Rewriting Techniques will take place in Bordeaux in May 1987. You may be aware that the first conference, held in May 1985, attracted over 100 researchers.

Papers for the second conference are solicited. Of particular interest are the following topics:

- Equational deduction
- Computer algebra
- Unification and matching algorithms
- Functional and logic programming
- Automated theorem proving
- Rewrite rules based on expert systems

Paper submission deadline is December 15, 1986. Papers will be published by Springer-Verlag in the *Lecture Notes in Computer Science* series. For further information, write to David Plaisted, RTA-87, Department of Computer Science, New West Hall 035-A, University of North Carolina, Chapel Hill, North Carolina 27514.

1987 Intn'l Symposium on Multiple-Valued Logic

The Multiple-Valued Logic Technical Committee of the IEEE Computer Society will hold its seventeenth annual symposium on May 26-28, 1987, in Boston, Massachusetts. The symposium is sponsored by the IEEE Computer Society, the University of Massachusetts in Boston, and Digital Equipment Corporation. Papers are welcome in the area of multiple-valued logic, including

- Automated reasoning
- Logic programming
- Fuzzy logic
- Logic design and switching theory
- Fault detection and diagnosis

Authors should submit four copies by November 1 to Prof. Ivo Rosenberg, Research Center for Applied Mathematics, University of Montreal, Montreal, Quebec, Canada, CP 6128, Succ. A, H3C 3J7.

TAPSOFT '87

The Second International Joint Conference on Theory and Practice of Software Development will be held in Pisa on March 23-27, 1987. The conference involves three main events:

• a colloquium on trees in algebra and programming,

• a colloquium on functional and logic programming and specifications, and

• an advanced seminar on foundations of innovative software development.

For further information, write to Dr. Piepaolo Degano, Dipartimento di Informatica, Universita di Pisa, 1 - 56100 Pisa, Corso, Italy, 40.

IJCAI-87

IJCAI-87, the 10th International Joint Conference on Artificial Intelligence, will be held in Milano, Italy, on August 23-28, 1987. Two distinct tracks will be emphasized: science (August 23-26) and engineering (August 25-28). Authors are invited to submit papers to one of the tracks within the following topic areas:

• Architectures and languages

• Reasoning

• Knowledge acquisition and learning

• Knowledge representation

• Cognitive modeling

• Natural language understanding

• Perception and signal understanding

• Robotics

Papers should be submitted no later than January 5, 1987. Inquiries about submissions and the program should be sent to the program chairman John McDermott, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213.

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.

Set Theory: A Source of Hard Problems for Theorem Provers (or, Resolution Can Do It All)

(from Bill McCune)

Bob Boyer recently brought to the attention of Rusty Lusk, Ross Overbeek, Mark Stickel, Larry Wos, and me that in 1940, Gödel published a finite first-order axiomatization of set theory [*The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of Set Theory*, Princeton University Press, 1940]. This allows for nearly all of known mathematics to be expressed in a finite number of first-order formulas, and therefore most mathematical questions can be attacked with a first-order theorem prover.

Because set theory is usually based on the schematic set-builder concept $\{x : p(x)\}$ where $p(x)$ can be any first-order formula, I had been under the impression that a first-order formulation of set theory requires an infinite number of formulas. Such is the case for the widely studied Zermelo-Fraenkel set theory. On the other hand, von Neumann-Bernays-Gödel set theory is able to avoid the set-builder axiom schema; the result is a finite first-order axiomatization in which the axiom schema of Zermelo-Fraenkel set theory are theorems. See W. Quine, *Set Theory and Its Logic*, Harvard University Press, 1969, for a comparison of various kinds of set theory.

During a recent visit of Boyer and Stickel to Argonne, the six of us came up with an axiomatization that closely follows Gödel's. Skolemization, CNF translation, and some simplification produce 103 clauses. In addition, some concepts from abstract algebra and number theory were easily defined within the set theory, and some interesting challenge problems and open problems were easily stated. All of this is presented in "Set Theory in First-Order Logic: Clauses for Gödel's Axioms," which appears in the Problem Corner of the *Journal of Automated Reasoning*, Vol. 2, no. 3, 1986.

We hope that this set of clauses will aid in the development of new techniques for automated deduction and will enable more attacks on deep mathematical questions by automated theorem-proving