ASSOCIATION FOR AUTOMATED REASONING NEWSLETTER

No. 8 April 1987

From the AAR President, Larry Wos...

This issue features three articles of particular interest to our readers—from our readers. Michael Rusinowitch discusses the need for functionally reflex axioms in experiments with paramodulation. Bill McCune challenges researchers to rewrite a complex expression using associativity and commutativity. And Leslie Burkholder presents a new problem to be used in testing a theorem-proving system.

We are delighted with these contributions, and we continue to urge AAR members to send in new questions or problems of this nature.

Workshops and Courses

ANL Tutorial/Workshop on Automated Reasoning

On June 23-24, 1987, the automated reasoning group of the Mathematics and Computer Science Division at Argonne National Laboratory is giving its sixth tutorial/workshop on automated reasoning. The workshop consists of a set of tutorials based on the book *Automated Reasoning: Introduction and Applications*, by Wos, Overbeek, Lusk, and Boyle. The focus will be on the elements and applications of automated reasoning, including research in mathematics, logic circuit design and validation, and proving properties of computer programs. Attendance is by invitation; anyone interested in attending may call Larry Wos (312-972-7224).

Course Offered in Artificial Intelligence

The European Coordinating Committee for AI is organizing the second Advanced Course in Artificial Intelligence, scheduled for July 28-August 7, 1987, in Oslo, Norway. The topics include

- inference methods
- parallel and rewriting systems
- machine learning
- constructive techniques
- knowledge acquisition
- expert systems methodology

For information, write to ACAI'87, P.O. Box 5030, Mojorstua, N-0301, Oslo 3, Norway.

New Trends in Automated Mathematical Reasoning

The University of Catania is offering a one-week course designed to give an account of some of the more advanced research in the field of automated deduction and proof verification. The course is structured in three six-hour conferences from Monday to Friday, June 8-12, 1987. The preliminary program is as follows:

- Automatic theorem proving in geometry and differential geometry
  Prof. Wu Wen-tsun

- Pragmatic issues in verification of programs and mathematical theorems
  Prof. Jacob T. Schwartz

- Proving by example in geometry
  Prof. Jiawei Hong

On the last day, the three lecturers will discuss open problems.

For further information, write to Dr. G. Gallo, Dipartimento di Matematica, Viale A. Doria 6, 95125, Catania, Italy.

Conferences

1987 International Symposium on Multiple-Valued Logic

IEEE is sponsoring a three-day meeting on multiple-valued logic, to be held in Boston, Massachusetts, on May 26-28, 1987. Sessions will cover topics such as algebra, circuits, logical design, and fuzzy logic. Of particular interest to AAR members, we note a talk by J. Kalman on "Testing Logical Matrices with the Help of Automated Reasoning."

For further information, write to the University of Massachusetts at Boston, Division of Continuing Education, Boston, Massachusetts 02125.

Logic in Computer Science

The second annual conference on logic in computer science will be held at Cornell University, Ithaca, New York, on June 22-25, 1987. The twelve sessions cover topics including logic programming, verification of concurrent programs, constructive type theory, lambda calculus, and reasoning with many processes. Session 12 may be of particular interest, as it includes a presentation on inference rules for rewrite-based first-order theorem proving and on theorem proving using rigid E-unification.

For further information, write to the conference chair Ashok K. Chandra, IBM Research, Yorktown Heights, New York 10598.

Applications of Artificial Intelligence in Engineering

The first international conference on the use of AI in engineering applications stimulated many presentations on tools and techniques. The second conference has now been scheduled for Boston during the first week in August 1987. Conference themes will include

- computer-aided design
- intelligent tutors
- knowledge-based systems

- database and graphics interfaces
- model-based problem solving

For further information, write to Dr. R. Adey, Computational Mechanics Institute, Suite 6200, 400 West Cummings Park, Woburn, MA 01801.

Austrian Meeting on Artificial Intelligence

The Austrian Society for Artificial Intelligence will hold its third meeting at the Technical University of Vienna on September 22-25, 1987. The scientific program will cover various aspects of artificial intelligence, including

AI tools	Cognitive science
Impacts of AI	Expert systems
Automatic programming	Learning
Vision	Robotics
Theorem proving	Knowledge representation

In addition to the scientific program, tutorials will be offered. For further information, contact E. Buchberger, OGAI, "OGAI-Tagung 1987," Postfach 177, A-1014 Wien, Austria.

Exploring the Knowledge-Based Society

The 1987 Fall Joint Computer Conference, sponsored by the ACM and the IEEE Computer Society, will be held on October 25-29, 1987, in Dallas, Texas. The conference is intended to provide a forum where corporate decision makers and other leaders in the field can discuss new and expected directions in the computer industry and profession. Conference topics will include

- expert systems
- architectures for AI
- robotics
- speech/vision recognition
- machine learning

For further information contact Dr. Stephen A. Szygenda, Conference Chairman, FJCC '87, The University of Texas, College of Engineering, Cockrell Hall 2.51OB, Austin, TX 78712-1653.

ESIB-87

The first annual conference on Expert Systems Applications in Business and Finance will be held in New York on November 10-12, 1987. Sponsored by Learned Information, publishers of Expert Systems--the International Journal of Knowledge Engineering, the conference will focus on the justification, development, and use of expert systems in the highly competitive, for-profit arenas of business and finance. For further information, contact Jerald Feinstein, Conference Chairman, ESIB-87, Learned Information Inc., 143 Old Marlton Pike, Medford, NJ.

CADE-9

The 9th International Conference on Automated Deduction is tentatively scheduled to be held on May 23-26, 1988, at Argonne National Laboratory, Argonne, Illinois. The conference is the primary international forum for reporting research in all aspects of automated deduction.

Authors interested in presenting papers at the CADE-9 conference should contact Ewing Lusk or Ross Overbeek, the program chairmen, at the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439-4844. While no formal dates have yet been set, we expect the deadline for submission of papers to be in early fall of 1987.

A New Newsletter

The Center for Design of Educational Computing at Carnegie Mellon University is pleased to announce the publication of the first issue of *The Computers and Philosophy Newsletter*. The newsletter is devoted to topics such as the use of computers in teaching logic and other subjects in philosophy, computers, and ethics, computers and the mind, and other issues concerning computers and philosophy. The newsletter will attempt to keep readers informed of successful instructional techniques, new software, new texts, conferences, and relevant research. It will carry articles, reviews, course syllabi, and news notes. Contributions should be directed to Leslie Burkholder, CDEC Carnegie-Mellon University, Building B, Pittsburgh, Pennsylvania 15213.

A New Database

(from Rick Stevens)

We are pleased to announce an on-line database of automated theorem-proving problems and proofs. The address to send requests is

aar-request@anl-mcs.arpa

Those who send mail to this address will be given instructions on how to retrieve and submit problems to the database.

We would also like to make available on-line distribution of the AAR newsletter, including past issues, and perhaps put together an e-mail mailing list for discussion of various topics of interest to researchers in automated reasoning. Any comments or suggesting for this mailing list are appreciated.

Functionally Reflexive Axioms and Linear Strategies: A Counterexample

(from Michael Rusinowitch)

For many refinements of paramodulation, the question of completeness is open, unless the following axioms [Wos and Robinson, "Paramodulation and Set of Support," Symposium on Automatic Deduction, Versailles, France, 1968, Springer-Verlag, pp. 276-310] are added.

Functional Reflexivity: given any f: f(x1,...,xn) = f(x1,...,xn)

For instance, the functionally reflexive axioms are absolutely needed to ensure the completeness of Horn semantic paramodulation [McCune and Henschen, "Experiments with semantic paramodulation," *JAR* 1(3), 1985, pp. 231-261].

Example 1 shows that the linear strategy for paramodulation and resolution (see [Loveland, *Automated Theorem Proving: Deduction Logical Basis*, North-Holland, Amsterdam, 1978]) is not complete without the functionally reflexive axioms.

Example 1.

1. P(x,x,f(a),f(b))
2. ¬P(f(a),f(b),x,x)
3. a=b

where x denotes a variable. This set of clauses is E-unsatisfiable. Here is a refutation:

4. P(x,x,f(a),f(a)) by paramodulation of 3 into 1
5. ¬P(f(a),f(a),x,x) by paramodulation of 3 into 2
6. □ by resolution of 4 and 5

But there is no way to get a linear deduction of the empty clause without a functionally reflexive axiom. This example can be compared with Example 2 given by Richard Statman [Statman, "Solution to a problem of Chang and Lee," *Notre Dame J. of Formal Logic*, 21(3), 1980] to show that input resolution with input paramodulation (IP) is strictly weaker than unit resolution with unit paramodulation (UP):

Example 2 (Statman).

1. P(a) | a=b
2. P(b)
3. ¬P(a) | Q
4. ¬P(a) | ¬Q

This E-unsatisfiable set of clauses admits a UP-refutation but no IP-refutation. However, the UP-refutation given by the author is also a linear refutation:

5. P(a) by paramodulation of 1 into 2
6. Q by resolution of 5 and 3
7. ¬P(a) by resolution of 6 and 4
8. □ by resolution of 5 and 7

The set of clauses in Example 1, although admitting a unit refutation, does not even admit a linear refutation. We can also remark that RUE-resolution [Digricoli and Harrison, "Equality-based binary resolution," *JACM* 33(2), 1986, pp. 253-289] provides a straightforward linear refutation in Example 1:

4'. a≠b | b≠ a innermost disagreement set between P(x,x,f(a),f(b)) and ¬P(f(a),f(b),x,x)
5'. a≠a | a≠ a by paramodulation of a=b into 4'
6'. □ by resolution with x=x

A 76th Automated Theorem Proving Problem

(by Leslie Burkholder)

This note provides an appropriate problem to be used in testing automated theorem proving systems. The problem is to be added to the list of 75 such problems given by Pelletier [1986]. For convenience, we review the notation used by Pelletier:

$\rightarrow$: if-then

$\neg$: not

& : and

A : for all

E : there exists

F, G, ..., P, Q, ... (perhaps with subscripts): predicate letters

x,y,z,w,... : individual variables

a,b,c,... : individual constants

The problem belongs in the section of Pelletier's list devoted to proof construction problems in full predicate logic without identity. Using his difficulty measure, this problem has a difficulty value of about 9 points.

The problem is to prove

(5) $\neg$ (Ex)[Gx & (Ay) (Py $\rightarrow$ (Az) Dxyz)]

from the following givens:

(1) (Ex)[Gx & (Ay)(Py $\rightarrow$ (Az)Dxyz)] $\rightarrow$ (Ew)[Pw & (Ay)(Py $\rightarrow$ (Az)Dwyz)]

(2) (Aw)([Pw & (Ay)(Py $\rightarrow$ (Az)Dwyz)] $\rightarrow$ (Ay)(Az)([(Py & H_2yz) $\rightarrow$ (H_2wyz & Owg)] & [(Py & $\neg$ H_2yz) $\rightarrow$ (H_3wyz & Owb)]))

(3) (Ew)[Pw & (Ay)([(Py & H_2yy) $\rightarrow$ (H_2wyy & Owg)] & [(Py & $\neg$ H_2yy) $\rightarrow$ (H_2wyy & Owb)])] $\rightarrow$ (Ev)[Pv & (Ay)([(Py & H_2yy) $\rightarrow$ (H_2vy & Ovg)] & [(Py & $\neg$ H_2yy) $\rightarrow$ (H_2vy & Ovb)])]

(4) (Ev)[Pv & (Ay)([(Py & H_2yy) $\rightarrow$ (H_2vy & Ovg)] & [(Py & $\neg$ H_2yy) $\rightarrow$ (H_2vy & Ovb)])] $\rightarrow$ (Eu)[Pu & (Ay)([(Py & H_2yy) -> $\neg$ H_2uy] $\rightarrow$ $\neg$ H_2uy] & [(Py & $\neg$ H_2yy) $\rightarrow$ (H_2yy) $\rightarrow$ (H_2uy & Oub)])].

buggy. See issue #17

Of what interest is this particular problem? Let

UD or domain of quantification: Unlimited

Gx: x is an algorithm

Px: x is a computer program in some programming language *L* (e.g., Pascal, a Turing machine programming language)

Dxyz: x is able to decide whether y halts, given input z

H_2xy : x halts, given input y

H_2xyz : *x* halts, given as input the pair y, z

Oxg: x outputs "good" (meaning: "Yes, it halts")

Oxb: x outputs "bad" (meaning: "No, it doesn't halt")

Then what is being proved, from some plausible premises, is that the halting problem is algorithmically unsolvable.

More discussion of this first-order predicate logic formulation of the proof of the unsolvability result and a "natural" proof are to be found in Burkholder [1987].

References

Burkholder, L. 1987. "The halting problem," SIGCSE Bulletin.

Pelletier, F. J. 1986. "Seventy-five problems for testing automatic theorem provers," JAR 2: 191-216.

A Challenge to Associative-Commutative Term Rewriters and Other Canonicalizers
(from Bill McCune)

The following set of 6 rewrite rules is a complete set of reductions for the theory of *exclusive or* and *and*. It assumes associative-commutative unification, and it is due to Hsiang. (It is also a complete set of reductions for free Boolean rings with unit.)

(1) eor(x,0) → x
(2) eor(x,x) → 0
(3) and(x,1) → x
(4) and(x,x) → x
(5) and(x,0) → 0
(6) and(x,eor(y,z)) → eor(and(x,y),and(x,z))

A sidenote: if the following two rewrite rules are included, the result is a decision procedure for the propositional calculus.

(7) or(x,y) → eor(x,eor(y,and(x,y)))
(8) not(x) → eor(x,1)

The challenge is to rewrite the following expression (from a verification problem being studied by Smith, Kljaich, Wojcik, and Chisholm at Argonne) with a general-purpose term rewriting system.

token(eor(eor(a2, eor(b2, eor(1, and(a2, b2)))), eor(eor(a3, b3), eor(1, eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a0, b0)), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), and(eor(1, and(a2, b2)), eor(cin, 1)))))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a0, b0)), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), and(eor(1, and(a2, b2)), eor(cin, 1)))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a0, b0)), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), and(eor(1, and(a2, b2)), eor(cin, 1)))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a0, b0)), and(eor(1, and(a1, b1)), and(eor(1, and(a2, b2)), eor(cin, 1))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), eor(1, and(a2, b2))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), eor(1, and(a2, b2)))))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a1, b1)), eor(1, and(a2, b2)))), eor(and(eor(a0, eor(b0, eor(1, and(a0, b0)))), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), eor(1, and(a2, b2))))), eor(and(eor(1, and(a0, b0)), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), and(eor(1, and(a2, b2)), eor(cin, 1)))))), eor(and(eor(1, and(a0, b0)), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(1, and(a1, b1)), and(eor(1, and(a2, b2)), eor(cin, 1))))) , eor(and(eor(1, and(a0, b0)), and(eor(1, and(a1, b1)), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), and(eor(1, and(a2, b2)), eor(cin, 1))))), eor(and(eor(1, and(a0, b0)), and(eor(1, and(a1, b1)), and(eor(1, and(a2, b2)), eor(cin, 1)))), eor(and(eor(a1, eor(b1, eor(1, and(a1, b1)))), eor(1, and(a2, b2))), and(eor(a1, eor(b1, eor(1, and(a1, b1)))), and(eor(a2, eor(b2, eor(1, and(a2, b2)))), eor(1, and(a2, b2))))))))))))))))))))

We are able to rewrite this expression using a technique (developed by Overbeek and colleagues at Argonne) *not* based on associative-commutative unification. Instead, we include permutative rewrite rules (such as commutativity) which are applied only if the result is smaller in a specified lexicographic ordering. Our best time so far on the expression is about 10 minutes on a Sun 3 workstation with a demodulation program written in Quintus Prolog. About 50,000 individual rewrites occur, resulting in the expression

token(eor(a3, eor(b3, eor(and(a0, and(b0, and(a1, a2))), eor(and(a0, and(b0, and(a1, b2))), eor(and(a0, and(b0, and(b1, a2))), eor(and(a0, and(b0, and(b1, b2))), eor(and(a0, and(a1, and(a2, cin))), eor(and(a0, and(a1, and(b2, cin))), eor(and(a0, and(b1, and(a2, cin))), eor(and(a0, and(b1, and(b2, cin))), eor(and(b0, and(a1, and(a2, cin))), eor(and(b0, and(a1, and(b2, cin))), eor(and(b0, and(b1, and(a2, cin))), eor(and(b0, and(b1, and(b2, cin))), eor(and(a1, and(b1, a2)), eor(and(a1, and(b1, b2)), and(a2, b2)))))))))))))))))

To obtain a copy of this note by electronic mail, send a request to mccune@anl-mcs.arpa .