ASSOCIATION FOR AUTOMATED REASONING NEWSLETTER

No. 20 May 1992

From the AAR President, Larry Wos... In this issue AAR members are asked to consider the incorporation of the Association. My intention is that incorporation will in no way adversely affect the Association: Dues will remain low, and reduced subscription rates to the *Journal of Automated Reasoning* will continue.

Why, then, incorporate at all? Many researchers active in automated reasoning have argued that incorporation—and a concomitant closer association with CADE—is necessary in order to give the field more recognition and influence.

I am, as you well know, a great promoter of automated reasoning. I believe that, at this time, incorporation of AAR is a logical way to promote the field (and I am also a great advocate of logical reasoning!). I ask you to read Woody Bledsoe's note about the incorporation, to study the bylaws, and to let me know if you have any concerns or questions.

The Incorporation of AAR

Woody Bledsoe

Larry Wos has done an excellent job as president of AAR during the past few years, and I hope that we can keep him in that position—at least for the near future. He has recently spent a great deal of his valuable time developing bylaws for the incorporation of AAR and CADE. The proposed bylaws for AAR, which are shown below, have been reviewed with lawyers experienced in this kind of activity. Also, Larry has discussed these with some members of the automated reasoning community, including Peter Andrews, Mark Stickel, Robert Boyer, Deepak Kapur, Donald Loveland, and myself. These discussions have resulted in iteration, with the result being completely agreeable to most of us.

As president of AAR, Larry Wos could go ahead with the incorporation of AAR, but he prefers delaying this action until the membership of AAR have had a chance to see the proposed bylaws and register any concerns that they might have. Furthermore, after consulting with colleagues to be sure we have a good choice, I have urged Larry to be the president of the incorporated AAR, and he has agreed.

I hope that the membership will back us on this incorporation of AAR and sustain Larry as the president (no vote is needed).

Association for Automated Reasoning Bylaws

The Association for Automated Reasoning (AAR) is a not-for-profit corporation whose objective is advancing the field of automated reasoning by means that include the dissemination of information to its members and the promotion of the exchange of information. This corporation will have the authority to incorporate other subcorporations. The Association for Automated Reasoning and all subcorporations will be independent of each other except for the respects covered in the included bylaws, and neither the corporation AAR nor its subcorporations will be liable for any acts or omissions of the other. The corporation AAR is subject to the following bylaws, and all other powers and provisions provided by law.

(1) Board of Directors

(A) The members of the Board of Directors of AAR consist of the AAR officers (president, vice-president, secretary, and treasurer) and the current Program Chairman of the Conference on Automated Deduction (CADE) and the Program Chairman of the preceding CADE. To be an officer of AAR, one must be a member of AAR. CADE is a not-for-profit subcorporation of AAR whose objective is to organize and hold conferences on automated reasoning.

(B) The term of office for officers of AAR is six years in duration; the term of office for the (current) Program Chairman of CADE is two years.

(C) An officer of AAR (president, vice-president, secretary, and treasurer) can hold consecutive terms, and is a member of the Board of Directors when holding the position of one of the officers of AAR. The Program Chairman of CADE cannot hold consecutive terms.

(D) If an officer of AAR is unable to complete a term, a replacement will be chosen by the Board of Directors of AAR; should a vote result in a tie, the president will cast the deciding vote. New officers will be elected by the members of AAR as their terms expire. Ballots must be mailed to AAR members at least 45 days before expiration of an officer's term. Ballots will list the persons nominated by the nominating committee, and will also provide for writing in names of other members of AAR.

(E) No salary will be paid to a member of the Board of Directors of AAR.

(2) Membership

(A) Members of AAR are those who have paid their dues, with the proviso that one can be as much as six months late and still be considered a member.

(B) Dues will be $5 per year or $9 for a two-year membership or $12 for a three-year membership, with the proviso that the dues can be increased in any year by as much as 10% if the Board of Directors makes that decision.

(3) Meetings

(A) No regular meetings of the Board of Directors and/or the entire membership will be held; a special meeting can be held if requested by a majority (defined here as 75%) of the respective group. The president can also call meetings. Voting on any issue will be by mail.

(B) A quorum is defined for the Board of Directors of AAR as 75%; for the entire membership, a quorum is also defined as 75%.

(4) Books, Records, and Fiscal Items

(A) The fiscal year will run from January 1 through December 31 of a calendar year.

(B) The treasurer will maintain records and books to account for dues and expenses, such as the costs incurred in producing a newsletter.

(C) The president will supervise the publication of the newsletter and have all other powers permitted by law. The president is also required to appoint a nominating committee to consist of at least three members of AAR.

(D) The secretary will maintain a file of all members and their status regarding dues and related matters, and will have all other duties and responsibilities assigned by the president to the secretary.

(E) The vice-president will assume the duties of the president should the president be unable to carry out the assigned responsibilities of the office of president. The holder of the offices of president and vice-president must be from different countries.

(F) AAR will in no way be responsible for any financial obligations for its subcorporations or affiliations, if any.

(5) Authorization and Amendments

(A) The treasurer is authorized to draw on the AAR funds to pay for any expenses; if the treasurer is unavailable, the secretary is given the authorization; if neither is available, the president has the authorization. The treasurer is also responsible for any duties assigned by the president to the treasurer.

(B) These bylaws may be amended by a vote of 75% of the AAR members, and the consent may be transmitted by mail or electronic mail or FAX.

Problems on Proving the Nonexistence of Finite Models of Fixed Sizes

Jian Zhang

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

1. Introduction

Experience has shown that it is beneficial to test mechanical theorem provers with hard problems. Typically, such problems are readily formulated in first-order logic. Some examples from number theory are given by Quaife (1991). In this paper, we introduce two hard problems from combinatorics.

The problems deal with the existence of certain objects having finite sizes, for example, Latin squares of order n. (A Latin square of order n is an n by n matrix in which each row (column) is a permutation of $1, 2, ..., n$.) Usually the number of candidates for the desired objects is finite. For example, the number of matrices that can be Latin squares is clearly less than $(n!)^n$. A straightforward approach to these problems is by exhaustive search (e.g., by backtracking), namely, enumerate all the candidates and check whether each has the desired properties. Sometimes this method does succeed in finding the desired objects, and the existence problem is answered affirmatively. However, if the oject does not exist, the exhaustive search may be too expensive. Even if one manages to examine all the candidates, it is difficult to assure other people of this fact. For a discussion of related issues, see Lam (1990).

If, on the other hand, automated theorem provers are used to obtain a proof of the nonexistence conjecture, and if the proof is not too long, then human mathematicians will be able to check the proof and accept it (in most cases).

2. The Problems

In this section, we state two combinatorial problems in the first-order logic language. For convenience, we introduce some notation.

For any predicate P and any fixed number n,

$\exists! x.P(x)$

there exists a unique x such that $P(x)$ holds.

$\exists x.[P(x) \wedge \forall y.(P(y) \rightarrow y = x)]$

$\exists!(x1, x2, ..., xn).P(x1, x2, ..., xn)$

there exist n distinct elements $xi (1 \leq i \leq n)$ such that $P(x1, x2, ..., xn)$ holds.

$\exists x1 \exists x2 ... \exists xn\ [... \wedge (xi \neq xj) \wedge ... \wedge P(x1, ..., xn)]$

$\exists \star (x, n).P(x)$

there exist exactly n distinct elements $xi (1 \leq i \leq n)$ such that each $P(xi)$ holds.

$\exists! x1 ... \exists! xn [... \wedge (xi \neq xj) \wedge ... \wedge P(x1) \wedge ... \wedge P(xn)]$

Problem 1. Idempotent Models of Quasigroup Identities

The first problem concerns the existence of certain models of quasigroup identities (Bennett 1989). A quasigroup is a set with a binary operation—the product, such that the equations $ax = b$ and $ya = b$ are uniquely solvable for every pair $< a, b >$ of elements. If a quasigroup satisfies an identity $l(x, y, ...) = r(x, y, ...)$, it is called a model of that identity. If, in addition, the identity $xx = x$ holds for every x, the model is called idempotent.

Models of the identity $((yx)y)y = x$ were studied by Bennett (1989). It was conjectured that there are no idempotent models of this identity having sizes 9, 13, ... Using backtracking search, Zhang (1991) "verified" the conjecture for size 9. But it still demands mathematical proof.

For this problem, the axioms are as follows (the function f denotes the product):

(QGI_1) $\forall u \forall v \exists! x.[f(u, x) = v]$
(QGI_2) $\forall u \forall v \exists! y.[f(y, u) = v]$
(QGI_3) $\forall x[f(x, x) = x]$
(QGI_4) $\forall x \forall y[f(f(f(y, x), y), y) = x]$

The conjecture is as follows.

Conjecture 1. The above formulas are not satisfiable in any domain that consists of n elements ($n = 9, 13, ...$).

(QGI_C) $\exists!(x1, ..., xn).\forall y[y = x1 \vee ... \vee y = xn]$

Problem 2. Finite Projective Planes

A projective plane consists of points and lines, in which some points lie on some line. A projective plane satisfies the following three axioms (Hall 1986: Sec. 12.3):

1. There is one and only one line containing two distinct points.
2. There is one and only one point common to two distinct lines.
3. There exist four points, no three of which are on a line.

A finite projective plane is said to be of order n if every line contains exactly $(n + 1)$ points. There is still no general method of determining the existence of finite projective planes of an arbitrary order n. After several years of computer search, C. W. H. Lam and his colleagues recently "solved" the nonexistence problem for the case $n = 10$, which had been a famous open problem in combinatorics. See Lam (1991) for details.

The above axioms can be translated directly into first-order formulas. (It may be more natural to use a many-sorted language.) We need three predicates: $P(x)$ — "x is a point"; $L(y)$ — "y is a line"; and $On(x, y)$ — "x is on y".

(PP_1) $\forall x1 \forall x2[P(x1) \wedge P(x2) \wedge (x1 \neq x2) \rightarrow \exists! y.[L(y) \wedge On(x1, y) \wedge On(x2, y)]]$

(PP_2) $\forall y1 \forall y2[L(y1) \wedge L(y2) \wedge (y1 \neq y2) \rightarrow \exists! x.[P(x) \wedge On(x, y1) \wedge On(x, y2)]]$

(PP_3) $\exists x1 \exists x2 \exists x3 \exists x4 [P(x1) \wedge P(x2) \wedge P(x3) \wedge P(x4) \wedge \neg Sh(x1, x2, x3)$
$\wedge \neg Sh(x1, x2, x4) \wedge \neg Sh(x1, x3, x4) \wedge \neg Sh(x2, x3, x4)]$
where $Sh(x, y, z) = \exists w[L(x) \wedge On(x, w) \wedge On(y, w) \wedge On(z, w)]$

In addition, we have

(A_1) $\forall x \neg [P(x) \wedge L(x)]$

(A_2) $\forall x \forall y [On(x, y) \rightarrow P(x) \wedge L(y)]$

The conjecture is as follows:

Conjecture 2. There is no projective plane of order n; that is, there is a line that does not contain exactly $(n + 1)$ points.

(PP_C) $\exists y[L(y) \wedge \neg \exists \star (x, (n + 1)).[P(x) \wedge On(x, y)]]$ $(n = 10, ...)$

3. Some Remarks

Most theorem provers work as follows: Negate the conjecture, add it to the set of axioms and lemmas, and then attempt to show that the resulting set is unsatisfiable. Moreover, most problems solved by such provers (see Wos 1988: Chapter 6) involve deduction of universal properties shared by any objects of a specific kind (e.g., any rings, any groups, ...). But the above two problems involve the unsatisfiability in domains of fixed sizes. Although they can also be embedded in the general framework of first-order logic, the representation is awkward. Is is practically possible to automatically prove the nonexistence of combinatorial objects of fixed sizes? Shall we develop some special methods? How can the problem be represented? And how can the combinatorial explosion be controlled?

Just as those presented by Quaife (1991), our problems are also difficult for human mathematicians. Some instances of the problems are open; others have been solved, perhaps with the aid of computers. Although we did not make any attempt to automatically prove the conjectures, because of the unavailability of powerful theorem provers, we believe it is worthwhile to do so, because it may offer convincing solutions to the open problems. Moreover, it will allow automated reasoning systems to compete not only with each other, but also with systems based on other techniques, such as backtracking search. As a starting point, one may try those smaller sizes for which a purely mathematical solution is known (e.g., $n < 9$ in problem 2).

References

Bennett, F. E. (1989) Quasigroup identities and Mendelsohn designs. *Canadian J. of Mathematics*, **41**, 341–368.

Hall, M., Jr. (1986) *Combinatorial Theory.* 2nd ed. John Wiley & Sons, New York.

Lam, C. W. H. (1990) How reliable is a computer-based prover? *Mathematical Intelligencer*, **12**, 8–12 (Winter).

Lam, C. W. H. (1991) The search for a finite projective plane of order 10. *The American Mathematical Monthly*, **98**, 305–318.

Quaife, A. (1991) Unsolved problems in elementary number theory. *J. Automated Reasoning,* **7**, 287–300.

Wos, L. (1988). *Automated Reasoning: 33 Basic Research Problems.* Prentice-Hall, Englewood Cliffs, New Jersey.

Zhang, J. (1991) Search for idempotent models of quasigroup identities. Unpublished manuscript.

Call for Papers

Logic at Work

Twenty years ago, logic was mainly applied to mathematical and philosophical problems. Today, the term *applied logic* has a far wider meaning, as numerous applications of logical methods in computer science, formal linguistics, and other fields testify. The aim of this conference, entitled Logic at Work, is to bring together those researchers—whether logicians with an interest in applications or workers in other fields with an interest in logical methods—whose work is best described as applied logic. The conference will be held at the University of Amsterdam on December 17–19, 1992.

The program committee welcomes theoretical and computational contributions in such areas as reasoning under change, reasoning under limited resources, reasoning under uncertainty, applications of logic in the social sciences, and methodological issues in applied logic.

The deadline for extended abstracts (8–12 pages) is September 15, 1992. Abstracts (preferably hard copy, prepared in LaTeX) should be sent to APLOC-conference, CCSOM, University of Amsterdam, Oude Turfmarkt 151, 1012 GC Amsterdam, The Netherlands (e-mail: aploc@ccsom.nl; fax: +31-20-626-4873).

AISB'93 Conference

The Society for the Study of Artificial Intelligence and the Simulation of Behaviour will hold its ninth biannual conference on March 29–April 2, 1993, at the University of Birmingham. The theme for invited papers is "Prospects for AI as the general science of intelligence."

Authors are welcome to submit papers or descriptions of work to be presented in a poster session. The deadline is September 1, 1992. For further information, send e-mail to aisb93-prog@cs.bham.ac.uk, or write to AIS'93, School of Computer Science, The University of Birmingham, Edgbaston, Birmingham, B152TT, U.K.

Preliminary Announcement: Eight Annual IEEE Symposium on LOGIC IN COMPUTER SCIENCE

June 20–23, 1993, Montreal, Quebec (Canada)

The LICS Symposium aims for wide coverage of theoretical and practical issues in computer science that relate to logic in a broad sense, including algebraic, categorical, and topological approaches.

Suggested, but not exclusive, 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, software specification, type systems, and verification.

The symposium has been sponsored by the IEEE Technical Committee on Mathematical Foundations of Computing in cooperation with the Association for Symbolic Logic and the European Association of Theoretical Computer Science.

Note that the conference will start on a Sunday.

LICS GENERAL CHAIR: Robert Constable, Department of Computer Science, Cornell University, Ithaca, NY 14253 (rc@cs.cornell.edu)

LICS'93 CONFERENCE CO-CHAIRS: Mitsu Okada, Computer Science Department, Concordia University, Montreal H3G 1M8, Quebec, Canada (okada@concour.cs.concordia.ca); and Prakash Panangaden, School of Computer Science, McGill University, Montreal H4B 2L5, Quebec, Canada (prakash@cs.mcgill.ca)

COMPLEX: A Reasoning System for Complexity Analysis

Shoshana Abel
San Francisco State University, San Francisco, Calif.

The following is an abstract of a paper that is planned for publication. Comments are welcome at sabel@toaster.sfsu.edu or (415) 338-2858.

Abstract: This paper elucidates a reasoning system called COMPLEX that uses divide-and-conquer algorithms and other list-processing algorithms as input and produces as output (1) their asymptotic complexity, (2) what these algorithms do, and (3) their typical-case complexity analysis. The algorithms are input in the form of pure Lisp and comprehended by the system by examining the input. The internal representation of the system's resulting knowledge of the algorithms takes the form of statements using epistemic/temporal logic. This representation permits the system to make inferences, using heuristics, that finally result in deducing the value of the time-complexity functions associated with such algorithms. The reasoning about asymptotic complexity in the case of divide-and-conquer algorithms is accomplished by utilizing the system's knowledge of the algorithm, together with the application of the results of a theorem incorporated in the knowledge base. The asymptotic complexity of the other list-processing algorithms is accomplished by the calculation of steps in reasoning.

Besides the asymptotic complexity analysis, COMPLEX also identifies what a given pure Lisp algorithm does (e.g., sorting), by reasoning, using heuristics, and by utilizing the system's knowledge of the algorithm. This information is necessary for the system to derive the typical-case complexity, by autonomously formulating the set of representative inputs that are suitable for the particular algorithm, according to what it does. The typical-case complexity is also obtained

by inference, given representative inputs, either by the user or generated by COMPLEX. An innovative feature of this system is that it has the capability of producing what a given algorithm does, as its output.

COMPLEX helps a user, who is typically a software developer, to obtain the complexity results of a collection of algorithms and also to procure comparative results, given several algorithms.

Incompleteness of the RUE/NRF inference systems

Maria Paola Bonacina and *Jieh Hsiang*
({bonacina,hsiang}@sbcs.sunysb.edu)

Introduction

The *Resolution with Unification and Equality (RUE)* and the *Negative Reflective Function (NRF)* inference rules were proposed in [2] as a generalization of *E-resolution* [3, 1]. These two rules are defined first in a very general form, called the *RUE-NRF inference system in open form*, which is proved to be refutationally complete for first-order logic with equality [2]. The RUE-NRF inference system in open form does not represent a mechanical inference system, because it does not specify completely how to compute the inferences. Thus, several refinements intended to turn the RUE-NRF schemes into concrete inference systems are given in [2]. In this note we show that these refinements are not complete.

The RUE-NRF inference rules

Given two clauses $\neg P(t_1 \ldots t_n) \vee A$ and $P(s_1 \ldots s_n) \vee B$, an RUE step consists in generating a resolvent $(A \vee B)\sigma \vee D$:

$$RUE\ \frac{\neg P(t_1 \ldots t_n) \vee A,\ P(s_1 \ldots s_n) \vee B}{(A \vee B)\sigma \vee D}$$

where σ is any substitution and D is a disjunction of inequalities, obtained by applying any number of *Decomposition* steps

$$Decompose\ \frac{f(l_1 \ldots l_n) \neq f(r_1 \ldots r_n)}{l_1 \neq r_1 \vee \ldots \vee l_n \neq r_n}$$

to the inequalities in the disjunction $(t_1 \neq s_1 \vee \ldots \vee t_n \neq s_n)\sigma$. If the two atoms $P(t_1 \ldots t_n)$ and $P(s_1 \ldots s_n)$ unify, the substitution σ is their most general unifier, the set D is empty and RUE reduces to resolution. Similarly, an NRF step

$$NRF\ \frac{t \neq s \vee A}{A\sigma \vee D}$$

applies a substitution σ and any number of Decomposition steps to the inequality $t\sigma \neq s\sigma$ in a given clause. If the two sides of the inequality $s \neq t$ unify, the substitution σ is their most general

unifier, the set D is empty and NRF reduces to resolution with $x = x$. The disjunction D is called a *disagreement set* in both RUE and NRF. These inference rules are said to be in *open form* [2], because they do not specify how the disagreement set D and the substitution σ are computed. The *RUE-NRF inference system in open form* [2] is obtained by adding to the RUE and the NRF inference rules a *Factoring* rule, modified in a similar way:

$$Factoring\ \frac{P(t_1 \ldots t_n) \vee P(s_1 \ldots s_n) \vee A}{(P(t_1 \ldots t_n) \vee A)\sigma \vee D}$$

where D is derived by applying any number of Decomposition steps to $(t_1 \neq s_1 \vee \ldots \vee t_n \neq s_n)\sigma$ like in RUE. This set of rules is complete for first-order logic with equality with no need of the axioms of equality [2]. Intuitively, this is because the axioms for reflexivity, transitivity, and substitutivity (the last ones are also called functional reflexive axioms) are sort of implicitly applied in the RUE-NRF rules. For instance, the basic reflexivity axiom $x = x$ is not needed, because resolution with $x = x$ is a special case of the NRF rule. On the other hand, the authors themselves observe that, in order to implement symmetry, RUE needs to be applied twice, whenever it is applied to a pair of complementary equality literals. Given clauses $s_1 = s_2 \vee A$ and $t_1 \neq t_2 \vee B$, one generally needs to derive two RUE-resolvents: $(A \vee B)\sigma \vee D$, where D is a disagreement set of $s_1 \neq t_1 \vee s_2 \neq t_2$, and $(A \vee B)\sigma' \vee D'$, where D' is a disagreement set of $s_1 \neq t_2 \vee s_2 \neq t_1$.

The RUE-NRF system in open form is complete without the axioms of equality, because it embeds them in a sort of straightforward way. The price to pay for completeness is the very high degree of generality and nondeterminism of the inference rules. The RUE-NRF system in open form is too general to be mechanized effectively, as the disagreement set D and the substitution σ can be arbitrary. Consequently, most of the work illustrated in [2] consists in designing criteria to compute D and σ.

The viability criterion is incomplete

The rule which selects the disagreement set determines the number of decomposition steps to be embedded in an RUE-NRF step. All the criteria proposed in [2] are based on a notion of *viable disagreement set*. A disagreement set $D = \{s_i \neq t_i\}_{i=1}^{n}$ is *viable* if and only if it can be partitioned into two sets D_1 and D_2, such that

1. for every side s of an inequality in D_1 there is a positive equality literal $l = r$ in S such that
   (a) either s and l unify
   (b) or s and l have the same topmost function symbol and there is a viable disagreement set below the pair (s, l)
2. there exists a unifier σ such that $s_i\sigma = t_i\sigma$ for all $s_i \neq t_i \in D_2$.

The viability condition seems to be designed with the purpose of guaranteeing that at least one step can be done in order to erase the generated inequalities. All the pairs (s_i, t_i) in D_2 can be unified and deleted in the same RUE-NRF step where they are generated. For a pair (s_i, t_i) in D_1, it is possible to perform an RUE step with the clause in which the literal $l = r$ occurs.

The basic refinement of the RUE-NRF system in open form, proposed in [2], consists in requiring that an RUE-NRF step is performed only if there is a viable disagreement set. Also, it is claimed that if the *topmost viable disagreement set* is selected at each step, the RUE-NRF system is still complete for first-order logic with equality without the axioms of equality. Choosing the topmost viable disagreement set means that decomposition is applied, within an RUE-NRF step, until it generates either an inequality $f(l_1 \ldots l_n) \neq g(r_1 \ldots r_n)$ with different topmost function symbols or an inequality $f(l_1 \ldots l_n) \neq f(r_1 \ldots r_n)$, which is viable. The following example contradicts the above claim, showing that if RUE-NRF is restricted to viable disagreement sets, the RUE-NRF system is not complete without the functional reflexive axioms:

Example 1 *Let S be $\{g(f(a)) = a, f(g(x)) \neq x\}$. This set is inconsistent, as can be seen by paramodulating $g(f(a)) = a$ into $f(g(x)) \neq x$:*

$$\frac{g(f(a)) = a, f(g(x)) \neq x}{f(a) \neq f(a)}$$

The applied unifier is $\{x \leftarrow f(a)\}$. The possible disagreement sets for an RUE step are $\{x \neq g(f(a)), f(g(x)) \neq a\}$ and $\{x \neq a, f(g(x)) \neq g(f(a))\}$. Neither one is viable, because of the inequalities including the term $f(g(x))$. Since there is no equation whose side starts with the function symbol f, the term $f(g(x))$ does not satisfy the viability requirements. The disagreement set $\{f(g(x)) \neq x\}$ for an NRF step is not viable either, since x occurs in $f(g(x))$, so that the two terms do not unify. No RUE-NRF step with a viable disagreement set can be applied. Thus, the restriction to viable disagreement sets is not complete. A topmost viable RUE-NRF refutation can be obtained by adding the functional reflexive axiom $f(y) = f(y)$: RUE applied to $x \neq f(g(x))$ and $f(y) = f(y)$ yields $x \neq f(y) \vee f(g(x)) \neq f(y)$, which is then reduced by NRF to $x \neq f(y) \vee g(x) \neq y$. This can be resolved with $g(f(a)) = a$ to yield the contradiction $f(a) \neq f(a)$.

The RUE-NRF system restricted to viable disagreement sets needs the functional reflexive axioms to handle *collapse equations*, that is, equations where one side is a variable. The equation $f(g(x)) = x$ which occurs negated in the above example is a collapse equation. The authors of [2] did observe that the filtering effect of the viability criterion is lost, if a collapse equation occurs in S: since every term unifies with a variable, every pair of terms is viable. However, they did not observe that completeness fails to hold in the presence of collapse equations. Intuitively, the equality reasoning in the RUE-NRF system is done by decomposition, which handles equations in a top down style. This kind of step does not apply to collapse equations, where one side is unknown. The functional reflexive axioms allow to build the term that needs to be assigned to the variable side in order to obtain a refutation. Unfortunately, the functional reflexive axioms make viable any disagreement pair, so that the need for the functional reflexive axioms defeats the purpose of the viability restriction.

The lowest reducible disagreement set heuristic is incomplete

A number of additional variations of the RUE-NRF system are described in [2]. Some of them are not claimed to preserve completeness and are given simply as heuristics. One such heuristic is to choose the *lowest reducible disagreement set.* Let E_S denote the set of all the positive equations

which appear, not necessarily as unit clauses, in the given set of clauses S. Decomposition within the RUE-NRF steps is applied until it generates either an inequality $f(l_1 \ldots l_n) \neq g(r_1 \ldots r_n)$ with different topmost function symbols or an inequality $f(l_1 \ldots l_n) \neq f(r_1 \ldots r_n)$, such that for some i, $1 \leq i \leq n$, $l_i \neq r_i$ is E_S-irreducible. This heuristic may help preventing the generation of trivially irreducible inequalities, such as $a \neq b$, where a and b are Skolem constants. However, if the RUE-NRF system is restricted to generate only disagreement sets of E_S-reducible inequalities, it becomes trivially incomplete:

Example 2 *Given* $S = \{f(g(a,x)) \neq f(g(x,a)), g(b,y) = g(y,b)\}$, *the inequality* $g(a,x) \neq g(x,a)$, *which leads to the refutation, cannot be generated because neither of its sides is* E_S*-reducible.*

All the other RUE-NRF systems that are claimed or conjectured to be complete in [2] include the restriction to viable disagreement sets and thus are not complete without the functional reflexive axioms. We conjecture that RUE-NRF with the topmost viable disagreement set may be complete for input sets of clauses S whose equational component E_S axiomatizes a *simple* theory. An equational theory is *simple* if any equation $s = t[s]$ (i.e., an equation where one side occurs as strict subterm of the other one) is unsatisfiable in the theory. Clearly, a simple theory does not contain collapse equations. However, a restriction to simple theories may be regarded as too strong, casting a shadow on the usefulness of such a result.

References

[1] R. Anderson, Completeness Results for E-resolution, in *Proceedings of the AFIPS 1970 Spring Joint Computer Conference*, AFIPS Press, Reston, Va., 653–656, 1970.

[2] V. J. Digricoli and M. C. Harrison, Equality-Based Binary Resolution, *Journal of the ACM*, Vol. 33, No. 2, 254–289, April 1986.

[3] J. B. Morris, E-resolution: An Extension of Resolution to include the Equality Relation, in D. E. Walker and N. M. Norton, eds., *Proceedings of the IJCAI'69*, 287–294, 1969.