Association of Automated Reasoning Newsletter No. 2

From the President

I am delighted to report that the membership of AAR has quadrupled. We now have more than 250 members, and the number continues to increase. Jorg Siekmann's suggestion of publicizing AAR through the IJCAI 83 mailing clearly is one of the causes for the increase in membership.

A New Journal

Another exciting development concerns a new journal for automated reasoning. Graham Wrightson conceived the idea two years ago and began exploring the possibility with various publishers. That possibility is now a reality. The *Journal of Automated Reasoning* will publish its first quarterly issue in January of 1985. The publisher is D. Reidel Publishing Company. We expect that the price of the journal will be surprisingly low, with the members of AAR entitled to a special rate. As editor-in-chief of the new journal, I contacted various experts from the appropriate fields to serve on the editorial board. To say that I was greeted with enthusiasm is an understatement. With almost no exceptions, each individual who was contacted agreed to serve.

The following paragraph gives a brief account of certain aspects and intentions of the journal:

"The new *Journal of Automated Reasoning* is an interdisciplinary journal that maintains a balance between theory and application. The spectrum of material ranges from the presentation of a new inference rule with proofs of its logical properties to a detailed account of a computer program designed to solve some problem from industry. The papers published in this journal are from, among others, the fields of automated theorem proving, logic programming, expert systems, program synthesis and validation, artificial intelligence (as related to automated reasoning), computational logic, and industrial applications of automated reasoning. The papers share the common feature of focusing on some aspect of automated reasoning, a field whose objective is the design and implementation of a computer program that serves as an assistant in solving problems and in answering questions that require reasoning".

The journal is issuing a call for papers. Until the final details for submission of papers are worked out, however, authors are encouraged to write or call

Lawrence Wos
Mathematics and Computer Science Department
7600 South Cass Avenue
Argonne, IL 60439
(312) 972-7224

Problem Sets

One of the prime motivations for forming AAR was to promote the existence of a problem set that can be used by various people to test programs. We have already had numerous requests for such a problem set, and Rob Shostak has agreed to serve as chairman of the committee to collect the problems. I would like the assistance of AAR members in formulating such problems, documenting them with clause notation or some other format, and sending them to

Rob Shostak
SRI International
Computer Science Laboratory
333 Ravenswood Ave.
Menlo Park, CA 94025

In the next newsletter, we intend to include a graduated set of problems that can be used to study and evaluate the treatment of equality by various automated reasoning programs. Small, graduated problem sets that focus on some aspect of automated reasoning are valuable additions to the field. A large set of problems focusing on some aspect of automated reasoning is also needed to test conjectures formulated from experiments with the smaller test sets.

Open Questions

Another motivation for forming AAR was to accrue a set of open questions to attack with various techniques from automated reasoning. Here is a sample open question. More precisely, we are checking with algebraists to see if this question is still open.

The question concerns Latin squares and quasigroups. A quasigroup is a structure that is closed under the operation of dot. For any elements a and b in the structure, you must be able to solve the equations ax = b and ya = b. For a given integer n, where n is the number of elements in the quasigroup, what is the maximum number of triples for which the associative law holds such that the structure is not a group?

Second Workshop on Automated Reasoning

The second Workshop on Automated Reasoning was held at Argonne National Laboratory in June. Fifty-seven people attended (more than twice the number at the first workshop). The questions showed an increased understanding of the basic concepts, possible applications, and potential of the field. A third workshop is in the planning stage.

Outstanding Work

Bob Boyer and J Moore were awarded the John McCarthy Prize for Program Verification at IJCAI 83 for their outstanding work in program verification. Their validation of an encryption algorithm provides an example of what can be done with automated reasoning and suggests to members from other fields the potential of our field.

Major Success for the Knuth-Bendix Procedure
(from M. Stickel)

The Knuth-Bendix complete-sets-of-reductions method has been used successfully to solve a challenge problem offered for theorem-proving programs by Woody Bledsoe in his August 1977 *Artificial Intelligence Journal* article "Non-Resolution Theorem Proving." The problem was to prove that if x*x*x=x in a ring, then the ring is Abelian.

The technique used was the Knuth-Bendix completion method using associative-commutative unification for + and incomplete associative unification for * as described in Peterson and Stickel's article "Complete Sets of Reductions for Some Equational

Theories" in the *Journal of the ACM*, April 1981. The program attempted to find a complete set of reductions beginning with a complete set of reductions for free rings plus the reduction x*x*x -> x. The program predictably failed to complete the set of reductions. Commutativity of *, a consequence of x*x*x=x, prevents there being a complete set of reductions unless commutativity of * is assumed. However, the program did discover the commutative equality x*y=y*x, thus proving the theorem. All reasoning was essentially forward reasoning with the program never having been told that the objective was proving x*y=y*x. The program also later succeeded in proving the same result with comparable effort by using ordinary rather than associative unification for * and building in associativity with the reduction (x*y)*z -> x*(y*z). The completion procedure was augmented by the use of + cancellation laws to simplify derived reductions. This greatly affected the feasibility of solving the problem.

When the commutativity of * was discovered, only 135 reductions had been created, of which 53 were retained. The economy of the Knuth-Bendix procedure is demonstrable from the fact that these 135 reductions were the result of simplifying 9,013 equations derived from matching 988 pairs of reductions. Most of the remaining equations were simplified to identities and discarded; a few were simplified to equalities like x*y+x*y+x*y=y*x+y*x+y*x that could not be converted into reductions and were also discarded. Total time was about 14.3 hours (including garbage collection time) on a Symbolics LM-2 Lisp Machine. This reflects slowness of the simplification procedure on numerous lengthy terms. The proof itself is 21 steps long, not counting input reductions or simplification steps.

This is a substantial success for the Knuth-Bendix completion method that had already shown promise in solving less difficult problems such as completing sets of reductions for various algebras like free groups and rings.

The only previous computer proof of this problem was done by Robert Veroff in 1981 using the Argonne National Laboratory - Northern Illinois University theorem-proving program (Argonne National Laboratory report ANL-81-6 "Canonicalization and Demodulation" provides some information on his approach). His solution required an impressive 2+ minutes on an IBM 3033. It is interesting to compare the approaches taken in these two proofs. Both rely heavily on equality reasoning. The process of fully simplifying equations with respect to a set of reductions is just demodulation. The Knuth-Bendix method's means for deriving equations from pairs of reductions is similar to the paramodulation operation used in the ANL-NIU prover. Nevertheless, solution by the Knuth-Bendix method required less preparation of the problem. The Knuth-Bendix program was given only the 11 reductions for free rings, the reduction x*x*x -> x, declarations of associativity and commutativity, and a reduction for the cancellation laws. The ANL-NIU program was provided with a total of about 60 clauses. Some clauses expressed information about associativity and commutativity that are handled by declarations in the Knuth-Bendix program. A large number were present to support a polynomial subtraction inference operation--e.g., to derive a+(-c)=0 from a+b=0 and b+c=0. Comparable operations are implicit in the Knuth-Bendix method that can infer a=c from the embedded reductions x+a+b -> x (from a+b -> 0) and y+b+c -> y (from b+c -> 0).

Belnap's Problem in Relevance Logic
(from H. Ohlbach and G. Wrightson)

Relevance Logic (RL) was first treated by Ackermann and Church and has been intensively refined and developed mainly by Anderson and Belnap. The primary motivation was to avoid certain paradoxes of implication that are present in classical formal logic. The problem shown below is known as "Belnap's Problem." It was given to Graham Wrightson at the University of Karlsruhe by N. D. Belnap of the University of Pittsburgh, one of the fathers of Relevance Logic. The proof is so tricky that even Belnap had forgotten how to do it. Michael McRobbie's theorem prover at La Trobe University, Melbourne, Australia solved it in less than a second, but his system works proof-theoretically and is designed to handle RL and related systems (i.e., it is a special-purpose theorem-proving system). Bob Meyer at the Australian National University is one of the very few specialists in Relevance Logic who was able to solve the problem by hand, using semantic tableaux.

The Karlsruhe group obtained a proof on their system using under 10 minutes of CPU time. The reasoning seemed to hinge on a remarkable restriction that limited the introduction of terms involving Skolem functions. The details of the proof can be acquired from either of the following:

Hans Jurgen Ohlbach
Universitat Karlsruhe
Institut fur Informatik
Postfach 6380
D-7500 Karlsruhe 1
West Germany

Graham Wrightson
University of Wellington
Department of Computer Science
Private Bag
Wellington, New Zealand

The problem is to show that the following set of clauses is unsatisfiable:

1. -R(x,y,z) | R(y,x,z)
2. R(0,x,x)
3. R(x,x,x)
4. -R(x,y,z) | -R(z,v,w) | R(x,v,F1(w,y,v,x))
5. -R(x,y,z) | -R(z,v,w) | R(F1(w,y,v,x),y,w)
6. -R(0,x,y) | -R(y,z,v) | R(x,z,v)
7. -T(Imp(x,y),z) | -R(z,v,w) | -T(x,v) | T(y,w)
8. T(Imp(x,y),z) | R(z,F3(z,y,x),F2(z,y,x))
9. T(Imp(x,y),z) | T(x,F3(z,y,x))
10. T(Imp(x,y),z) | -T(y,F2(z,y,x))
11. -T(Imp(Imp(a,Imp(b,b)),Imp(a,Imp(a,Imp(b,b)))),0)

The proof is virtually unreadable unless you include the following definitions:

F3(0,Imp(a,Imp(a,Imp(b,b))),Imp(a,Imp(b,b))) = A
F2(0,Imp(a,Imp(a,Imp(b,b))),Imp(a,Imp(b,b))) = B
F3(B,Imp(a,Imp(b,b)),a) = C
F2(B,Imp(a,Imp(b,b)),a) = D
F3(D,Imp(b,b),a) = E
F2(D,Imp(b,b),a) = F
F3(F,b,b) = G
F2(F,b,b) = H
F1(D,A,C,A) = I
F1(F,I,E,A) = J
F1(H,I,G,J) = K

Given these definitions the proof goes as follows:

12. R(0,A,B) 11 8
13. T(Imp(a,Imp(b,b)),A) 11 9
14. -T(Imp(a,Imp(a,Imp(b,b))),B) 11 10
15. R(B,C,D) 14 8
16. T(a,C) 14 9
17. -T(Imp(a,Imp(b,b)),D) 14 10
18. R(D,E,F) 17 8
19. T(a,E) 17 9
20. -T(Imp(b,b),F) 17 10
21. R(F,G,H) 20 8
22. T(b,G) 20
23. -T(b,H) 20 10
24. R(A,C,D) 15 6 12
25. R(A,C,I) 24 4 3

26. R(I,A,D) 24 5 3
27. T(Imp(b,b),I) 25 7 13 16
28. R(A,I,D) 26 1
29. R(A,E,J) 28 4 18
30. R(J,I,F) 28 5 18
31. T(Imp(b,b),J) 29 7 13 19
32. R(J,G,K) 30 4 21
33. R(K,I,H) 30 5 21
34. T(b,K) 32 7 31 22
35. -R(I,K,H) 34 7 27 23
36. R(I,K,H) 33 1
37. null 36 35

The problem is undoubtedly difficult. In our opinion it indicates that the heuristic used by the Margraf Karl Refutation Procedure (the name of the system at Karlsruhe) to limit the introduction of terms with Skolem functions may be quite valuable.

A Problem from Lewis Carroll
(from E. Lusk)

The following problem was given by Lewis Carroll as a challenge problem.

"All the boys, in a certain school, sit together in one large room every evening. They are of no less than five nationalities — English, Scotch, Welsh, Irish, and German. One of the Monitors (who is a great reader of Wilkie Collins' novels) is very observant and takes MS. notes of almost everything that happens, with the view of being a good sensational witness, in case any conspiracy to commit a murder should be on foot. The following are some of his notes:

(1) Whenever some of the English boys are singing 'Rule, Britannia,' and some not, some of the Monitors are wide awake;

(2) Whenever some of the Scotch are dancing reels, and some of the Irish fighting, some of the Welsh are eating toasted cheese;

(3) Whenever all the Germans are playing chess, some of the Eleven are not oiling their bats;

(4) Whenever some of the Monitors are asleep, and some not, some of the Irish are fighting.

(5) Whenever some of the Germans are playing chess, and none of the Scotch are dancing reels, some of the Welsh are not eating toasted cheese.

(6) Whenever some of the Scotch are not dancing reels, and some of the Irish are not fighting, some of the Germans are playing chess;

(7) Whenever some of the Monitors are awake, and some of the Welsh are eating toasted cheese, none of the Scotch are dancing reels;

(8) Whenever some of the Germans are not playing chess, and some of the Welsh are not eating toasted cheese, none of the Irish are fighting;

(9) Whenever all of the English are singing 'Rule, Britannia,' and some of the Scotch are not dancing reels, none of the Germans are playing chess;

(10) Whenever some of the English are singing 'Rule, Britannia,' and some of the Monitors are asleep, some of the Irish are not fighting;

(11) Whenever some of the Monitors are awake, and some of the Eleven are not oiling their bats, some of the Scotch are dancing reels;

(12) Whenever some of the English are singing 'Rule, Britannia,' and some of the Scotch are not dancing reels, ...

Here the MS. breaks off suddenly. The problem is to complete the sentence, if possible."

The problem can be formulated in either the predicate calculus or a propositional form. The first order version uses the following dictionary of array predicates:

a = English boys
b = singing "Rule, Britannia"
c = monitor
d = asleep
e = Scottish boys
f = dancing reels
g = Irish boys
h = fighting
j = Welsh boys
k = eating toasted cheese
l = members of the Eleven
m = oiling their bats
n = German boys
p = playing chess

This gives the following abstraction of the problem:

1. If some a are b and some a are not b, then some c are not d.
2. If some e are f and some g are h, then some j are k.
3. If all n are p, then some l are not m.
4. If some c are d and some c are not d, then some g are h.
5. If some n are p and no e are f, then some j are not k.
6. If some e are not f and some g are not h, then some n are p.
7. If some c are not d and some j are k, then no e are f.
8. If some n are not p and some j are not k, then no g are h.
9. If all a are b and some e are not f, then no n are p.
10. If some a are b and some c are d, then some g are not h.
11. If some c are not d and some l are not m, then some e are f.
12. If some a are b and some e are not f, ...

If you prefer the problem in clausal form, it is as follows:

```
-a(x) | -b(x) | -a(y) | b(y) | c(S1)
-a(x) | -b(x) | -a(y) | b(y) | -d(S1)
-e(x) | -f(x) | -g(y) | -h(y) | j(S2)
-e(x) | -f(x) | -g(y) | -h(y) | k(S2)
n(S3) | l(S4)
n(S3) | -m(S4)
-p(S3) | l(S4)
-p(S3) | -m(S4)
-c(x) | -d(x) | -c(y) | d(y) | g(S5)
-c(x) | -d(x) | -c(y) | d(y) | h(S5)
-n(x) | -p(x) | e(S6) | j(S7)
-n(x) | -p(x) | e(S6) | -k(S7)
-n(x) | -p(x) | f(S6) | j(S7)
-n(x) | -p(x) | f(S6) | -k(S7)
-e(x) | f(x) | -g(y) | h(y) | n(S8)
-e(x) | f(x) | -g(y) | h(y) | p(S8)
-c(x) | d(x) | -j(y) | -k(y) | -e(z) | -f(z)
-n(x) | p(x) | -j(y) | k(y) | -g(z) | -h(z)
a(S9) | -e(y) | f(y) | -n(z) | -p(z)
-b(S9) | -e(y) | f(y) | -n(z) | -p(z)
```

-a(x) | -b(x) | -c(y) | -d(y) | g(S10)
-a(x) | -b(x) | -c(y) | -d(y) | -h(S10)
-c(x) | d(x) | -l(y) | m(y) | e(S11)
-c(x) | d(x) | -l(y) | m(y) | f(S11)
a(S12)
b(S12)
e(S13)
-f(S13)

Note that this set of clauses is not unsatisfiable. The problem is to complete the sentence in the manuscript. We will give the exact theorem that represents the appropriate solution in our next issue.

The second formulation (which evidently would not have been acceptable to Carroll) is propositional and uses the following dictionary:

a = some English sing
b = some English sing not
c = some monitors are awake
d = some monitors are not awake
e = some Scotch dance
h = some Scotch dance not
k = some Irish fight
l = some Irish fight not
m = some Welsh eat
n = some Welsh eat not
r = some Germans play
s = some Germans play not
t = some Eleven are not oiling

This gives

1. If a and b, then c.
2. If e and k, then m.
3. If r and not s, then t.
4. If c and d, then k.
5. If r and not e, then n.
6. If h and l, then r.
7. If c and m, then not e.
8. If s and n, then not k.
9. If a and not b and h, then not r.
10. If a and d, then l.
11. If c and t, then e.
12. If a and h, then ?

The clause version here is as follows:

-a | -b | c
-e | -k | m
-r | s | t
-c | -d | k
-r | e | n
-h | -l | r
-c | -m | -e
-s | -n | -k
-a | b | -h | -r
-a | -d | l
-c | -t | e

a
h
a | b
c | d
e | h
k | l
m | n
r | s

A Proposal for CADE Competitions
(from R. Overbeek)

Ross Overbeek has proposed a competition for automated reasoning systems, to start with the 8th Conference on Automated Deduction in 1986. Here are his suggestions for possible ground rules:

A list of several competition categories would be created at CADE 7, which will be held in May 1984. These categories might include puzzles, nonstandard logics, state space problems, geometry, analysis, circuit verification, program verification, algebra, polynomial manipulation, and medical diagnosis. Competitions would be held in those areas in which three or more research groups announced an intention to compete. At CADE 7, at least one referee would be selected for each area in which enough groups were interested. The referee, along with the contestants, would be responsible for determining the details of how each contest should be conducted.

The main goals of the competition should be to "display wares" and to determine appropriate formats for future competitions. Prizes should consist of pitchers of beer. By allowing the referee and contestants to determine appropriate formats, a variety of approaches may emerge. The different approaches could then be evaluated as a basis for a more serious competition at CADE 9.

Anyone interested in helping to organize such a competition or in contributing sample problems should contact Ross Overbeek@anl-mcs (telephone: 312-972-7856).

Kibbitzing
(from Don Cohen)

Don Cohen (U.S.C. Information Sciences Institute) has sent the following letter:

I'm working on a "specification kibbitzer" - a program that reads specifications (in our own formal specification language, Gist) and remarks on any interesting or surprising properties it finds of them. This is meant as a tool for validating/debugging specifications. The specification may be regarded as a set of axioms in a temporal predicate logic. The behaviors which meet the specification correspond to the models of that set of axioms. Kibbitzing amounts to 1) reasoning forward to find consequences of the specification and 2) presenting some of those consequences to a user. Both tasks have to be sensitive to the problem of what is "interesting." I'm working on the first task. Here the goal is to find as many of the "interesting" consequences (I have no formal definition of that term) as possible in reasonable time. Naturally one wants to avoid uninteresting results both in the interest of time and in the interest of making the job easier for the presentation module.

I seem to be making progress in this goal, which is another way of saying that my program used to work worse than it does now. Some of the mechanisms that I use are well known, such as subsumption. Others may be generally useful in other forward inference systems, or may work well due to the particulars of the specification language, or even the particular specifications I've been working on.

I am interested in other work that might be relevent to mine. Anyone who has experience with forward inference, ideas, suggestions, related experience (maybe even clause deletion strategies?), questions, etc. is encouraged to contact me.

Don Cohen
U.S.C. Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291
(213) 822-1511 ext. 226
Arpanet: DonC@ISIF