ASSOCIATION FOR AUTOMATED REASONING
NEWSLETTER
No. 32 March 1996

From the AAR President, Larry Wos...

This issue begins with three notes about CADE. The first is a reminder that an important vote on the new bylaws will be held at the 1996 meeting. The second note is about the proposed site for CADE-15. The third is a call for nominations for the prestigious Herbrand award, to be presented at CADE-13. Since CADE is one of the leading conferences in automated deduction, I urge AAR members to read each note most carefully.

Also included in this issue of the *AAR newsletter* is an article by one of our frequent contributors, Li Dafa. Using his ANDP system, he has now produced an automated natural deduction proof of the formalization of the so-called halting problem—a problem that has appeared periodically in our newsletter since 1987.

A request: I would like to know whether you, as an AAR member, would prefer to "receive" our newsletter on the Web rather than in paper form. Specifically, we would have *AAR Newsletter* Web pages that would include new articles as they were accepted. We would give hot links to appropriate calls for papers. Please let our editor Gail Pieper (pieper@mcs.anl.gov) know your opinion on this matter.

CADE-13 to Hold Vote on New Bylaws

A vote will be held at the 1996 meeting of CADE to determine whether a new set of bylaws should be adopted for CADE. The vote will be by secret ballot.

So that the voters can make an informed decision, we have put the proposed bylaws and the existing ones on the World Wide Web. See http://www.cs.albany.edu/~nvm/cade.html.

Proposals for Sites for CADE-15 Solicited

Alan Bundy, President, CADE Inc.

CADE Inc. invites proposals to host the 15th Conference on Automated Deduction (CADE-15). CADE-15 will be held in early to mid-summer 1998 in Europe. Proposals are due by July 1, 1996, and a final decision will be made by September 1, 1996. Proposals will be evaluated in relation to a number of site selection criteria, which include suitability of site and facilities, strength of local automated reasoning research, costs, and availability of local sponsorship. Further details are available on request from the CADE Inc. Secretary, Neil Murray (nvm@cs.albany.edu).

Nominations for Herbrand Award

Alan Bundy, President, CADE Inc.

The Herbrand Award is given by CADE Inc. to honor an person or a group of people for exceptional contributions to the field of automated deduction. Previous awards have been made at CADE-11 to Larry Wos and at CADE-12 to Woody Bledsoe. Nominations for the award can be made at any time to the CADE Inc. president, Alan Bundy (A.Bundy@ed.ac.uk). Nominations should consist of a letter of up to 2000 words from the principal nominator, describing the nominee's contribution, along with letters of up to 2000 words of endorsement from two other seconders. The winner is selected by the CADE Trustees, the current Programme Committee, and the previous winners.

In order to ensure enough time for selection in time for CADE-13, nominations should reach Bundy by April 30. E-mail nominations are preferred.

Call for Papers

FMCAD '96

The International Conference on Formal Methods in Computer-Aided Design '96 (FMCAD '96) will be held in Palo Alto, California, on November 6–8, 1996. FMCAD '96 is a forum for presenting state-of-the-art tools and techniques based on formal methods for computer-aided design of hardware. A special focus of this conference will be on the integration of complementary techniques and tools. Specific areas of interest to AAR members include the following.

- New hardware verification techniques based on theorem proving, state exploration, model-checking, and BDDs
- Hybrid approaches that integrate synthesis and verification or different verification techniques
- Formal verification techniques for hardware description languages, such as VHDL, Verilog
- Case studies and application of formal methods in industry

This conference is a sequel in a series of IFIP WG 10.2/10.5 sponsored conferences with similar themes that have been held most recently in 1992 and 1994 under the banner "Theorem Provers in Circuit Design." The intended audience includes workers in the field of hardware verification and synthesis as well as practicing digital designers with an interest in formal methods.

Authors may submit research papers (15 pages) or tutorials (15 pages) in Postscript to fmcad96@csl.sri.com, or may send seven hard-copies to the following (submission deadline is April 15, 1996):

Papers	Tutorials
Mandayam Srivas	Albert Camilleri
Re: FMCAD '96	Re: FMCAD '96
SRI International (EL-262)	Hewlett-Packard Co. M/S 5596
333 Ravenswood Avenue	8000 Foothills Boulevard
Menlo Park, CA 94025	Roseville, CA 95747-5596
E-mail: srivas@csl.sri.com	E-mail: ac@hprpcd.rose.hp.com
Tel: +1 415-859-6136	Tel : +1 916 785 8488
Fax: +1 415-859-2844	Fax : +1 916 785 3096

Theorem Proving in Higher-Order Logics

The 1996 International Conference on Theorem Proving in Higher-Order Logics will be held on August 27–30, 1996, in Turku, Finland. Authors are invited to submit papers on all aspects of theorem proving, particularly those relating to higher-order logics or to proof systems based on secure mechanizations of logic. These include advances in theorem-proving technology, proof automation and decision procedures, applications of mechanized theorem proving, development and extension of higher-order logics, and novel industrial applications of theorem provers.

Submissions are invited in two categories: **A** - full research paper, and **B** - informal progress report. Category A papers are due March 15, 1996; these will be refereed and, if accepted, published in the conference proceedings. Category B papers will be distributed in an informal proceedings at the workshop. All papers are due April 14, 1996.

E-mail submissions to orgcom@abo.fi (in PostScript form) are encouraged. Paper copies may be sent to the Department of Computer Science, Abo Akademi University, Lemminkaisenkatu 14a, FIN-20520 Turku, Finland.

FroCoS'96

The first international workshop on Frontiers of Combining Systems will be held on March 26–29, 1996, in Munich, Germany. In various areas of logic, computation, language processing, and artificial intelligence there is an obvious need for using specialized formalisms and inference mechanisms for special tasks. In order to be usable in practice, these specialized systems must be combined, and they must be integrated into general-purpose systems. The development of general techniques for the combination and integration of special systems has been initiated in many areas, and the workshop Frontiers of Combining Systems intends to offer a common forum for these research activities.

Topics of the workshop are

- combination of logics (e.g., modal logics, logics in AI)
- combination of constraint solving techniques
- integration of equational and other theories into deductive systems
- combination of term rewriting systems

- integration of data structures into CLP formalisms and deduction processes
- hybrid systems in computational linguistics, knowledge representation, natural language semantics, and human computer interaction
- logic modeling of multi-agent systems.

Invited speakers include A. Colmerauer, D. Gabbay, U. Glaesser, and M. Stickel. For further information, see http://www.cis.uni-muenchen.de/hot/frocos96.html or contact K. U. Schulz, CIS, University of Munich, Wagmuellerstr. 23, D-80538 Muenchen, Germany; e-mail: schulz@cis.uni-muenchen.de.

An Automated Natural Deduction Proof of the Formalization of the Halting Problem

Li Dafa
Dept. of Applied Mathematics
Tsinghua University
Beijing 100084
CHINA(PRC)
e-mail: ldf@s1000e.dcs.tsinghua.edu.cn

In [5] we suggested using the following three formulas as premises of the halting problem, the first two of which are from [1,2,3]. Let P_i stand for the ith premise, $i = 1, 2, 3$. Then the formalization of the halting problem is as follows.

The English statement for the halting problem is given in [1,2,3]. The notation is as follows:

Ax : x is an algorithm
Cx : x is a computer program in some programming language
Dxyz : x is able to decide whether y halts given input z
H_2xy : x halts on given input y
H_3xyz : x halts on given input the pair $< y, z >$
Oxy :x outputs y

P_1 (for premise 1):

$$\exists x[Ax \wedge \forall y[Cy \rightarrow \forall z Dxyz]] \rightarrow \exists w[Cw \wedge \forall y[Cy \rightarrow \forall z Dwyz]]$$

P_2 (for premise 2):

$$\forall w[[Cw \wedge \forall u[Cu \rightarrow \forall v Dwuv]] \rightarrow$$
$$\forall y \forall z[[[Cy \wedge H_2yz] \rightarrow [H_3wyz \wedge Owg]] \wedge [[Cy \wedge \sim H_2yz] \rightarrow [H_3wyz \wedge Owb]]]]$$

P_3 (for premise 3):

$$\forall w[Cw \& \forall y \forall z[[Cy \& H_2yz \rightarrow H_3wyz \& Owg] \& [Cy \& \sim H_2yz \rightarrow H_3wyz \& Owb]]$$
$$\rightarrow \exists v[Cv \& \forall y[[Cy \& H_3wyy \& Owg \rightarrow \sim H_2vy] \& [Cy \& H_3wyy \& Owb \rightarrow H_2vy \& Ovb]]]]$$

The conclusion is that an algorithm to solve the halting problem does not exist.

That is, $\sim \exists x[Ax \wedge \forall y[Cy \rightarrow \forall z Dxyz]]$.

The problem is to prove that $P_1 \wedge P_2 \wedge P_3 \rightarrow \sim \exists x[Ax \wedge \forall y[Cy \rightarrow \forall z Dxyz]]$ is valid.

We report here a mechanical proof, in natural deduction (ND) style, of the new formalization above. The proof was found automatically by our ANDP system. It is a direct proof and consists of 74 natural deduction steps. Clearly the ND proof is readable.

In [7] Uwe Egly and Thomas Rath reported the first mechanical resolution proof of the new formalization of the halting problem. In [4] we presented a mechanical proof in natural deduction style of Burkholder's formalization of the halting problem. However, our ANDP failed to find a mechanical proof of the new formalization; we were able to give only a hand-crafted ND proof of the formalization [5]. Why did ANDP fail to prove it? After many experiments, we found that one of the reasons was that the rule CASES was applied limitlessly. If the rule CASES is applied to a disjunction, two disjuncts of it will be used as new hypotheses. We conjectured that it might produce new constants from the new hypotheses, hence many new Herbrand terms and irrelevant and redundant formulas. To address that problem, we developed the following strategies:

1. The rule CASES is first applied to premises.
2. The rule CASES is then applied to the disjunctions from which it will not produce new constants.
3. The rule CASES is then applied to short formulas.

The strategies will not affect the completeness. Numerous experiments proved that the strategies were general. Using the strategies, ANDP not only found a mechanical proof in natural deduction style of the new formalization of the halting problem but also produced the small search spaces for Burkholder's original formalization of the halting problem in [1, 2, 3] and Pelletier's 75-problems [6].

Acknowledgment

The project was supported by NSFC.

Appendix: The Mechanical Proof of the Formulation of the Halting Problem

We use (Ex), (Ax) to stand for existential quantifier and universal quantifier respectively.

```
1.  P1 & P2 & P3                                       ASSUMED-PREMISE
2.  (Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]
 -> (Ew)[Cw & (Ay)[Cy -> (Az)Dwyz]]                              SIMP 1
3.  (Aw)[Cw & (Au)[Cu -> (Av)Dwuv]
 -> (Ay)(Az)[[Cy & H2yz -> H3wyz & Owg] & [Cy & ~H2yz
 -> H3wyz & Owb]]]                                               SIMP 1
```

```
4.  (Aw)[Cw & (Ay)(Az)[[Cy & H2yz -> H3wyz & Owg] &
 [Cy & ~H2yz -> H3wyz & Owb]]
 -> (Ev)[Cv & (Ay)[[[Cy & H3wyy] & Owg -> ~H2vy] &
 [[Cy & H3wyy] & Owb -> H2vy & Ovb]]]]                       SIMP 1
5.  ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]                         CASE  2
6.  (Ew)[Cw & (Ay)[Cy -> (Az)Dwyz]]                          CASE  2
7.  Ca1 & (Ay)[Cy -> (Az)Da1yz]                               HYPO 6
8.  Ca1                                                       SIMP 7
9.  (Ay)[Cy -> (Az)Da1yz]                                     SIMP 7
10. Ca1 & (Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
 [Cy & ~H2yz -> H3a1yz & Oa1b]]
 -> (Ev)[Cv & (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2vy] &
 [[Cy & H3a1yy] & Oa1b -> H2vy & Ovb]]]                  US (a1 w) 4
11. ~Ca1 v [~(Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
 [Cy & ~H2yz -> H3a1yz & Oa1b]] v
 (Ev)[Cv & (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2vy] &
 [[Cy & H3a1yy] & Oa1b -> H2vy & Ovb]]]]              IMPLICATION 10
12. ~(Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
 [Cy & ~H2yz -> H3a1yz & Oa1b]] v
 (Ev)[Cv & (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2vy] &
 [[Cy & H3a1yy] & Oa1b -> H2vy & Ovb]]]                      LDS 11 8
13. Ca1 & (Au)[Cu -> (Av)Da1uv]
 -> (Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
             [Cy & ~H2yz -> H3a1yz & Oa1b]]              US (a1 w) 3
14. ~Ca1 v [~(Au)[Cu -> (Av)Da1uv] v
 (Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
          [Cy & ~H2yz -> H3a1yz & Oa1b]]]             IMPLICATION 13
15. ~(Au)[Cu -> (Av)Da1uv] v
  (Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
           [Cy & ~H2yz -> H3a1yz & Oa1b]]                    LDS 14 8
16.  (Ay)(Az)[[Cy & H2yz -> H3a1yz & Oa1g] &
             [Cy & ~H2yz -> H3a1yz & Oa1b]]                  LDS 15 9
17.  (Ev)[Cv & (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2vy] &
            [[Cy & H3a1yy] & Oa1b -> H2vy & Ovb]]]          LDS 12 16
18.   Ca2 & (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2a2y] &
      [[Cy & H3a1yy] & Oa1b -> H2a2y & Oa2b]]                HYPO 17
19.  Ca2                                                     SIMP 18
20.  (Ay)[[[Cy & H3a1yy] & Oa1g -> ~H2a2y] &
          [[Cy & H3a1yy] & Oa1b -> H2a2y & Oa2b]]            SIMP 18
21.  [[Ca1 & H3a1a1a1] & Oa1g -> ~H2a2a1] &
     [[Ca1 & H3a1a1a1] & Oa1b -> H2a2a1 & Oa2b]        US (a1 y) 20
22.  [Ca1 & H3a1a1a1] & Oa1g -> ~H2a2a1                      SIMP 21
23.  [Ca1 & H3a1a1a1] & Oa1b -> H2a2a1 & Oa2b                SIMP 21
```

```
24.  ~Ca1 v [ [~H3a1a1a1 v ~Oa1b] v H2a2a1 & Oa2b]
                                                    IMPLICATION 23
25.  [~H3a1a1a1 v ~Oa1b] v H2a2a1 & Oa2b                  LDS 24 8
26.  ~Ca1 v [[~H3a1a1a1 v ~Oa1g] v ~H2a2a1]         IMPLICATION 22
27.  [~H3a1a1a1 v ~Oa1g] v ~H2a2a1                        LDS 26 8
28.  [[~H3a1a1a1 v ~Oa1b] v H2a2a1] &
  [[~H3a1a1a1 v ~Oa1b] v Oa2b]                DISTRIBUTIVE-LAW 25
29.  [~H3a1a1a1 v ~Oa1b] v H2a2a1                          SIMP 28
30.  ~H3a1a1a1 v ~Oa1g                                     CASE 27
31.  ~H2a2a1                                               CASE 27
32.  ~H3a1a1a1 v ~Oa1b                                   RDS 29 31
33.  (Az)[[Ca2 & H2a2z -> H3a1a2z & Oa1g] &
          [Ca2 & ~H2a2z -> H3a1a2z & Oa1b]]          US (a2 y) 16
34.  [Ca2 & H2a2a1 -> H3a1a2a1 & Oa1g] &
     [Ca2 & ~H2a2a1 -> H3a1a2a1 & Oa1b]              US (a1 z) 33
35.  Ca2 & H2a2a1 -> H3a1a2a1 & Oa1g                       SIMP 34
36.  Ca2 & ~H2a2a1 -> H3a1a2a1 & Oa1b                      SIMP 34
37.  ~Ca2 v [H2a2a1 v H3a1a2a1 & Oa1b]              IMPLICATION 36
38.  H2a2a1 v H3a1a2a1 & Oa1b                            LDS 37 19
39.  H3a1a2a1 & Oa1b                                     LDS 38 31
40.  Oa1b                                                  SIMP 39
41.  ~H3a1a1a1                                           RDS 32 40
42. ~Ca2 v [~H2a2a1 v H3a1a2a1 & Oa1g]              IMPLICATION 35
43.  ~H2a2a1 v H3a1a2a1 & Oa1g                           LDS 42 19
44. [~H2a2a1 v H3a1a2a1] & [~H2a2a1 v Oa1g]     DISTRIBUTIVE-LAW 43
45. ~H2a2a1 v Oa1g                                         SIMP 44
46. (Az)[[Ca1 & H2a1z -> H3a1a1z & Oa1g] &
         [Ca1 & ~H2a1z -> H3a1a1z & Oa1b]]           US (a1 y) 16
47. [Ca1 & H2a1a1 -> H3a1a1a1 & Oa1g] &
    [Ca1 & ~H2a1a1 -> H3a1a1a1 & Oa1b]               US (a1 z) 46
48. Ca1 & H2a1a1 -> H3a1a1a1 & Oa1g                        SIMP 47
49. Ca1 & ~H2a1a1 -> H3a1a1a1 & Oa1b                       SIMP 47
50. ~Ca1 v [H2a1a1 v H3a1a1a1 & Oa1b]               IMPLICATION 49
51. H2a1a1 v H3a1a1a1 & Oa1b                              LDS 50 8
52. ~Ca1 v [~H2a1a1 v H3a1a1a1 & Oa1g]              IMPLICATION 48
53. ~H2a1a1 v H3a1a1a1 & Oa1g                             LDS 52 8
54. [H2a1a1 v H3a1a1a1] & [H2a1a1 v Oa1b]       DISTRIBUTIVE-LAW 51
55. H2a1a1 v H3a1a1a1                                      SIMP 54
56. H2a1a1 v Oa1b                                          SIMP 54
57. H2a1a1                                               RDS 55 41
58. H3a1a1a1 & Oa1g                                      LDS 53 57
59. H3a1a1a1                                               SIMP 58
60. ~H2a1a1                                              RDS 30 53
```

```
61. Oa1b                                                 LDS 56 60
62. H3a1a1a1                                             LDS 55 60
63. ~Oa1g                                                LDS 30 62
64. ~H2a2a1                                              RDS 45 63
65. ~H3a1a1a1 v ~Oa1b                                    RDS 29 64
66. ~Oa1b                                                LDS 65 62
67. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]          ~-ELIMINATION 59 41
68. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]          ~-ELIMINATION 61 66
69. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]               CASES 27 68 67
70. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]                     EE 17 69
71. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]                      EE 6 70
72. ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]                       SAME 5
73.  ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]               CASES 2 72 71
74. P1 & P2 & P3  -> ~(Ex)[Ax & (Ay)[Cy -> (Az)Dxyz]]       CP 73
```

References

[1] Bruschi, M., The Halting Problem, *AAR Newsletter* 17, March 1991.

[2] Burkholder, L., The Halting Problem, *SIGACT News* 18, no. 3, Spring 1987.

[3] Burkholder, L., A 76th Automated Theorem Proving Problem, *AAR Newsletter* 8, April 1987.

[4] Li Dafa, A Mechanical Proof of the Halting Problem, *AAR Newsletter* 23, June 1993.

[5] Li Dafa, The Formalization of the Halting Problem Is Not Suitable for Describing the Halting Problem, *AAR Newsletter* 27, October 1994.

[6] Pelletier, F. J., Seventy-Five Problems for Testing Automatic Theorem Provers, *J. Automated Reasoning* 2 (1986) 191–216.

[7] Egly, U., and Rath, T., The Halting Problem: An Automatically Generated Proof, *AAR Newsletter* 30, August 1995.