Automated Deduction and its Application to Mathematics (11w2170)

R. Padmanabhan (University of Manitoba), Robert Veroff (University of New Mexico)

June 24–June 26, 2011

1 Overview

The goal of researchers in automated deduction is to develop methods and tools to assist mathematicians, scientists, and engineers with some of the deductive aspects of their work. In contrast to other symbolic computation systems, the core methods in automated deduction focus on searching for proofs and counterexamples. Automated deduction can be applied to wide-ranging problems in nearly any formal language. However, for problems in abstract mathematics and logic, it has been most successful to date when applied to problems stated in first-order and equational logic.

This workshop was the continuation of a series of yearly workshops that began in the summer of 2001 (`http://www.cs.unm.edu/~veroff/ADAM/`). The workshops have provided an opportunity to bring together researchers and students from both the mathematics and automated deduction communities to consider the application of automated deduction to problems in mathematics and logic. The objectives of the workshops include: collaborations on specific math and logic problems; expanding the community of mathematicians interested in applying automated deduction methods to their own research problems; and the continued development of automated deduction tools and strategies.

This year's workshop was dedicated to the memory of Bill McCune, whose untimely passing in May 2011 shocked and saddened his many friends and colleagues. Bill is perhaps best known for his expertise in the design and implementation of automated reasoning programs. His programs, including Otter [8], Prover9 and Mace4 [9], have been used to solve numerous open questions in mathematics and logic and continue to be used by researchers from various disciplines. Indeed, many of the presentations and discussions in this workshop were directly or indirectly influenced by Bill's work.

2 Presentation Highlights

The unifying theme of the workshop was the application of automated deduction methods to research problems in mathematics. One consequence is that the presentations—especially those that focused on the math applications—were disparate in nature. The presentations can roughly be put into two categories, those focusing on automated deduction methods and tools and those focusing on the math applications themselves.

2.1 Automated Deduction Methods and Tools

The TPTP Typed First-order Form with Arithmetic (Sutcliffe). The TPTP World [15] is a well established infrastructure that supports research, development, and deployment of Automated Theorem Proving (ATP)

systems. The TPTP World is based on the Thousands of Problems for Theorem Provers (TPTP) problem library [14], and includes the TPTP language, the SZS ontologies, the Thousands of Solutions from Theorem Provers (TSTP) solution library, various tools associated with the libraries, and the CADE ATP System Competition (CASC). This infrastructure has been central to the progress that has been made in the development of high performance first-order ATP systems—most state of the art systems natively read the TPTP language, many produce proofs or models in the TSTP format, much testing and development is done using the TPTP problem library, and CASC is an annual focal point where developers meet to discuss new ideas and advances in ATP techniques.

Originally the TPTP supported only first-order problems in clause normal form (CNF) [13]. Over the years support for full first-order formulae (FOF) [14] and typed higher-order formulae (THF) [16] has been added. In this talk, we introduce simply typed first-order formulae (TFF) into the TPTP World. TFF in turn is used as the basis for supporting arithmetic. Problems that use these new features have been added to the TPTP problem library. This will provide the impetus for the corresponding development of ATP systems. In particular, the integration of arithmetic capabilities into ATP systems will answer a long-standing demand from ATP users.

The key steps of these developments have been:

- The design of the TPTP TFF language.
- The choice and design of arithmetic features to be written in TFF.
- Collection of problems in TFF, especially problems with arithmetic.
- Building and adapting ATP systems to solve TFF problems.
- Extending the TPTP software infrastructure.

Our presentation described these developments, with the aim of publicizing the developments to working mathematicians, who might then be able to use these new capabilities in their mathematical endeavours.

A Syntactic Approach to Automated Deduction (Ernst). The best strategies for solving open problems in mathematics using first-order theorem-provers rely upon the practitioner having domain specific knowledge of the problem. For example, if it is possible to identify lemmas that are likely to appear in a proof, or if the theory is part of a well-understood hierarchy of related theories, then those facts can be leveraged to guide the proof search. However, it is often the case—especially for difficult open problems—that such information is unknown or unavailable. For this reason, it is necessary to consider search strategies that rely only upon the syntax of the problem representation, because that is the only information that practitioners are guaranteed to have. This talk outlined one approach for using the syntax of the problem to guide the proof search and presented two cases in which significant increases in efficiency were obtained without deploying any domain-specific knowledge of the problem.

Working Our Way Up a Theory Hierarchy (Veroff). We discussed two automated deduction methods for using theory hierarchies to help search for proofs of a theorem t in a target theory T. Using *semantic guidance*, we consider models that falsify t in a simplified theory—that is, with one or more axioms deleted from T. Using the method of *proof sketches* [17], we consider proofs of t in an extended theory—that is, with extra assumptions added to T. In both cases, we use the additional information—models and proofs—to guide the search for a proof of t in the original target theory T.

We also summarized results for a successful application of these methods to a set of problems in loop theory, including the solution to some open questions. Some of the found proofs are several thousand steps long. See [5] for general background on the problem.

2.2 Math Applications

Normal Forms in Graded Lie Algebras (Churchill). The success of Prover9 in establishing particular cases of Jacobson's $x^n = x \Rightarrow$ commutativity theorem in ring theory led us to suspect that such techniques could be applied to normal form problems which can be formulated in a graded Lie algebra context. To communicate these ideas to the other participants we delivered a general background lecture in that area, focusing for simplicity on the case of upper triangular matrices. Following that talk, we had one-on-one discussions with

several participants on more complicated problems which also fit that perspective. See [1, 2, 4] for general background on the problem.

Group Embeddings of Configurations with Prover9 (Ens and Padmanabhan). A configuration is a finite set of elements (called "points" just to have a guiding analogy with the plane geometry) and a finite set of blocks (again, we call them "lines") such that each point is incident with the same number of lines and each line is incident with same number of points. Motivated by the geometric definition of a group law on non-singular cubic curves, we define the concept of group embeddability of (n, k) configurations and classify the set of all (11, 3) configurations that can be embedded into abelian groups in such a way that whenever {P,Q,R} is a line in the configuration then P+Q+R = 0 in the corresponding abelian group. It is precisely in this sense that the set of all inflexion points of a complex cubic turns out to be isomorphic to the abelian group Z[3] $\times$ Z[3]. In this paper we employ Prover9—a first-order theorem prover developed by William McCune—to determine the embeddability of (11,3) configurations. Naturally, there are two kinds of theorems we need to prove: for a given configuration, we have either a concrete group representation or else a proof that no such representation exists. Prover9 is successfully employed to get the proofs of both kinds. Finally, we apply the positive results to obtain a concrete geometric realizability (over a projective plane) of these configurations.

Commutativity Theorems in CL-Semirings (Padmanabhan and Zhang). There are many conditions known which force a ring to be commutative. Such theorems are known as "commutativity theorems". Here we generalize some of the commutativity theorems to cancellative semirings, i.e., semirings in which the addition is cancellative. We use Prover9 to give first-order proofs without actually going through the quotient construction.

Bol-Moufang Groupoids of "Group-like" Type (Phillips). An identity involving one binary operation is of *Bol-Moufang type* if it contains three variables, two which of which occur once, one of which occurs twice, on both sides of the equal sign, and in the same order. These include the well-known Moufang and Bol laws, whence the name. They have been widely investigated. In this talk, we investigate conditions under which Bol-Moufang groupoids axiomatized as algebras of type $< 2, 1, 0 >$ (i.e., with two-sided identity and inverses, in the manner of groups), are, in fact, loops. We also look at "localized" versions of the Moufang laws in groupoids of this type. See [5, 10, 12].

Model Builders, Automated Deduction and Automorphic Loops (Vojtěchovský). A set Q with a binary operation $\cdot$ and an element $1 \in Q$ is a *loop* if $1 \cdot x = x \cdot 1 = x$ for every $x \in Q$, and if for every x, $y \in Q$ there are unique u, $v \in Q$ such that $x \cdot u = y$, $v \cdot x = y$. A loop is *automorphic* if all its inner mappings are automorphisms.

The structural theory of automorphic loops emerged in the last three years, in large part thanks to automated provers and model builders. It presents a sweeping generalization of some classical results of group theory. In this talk we (i) prove the Odd Order Theorem and Lagrange Theorem for commutative automorphic loops, pointing to a crucial lemma obtained with Prover9, and (ii) construct a class of automorphic loops of order p^3 with trivial center, all originating from a single example of order 27 obtained with Mace4.

See [3, 6, 7] for basic information on loops and automorphic loops.

3 Outcome of the Workshop

Our yearly workshops are workshops in the truest sense. Although presentations help establish some context, the most significant value is in the many group and one-on-one discussions that are motivated by the presentations. In this regard, the workshop was very successful. The mathematicians collaborated on their specific research problems; the computer scientists worked with the mathematicians on specific applications; and there was substantial discussion defining and designing new features for automated deduction tools. For one example, there was some discussion about adding support for the inference rule *gL* for cubic curves [11] to Prover9. There was a working prototype for the added functionality shortly after the end of the workshop.

4 List of Partipants

Richard Churchill (City College of New York)
Eric Ens (University of Manitoba)
Zachary Ernst (University of Missouri-Columbia)
R. Padmanabhan (University of Manitoba)
J. D. Phillips (Northern Michigan University)
Geoff Sutcliffe (University of Miami)
Robert Veroff (University of New Mexico)
Petr Vojtěchovský (University of Denver)
Qiduan Yang (University of British Columbia)
Yang Zhang (University of Manitoba)

References

[1] A. Baider, Unique normal forms for vector fields and Hamiltonians, *J. Differential Equations* **78** (1989), 33–52.

[2] M. Bendersky and R.C. Churchill, Normal forms in a cyclically graded Lie algebra, *J. Symbolic Computation* **41**(6) (2006), 633-662.

[3] R. H. Bruck, A survey of binary systems, third printing, corrected, *Ergebnisse der Mathematik und ihrer Grenzgebiete* **20**, Springer-Verlag, Berlin, 1971.

[4] R.C. Churchill and M. Kummer, A unified approach to linear and non-linear normal forms for Hamiltonian systems, *J. Symbolic Computation* **27**(1) (1999), 49–131.

[5] P. Csörgő, Abelian inner mappings and nilpotency class greater than two, *European J. Combin.* **28** (2007), 858–868.

[6] P. Jedlička, M. K. Kinyon and P. Vojtěchovský, Constructions of commutative automorphic loops, *Comm. Algebra* **38**(9) (2010), 3243–3267.

[7] P. Jedlička, M. K. Kinyon and P. Vojtěchovský, The structure of commutative automorphic loops, *Trans. Amer. Math. Soc.* **363** (2011), 365–384.

[8] W. McCune, Otter 3.0 Reference Manual and Guide, Tech. Report ANL-94/6, Argonne National Laboratory, Argonne, IL, 1994.

[9] W. McCune, Prover9, `http://www.cs.unm.edu/~mccune/prover9/`.

[10] G. Nagy and P. Vojtěchovský, Moufang loops with commuting inner mappings, *J. Pure Appl. Algebra* **213**(11) (2009) 2177–2186.

[11] R. Padmanabhan and W. McCune, Automated reasoning about cubic curves, *Computers and Mathematics with Applications* **29**(2) (1995) 17–26.

[12] J.D. Phillips and D. Stanovsky, Bruck loops with Abelian inner mapping groups, *Comm. Alg.*, to appear.

[13] G. Sutcliffe and C.B. Suttner, The TPTP problem library: CNF release v1.2.1, *J. Automated Reasoning* **21**(2) (1998), 177–203.

[14] G. Sutcliffe, The TPTP problem library and associated infrastructure. The FOF and CNF parts, v3.5.0, *J. Automated Reasoning* **43**(4) (2009), 337–362.

[15] G. Sutcliffe, The TPTP world - infrastructure for automated reasoning, In *Proceedings of the 16th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, Lecture Notes in Artificial Intelligence, **6355**, 1–12, Springer-Verlag, 2010.

[16] G. Sutcliffe and C. Benzmüller, Automated reasoning in higher-order logic using the TPTP, THF infrastructure, *"J. Formalized Reasoning* **3**(1) (2010), 1–27.

[17] R. Veroff, Solving open questions and other challenge problems using proof sketches, *J. Automated Reasoning* **27**(2) (2001), 157–174.