ASSOCIATION FOR AUTOMATED REASONING
NEWSLETTER

No. 22 March 1993

From the AAR President, Larry Wos...

This issue is somewhat unusual in the large number of conference announcements. I am delighted to see that research in automated reasoning and related areas is so active.

I am equally pleased to include the announcement about the new journal *Experimental Mathematics.* As AAR members well know, I have long been an advocate of the importance of experimentation. Too often, researchers in automated reasoning have dwelt on theoretical advances alone. I, on the other hand, believe strongly that advances in the field of automated reasoning will come only with advances in all three areas—theory, implementation, and experimentation.

To encourage you to experiment, we include in this issue two articles by our readers. The first, immediately following my comments, discusses a decision procedure called Sato. The author, Hantao Zhang, invites you to obtain a copy of Sato by anonymous ftp and to use it as a research tool to attack open questions or to implement new strategies. The second article, at the end of this issue, discusses LD-algebras. Thomas Jech reports on his experiences with OTTER and presents several open problems in embedding algebras. I encourage you to try these problems and to let us know of your own results.

Sato: A Decision Procedure for Propositional Logic

Hantao Zhang
hzhang@cs.uiowa.edu

A decision procedure called Sato for testing satisfiability of ground clauses is available at the University of Iowa. Sato, written in C, runs on any Unix system. Sato accepts as input a list of clauses, each of which is abstracted as a list of integers, and tells the user whether the input clauses are satisfiable. The user has options to let Sato search/print one or more models when the clause set is satisfiable.

The main algorithm in Sato is essentially the Davis-Putnam procedure [1] with some ideas from the constraint satisfaction approaches. In particular, the clauses are organized as a discrimination tree instead of list. Some resolutions are performed on the tree before the main search procedure is started. The details of Sato's algorithms will be given in a forthcoming report.

The performance of Sato has been manifested by a number of benchmark problems such as logic puzzles or constraint satisfaction problems. In [2], Lee and Plaisted collected a list of propositional and near-propositional problems to illustrate that their hyper-linking prover, CLIN, has much better performance than OTTER and PTTP. Sato is about 50 times faster than CLIN on these problems (a large portion of the difference may be due to the difference of the implementation languages: CLIN is written in Prolog while Sato is in C). See the table for the performance comparison.

	Sato		CLIN	
problem	clauses	run time	clauses	run time
ph4	22	0.02	22	0.60
ph5	45	0.04	45	1.80
ph6	81	0.16	81	7.30
ph7	133	0.78	133	39.60
ph8	204	4.92	204	276.46
ph9	297	40.36	297	2266.61
ph10	415	378.20		
nonobv	102	0.10	6	17.18
latinsq	173	0.10	16	56.35
school	20	0.04		
jobs	273	4.70	45	331.00
salt	63	0.08	44	28.01
zebra	2975	53.62	128	866.11
lion	163	0.10	51	231.00

The run times (in seconds) are all measured on a Sun 3/60 workstation with 12 MB of memory. CLIN accepts general clauses while Sato accepts only ground clauses. For near-propositional problems, we have to instantiate general clauses before Sato is called. That is why the number of clauses for the two provers are different. The time for instantiating general clauses into ground ones is not counted for Sato.

The first seven problems, 'ph4' through 'ph10', are the pigeonhole problems ('ph4' means 4 objects in 3 holes). They are propositional problems. 'nonobv' is Pelletier and Rudnicki's nonobvious problem. 'latinsq' is the latin square problem. 'school' (the school boy's puzzle), 'salt' (the salt and mustard puzzle), 'jobs' (the job assignment puzzle) and 'zebra' are Lewis Carroll's famous puzzles. 'lion' is Raymond Smullyan's lion and unicorn puzzle. Please refer to [2] for the sources of these problems.

A good feature of Sato is that it not only tells whether a set of clauses is satisfiable, but also outputs a model when the clause set is satisfiable. For instance, after we encoded the n-queen problem into the propositional logic, Sato then found all the solutions of the n-queen problem (at the rate of 100 solutions per second on average on a sparc workstation 2 for large n's).

Sato could serve as a research tool in the future. For instance, it was an open problem two year ago whether there exists a Bennett quasigroup of order 9. Sato becomes the fourth system to solve this problem (after encoding the problem by 61498 ground clauses and running for 25 hours on a sparc 2 workstation). Sato also can be integrated into a resolution-based theorem prover like OTTER [3] or used to implement Lee and Plaisted's hyper-linking strategy [2].

To obtain a copy by ftp, connect to herky.cs.uiowa.edu, with username anonymous, and with your e-mail address as password. Go to public/sato, and follow the directions in REAMDE. For further information, write to Hantao Zhang (hzhang@cs.uiowa.edu), Dept. of Computer Science, University of Iowa, Iowa City, IA 52242.

Acknowledgments: Some of OTTER's utility functions [3] were used in Sato. The work was partially supported by NSF (INT-9016100, CCR-9202838).

References

[1] Davis, M., Putnam, H.: (1960) A computing procedure for quantification theory. J. of ACM, 7, 201–215.

[2] Lee, S. J., Plaisted, D. A.: (1992) Eliminating duplication with the hyper-linking strategy. J. of Automated Reasoning, Vol. 9, no.1, 25–42.

[3] McCune, W. W.: (1990) OTTER 2.0 users' guide. Argonne National Laboratory Report ANL-90/9, Argonne, Illinois.

Call for Papers

CSL'93

The conference *Computer Science Logic '93* will be held September 13–17, 1993, at the University College of Swansea, Wales.

CSL '93 is the 1993 annual conference of the newly formed European Association for Computer Science Logic (EACSL). The conference took place in previous years under the title Workshop on Computer Science Logic at Karlsruhe (1987), Duisburg (1988), Kaiserslautern (1989), Heidelberg (1990), Bern (1991) and Pisa (1992). The conference is intended for computer scientists whose research activities involve logic, as well as for logicians working in areas related to computer science.

The scientific program will consist of invited lectures and contributed papers selected from those submitted. Invited speakers include J. A. Bergstra (Amsterdam), A. Blass (Michigan), E. Grädel (Basle), I. Hodkinson (Imperial College), P. Martin-Löf (Stockholm), R. Milner (Edinburgh), W. Thomas (Kiel), and S. S. Wainer (Leeds).

All contributions will be refereed for a proceedings volume, to be published by Springer Verlag in the *Lecture Notes in Computer Science Series.* Six copies of an extended abstract (up to 5 pages) of papers to be submitted should be sent to the program committee chairman (K. Meinke) by June 1, 1993. Correspondence should be sent to Dr. K. Meinke (CSL '93), Department of Computer Science, University College of Swansea, Swansea SA2 8PP, Great Britain (e-mail: csl93@pyr.swan.ac.uk).

HOA '93

An international workshop on higher-order algebra, logic, and term rewriting will take place September 23–24, 1993, in Amsterdam, The Netherlands.

Higher-order methods are being increasingly applied in functional and logic programming languages, as well as in specification and verification of programs. The algebra and model theory of higher-order languages, computational logic techniques including resolution and term rewriting, and specification and verification case studies using higher-order techniques are particularly vigorous areas of research at present. This workshop aims to provide an overview of current areas of research, and to suggest new research directions. The scope of the workshop includes higher-order aspects of

- algebra and model theory,
- term rewriting,
- specification and verification languages,
- computational logics, and
- system implementations.

Three copies of an extended abstract (up to 4 pages) of papers to be submitted should be sent to the program committee chairman, B. Moller, to arrive not later than June 15, 1993. Complete drafts will be externally refereed for the final conference proceedings.

Prof. Dr. B. Moller (HOA '93)
Institut fur Mathematik
Universitaet Augsburg
Universitaetsstr. 2 W-8900 Augsburg
Germany
E-mail: moeller@uni-augsburg.de
Fax: +49 821 598 2200

Symposium on Quantifier Elimination and Cylindrical Algebraic Decomposition

The first symposium on quantifer elimination and cylindrical algebraic decomposition will be held on October 6–8, 1993, in Linz, Austria, at the Research Institute for Symbolic Computation, Johannes Kepler Universitaet. The symposium is in honor of the sixty-fifth birthday of George E. Collins, one of the most important contributors to research on algorithms for quantifier elimination, cylindrical algebraic decomposition, and computer algebra in general.

The organizing committee invites submission of papers presenting original research on topics related to the symposium themes, including

- algorithmic aspects of quantifier elimination

- cylindrical algebraic decomposition
- algorithmic aspects of real geometry
- complexity of decision theories for elementary algebra and geometry
- related computer algebra algorithms
- implementations and systems in the symposium theme areas
- applications of these techniques to mathematics and to high-tech problems

The symposium will have two parts. The first part, on October 6, will be an introductory workshop on quantifier elimination and cylindrical algebraic decomposition. The workshop will be conducted by Prof. Collins and members of his research group. On Thursday and Friday, the second part of the symposium will be devoted to research presentations — both invited and submitted.

Authors should send three copies of papers before April 15, 1993, to the co-chair of the programming committee, at Jeremy R. Johnson, Department of Mathematics and Computer Science, Drexel University, Philadelphia, PA 19104 USA (e-mail: jjohnson@mcs.drexel.edu; phone: (215) 895-2893).

The symposium proceedings will be published by Springer-Verlag.

ILPS'93

The International Logic Programming Symposium'93 will be held in Vancouver, Canada, on October 26–29, 1993. This is is the tenth in a series of international symposia on logic programming held annually in North America. The technical program for the conference will include tutorials, invited lectures, and presentations of refereed papers and posters. Workshops will be scheduled immediately following the conference.

Submissions of papers are invited on all aspects of logic programming, including

algorithmic analysis	deductive databases
artificial intelligence	logical extensions
compilation techniques	natural language
constraints	proof theory
concurrency and parallelism	semantic analysis
environmental support	programming methodologies
implementations and architectures	programming language design
meta and higher-order programming	

Papers must be 15 pages or less and must contain a cover page including a 200-word abstract, keywords, and postal and electronic mailing addresses (as well as phone numbers and fax numbers, when available. The proceedings will be published by MIT Press. Please mail six copies of each submission by April 16, 1993, to

Dale Miller
Department of Computer Science
200 South 33rd Street
University of Pennsylvania
Philadelphia, PA 19104-6389 USA

Phone: +1 215 898 1593
FAX: +1 215 898 0587
dale@saul.cis.upenn.edu

Special issue of *Fundamenta Informaticae* on Term Rewriting Systems

A special issue of *Fundamenta Informaticae*, on term rewriting systems, is scheduled to appear in 1994. The editor is David Plaisted.

For this issue, all aspects of term-rewriting systems will be considered. Interested authors should send four copies of their manuscripts to the following address. Please include your e-mail address if appropriate.

David A. Plaisted
Department of Computer Science
CB No. 3175
352 Sitterson Hall
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina 27599-3175
plaisted@cs.unc.edu.

The deadline for submission is August 1, 1993.

Contributions should be in English. Manuscripts should be double spaced with wide margins. A duplicate copy should be retained by the author. The author's mailing address should appear on the manuscript.

To assure rapid dissemination, publication in Fundamenta Informaticae is based on camera-ready copy supplied by the authors. Upon acceptance of a paper, detailed instructions for the format of this camera-ready copy will be given. There are no page charges.

Fifth International School for Computer Science Researchers
Specification and Validation Methods for
Programming Languages and Systems
Lipari Island, June 21 - July 3, 1993

The Fifth School for Computer Science Researchers addresses Ph.D. students and young researchers who wish to get exposed to the forefront of research activity in the field of specification and validation methods. The school will be held in the beautiful surroundings of the island of Lipari, which can be reached by ferry from Naples, Milazzo, Messina, Reggio Calabria, and Palermo.

Six courses will be offered, of which each student must choose four. A proficiency final exam at the end of each chosen course is mandatory. Saturday of the first week will be entirely dedicated to open research problems and discussion.

Inexpensive accommodations for students will be provided (20,000 lira per night). Registration fees for non-Italian students are 200,000 lira; for Italian students, 500,000 lira. A limited number of fellowships for students coming from disadvantaged regions will be available. A maximum of thirty students will be allowed.

Applications should be sent by March 31, 1993, to

Prof. Alfredo Ferro
Dipartimento di Matematica
Citta' Universitaria
Viale A. Doria, 6
95125 Catania, Italy
Tel. 39-95-222222 / 330533 (ext.663)
Fax: 39-95-330094
e-mail:SCHOOL@MATHCT.CINECA.IT

The application should contain a curriculum vitae plus at least two letters of recommendation (but no papers). Students will be informed about acceptance by April 20, 1993. Those interested in getting a quick response should include an e-mail address.

List of speakers:

A. Pnueli (Israel): "Application of temporal logic to the specification and verification of reactive and real-time systems"

Y. Gurevich (USA): "Operational semantics based on evolving algebras"

K. Apt (The Netherlands): "Declarative and procedural interpretations for logic programming languages"

M. Fourman (United Kingdom): "System specification and development using higher-order logic"

W. Damm (Germany): "Specification and verification of VDHL-based hardware design"

E. Boerger (Italy): "Evolving algebra based specification and verification of logic programming systems"

Directors of the school:

Alfredo Ferro	Egon Boerger
Dipartimento di Matematica	Dipartimento di Informatica
Catania	Pisa

The school is organized under the auspices of the European Association for Computer Science Logic (EACSL).

News about Journals

Annals of Pure and Applied Logic

The editors of the *Annals of Pure and Applied Logic* welcome submissions in theoretical computer science and the foundations of computing. Many articles in theoretical computer science have already been published on such topics as domain theory, linear logic, lambda calculus, type-theory and complexity theory—including a special issue (and another forthcoming) of selected papers presented at the IEEE Symposium on Logic in Computer Science. The current backlog (time from final acceptance to publication) is six months—considerably less than most other journals covering the area. For further information, please contact the publisher:

A. Sevenster
Mathematics, Computer Science and Cognitive Science
Elsevier Science Publishers
P.O. Box 103
1000 AC AMSTERDAM
e-mail: A.SEVENSTER@elsevier.nl

Experimental Mathematics

Experimental Mathematics is a new journal founded in the belief that theory and experiment feed on each other and that the mathematical community stands to benefit from a more complete exposure to the experimental process.

The essential ingredients of a paper published in *EM* are two: some experimental aspect, and relevance to mathematics proper. The word "experimental" is conceived broadly: many mathematical experiments these days are carried out on computers, but others are still the result of pencil-and-paper work, and there are other experimental techniques, like building physical models. As for the second ingredient, the focus is on work that will have a theoretical impact and contribute to the development of mathematical ideas.

Of particular interest are

- experiments that give rise to new theorems or new conjectures, or lend support to existing conjectures, or point to areas that ought to be investigated

- new theorems proved with the help of experimental results. (Authors should submit both the formal proofs and information about the experiments.) Ideally, the programs used for the experiment should be made freely available in electronic form to other researchers.
- algorithms for the solution or exploration of mathematical problems, including theoretical or experimental analyses of complexity
- practical issues, such as techniques and pitfalls involved in experimentation, if they present an original contribution and have a core of mathematical interest
- descriptions of computer programs for mathematical investigation.
- surveys of particular areas of mathematics, from the experimental point of view
- essays, humor, polemic, correspondence.

For a free copy of the first issue (while supplies last), subscription orders, author guidelines, or any other information, contact

Experimental Mathematics
AK Peters
289 Linden St
Wellesley, MA 02181

phone: (617) 235-2210
fax: (617) 235-2404

e-mail: `expmath@geom.umn.edu`

LD-algebras

Thomas Jech
jech@math.psu.edu

We consider algebras with one binary operation $\cdot$ satisfying the left distributive law $a\cdot(b\cdot c) = (a\cdot b)\cdot(a\cdot c)$. Whenever possible, we use concatenation ab in place of $a\cdot b$, and adopt the convention

$$abc = (ab)c.$$

Therefore, $abcde = (((ab)c)d)e$, etc. A left-distributive algebra is *monogenic* if it is generated on one generator, and the generator is usually denoted by the symbol 1. One can also introduce terms $2 = 1\cdot 1$, $3 = 2\cdot 1$, $4 = 3\cdot 1$ and so on; note that 3 is not necessarily equal to $1\cdot 2 = 1\cdot(1\cdot 1) = 2\cdot 2$.

LD-algebras (left-distributive algebras) have been studied extensively by set theorists in the past few years, particularly the free monogenic LD-algebra that is related to the subject of large cardinals. As many results have been proved using just a few simple assumptions, the subject is a natural candidate for an application of an automated reasoning program. In this article I describe some of my attempts to use OTTER.

The word problem for the monogenic LD-algebra has been solved first by Laver using large cardinals, and last year by Dehornoy, whose proof is elementary (precisely, can be formalized in primitive recursive arithmetic). It is unrealistic to expect OTTER to solve the word problem, but it should be able to verify true equations and possibly prove some simple rules.

I gave OTTER various simple (true) equations to verify and was disappointed to find that it is very slow at this. The following two problems are typical examples:

Problem 1. Verify $3 \cdot 2 \cdot U = UUU$, where $U = 2 \cdot 2$.

This is fairly straightforward, yet OTTER took about 40 seconds to prove it.

Problem 2. Verify $3 \cdot 2(U2)(UU(UU)) = U1(U3)(UU(UU))$.

OTTER could not prove this in 180 seconds. (I imposed a limit of 180 seconds in all the proofs but in many cases raised the limit and brought my Sun workstation to a halt.)

Left Segments

The concept of left segments has been instrumental in many results on LD-algebras. We say that x is a *left segment* of y if $y = xu_1u_2...u_k$ for some $u_1, ...u_k$. Dehornoy proved that the free monogenic LD-algebra is linearly ordered by this relation. Again, it is unrealistic to expect OTTER to prove this, so I gave it to verify various instances of words that are left segments, with the same mixed results. The following two examples are typical:

Problem 3. Show that 3 is a left segment of $U = 2 \cdot 2$.

This rather trivial fact took OTTER 10 seconds, while it was unable to prove the following (which is not quite easy):

Problem 4. Show that $3 \cdot 2(U2)$ is a left segment of $U1(U3)$.

Embedding Algebras

As I mentioned earlier, LD-algebras are related to large cardinals. Under a very strong large cardinal assumption, the free monogenic LD-algebra can be represented by an algebra of elementary embeddings. Theorems about this algebra can be proved from a small number of properties, suggesting the definition of an *embedding algebra.* Even though axioms for embedding algebras include additional properties to those listed below, many results can be proved from these axioms. The axioms are simple enough to hope that a theorem prover might be useful.

Definition. An embedding algebra is an LD-algebra whose members, *embeddings,* are increasing functions on a linearly ordered set of *ordinals.* Every embedding a has a *critical point* $\mathrm{cr}(a)$, an ordinal such that $a(\gamma) = \gamma$ if $\gamma < \mathrm{cr}(a)$ and $a(\gamma) > \gamma$ if $\gamma \geq \mathrm{cr}(a)$. Embeddings satisfy the following equations:

$$a(b\gamma) = ab(a\gamma), \quad \mathrm{cr}(ab) = a\mathrm{cr}(b).$$

(Here and thereafter we write $a\gamma$ for $a(\gamma)$, etc.)

This definition can be formally presented in the following form, where we do not distinguish between embeddings and ordinals, so xy means either $x \cdot y$ (when both x and y are embeddings) or $x(y)$ (when x is an embedding and y is an ordinal).

Axioms

(A1) $x(yz) = xy(xz)$

(A2) $\mathrm{cr}(uv) = u\mathrm{cr}(v)$

(B1) $x \not< x$

(B2) $x < y \vee y < x \vee x = y$

(B3) $x < y < z \rightarrow x < z$

(B4) $x < y \rightarrow ux < uy$

(C1) $x = ux \vee x < ux$

(C2) $x < \mathrm{cr}(u) \rightarrow ux = x$

(C3) $x \not< \mathrm{cr}(u) \rightarrow x < ux$

(D1) $x < y \rightarrow x < uy$

(D2) $ux = x \rightarrow vux = x$

The axioms (C1), (D1), and (D2) are redundant. The reason why I include them is that axioms (B2) and (B3) cause enormous problems when applying OTTER. So I always leave them out, but I need (C1) quite often, which cannot be proved without (B2). Occasionally I use (D1) or its consequence (D2); neither can be proved without (B3), and (D2) is easier to use than (D1), which I also leave out whenever possible.

I have been experimenting with OTTER, applying it to simple formulas that we know can be proved from the axioms. Problems 5–8 describe my attempts at two such formulas.

Problem 5. Let $\gamma = \mathrm{cr}(t)$. Show that $tt(ts\gamma) < t(ts\gamma)$ (for any s).

Problem 6. Again, let $\gamma = \mathrm{cr}(t)$, and show that $ts\gamma$ is not in the range of t, that is, $ts\gamma \neq t\alpha$, for any α.

Problem 6 is a very easy consequence of Problem 5, yet OTTER cannot prove either (at least I did not succeed in making it prove it in less than 180 seconds). When dissecting the proof (by a human), I found that OTTER did not find the equation in Problem 7:

Problem 7. $t(ts\gamma) = tt(ts)(t\gamma)$.

I found it incredible; when I left out all axioms except (A1), OTTER proved it easily:

Problem 8. Prove the equation in Problem 7 just from (A1).

It seems that OTTER becomes much less able when irrelevant information is included. This seems to prevent it from solving Problem 5. I think that I cannot make much progress unless I know how to handle this.

Problem 9 is another easy consequence of the axioms that OTTER could not prove:

Problem 9. Assume that $\mathrm{cr}(t) = \mathrm{cr}(T = \gamma$. If $T\alpha < Ts\gamma$, then $t\alpha < ts\gamma$.

I spent a lot of time on Problem 10; I hoped to learn how OTTER works, and the choice of axioms and flags is the result of this experimentation. I am very grateful to Dale Myers for answering my many questions about OTTER.

Problem 10. Let $\gamma = \mathrm{cr}(t)$. If $\gamma < s\gamma$, then $st(ts)\gamma < stt(s\gamma)$.

OTTER still cannot prove this. When analyzing the proof, I discovered a curious difficulty:

Problem 11. Same assumptions as in 10; prove $stts(sttt)(stts)\gamma < stt(s\gamma)$.

This OTTER could not prove, even though it proved Problem 12 and Problem 13, both rather quickly:

Problem 12. Under same assumptions, $stts(sttt)(stts)\alpha < stt(s\gamma)$, where $\alpha = stts(sttt)\gamma$.

Problem 13. Under same assumptions, $stts(sttt)\gamma = \gamma$.

So, while proving Formula 12 involving a term α and also proving that $\alpha = \gamma$, OTTER was not able to prove the formula for γ!

Open Problems

If a suitable choice of flags can be found so that OTTER is comfortable with the above problems, one might hope that it could prove some of the conjectures we have made about embedding algebras. I present three; in each problem we assume that the embedding algebra is monogenic (which, of course, is an assumption of a different nature from the axioms A–C).

Conjecture 1. If $\gamma = \mathrm{cr}(t)$, then $aa\gamma \leq a\gamma$.

This is in fact true for every elementary embedding (in the large cardinal theory) and for every ordinal. We are looking for an algebraic proof. Such is lacking even when a is the generator (in other words, prove that $2\gamma \leq 1\gamma$.)

Conjecture 2. Let $\gamma = \mathrm{cr}(t)$ and $\delta = \mathrm{cr}(s)$. If $t1\gamma < t\delta$, then $1\gamma \leq \delta$.

There is much experimental evidence for Conjecture 2. Conjecture 3 is a translation into the language of embedding algebras of something that is suspected and that would considerably simplify the theory:

Conjecture 3. For all a and c there exist b and d such that

$$\mathrm{cr}(c(adc) < \mathrm{cr}(c(a(dbd)))$$

and

$$\mathrm{cr}(dbd(da)(dbd)) < \mathrm{cr}(dbd(dc)).$$