Using Automated Theorem Provers in Nonassociative Algebra

J. D. Phillips
Wabash College, Crawfordsville, IN, USA
phillipj@wabash.edu

David Stanovský *
Charles University, Prague, Czech Republic
stanovsk@karlin.mff.cuni.cz

Abstract

We present a case study on how mathematicians use automated theorem provers to solve open problems in (non-associative) algebra.

1 Introduction

In recent years, a growing number of mathematicians have begun to learn about automated reasoning. It has become increasingly useful for their research due to both development of software tools and increasing power of computers. A great deal of attention is paid to formal verification (although mostly by computer scientists, rather than pure mathematicians), but first order automated theorem proving itself has become successful, too. In this paper, we survey some novel results (including solutions to several longstanding open problems) in pure algebra obtained over last decade with the assistance of automated theorem provers, with emphasis on non-associative structures.

We start with a short description of how algebraists actually usually use these tools. Interesting and important problems are almost never stated in a form that can be directly "fed" into a first order theorem prover. So, one particular skill involved is *first order formalization*, and often simplification, of the original goal. This is sometimes straightforward, but some formalizations may require as many as several pages of correctness proof (such as jobs around inner mapping groups and nilpotency, see below). This is related also to the question of which formalization is optimal — a short one or a one with redundant but nontrivial information added? One with less symbols but longer formulas? One with many additinal concepts and compact statements, etc. The answer is ambiguous and a solution very much depends on both experience and the problem at hand.

Now, assume, a formalization is given. Very few interesting problems can be proven directly by any prover in a few minutes, and usually not even in a few days. It is always necessary to try various combinations of *parameters* for proof searches — the most important one indeed is the ordering. Again, there is no general rule, and several possibilities are worth a try. Last but not least, many open problems were solved only by using the *hints strategy*, or sketches [V01], implemented in Prover9 (it would be heplful if other provers had implemented hints, too).

Most problems are not attacked directly. In most cases, the proof of the main result was assisted by theorem provers only partly. Very often a prover handles only several technical steps (which can be still quite difficult). Sometimes, only a particular case of a theorem can be proven automatically, and a general result is sussed out from partial proofs. For several concrete examples, see Section 3 of [PS08].

In general one can recognize the following types of computation that mathematicians perform:

- direct proofs of difficult open problems (very rarely successful);
- proving tedious technical steps;
- quickly checking easy conjectures, particularly those easily formalizable (typically, we find a small counterexample, and thus aren't interested either in the conjecture, nor in the example, anymore);

*This work is a part of the research project MSM 0021620839 financed by MŠMT ČR. The second author was partly supported by the GAČR grant #201/08/P056.

- exhaustive search (typically a lot of almost trivial tasks).

Finally, we wish to stress that mathematicians want to *understand the proof*. Most of the papers (except when an exhaustive search was carried) contain a "human proof". It is usually obtained either by a simple translation of the computer generated proof (which is feasible for little lemmas), or, probably more often, by redoing the proof along the lines suggested by computer. Original computer generated proofs are often significantly shortened using various tricks.

2 Results

In general, one can say that automated theorem proving is particularly useful when one works in a not fully developed environment — e.g., various kinds of weak associativity, such as in loops; or a complicated structure added on top of a classical object, such as lattices with operators in algebraic logic. Sadly, we don't know of any result obtained with ATP that could be called mainstream algebra. This is probably due to the fact that such problems almost always include difficult arithmetics and none of them can be easily formalized.

There are some ATP results about groups and Boolean algebras, though, for instance, various single axiom projects, acheived mostly by the Argonne group and their collaborators in the 1990's, e.g. [MPV03], [MVFHFW02].

Several open problems were solved by using ATP's in the domain of lattices with operators (such as Boolean algebras and their many generalizations), the most prominent one being the Robbins problem [M97]. Recently, many interesting questions that can be approached automatically are coming from algebraic logic, e.g. [VS06], [SV08].

In this paper, we focus on the progress in non-associative algebra. This mostly concerns quasigroups and loops, where automated reasoning has had perhaps its greatest impact over the past decade.

2.1 Quasigroups and loops

It is widely believed that the recent achievements of automated theorem provers have transformed loop theory, both as a collection of deep results, as well as the mode of inquiry itself. Automated reasoning tools are now standard in loop theory.

To highlight the milestones: the first paper assisted by ATP (Otter, at the time), was K. Kunen's 1996 result that Moufang quasigroups are loops. In the early 00's, several other loop theorists started to use Otter and achieved some remarkable results, including one of the greatest open problems in loop theory at the time. After the first author's tutorial at Loops'04 conference, ATP became a standard tool in loops. Up to now, there are 21 papers (and several yet unpublished results) containing results obtained with the assistance of ATP, including several longstanding open problems and significant new results in various projects. All published proofs were obtained with Otter and Prover9, often with non-trivial parameter setting and/or extensive use of the hints strategy.

The progress is summarized in a recent paper [PS08], which contains a commented list of these results. We created a library called QPTP (Quasigroup Problems for Theorem Provers) consisting of a representative selection of 80 problems (68 equational); the problems vary from famous problems open for many years, to little lemmas used in a proof of a much larger result.

We benchmarked selected state-of-the-art provers on QPTP. 71 problems were solved by at least one prover, 38 by all of them. The overall performance of the provers is summarized in Figure 1; for technical details see the original paper.

One can see that Waldmeister outperforms other provers on equational problems by an order of magnitude. This is partly due to how we organized the test (default CASC setting, no advanced strategies).

prover	E 0.999	Prover9 1207	Spass 3.0	Vampire 8.0	Waldmeister 806
proofs in 360s	53	46	31	44	46
proofs in 3600s	59	53	35	57	56
proofs in 86400s	62	61	39	60	59
timeouts	18	19	41	20	9

Figure 1: Benchmarking on QPTP.

On the other hand, focusing solely on one prover in the past was a mistake. Here, we can announce two novel theorems proved by Waldmeister. They fall into the project about how the structure of the inner mapping group determines the structure of the corresponding loop.

1. Bruck loops with abelian inner mapping group are centrally nilpotent of class two.

2. Uniquely 2-divisible loops with abelian inner mapping group of exponent 2 are actually abelian groups.

In particular, the first statement forms a natural complement to a recent result by Nagy and Vojěchovský [NV] claiming the same property when “Bruck” is replaced by “Moufang of odd order”.

The two theorems were obtained by Waldmeister as the result of a large number of computations converging to the current formulations. The final proof of 1. took almost a day of CPU time, resulting in a 2MB output (about 1500 pages), excluding some handwork to prove that what the computer computes is actually equivalent to the English sentence above. This is probably the most complicated proof ever obtained by computer in loop theory.

2.2 Other areas

Few mathematicians work in non-associative algebra outside loop theory, and even fewer use computer assistance. However, we believe that this is a perfect playground for ATP, as the problems approached are often technical and unintuitive. We quickly summarize all five related papers we know about.

In [APS], Prover9 was used to prove some of the partial cases for a general conjecture that a complex condition implies the entropic property.

In [P06], Prover9 helped to sharpen a result of D. A. Bredikhin (1992) by finding a short equational bases for two varieties of groupoids associated with involuted restrictive bisemigroups of binary relations.

ATPs are indeed the perfect tools for supplying direct proofs for results that have been known true, but with a complicated proof possibly involving additional assumptions (such as the axiom of choice). Veroff and McCune [VM] reproved—much more compactly—a result by Kolibiar and Marcisová (1974) on median algebras, certain ternary algebras coming from modular lattices. Another example is a recent paper [S08] providing a direct proof of a decompostion result for selfdistributive groupoids by Ježek and Kepka (1982).

An interesting use of automated reasoning can be found in [DJMKS07]. We investigated equational theories with one binary operation, where each term is equivalent to exactly one linear term. A subgoal (that eventually lead to a solution of the problem) was, to search for theories which have the property for all terms in at most n variables. Such theories are determined by their n-generated free algebras, and those have a known carrier: exactly all linear terms in n variables (the sizes are 1, 4, 21, 184, etc.). What remains is to fill in the multiplication table. The search was carried out independently by a mathematician and by a computer. We wrote a Perl script that was completing the multiplication table and calling Otter to check whether the theory collapses some linear terms. It took about 1 minute to compute 2-generated

free algebras (they appeared earlier in the literature). It took several days by hand and about 2 hours by computer to compute 3-generated free algebras. And using some clever tricks, it wasn't so difficult to find all 4-generated extensions, while the computer search took about two months.

3 Conclusions

We believe this is just beginning of the story. The point we want to make is that, yes, we mathematicians really want to use automated theorem provers. They can help us with some tedious work and, occasionally, even prove difficult theorems. In order to attract even more mathematicians to ATP's we suggest the following:

- Make them as easy to use as major computer algebra systems. (Most of them in current use are not especially user friendly.)
- Care about output; we want to understand the proof!
- Make the provers work efficiently with large libraries of results (and create such libraries).

We believe that automated theorem provers will, sooner or later, become as widespread as computer algebra systems are today (or, perhaps, integrated into them), to assist mathematicians (or at least algebraists) in their work.

References

[APS] K. Adaricheva, A. Pilitowska, D. Stanovský, *On complex algebras of subalgebras*, to appear in Algebra i logika (in Russian).

[DJMKS07] P. Djapić, J. Ježek, P. Marković, R. McKenzie, D. Stanovský, *Star-linear theories of groupoids*, Algebra Universalis 56/3-4 (2007), 357–397.

[M97] W. McCune, *Solution of the Robbins problem*, J. Autom. Reasoning 19, No.3, 263-276 (1997).

[MPV03] W. McCune, R. Padmanabhan, R. Veroff, *Yet another single law for lattices*, Algebra Universalis 50 (2003), no. 2, 165–169.

[MVFHFW02] W. McCune, R. Veroff, B. Fitelson, K. Harris, A. Feist, L. Wos, *Short single axioms for Boolean algebra*, J. Automat. Reason. 29 (2002), no. 1, 1–16.

[NV] G. Nagy, P. Vojtěchovský, *Moufang loops with commuting inner mappings*, submitted to J. London Math. Soc.

[P06] J.D. Phillips, *Short equational bases for two varieties of groupoids associated with involuted restrictive bisemigroups of binary relations*, Semigroup Forum 73, No. 2, 308-312 (2006).

[PS08] J.D. Phillips, D. Stanovský, *Automated theorem proving in loop theory*, proceedings of the ESARM workshop, Birmingham, 2008.

[SV08] M. Spinks, R. Veroff, *Constructive logic with strong negation is a substructural logic*, Studia Logica 88 (2008), no. 3, 325–348.

[S08] D. Stanovský, *Distributive groupoids are symmetric-by-medial: An elementary proof*, to appear in Comment. Math. Univ. Carolinae.

[V01] R. Veroff, *Solving open questions and other challenge problems using proof sketches*, J. Automated Reasoning 27(2) (2001), 157–174.

[VM] R. Veroff, W. McCune, `http://www.cs.unm.edu/~veroff/MEDIAN_ALGEBRA/`

[VS06] R. Veroff, M. Spinks, *Axiomatizing the skew Boolean propositional calculus*, J. Automat. Reason. 37 (2006), no. 1-2, 3–20 (2007).