ASSOCIATION FOR AUTOMATED REASONING
NEWSLETTER

No. 21 September 1992

From the AAR President, Larry Wos...

This issue of the *AAR Newsletter* is most impressive, for it features a variety of articles that present new research, respond to issues raised in previous newsletters, and offer our readers challenge problems. The problems discussed illustrate the broad spectrum of areas being successfully addressed by automated reasoning programs today—areas ranging from combinatory logic and sentential calculus to group theory. Further, the articles underscore the three-pronged approach to automated reasoning that we find so beneficial to the field. Specifically, some of the articles focus on implementation, others on theory, and still others on experimentation.

We invite our readers to peruse these articles, to tackle the challenges raised, to add to the growing number of successes in the field of automated reasoning, and to contribute items to the *AAR Newsletter.*

New Journals

Journal of Logic, Language and Information

The goal of the newly established *Journal of Logic, Language and Information* is to contribute toward the construction of systematic and principled theories of cognitive activities and human information processing. The journal is intended to be interdisciplinary, promoting the coordination of research within different subareas, especially philosophy, linguistics, computer science, and cognitive science.

The journal is published quarterly; the cost of an annual subscription is $150.00. A special rate is available for members of the European Foundation for Logic, Language and Information. For further information, write to Kluwer Academic Publishers, Order Department, P.O. Box 358, Accord Station, Hingham, MA 02018-0358; or Kluwer Academic Publishers, Spuiboulevard 50, P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

Applicable Algebra in Engineering, Communication and Computing

Applicable Algebra in Engineering, Communication and Computing is a new journal published by Springer-Verlag. Among others, the journal focuses on such subjects as equational theories, rewriting techniques, unification, and theorem proving. More generally, it covers every application of term manipulation to automated deduction and computer science. The refereeing process is kept as short as possible: the guideline is that a paper without problems must be published no later than one year after its submission.

Submissions can be sent to the managing editor or directly to one of the field editors.

Managing Editor:

Prof. Dr. J. Calmet
University of Karlsruhe
Institute of Algorithms and Cognitive Systems
Postfach 6980
7500 Karlsruhe 1, Germany

tel: +49 (0)721 6084208
e-mail: KG02@DKAUNI2.BITNET
fax: +49/(0)721 696893

Editors:

Prof. Deepak Kapur
LI67A, Dept. of Computer Science
State University of New York
Albany, NY 12222

e-mail: kapur@cs.albany.edu
tel. +1 518 442 4281

Pierre Lescanne
CRIN (CNRS) and INRIA-Lorraine
BP 239
F54506 Vandoeuvre-les-Nancy Cedex, France

tel: +33 83 59 30 07
e-mail: lescanne@loria.fr
fax: +33 83 27 83 19

Call for Papers

DISCO '93

The third international symposium on Design and Implementation of Symbolic Computation Systems will be held in Gmunden, Austria, on September 15–17, 1993. DISCO '93 will focus on innovative methodological and technological aspects of hardware and software system design and implementation for symbolic and algebraic computation, geometric modeling and computation, automated reasoning, and automatic programming.

Papers are expected on the following topics:

Theory: specification and verification logics, abstract data types and type inference, specification languages, and executable specifications.

Languages: language design and implementation, constructs for programming paradigms, and integration of programming paradigms.

Software environments: system design and implementation, systems for different computing paradigms, software development tools, visual and graphic tools, and user interfaces.

Architectures: parallel and specialized architectures, software and hardware interfaces.

Papers should be no more than 12 pages. Send four copies of a one-page abstracts and four copies of the full paper, by February 1, 1993, to Prof. Alfonso Miola, DIS, Università di Roma "La Sapienza", 00198 Roma, Italy; e-mail miola@iasi.rm.cnr.it; fax Italy (6) 8442383; tel. Italy (6) 8841950.

LICS

The Eighth Annual IEEE Symposium on Logic in Computer Science will be held in Montreal, Quebec (Canada), on June 20–23, 1993. The symposium aims for wide coverage of theoretical and practical issues in computer science broadly related, including algebraic, categorical, and topological approaches.

Topics of interest include abstract data types, automated deduction, concurrency, constructive mathematics, data base theory, finite-model theory, knowledge representation, lambda and combinatory calculi, logical aspects of computational complexity, logics in artificial intelligence, logic programming, modal and temporal logics, program logic and semantics, rewrite rules, logical aspects of symbolic computing, problem solving environments, software specification, type systems, and verification.

The program chairman is Moshe Y. Vardi, IBM Research, Almaden Research Center, K53-802, 650 Harry Rd., San Jose, CA 95120-6099; vardi@almaden.ibm.com, vardi@almaden.bitnet; phone (408) 927-1784; fax (408) 927-2100.

Ten hard copies of a detailed abstract (not a full paper) and 20 additional copies of the cover page should be *received* by December 8, 1992, by the Program Chair (Attn: LICS). Authors without access to reproduction facilities may submit a single copy of their submission. Authors will be notified of acceptance by February 14, 1992. Accepted papers in the specified format for the symposium proceedings will be due April 6, 1993.

The cover page of the submission should include the title, authors, a brief synopsis, and the corresponding author's name, address, phone number, fax number, and e-mail address if available. Abstracts must be in English, clearly written, and provide sufficient detail to allow the program committee to assess the merits of the paper. References and comparisons with related work should be included. It is recommended that each submission begin with a succinct statement of the problem, a summary of the main results, and a brief explanation of their significance and relevance to the conference, all suitable for the non-specialist. Technical development of the work, directed to the specialist, should follow. Abstracts of fewer than 1500 words are rarely adequate, but the total abstract should not be longer than 10 typed pages with roughly 35 lines per page. If the authors believe that more details are essential to substantiate the main claims of the paper, they may include a clearly marked appendix to be read at the discretion of the committee. Late abstracts, or those departing significantly from these guidelines, run a high risk of rejection.

The results in the submission must be unpublished and not submitted for publication elsewhere, including proceedings of other symposia or workshops. All authors of accepted papers will be expected to sign copyright release forms, and one author of each accepted paper will be expected to present the paper at the conference.

On the Use and Completeness of the RUE-NRF System of Inference

Vincent J. Digricoli
Department of Computer Science, Fordham University, New York
digricoli@murray.fordham.edu

The paper "Equality-based Binary Resolution" [1] elaborates RUE-NRF theory on two levels: open form and strong form. It presents a proof of completeness in open form and a conjecture for completeness in strong form. In addition, it describes 19 experiments using the open form under a specific, defined heuristic.

Bonacina and Hsiang, in the May 1992 issue of the *AAR Newsletter*, [2] present two counterexamples to the conjecture on completeness for RUE-NRF in strong form. In [1] the strong form theory is developed as follows:

- A viability test is defined that is a necessary condition for an inequality to participate in a refutation.
- An equality restriction is defined as a condition to be satisfied before resolving complementary equality literals.
- An RUE-NRF unifier is conjectured but not shown to be complete.

Let us now consider the counterexamples offered by Bonacina and Hsiang:

```
S:   1. f(g(x)) ~= x     2. g(f(a)) = a.
```

We apply RUE in strong form to these clauses:

```
f(g(x)) ~= x
      |
      |----   g(f(a)) = a
      |
```

applying the RUE unifier a/x, we can resolve only to

```
f(g(a)) ~=  g(f(a))
```

which is not viable, and thus this resolution is suppressed. The viability test has correctly blocked the generation of an inequality that cannot be used to refute S since we cannot derive $f(g(a)) = g(f(a))$ from S. It remains to apply the NRF rule to $f(g(x)) = x$. The NRF unifier is the null substitution, and the NRF cannot be applied. Hence, the RUE-NRF inference rules in strong form cannot refute S which is E-unsatisfiable.

The failure occurs with the NRF unifier (not with the viability test) since the substitution $f(y)/x$ before NRF leads to a refutation:

```
f(g(x)) ~= x   (apply the substitution: f(y)/x)
f(g(f(y))) ~= f(y)   (apply NRF)
g(f(y) ~= y    (which is viable)
```

This inequality resolves with $g(f(a)) = a$ to produce the empty clause. Hence, this counterexample shows that the RUE-NRF unifier as defined in [1] is incomplete and must be modified by the following extension:

> "When an inequality has the form $x = f[...x...]$ (x appears one or more times in the argument list of f), the NRF unifier becomes $f(y)/x$."

With this change in the definition of the RUE-NRF unifier, S can now be refuted by the inference rules in strong form as shown above.

Bonacina and Hsiang obtain the substitution $f(y)/x$ by resolving

```
 x ~= f(g(x))
       |
       |-- f(y) = f(y)     (substitute f(y)/x)
       |
g(f(y)) ~= y  (topmost viable dis.set)
```

which implies adding the functionally reflexive axioms to S. My feeling is that this approach is not necessary if we extend the definition of the unifier as stated above. In extending the definition of the RUE-NRF unifier, we have made an implied use of the functionally reflexive axioms.

The central problem of RUE resolution in strong form is defining the substitution to be used before each RUE-NRF step, whose stepwise use will define the composite substitution that underlies the refutation shown to exist in the proof of completeness in open form. The original conjecture for the RUE unifier in [1] has been shown to be inadequate, and we now have an extended conjecture for further study.

The second counterexample uses the clause set

(2) S: 1. f(g(a,x)) ~= f(g(x,a) 2. g(y,b) = g(b,y)

The first clause alone is E-unsatisfiable since x is universally quantified and the strong form can be applied to it as follows:

```
f(g(a,x)) ~= f(g(x,a))
(the NRF unifier is the null substitution)
          |
          | apply nrf
 g(a,x)) ~= g(x,a)  (viable)
 (the NRF unifier is: a/x)
          |
          | apply nrf
    empty clause.
```

S can also be refuted by the strong form using clause 2:

```
 g(a,x)) ~= g(x,a)  (from above)
          |
          |---  g(y,b) = g(b,y)
  (RUE unifier is: a/y)
          |
     empty clause.
```

Hence, this clause set presents no problem for RUE-NRF in strong form.

Furthermore, in respect to this example, if in place of the strong form we simply apply the *mgpu* substitution, we obtain the refutation:

```
f(g(a,x)) ~= f(g(x,a)
  (the mgpu is: a/x)
          |
 f(g(a,a)) ~= f(g(a,a)
          | nrf
     empty clause.
```

Here, the viability test is not being used. There is also no need to apply the notion of an irreducible inequality. We wish to further clarify this notion, which was not associated with viability in [1].

In the experiments of [1], since input clause sets made all inequalities viable, we supplied the theorem prover in each run with a table of irreducible inequalities, that is, inequalities for which it was evident by inspection of the input set that the corresponding equality could not be derived. Examples of this are given in [1,4,5].

I now comment on the statement that

> "the RUE-NRF system in open form is too general to be mechanized effectively, as the disagreement set and the substitution can be arbitrary."

Using RUE-NRF in open form, we have performed 26 experiments in the fields of Boolean algebra, group theory, ring theory, and, most recently, the LIM+ challenge problems of Bledsoe [3,4]. In every experiment the input clause set contained equality literals of the form $X = ...$ (a simple variable as the argument of equality) which made all inequalities viable, and thus the viability test could not be used to prune the search. As a consequence, RUE-NRF in strong form was not applied in any of the experiments. In fact, the NRF rule was not used. Instead, we employed the RUE rule of inference in open form where we are free to heuristically select both a substitution and disagreement set. The central heuristic in all experiments was to apply the *mgpu* substitution and select the innermost disagreement set not containing an irreducible literal. These terms are defined in [1]. We derived refutations within efficient run-times, with the refutation length ranging between 2 and 42 steps. Hence, we succeeded in defining effective heuristics for convergent proof search.

In this context, the statement that the completeness of RUE-NRF in open form cannot be implemented by level saturation since the substitution is not specified is not empirically important. In point of fact, the disagreement sets of complementary literals are defined and finite in number and raise no problem for implementation. The challenge of the open form is to heuristically define the substitution and disagreement set to be used in the proof search. This successfully occurred in the above experiments. In the practical order of theorem proving, since completeness cannot be guaranteed within acceptable computer time, the central issue becomes the definition of proof-finding heuristics and whether an inference system supports or obstructs this development. RUE resolution presents a new environment for heuristics which has been effectively exploited in experiments performed.

References

1. V. J. Digricoli and M. C. Harrison, "Equality-based Binary Resolution," *J. ACM,* **33,** no. 2 (April 1986), 253–289.

2. Maria Bonacina and Jieh Hsiang, "Incompleteness of the RUE/INF Inference Systems," *AAR Newsletter*, May 1992.

3. W. W. Bledsoe, "Challenge Problems in Elementary Calculus," *J. Automated Reasoning,* **16,** no. 3 (September 1990), 341–359.

4. V. J. Digricoli and Kochendorfer, "LIM+ Challenge Problems by RUE Hyper-Resolution," *Proc. CADE-11,* June 1992, 239–252.

5. V. J. Digricoli, "The Management of Heuristic Search in Boolean Experiments with RUE Resolution," *Proc. IJCAI-85,* 1154–1161.

A Challenging Theorem of Levi

Bill McCune and Rusty Lusk
Argonne National Laboratory, Argonne, Illinois 60439
{mccune,lusk}@mcs.anl.gov

While attending LPAR'92 in Russia in July, one of us (RL) came across a group theory problem that is quite challenging to automated theorem provers. Let the commutator of two elements be defined as $[x, y] = x^{-1} \cdot y^{-1} \cdot x \cdot y$.

THEOREM (Levi). *The commutator operation is associative if and only if the commutator of any two elements lies in the center of the group.* In other words,
$[[x, y], z] = [x, [y, z]]$ if and only if $[u, v] \cdot w = w \cdot [u, v]$.

A textbook proof of the theorem can be found in [1].

One direction ($\leftarrow$) is easy for our theorem prover OTTER, but ($\rightarrow$) is difficult. On the fourth or fifth attempt with various strategies, OTTER found a 137-step proof of ($\rightarrow$) in about two hours on a SPARC 1 computer. The following strategy led to the proof:

- Paramodulation and demodulation similar to the Knuth-Bendix completion method.
- A term ordering that prefers commutator expressions, that is, using the definition of commutator as an equality and as a rewrite rule in the *reverse* direction from above. (This is counter to the standard strategy of using the definition to rewrite and eliminate commutator expressions at the start of the search.)
- A combination of shortest-first and breadth-first search (ratio 5:1).
- Discarding equalities with more than 20 symbols.
- Discarding equalities that contain a subterm of one of the following forms: commutator applied to commutator applied to commutator, commutator applied to product applied to commutator, commutator applied to product applied to product.

The current challenge is to find a proof automatically, without such a specialized strategy.

References

1. Kurosh, A. G., *The Theory of Groups, Vol. I*, Chelsea, New York, 1956, pp. 99–100.

Prolog-D-Linda

Geoff Sutcliffe and James Pinakis
Department of Computer Science
The University of Western Australia
geoff@cs.uwa.edu.au

Overview

Prolog-D-Linda (Prolog-Distributed-Linda) is an embedding of the Linda paradigm into SIC-Stus Prolog. (Linda is a programming framework of language-independent operators that are injected into the syntax of existing programming languages to produce new parallel programming languages. Linda permits cooperation between parallel processes by controlling access to a shared tuple space. The tuple space is accessed using the Linda operators.) Prolog-D-Linda supports a distributed tuple space, unification and Prolog style deduction in the tuple space, and a control hierarchy that provides remote I/O facilities for client processes.

Implementation

Prolog-D-Linda's tuple space and associated operations are implemented in server processes. Multiple servers can be used, each being responsible for a part of the tuple space. The partitioning is controlled by a user-supplied Prolog program. Each server stores its tuples as a collection of Prolog clauses in the Prolog database. Both Prolog rules and facts can exist in the tuple space, thus making the tuple space deductive. Linda operations in client processes are passed to an appropriate server. Client processes may be started independently at a terminal or via the eval operator. Prolog-D-Linda is controlled by a single controller process. The controller provides the remote I/O facilities.

Remarks

Prolog-D-Linda is a truly distributed logic-programming environment. The distribution allows applications to take advantage of the added computing power available in a structured fashion.

Prolog-D-Linda has been used to implement a distributed deduction system, which incorporates three different deduction formats [1]. The coarse-grained parallelism offered by Prolog-D-Linda makes it particularly suited to such applications. There appears to be significant scope for further research into such deduction systems.

Availability

Prolog-D-Linda and a full research report are available by ftp from ftp.cs.uwa.edu.au. Queries can be addressed to Geoff Sutcliffe, e-mail geoff@cs.uwa.edu.au.

Reference

1. Sutcliffe, G. (1992). A Heterogeneous Parallel Deduction System, in *Proceedings of the Workshop on Automated Deduction: Logic Programming and Parallel Computing Approaches,* Tokyo, 1992, Institute for New Generation Computer Technology.

An Opportunity to Test Your Skills—and the Power of Your Automated Reasoning Program

Larry Wos
Argonne National Laboratory, Argonne, Illinois 60439
wos@mcs.anl.gov

Because we enjoy challenging problems—both solving them ourselves and suggesting them for others to solve—we present here two challenge problems, one from logic and one from mathematics. For the first, equality plays no role; the second depends exclusively on equality. Both can be attacked with an automated reasoning program. Indeed, we have had some success with these problems using our own powerful program OTTER. But important questions remain that test the inventive skills of the researcher, as well as the power of the automated reasoning program selected.

A Problem from Formal Logic

The first problem comes from the two-valued sentential calculus, a field of formal logic concerned with the notions of implication and negation. The problem under consideration asks one to prove, from the Lukasiewicz system consisting of theses L1–L3, Church's axiom system consisting of theses 18, 35, and 49.

(CD) -P(i(x,y)) | -P(x) | P(y)
(L1) P(i(i(x,y),i(i(y,z),i(x,z))))
(L2) P(i(i(n(x),x),x))
(L3) P(i(x,i(n(x),y)))

(thesis_18) P(i(x,i(y,x)))
(thesis_35) P(i(i(x,i(y,z)),i(i(x,y),i(x,z))))
(thesis_49) P(i(i(n(x),n(y)),i(y,x)))

With the use of sophisticated strategy, an automated reasoning program can derive the Church system. For example, McCune's *tail strategy* [1] has been used to cause OTTER to focus on certain parts of a formula. The program did find a proof that Church's system (consisting of 18, 35, and 49) is complete. As evidence of the difficulty of the problem, OTTER requires approximately 7 CPU-hours on a SPARC-2 workstation.

As an alternative technique, we investigated the use of the *resonance strategy* [3]. In this case, the steps of a related or earlier proof are used as patterns to guide the search for a proof. With the resonance strategy, OTTER found a proof far more quickly than it had done with the tail strategy. But this proof—while certainly more than proof checking—still required an element of assistance (we used Lukasiewicz's proof steps as "guides") that might not be needed.

Indeed, the challenge we pose to you is to find a *general strategy* that can use theses L1, L2, and L3 to prove theses 18, 35, and 49. The most challenging aspect is to prove thesis 35–indeed challenging without advice from the researcher.

A Problem from Mathematics

The second problem is one that has puzzled mathematicians (including Tarski) for many years: Is every Robbins algebra a Boolean algebra? The axioms for a Robbins algebra are

$$o(x,y) = o(y,x)$$
$$o(o(x,y),z) = o(x,o(y,z))$$
$$n(o(n(o(x,y)),n(o(x,n(y))))) = x$$

We have studied the following: Can one find a property such that, when adjoined to the axioms for a Robbins algebra, the resulting algebra is Boolean?

Using OTTER, we can easily prove that the adjunction of $\exists c\ o(c,c) = c$ suffices, where the function o denotes plus. More significant, our colleague Steve Winker recently proved [2] that the adjunction of $\exists c \exists d\ o(c,d) = d$ suffices. But that proof required much assistance from Winker, and numerous guesses at the lemmas to prove.

The challenge to you, then, is to obtain a computer proof *unaided by human intervention.*

Some Help

Both problems—and a whole family of others like them that have occupied the attention of such eminent researchers as Frege, Russell, and Hilbert—are discussed in the new and thoroughly revised edition of *Automated Reasoning: Introduction and Applications*, by L. Wos, R. Overbeek, E. Lusk, and J. Boyle, published by McGraw-Hill in 1992. In particular, Chapters 9 and 10 provide input files for using OTTER to attack such questions and to attempt to find new axiom systems. The book includes a diskette of OTTER that runs on UNIX workstations such as the Sun and on IBM-compatible personal computers. In addition to users manuals offered on the diskette, the new book contains a chapter devoted solely to a tutorial on the use of OTTER.

References

1. Wos, L., "Meeting the Challenge of Fifty Years of Logic, " *J. Automated Reasoning* **6** (1990) 213–232.

2. Winker, S. "Robbins Algebra: Conditions That Make a Near-Boolean Algebra Boolean," *J. Automated Reasoning* **6** (1990) 465–489.

3. Wos, L. "Automated Reasoning Answers Open Questions," *Notices of the AMS*, to appear November 1992.

Solution to an Open Question in Combinatory Logic

Jian Zhang
Institute of Software, Academia Sinica, Beijing, P.R. China

In a paper by L. Wos entitled "The kernel strategy and its use for the study of combinatory logic" [1], the following open problem is listed:

Does there exist a finite model that satisfies L and Q ($Lxy = x(yy)$ and $Qxyz = y(xz)$) but for which the strong fixed point property, $\exists\Theta\forall x \ \ \Theta x = x(\Theta x)$, does not hold?

My answer is yes. The multiplication table for the four-element model is as follows:

	Q	L	C_1	C_2
Q	Q	Q	C_1	C_1
L	Q	Q	C_1	C_1
C_1	L	L	C_2	C_2
C_2	L	L	C_2	C_2

That paper also lists the question: Does there exist a finite model that satisfies S and W ($Sxyz = xz(yz)$ and $Wxy = xyy$) but for which the weak fixed point property, $\forall x\exists y \ \ y = xy$, does not hold?

I have found a three-element model for this problem, as described by the following table:

	a	b	c
a	a	a	a
b	a	a	a
c	b	a	a

Here the element a has the property of S, and the element b had the property of W. For the element c there is no element y such that $y = cy$. Hence, the weak fixed point property does not hold.

References

[1] Wos, L., "The kernel strategy and its use for the study of combinatory logic," *J. Automated Reasoning*, to appear.