ASSOCIATION FOR AUTOMATED REASONING NEWSLETTER

No. 34 October 1996

From the AAR President, Larry Wos...

Maiora meliora. This issue is certainly both—bigger and better. Of especial interest are two articles on the use of automated theorem provers (IPR and ILF) and one article on Gödel's algorithm for class formation. This mix of theory, application, and implementation is most pleasing.

I am also delighted to invite AAR members—and anyone else who wishes to use the (perhaps) first-ever completely push-button theorem prover—to check out *Son of Bird Brain* at http://www.mcs.anl.gov/home/mccune/ar/sobb/. It lets the user construct some simple conjectures and run them through Otter (for proofs) and MACE (for models and counterexamples), two of Argonne's automated deduction systems—all within a few seconds. It's designed to hook the tyro—and I think it will!

Results of the CADE-13 ATP System Competition

Geoff Sutcliffe
James Cook University, Australia
geoff@cs.jcu.edu.au

The CADE-13 ATP System Competition was held on August 1, 1996, at CADE-13. The competition had several categories, according to the hardware, problems, and ATP system type.

In the *General Hardware* categories, where all the systems used the same hardware, the following systems were the winners:

- Mixed problems - Monolithic systems: E-SETHEO, entered by Reinhold Letz from TU München
- Mixed problems - Compositional systems: SPASS, entered by Christoph Weidenbach from MPI Saarbrüecken
- Unit equality problems: Otter 3.0.4z, entered by William McCune from Argonne National Laboratory

In the *Special Hardware* categories, no winners were announced because of the large differences between the hardware used.

Details of the results are now available through the competition Web page:

```
http://wwwjessen.informatik.tu-muenchen.de/~suttner/Competition.html
```

In addition, a full archive of the competition is available, including entrants' binaries; the soundness testing problems and output files; the eligible problem files, precisely as used in the competition; and system output files from the competition.

Minutes of the CADE-13 Business Meeting

Wednesday, July 31, 1996, New Brunswick, N.J.

1. Report on CADE-13 (Slaney)

There were 131 research paper submissions (down from 170+) and 19 system descriptions. Of these, 46 and 16 were accepted, respectively. Both types of submission were refereed. Electronic submissions were allowed in hardship cases.

Most referee reports were on time and of high quality. Problematic papers were discussed before the Program Committee (PC) meeting. (A Web site was considered, but the idea was dropped because of security issues.)

The Program Committee met by e-mail. One advantage was that almost all PC members were involved. However, the process took too long—several days. Moreover, not all PC members got to see all relevant discussion.

Referee reports were returned via e-mail. This procedure resulted in several problems, particularly with the software used to automatically process the reports (e.g., multiple reports in one message, uuencoded and gzipped messages).

For all of FLoC, there were 625 registrations (465 individuals). CADE registration was 202. There were 226 registered for LICS, 95 for RTA, and 170 for CAV.

Bundy asked for comments on CADE-13. The program was quite successful. It was mentioned, however, that the meeting rooms were not optimally located.

2. President's Report (Bundy)

The site for CADE-14 will be Australia (chosen over Rome and Utrecht). The workshop Local Arrangements Chair will be J. Sutcliffe; the Program Chair will be W. McCune.

Alan Robinson was chosen to receive the Herbrand Award. The Bledsoe Travel Award has been initiated in memory of Woody Bledsoe; Woody and Alan Robinson donated their Herbrand Award to this. The winners are Fabio Massacci and Mahadevan Subramaniam.

3. Report on CADE-14 (McCune, Sutcliffe)

CADE-14 will take place in one year (the conferences will now be annual). The Call for Papers will be sent out in September. Electronic submission will be accepted. From the floor it was suggested that having 50 Program Committee members, as was done for CADE-13, was excessive. McCune said that he was thinking of more like 35. Someone asked how many papers were submitted by PC members. The discussion then turned to the issue of assuring that PC members not receive preferential treatment for their submitted papers. The use of e-mail for PC meetings and the restriction of the discussion to PC members who are not authors of a paper were considered. These procedures were followed for CADE-13 (for the most part) and were promised for CADE-14. Slaney mentioned that a number of papers authored by PC members were rejected and that, with a small PC, the workload is significant. If the PC is large, then excluding PC members from submissions is impractical. Bundy noted that while it is desirable to have younger PC members involved, a larger committee is then needed because these young researchers do not yet have the contacts to whom a large number of submissions can be farmed out for reviewing. Bundy directed McCune to consider the points.

4. CADE-15 — Germany or Italy (Bundy)

Bidding for the site of CADE-15 has taken place. Materials from each bid were available in Priscilla's office for review by members during the conference. Both the German and the Italian bids were presented briefly by respective representatives. Bundy then mentioned that another FLoC was being considered for 1999.

5. Treasurer's Report (Murray)

CADE funds are in money markets. A brief history of the CADE funds was presented; the process of giving seed money to LAC's was also reviewed. Profits from CADE-9, -10, -11, and -12 were briefly reviewed. The current balance is about $15,000, held in a corporate account.

6. Bylaws (McRobbie)

Bundy handed over to McRobbie the chairmanship of the meeting. McRobbie pointed out that any new bylaws must be consistent with the laws of Illinois. It was asked if anyone has proxy votes. The two members with proxies decided not to use them. It was requested that people speaking about the bylaws be brief.

Plaisted then presented his support of his proposed bylaws. Maximum democracy was emphasized. Plaisted argued that current Program Chairs and the Secretary-Treasurer should not be voting on future Chairs or other issues as regular Trustees.

Bundy then argued in support of the Trustees' proposed revision. It was noted that democracy was most useful when opposing groups competed for limited resources in an environment that all had to share peacefully. The awkwardness of having the current Program Chair(s) and Secretary-Treasurer involved in crucial CADE discussions and yet treating them as less than full Trustees was bought out. Bundy stated that the Trustees were going to remove their power to change bylaws regardless of the outcome of the current revision. Also, he argued for the super-majority requirement for changing the bylaws, and the STV voting procedure.

Rosenthal suggested that constitutional changes always be put to the entire membership, not to the quorum at a business meeting. Loveland eventually convinced the membership that motions passed at the business meeting should be advisory only, with any rejection by the Trustees to be explained in the next issue of the *AAR Newsletter.*

Slaney introduced some simplifications into the language defining terms of service of Trustees, limits, and nominations. These were accepted into both Plaisted's and the Trustees' amendments to the bylaws.

Rosenthal, Walther, and others clarified that bylaws should be changed only by a two-thirds majority vote of the membership and that at least 30% of the membership must vote. Also, elected Trustees must be elected by the entire membership.

An alternative suggestion was that a vote on changing the bylaws be taken and, if passed, that the changes be discussed over the net one by one for the next year or so to determine exactly what they should be. Murray expressed concern that this approach would be too time consuming for him to moderate as Secretary and that two years already have been spent dealing with bylaws revision issues.

A vote was taken on the Slaney/Rosenthal/Walther/et al. proposals, and all passed.

A vote was taken by secret ballot for choosing between Plaisted's proposal and the Trustees' proposal for changing the bylaws (as amended by Slaney, Rosenthal, and Walther). The results were Trustees' version–35, Plaisted's version–20.

The Trustees' proposed revision of the bylaws was passed.

Plaisted then proposed that the Trustees' version be further modified so that unelected Trustees (current Program Chair and Secretary-Treasurer) not be allowed to vote on future Program Chairs but do be allowed to vote on anything else. This proposal was opposed by several members. A vote was taken: 9 voted in favor, but when 10 opposing votes were counted, the counting was discontinued and the motion defeated.

The discussion then turned to the question of whether to change the bylaws at all. Murray briefly argued for no change, pointing out that the current bylaws did provide for democracy, as evidenced by the current voting. It was agreed that secret ballot was no longer necessary.

A vote was taken by a show of hands, and the accepted revised bylaws (Trustees' version) were preferred to the current bylaws.

7. Membership and Officers of Trustees (Bundy)

Bundy went over the terms of the current Trustees and pointed out that two new Trustees would be elected at CADE-14. The outgoing Trustees were thanked and given a round of applause.

A Significant Step Forward

David A. Plaisted

The bylaws revision passed at the recent CADE business meeting is a significant step forward for CADE, because it gives the AAR membership a regular voice in the running of CADE. I thank the trustees and the supporters of my proposal who have helped to make this revision possible. However, it is curious that the attendees at the CADE business meeting voted to give themselves less of a voice in the running of CADE than they would have had under my fully democratic proposal. Also, if the trustees would propose a fully democratic system for CADE, it would probably pass, and I encourage and challenge them to do so. It should not be necessary to argue the benefits of a fully democratic system, but this should be accepted from the start. I have proposed another democratic amendment to the bylaws, and it may be viewed at the Web site http://www.cs.unc.edu/Research/mi/ along with other information about the bylaws issue; AAR members (and others) are encouraged to visit this site.

Challenge Problems in First-Order Theories

Benjamin Shults
The University of Texas at Austin
bshults@math.utexas.edu

1. IPR

IPR is a tableau-base theorem prover that implements rules for handling a knowledge base of axioms, theorems and definitions. It tries to be selective in its choice of theorems to apply and how to apply them. It does this by taking advantage of the representation of the knowledge in the knowledge base and common-sense restrictions to fetching.

When selecting a theorem to apply in a proof, IPR generally follows the following guidelines:

- The theorem should have something to do with the problem at hand.
- The theorem should (generally) not add more things to be proved but preferably finish part of the proof.

IPR handles equality using an incomplete substitution method advocated by Frank Brown [2].

IPR also has a very nice interface. When run interactively, the unused formulas from a branch of the tree are presented in English with the header “Suppose” over the positive formulas and “Show one of the following” over the negative formulas. When a proof is found, the *condense* algorithm is applied [5], and the proof is output in English.

More details about the IPR prover can be found at

`http://www.ma.utexas.edu/users/bshults/IPR/tab-draft.ps0`

2. Examples

Here we briefly describe some examples of theorems proved by the IPR system in the presence of relatively large knowledge bases. In each example, the knowledge base is designed to have the property that for each sequent, there is a sequence of sequents in the knowledge base that form a chain relating some predicate in the sequent to some predicate in the challenge problem. This design ensures that it is *possible* for even the restricted rule used by IPR to apply each theorem in the knowledge base.

In the first example, the proof itself is very short and easy to find. The difficulty of this problem comes from the fact that IPR found this proof in the presence of a knowledge base of over 100 sequents, each taken from earlier sections of Kelley’s text. In the second and following examples, the proof itself is rather complex.

Example 1 The challenge is part of the 101st labeled theorem from John Kelley’s *General Topology* [3]. This is Theorem 19 on page 147.

If a product is locally compact, then each coordinate space is locally compact.

This is formalized as

$$(\forall X)(\forall A)(\text{locally-compact}(\prod_{A}^{\tau} X) \supset (\forall a)\text{locally-compact}(X_a))$$

This is true in the following theory:

$$(\forall X)(\forall A)(\forall a)\text{continuous-from-to}(\pi_a, \textstyle\prod_A^{\tau} X, X_a)$$

$$(\forall X)(\forall A)(\forall a)\text{open-from-to}(\pi_a, \textstyle\prod_A^{\tau} X, X_a)$$

$$\begin{gathered}(\forall f)(\forall A)(\forall B)((\text{open-from-to}(f, A, B) \wedge \text{continuous-from-to}(f, A, B) \wedge \\ \text{locally-compact}(A)) \supset \text{locally-compact}(B))\end{gathered}$$

Here, $\prod_A^{\tau} X$ represents the product topology where X is a bijection from the index set A to a set of topologies. We use $\prod_A^{\tau}$ rather than $\prod_A$ to distinguish the topology from the underlying set.

Because this proof is short, we present here the English proof output by IPR. Bound variables are symbols preceded by the underscore character _. Skolem constants are also preceded by the underscore character and surrounded by parentheses. If f is a function, the application of f to x is denoted $\{f\}(x)$. The three theorems used are labeled in the knowledge base by the strings "a statement on page 147 of Kelley," "Theorem 3.2 in Kelley" and "a statement on page 90 of Kelley." The rest of the theorems in the knowledge base were taken from the earlier parts of the same text relating the predicates involved in the statement of the challenge. The complete input is available from the author.

```
Theorem:
If the product of (_X_) over the index set (_A_) is locally compact
then for every _A  {(_X_)}(_A) is locally compact.

Proof:
Suppose that
 the product of (_X_) over the index set (_A_) is locally compact
and show that for every _A  {(_X_)}(_A) is locally compact.
Replace the first bound variable in the formula:
  for every _A  {(_X_)}(_A) is locally compact with the new term (_A).
Since we know that
 the product of (_X_) over the index set (_A_) is locally compact
 and we are trying to show that {(_X_)}((_A)) is locally compact
we can apply a statement on page 147 of Kelley
 Now we only need to show that
 the (_A)th projection function of (_X_) over (_A_) is an open
 function from the product of (_X_) over the index set (_A_) onto
 {(_X_)}((_A))
 and
  the (_A)th projection function of (_X_) over (_A_) is a continuous
  function from the product of (_X_) over the index set (_A_) to
  {(_X_)}((_A)).

1. Since we are trying to show that
```

```
 the (_A)th projection function of (_X_) over (_A_) is an open
 function from the product of (_X_) over the index set (_A_) onto
 {(_X_)}((_A))
we can apply Theorem 3.2 in Kelley
which finishes that branch of the proof.

2. Since we are trying to show that
 the (_A)th projection function of (_X_) over (_A_) is a continuous
 function from the product of (_X_)
 over the index set (_A_) to {(_X_)}((_A))
we can apply a statement on page 90 of Kelley
  which finishes that branch of the proof.
```

Example 2 We wish to prove

$$(\forall S)(\text{Hausdorff}(S) \supset \text{closed-in}(\text{top-to-class}(\text{the-diagonal-of}(S)), S \times_{\mathcal{T}} S)).$$

The following seven formulas are all that is needed in the proof. They actually contain a bit more information than what is needed.

$$\begin{gathered}(\forall X)(\forall S)(X \in \text{top-to-class}(\text{the-diagonal-of}(S)) \leftrightarrow \\ (\exists A)(A \in \text{top-to-class}(S) \wedge X = \langle A, A \rangle))\end{gathered}$$

$$(\forall A)(\forall B)(\forall C)(\forall D)(\langle A, B \rangle = \langle C, D \rangle \supset (B = D \wedge A = C))$$

$$(\forall A)(\forall B)(\text{disjoint}(A, B) \leftrightarrow \neg(\exists Y)(Y \in A \wedge Y \in B))$$

$$(\forall X)(\forall S)(\forall T)(X \in S \times T \leftrightarrow (\exists A)(\exists B)(A \in S \wedge B \in T \wedge X = \langle A, B \rangle))$$

$$\begin{gathered}(\forall X)(\forall S)(\forall T)(X \in \text{top-to-class}(S \times_{\mathcal{T}} T) \leftrightarrow \\ (\exists A)(\exists B)(A \in \text{top-to-class}(S) \wedge B \in \text{top-to-class}(T) \wedge X = \langle A, B \rangle))\end{gathered}$$

$$\begin{gathered}(\forall X)(\text{Hausdorff}(X) \leftrightarrow (\forall A)(\forall B)((A \in \text{top-to-class}(X) \wedge B \in \text{top-to-class}(X) \wedge \\ A \neq B) \supset (\exists G1)(\exists G2)(\text{open-in}(G1, X) \wedge \text{open-in}(G2, X) \wedge A \in G1 \wedge \\ B \in G2 \wedge \text{disjoint}(G1, G2))))\end{gathered}$$

$$\begin{gathered}(\forall X)(\forall A)(\text{closed-in}(A, X) \leftrightarrow (\forall y)(y \in \text{top-to-class}(X) \wedge y \notin A) \supset \\ (\exists G)(y \in G \wedge \text{open-in}(G, X) \wedge \text{disjoint}(G, A)))\end{gathered}$$

Notice that $\times_{\mathcal{T}}$ is the product topology on the product of topological spaces, whereas $\times$ is simple Cartesian product of sets. Also the-diagonal-of a topological space, S, represents a subspace (rather than a subset) of $S \times_{\mathcal{T}} S$.

The proof is found, pared down to its shortest form, and printed in English by IPR in about 30 seconds.

Example 3 Here is an example from the theory of vector spaces [1].

$$\begin{gathered}(\forall W)(\forall V)((\text{a-vector-subspace}(W, V) \wedge \text{a-vector-space}(V)) \supset \\ (\exists E)(\exists F)(\text{basis-of}(E \cup F, V) \wedge \text{basis-of}(E, W)))\end{gathered}$$

IPR finds the proof using the knowledge base formed from the following five formulas.

$$(\forall b)(\forall V)(\text{basis-of}(b,V) \supset (\text{lin-ind-subset}(b,V) \wedge b \subset \text{vec-to-class}(V)))$$

$$(\forall s)(\forall V)(\forall t)(\text{lin-ind-subset}(s,V) \wedge \text{basis-of}(t,V) \supset \\ (\exists u)(u \subset t \wedge \text{basis-of}(s \cup u, V)))$$

$$(\forall A)(\text{a-vector-space}(A) \supset (\exists b)\text{basis-of}(b,A))$$

$$(\forall A)(\forall B)(\text{a-vector-subspace}(A,B) \supset \text{a-vector-space}(A))$$

$$(\forall W)(\forall V)(\forall e)((\text{a-vector-subspace}(W,V) \wedge e \subset \text{vec-to-space}(W)) \supset \\ (\text{lin-ind-subset}(e,W) \leftrightarrow \text{lin-ind-subset}(e,V)))$$

Example 4 Here is an example from homotopy theory [4].

$$(\forall X)(\forall x_0)(\forall x_1)((\text{path-connected}(X) \wedge x_0 \in X \wedge x_1 \in X) \supset \\ \text{isomorphic-groups}(H_1(X,x_0), H_1(X,x_1)))$$

The knowledge base formed from the following formulas is sufficient, although a bit more than necessary for the proof. The proof was found with these theorems (which are more than needed) and others in the knowledge base by IPR.

$$(\forall A)(\forall B)(\text{isomorphic-groups}(A,B) \leftrightarrow (\exists f)\text{group-isomorphism}(f,A,B))$$

$$(\forall X)(\forall x_0)(\forall x_1)((\text{path-connected}(X) \wedge x_0 \in X \wedge x_1 \in X) \supset \\ (\exists p)\text{path-from-to}(p,x_0,x_1,X))$$

$$(\forall X)((\forall x_0)(\forall x_1)((x_0 \in X \wedge x_1 \in X) \supset (\exists p)\text{path-from-to}(p,x_0,x_1,X)) \supset \\ \text{path-connected}(X))$$

$$(\forall a)(\forall x_0)(\forall x_1)(\forall X)\text{path-from-to}(a,x_0,x_1,X) \supset \\ \text{group-isomorphism}(\widehat{a}, H_1(X,x_0), H_1(X,x_1))$$

The source code for the implementation and the input for these examples are available from the author.

References

[1] Richard L. Bishop and Samuel I. Goldberg. *Tensor Analysis on Manifolds.* Dover, 1980.

[2] Frank M. Brown. Towards the automation of set theory and its logic. *Artificial Intelligence*, 10(3):281–316, 1978.

[3] John Kelley. *General Topology.* The University Series in Higher Mathematics. D. Van Nostrand, 1955.

[4] James R. Munkres. *Topology: A First Course.* Prentice-Hall, 1975.

[5] F. Oppacher and E. Suen. HARP: A tableau-based theorem prover. *J. Automated Reasoning*, 4:69–100, 1988.

Presentation of Otter Proofs

Bernd I. Dahn
Mathematical Institute of the Humboldt University Berlin
dahn@mathematik.hu-berlin.de

Since May 1995 the mail server of the ILF system has received block-structured proofs, model elimination proofs, and native proofs from the provers DISCOUNT [2], KOMET [1], and SETHEO [4]. These proofs are converted into block-structured proofs in a standard format. Then several proof transformation procedures are applied in order to enhance the readability of the proof. Finally, the proof is typeset in LaTeX according to user-defined directives and returned by e-mail. Details of the technology used by the server are explained in [3].

This service is now is available for tests with proofs from the Otter [5] system. As a refutational prover, Otter starts from a set of formulas containing the negation of the goal to be proved. In order to obtain a natural proof presentation, the ILF server has to know which axiom is the negated goal. The user can include this information by adding a comment `NAME: goal` before the relevant axiom in the Otter input file. The other axioms can be named in a similar way. These names will be used in the final proof presentation as references to the axioms.

Hyperresolution performed by Otter in a single step can be hard to understand, especially in equational proofs. Therefore, the ILF server uses Otter's proof object in order to explain these steps. This requires that the flag `build_proof_object` has been set in the Otter input file. However, in the current version 3.0.4 of Otter, this implies some restrictions. For example, it excludes the generation of answer substitutions. Moreover, it is incompatible with the flag `formula_history`. Therefore, there is no reliable way to trace the clauses from a specific non-clausal axiom up to their use in the proof object. Hence, only proofs from clausal theories can be handled by the ILF server.

When the user has the input and output files and optionally a file with typesetting declarations, he can combine them into a mail to

`ilf-serv@mathematik.hu-berlin.de`.

This is automated by a script that can be downloaded.

Depending on the complexity of the proof and the load of the server, the automatic conversion of the proof may take between a few minutes and a few hours. A mail is returned that can be executed to produce a `.tex` file containing the presentation of the proof. The user may edit this file manually for easier reading or, for example, if the text contains very long formulas that cannot be handled satisfactorily by LaTeX.

General information on the ILF server is on

`http://www-irm.mathematik.hu-berlin.de/ilf-serv`.

From this page an Otter-specific version of the manual can be obtained. Individual support can be requested by mailing to ilf-serv-request@mathematik.hu-berlin.de.

In the future, the ILF server may be enhanced by proof transformation procedures that exploit more specific properties of Otter's proofs if requested by the users. A tool for the interactive manipulation of block-structured proofs is in preparation.

References

[1] W. Bibel, S. Brüning, U. Egly, and T. Rath. KoMeT. In A. Bundy, editor, *Proceedings of CADE 12*, vol. 814, pages 783–787, *Lecture Notes in Artificial Intelligence,* Springer, 1994.

[2] J. Denzinger and W. Pitz. Das Discount-System: Benutzerhandbuch. SEKI Working Paper SR-92-16, Univ. Kaiserslautern, 92.

[3] B. I. Dahn and A. Wolf. Natural language presentation and combination of automatically generated proofs. In *Proceedings of the Conference on Frontiers of Combining Systems.* Kluwer, 1996.

[4] R. Letz, J. Schumann, S. Bayerl, and W. Bibel. Setheo: A high-performance theorem prover. *J. Automated Reasoning*, 8:183–212, 1992.

[5] W. McCune. Otter 2.0. In M. E. Stickel, editor, *Proceedings of the 10th CADE*, pages 663–664. Springer, 1990.

On a Modification of Gödel's Algorithm for Class Formation

Johan G. F. Belinfante
belinfan@math.gatech.edu

1. Introduction

The purpose of this brief note is to comment on a modification of Kurt Gödel's algorithm for class formation and to indicate some simple applications of this algorithm to automated theorem proving in first-order set theory.

The fastest and most convenient theorem provers are the simplest ones, which make use of only first-order logic. Robert Boyer, Ewing Lusk, William McCune, Ross Overbeek, Mark Stickel, and Larry Wos [1] showed how Kurt Gödel's finite axiomatization for set theory can be employed to enable Otter to prove theorems in set theory within first order logic. According to Wos [2], the basic idea of using Gödel's axioms was due to Robert Boyer. In his remarkable thesis, Art Quaife [3] has greatly simplified this formalism.

One of the obvious drawbacks in using Gödel's formalism is that the usual class formation notation $\{x \mid p(x)\}$ is not available; instead, one must reformulate such expressions in terms of Gödel's primitives, some of which, such as *flip*(x) and *rotate*(x) are not even in the average mathematician's vocabulary. On pages 9–11 of Gödel's famous monograph [4] on the Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis, Gödel proves what is called the Class Existence theorem, which roughly says that any normal expression formulated using class formation can also be reformulated in terms of Gödel's primitives.

Gödel's proof of the Class Existence theorem is constructive; he gives what amounts to an explicit algorithm for performing the conversion, together with a proof that the algorithm always terminates. It is not difficult to implement Gödel's algorithm as a computer program, which we have done, using Mathematica$^{\mathrm{TM}}$, taking full advantage of Mathematica's speed and superb pattern matching abilities, including commutative-associative pattern matching capabilities [5]. Our main purpose in creating this program was to use it as a helpful companion to Otter, but the program appears also to be of interest in its own right, independent of this intended application.

Typically, the expressions that are produced by Gödel's algorithm can be exceedingly complicated, primarily because of Gödel's use of Kuratowski's definition of ordered pairs. We have made a small modification of Gödel's algorithm so that Kuratowski's definition is bypassed. We also further process the output so that the *flip* and *rotate* primitives are completely eliminated in favor of more conventional notions, such as composition, the functions $FIRST$ and $SECOND$ (which project out the first and second components of an ordered pair), and the function $SWAP$ (which interchanges the components of an ordered pair). Finally, we have added many simplification rules to produce concise expressions.

It is not the purpose of this short note to spell out the full details of our modification of Gödel's algorithm, but merely to indicate a few simple results that have been obtained with its use that have a direct bearing on some of Quaife's earliest groups of theorems.

2. Quaife's *memb* functor

In Quaife's SS group of theorems that concern the singleton functor, he had introduced the functor $memb(x)$, which picks out the sole member of the class x when the class x is the singleton of a set, and gives back the class x itself when x is not the singleton of a set. The *memb* functor is used only sparingly in Quaife's work, and one comes away with the feeling that this new primitive is not really needed. This is indeed the case; using our modified Gödel algorithm, we readily found that functor *memb* can be eliminated in terms of other primitives:

$$memb(x) = (U(x) \cap image(V, (x \cap image(Id', x)'))) \cup (x \cap image(V, (x \cap image(Id', x)'))').$$

Here V denotes the universal class, Id denotes the identity relation, and primes are used to denote complements. We write $image(z, x)$ for the class $\{v \mid \exists u\, (u \in x \,\&\, \langle u, v\rangle \in z)\}$, and $U(x)$ for the sum class $\{u \mid \exists v\, (u \in v \,\&\, v \in x)\}$.

The quantity $image(V, x)$ is the empty set 0 for $x = 0$, and V otherwise; Quaife made it the subject of his theorem IM10. We have found that this quantity is extremely useful not only for the formulation of conditional definitions, but it also for the simplification of assertions. To simplify an assertion, we first convert the assertion into a class. If p is any assertion, and if w is any variable that does not occur in p, then the class $\{w \mid p\}$ is V if p is true and 0 if p is false. For example, to the assertion $x \in y$ we associate the class $\{w \mid x \in y\}$, which is converted by Gödel's algorithm to $image(V, \{x\} \cap y)$. The simplification rules that we use for classes can therefore also be used to simplify assertions.

Although Gödel's algorithm plus simplification rules certainly does not constitute a general-purpose theorem prover, this combination can occasionally manage to prove assertions by simplifying them to true. When it does not succeed in doing so, for example, when the assertion is not even true, the simplified assertion often suggests an extra hypothesis to add. We have found this technique to be useful in ferreting out inadvertent omissions of hypotheses, as well as for suggesting completely new theorems.

The quantity $image(Id', x)$ in the above formula is equally interesting. This quantity can be used to distinguish singletons from classes with more than one element. It is equal to 0 when $x = 0$, and to V for all nonempty classes except singletons; when x is the singleton of a set, $image(Id', x)$ is x'. Note also that since images preserve unions, and $image(Id, x) = x$, one has the useful formula

$$image(V, x) = x \cup image(Id', x).$$

3. Quaife's *first* and *second* functors

In Quaife's OP group of theorems, the functors $first(x)$ and $second(x)$ are introduced. These two functors serve to pick out the first and second components of an ordered pair of sets, and yield x otherwise. Quaife introduces these functors in theorem OP6, but they are also needed already for the statement of Axiom B-5'b, a slightly awkward state of affairs.

Using the modified Gödel algorithm for class formation, we obtained the following formulas, which show that these functors can be defined in terms of other primitives:

$$\begin{aligned} first(z) &= (z \cap image(V, D(\{z\}))') \cup (image(V, D(\{z\})) \cap U(D(\{z\}))) \\ second(z) &= (z \cap image(V, D(\{z\}))') \cup (image(V, D(\{z\})) \cap U(R(\{z\}))) \end{aligned}$$

Here $D(x)$ and $R(x)$ denote the domain and range of x, respectively. The class $image(V, D(\{z\}))$ which appears in both formulas detects whether z is an ordered pair of sets or not. When $z = \langle x, y\rangle$, where both x and y are sets, we have $D(\{z\}) = \{x\}$ and $R(\{z\}) = \{y\}$. Thus $D(\{z\})$ is not empty, and so $image(V, D(\{z\})) = V$. In this case the above formulas reduce to the statements $first(\langle x, y\rangle) = U(\{x\}) = x$ and $second(\langle x, y\rangle) = U(\{y\}) = y$. On the other hand, when z is not an ordered pair of sets, the quantity $\{z\}$ is not contained in $(V \times V)$, and hence $D(\{z\})$ is 0. In this case, $image(V, D(\{z\})) = 0$ and the above formulas instead reduce to $first(z) = second(z) = z$.

4. Cartesian Products

In practice we have found it useful to add a large number of ad hoc simplification rules to Gödel's basic algorithm in order to produce concise formulas for commonly needed classes. Gödel's proof of termination, of course, does not extend to the simplification algorithm that we use. Several test suites were devised to develop some confidence all the added simplification rules are really correct, and that one does not go around in circles trying to simplify expressions. A few of the rules have been formally verified using Otter, but many are supported merely by hand proofs.

Since it is very easy to overlook special situations that invalidate hand proofs, having also a formal proof produced by Otter often inspires confidence that nothing has been overlooked.

But people can produce faulty input files, and merely using a fine program like Otter does not guarantee that the theorems one thinks one has proved are really correct. Just for fun, all of the assertions made by Quaife in his Appendix 2 were fed into our Gödel program to see if they would simplify to true. A great many of these assertions did simplify to true, and those that did not often suggested interesting new simplification rules to add. But this exercise did reveal that Quaife's theorems CP11 and CP12, his left and right cancellation laws for Cartesian products, are not valid as stated; the case $(0 \times V) = (V \times 0)$ provides a simple counterexample. Substitutes for these theorems were found that the Gödel program did simplify to true:

$$\begin{array}{ll} \text{CP11'} & -((u \times v) \subseteq (w \times x)) \mid (u = 0) \mid (v \subseteq x). \\ \text{CP12'} & -((u \times v) \subseteq (w \times x)) \mid (v = 0) \mid (u \subseteq w). \end{array}$$

Quaife's corollary CP13 concerning Cartesian squares nonetheless remains valid. One can prove this corollary without using CP11 and CP12, or one can wait a bit and obtain CP13 as a corollary of his theorem DO6.

5. Composition and Restriction

The definition of the composite of classes x and y is initially converted by Gödel's algorithm to the expression

$$x \circ y = D(rotate(flip(x \times V)) \cap flip(rotate(y \times V))).$$

Contrary to what one might expect, this formula turns out to be a perfectly feasible starting point for the development of the basic theorems about composites. Despite the apparent complexity of this definition, one can use Otter with all inference rules turned off except for paramodulation to prove enough theorems about composites to get under way. Among the early theorems one can prove in this way are that composition preserves unions, and the following useful replacement for the faulty third and fourth clauses of Quaife's theorem CO8,

$$\begin{aligned} (x \times y) \circ z &= (image(inverse(z), x) \times y) \\ x \circ (y \times z) &= (y \times image(x, z)). \end{aligned}$$

These equations serve as important demodulators. Quaife's theorem CO3, the demodulators for the identity function, are also misprinted by the way, but we already knew that before developing the Gödel program. For example, $Id \circ V = V \circ Id = (V \times V)$ is not equal to V. The following replacement demodulators proved useful.

$$Id \circ Id = Id$$
$$x \circ Id = Id \circ x$$
$$x \cap (V \times V) = Id \circ x$$

The rule $Id \circ Id = Id$ is necessary to prevent the second rule from causing a recursion limit in Mathematica.

Once composition has been introduced, it appears to be useful to eliminate the restrict functor

$$restrict(z, x, y) = z \cap (x \times y),$$

by using a demodulator to convert it to a composite

$$restrict(z, x, y) = id(y) \circ z \circ id(x),$$

where

$$id(x) = Id \cap (x \times x) = restrict(Id, x, x).$$

6. Eliminating *flip* and *rotate*

To eliminate the *flip* and *rotate* functors produced by Gödel's algorithm, it is convenient to introduce the functions $SWAP$, $FIRST$, and $SECOND$. Their conventional definitions

$$\begin{aligned} SWAP &= \{\langle\langle u, v\rangle, \langle x, y\rangle\rangle \mid u = y \;\&\; v = x\} \\ FIRST &= \{\langle\langle u, v\rangle, w\rangle \mid u = w\} \\ SECOND &= \{\langle\langle u, v\rangle, w\rangle \mid v = w\} \end{aligned}$$

are converted by Gödel's algorithm to

$$\begin{aligned} SWAP &= flip(Id) \\ FIRST &= flip(rotate(Id \times V)) \\ SECOND &= rotate(Id \times V). \end{aligned}$$

If one is starting with Gödel's axioms, as in Quaife's development, these definitions, or perhaps the equivalent formulas

$$FIRST = D(rotate(Id)), \quad SECOND = flip(FIRST),$$

can be used to introduce these three functions and to prove their elementary properties. Once this has been done, however, one can turn the tables and use these functions and composites to eliminate the *flip* and *rotate* functors:

$$\begin{aligned} flip(x) &= x \circ SWAP \\ rotate(x) &= SECOND \circ ((inverse(FIRST) \circ SECOND) \cap (inverse(x) \circ FIRST)). \end{aligned}$$

As a matter of fact, Gödel's algorithm never actually requires this full-blown formula for $rotate(x)$. All that one really needs are the following somewhat simpler special facts:

$$\begin{aligned} rotate(0) &= 0 \\ rotate(x \times y) &= x \circ SECOND \circ id(y \times V) \\ rotate(x \cup y) &= rotate(x) \cup rotate(y). \end{aligned}$$

Central to our simplification rules is the functor $fix(x) = D(x \cap Id)$ which is the same as the complement of Quaife's $diag(x)$. Like the domain and range functors D and R, the fix functor preserves unions. But unlike domain and range, the fix functor also preserves complements and intersections, and $fix(inverse(x)) = fix(x)$. One of our central simplification formulas is $D(x \cap y) = fix(inverse(x) \circ y)$. Since Gödel's algorithm first converts the definition

$$x \circ y = \{\langle u, w\rangle \mid \exists v\,(\langle u, v\rangle \in y \;\&\; \langle v, w\rangle \in x)\}$$

into the domain of an intersection, this simplification rule then turns that into the fix of a composite. All that one needs to recover the usual expression $x \circ y$ are some more simplification rules for fix.

7. The Successor Relation $SUCC$

Quaife remarks that the formula he obtained for the successor function $SUCC$ using Gödel's algorithm was too complicated to be useful. Using our enhancements to Gödel's algorithm, we obtain the simpler formula

$$SUCC = E \cap S \cap (E \circ (Id' \cap inverse(E)'))'.$$

In this case, an even simpler formula was obtained earlier by hand,

$$SUCC = ES \cap (PS \circ ES)',$$

where $ES = E \cap S$ and $PS = S \cap Id'$ is the proper subset relation.

8. Concluding Remarks

Automated theorem proving is by no means confined to mathematical theorems, but even a moderate success with computer-assisted proofs of mathematical theorems greatly enhances the credibility of the whole enterprise. It is natural to begin with set theory, because set theory forms the basis for the rest of modern mathematics. At a minimum one would like to be able to prove the theorems that mathematicians generally set forward in Chapter 0 of their texts. It is frustrating that despite many successes, even this modest goal remains fairly challenging. But it is perhaps not the fault of the computer programs, but rather with finding an appropriate formulation of the results to be proved.

References

[1] Robert Boyer, Ewing Lusk, William McCune, Ross Overbeek, Mark Stickel, and Larry Wos, Set theory in first order logic: Clauses for Gödel's axioms, *J. Automated Reasoning,* 2:287–327, 1986.

[2] Larry Wos, *Automated Reasoning: 33 Basic Research Problems,* Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[3] Art Quaife, *Automated Development of Fundamental Mathematical Theories*, Ph.D. thesis, University of California at Berkeley, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1992.

[4] Kurt Gödel, *The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis,* Princeton University Press, Princeton, 1940.

[5] Stephen Wolfram, *Mathematica™, A System for Doing Mathematics by Computer,* 2nd ed. Addison-Wesley, New York, 1991.

Call for Papers

CADE-14

The 14th International Conference on Automated Deduction (CADE-14) will be held on July 13–17, 1997, in Townsville, Australia. CADE is the major forum for presentation of research in all aspects of automated deduction.

Call for CADE-14 Papers: Original research papers and descriptions of working automated deduction systems are solicited for the 14th CADE. Topics of interest include logics, methods (resolution, paramodulation, unification, term rewriting, tableaux, constraints, decision procedures, induction, interactive systems, and frameworks), and applications. Special topics of interest include proof translation, human-computer interfaces, distributed deduction, and search heuristics. Papers on applications of automated deduction are especially encouraged. Research papers can be up to 15 proceedings pages, and system descriptions can be up to 4 pages. The proceedings of CADE-14 will be published by Springer-Verlag in the LNAI series. All submissions must be received by December 4, 1996.

Call for CADE-14 Workshops and Tutorials: Proposals for workshops and tutorials, which are to be held Sunday, July 13, are solicited for CADE-14. Workshops will run the whole day, and tutorials for half a day. Tutorials may be introductory, intermediate, or advanced. Anyone wishing to organize a workshop or tutorial in conjunction with CADE-14 should send (e-mail preferred) a proposal no longer than two pages to the program chair by January 15, 1997.

Program Chair: William McCune, Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL 60439-4844 (Phone: +1 630 252 3065; FAX: +1 630 252 5986; E-mail: cade14-chair@mcs.anl.gov).

Details can be found at the CADE-14 Web site: http://www.cs.jcu.edu.au/~cade-14/

TABLEAUX'97

TABLEAUX'97 will be held May 13–16, 1997, in Pont-à-Mousson (Abbaye des Premontres) near Nancy, France. The conference intends to bring together researchers interested in all the aspects of mechanization of reasoning with tableaux and related methods (e.g., sequent calculi, connection method and model elimination) and working on theoretical foundations of methods, implementation techniques, systems development, and applications. Beside more traditional aspects of tableaux reasoning in various underlying logics as classical logic and nonclassical logics, works dealing with other related approaches to automated reasoning are also solicited.

Submissions are invited in three categories: (1) original research papers (up to 15 pages), (2) original papers about system descriptions (up to 5 pages), and (3) position papers or work in progress, not necessarily original (up to 6 pages). It is intended to publish the conference proceedings (accepted papers of categories 1 and 2) within the LNAI series of Springer.

Authors must submit PostScript contributions by e-mail, preferably in LaTeX llncs style, to the program chair (Didier.Galmiche@loria.fr or tab97@loria.fr) by November 22, 1996. For additional information, see http://www.loria.fr/tab97.