Automated Theorem Proving in Quasigroup and Loop Theory *

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 survey all results in the area of quasigroup and loop theory, known to have been obtained with the assistance of automated theorem provers. We provide both informal and formal description of selected problems, and compare the performance of selected state-of-the art first order theorem provers on them. Our analysis yields some surprising results, e.g., the theorem prover most often used by loop theorists doesn't necessarily yield the best performance.

1 Introduction

In recent years, a growing number of mathematicians have begun to learn about automated reasoning. It's becoming 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 (first order) automated theorem provers, with emphasis on quasigroups and loops.

Automated reasoning have had great impact on loop theory over the past decade, both in finding proofs and in constructing examples. It is widely believed that these achievements 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.

While [Phi03] is an introduction to automated reasoning for loop theorists, the present paper is intended as its complement: for computer scientists as an introduction to one of the areas in algebra, namely loop theory, in which automated reasoning tools have had perhaps the greatest impact. The paper is self-contained in that we don't assume the reader is familiar with loop theory.

Our goals are twofold. Firstly, we catalogue the quasigroup and loop theory results to date that have been obtained with the assistance of automated theorem provers. Secondly, we lay the groundwork for developing benchmarks for automated theorem provers on genuine research problems from mathematics. Toward that end, we create a library called QPTP (Quasigroup Problems for Theorem Provers) and test the problems on selected automated theorem provers. Note that we don't intend to mirror the TPTP library [SS98] and the CASC competition [SS06]. Rather, we select a representative subset of problems that mathematicians approached by automated reasoning in their research. In fact, QPTP problems were just submitted for the TPTP library.

We now give an outline of the paper.

Section 2 contains a brief introduction to quasigroups and loops, with an emphasis on formal definitions (as opposed to motivation, history, applications, etc.). We think this self-contained introduction is the right approach for our intended audience: computer scientists interested in applications of automated reasoning in mathematics. For a short history, examples and motivation, see e.g. [Pfl00]. For a more comprehensive introduction to the theory of quasigroups and loops, see [Bel67], [Bru71], or [Pfl90]. The

*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.

reader may skip this section and use it for later reference. Most of the notions are used throughout the paper.

In Section 3, we shortly survey techniques algebraists use to support their research by automated theorem provers. This includes formalization of a given conjecture in first order logic, proving the problem and understanding its proof.

Section 4 contains a catalogue of all the theorems from quasigroup and loop theory that we used in our analysis. Taken together, the papers that contain these theorems—and we give full citations for all of them—constitute a complete list of those results on quasigroups and loops that have been achieved to date with the assistance of automated theorem provers.

Section 5 is devoted to technical details of the QPTP database and a benchmark of selected theorem provers on QPTP problems.

In Section 6, we discuss using automated theorem provers in other fields of general algebra. Section 7 contains final thoughts as well as suggested directions for future work.

Additional information on our library, the problem files and the output files may be found on the website

`http://www.karlin.mff.cuni.cz/~stanovsk/qptp`

The present paper is a significant extension of [PS08], presented at the ESARM workshop in Birmingham, in summer 2008.

2 Basic Loop Theory

We call a set with a single binary operation and with a 2-sided identity element 1 a *magma*. There are two natural paths from magmas to groups, as illustrated in Figure 1.

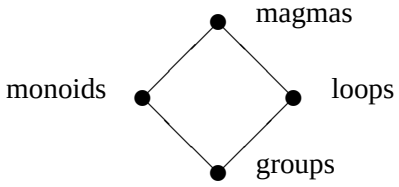

Figure 1: Two paths from magmas to groups.

One path leads through the *monoids*—these are the associative magmas, familiar to every computer scientist. The other path leads through the *loops*—these are magmas in which every equation

$$x \cdot y = z$$

has a unique solution whenever two of the elements x, y, z are specified. Since groups are precisely loops that are also monoids, loops are known colloquially as "nonassociative groups", and via this diagram, they may be thought of as dual to monoids. Many results in loop theory may by regarded as a generalization of results about groups.

As with the class of monoids, the class of loops is too large and general to yield many of its secrets to algebraic inquiry that doesn't focus on narrower subclasses. Here, we simply catalog a few of the most important of these subclasses (the abundant evidence arguing for their importance may be found in many loop theory sources).

First, a comment about notation: we use a multiplication symbol for the binary operation. We usually write xy instead of $x \cdot y$, and reserve $\cdot$ to have lower priority than juxtaposition among factors to be

multiplied, for instance, $y(x \cdot yz)$ stands for $y \cdot (x \cdot (y \cdot z))$. We use binary operations $\backslash, /$ of *left* and *right division* to denote the unique solutions of the equation $x \cdot y = z$, ie., $y = x\backslash z$ and $x = z/y$. Loops can thus be axiomatized by the following six identities:

$$x \cdot 1 = x, \quad 1 \cdot x = x,$$

$$x\backslash(xy) = y, \quad x(x\backslash y) = y, \quad (yx)/x = y, \quad (y/x)x = y.$$

Loops without the unit element 1 are refered to as *quasigroups*; in the finite case, they correspond to Latin squares, via their multiplication table. In the introduction, we emphasise loops over quasigroups, since most automated reasoning results relate to this class, perhaps due to more combinatorial (than algebraic) nature of quasigroup theory.

2.1 Weakening associativity

A *left Bol loop* is a loop satisfying the identity

$$x(y \cdot xz) = (x \cdot yx)z; \tag{lBol}$$

right Bol loops satisfy the mirror identity, namely

$$z(xy \cdot x) = (zx \cdot y)x. \tag{rBol}$$

In the sequel, if we don't specify right or left, and simply write "Bol loop", we mean a left Bol loop.

A left Bol loop that is also a right Bol loop is called *Moufang loop*. Moufang loops are often axiomatized as loops that satisfy any one of the following four equivalent (in loops) identities:

$$x(y \cdot xz) = (xy \cdot x)z, \quad z(x \cdot yx) = (zx \cdot y)x, \quad xy \cdot zx = x(yz \cdot x), \quad xy \cdot zx = (x \cdot yz)x.$$

Generalizing from the features common to both the Bol and the Moufang identities, an identity $\alpha = \beta$ is said to be of *Bol-Moufang type* if: (i) the only operation appearing in $\alpha = \beta$ is multiplication, (ii) the number of distinct variables appearing in α, β is 3, (iii) the number of variables appearing in α, β is 4, (iv) the order in which the variables appear in α coincides with the order in which they appear in β. Such identities can be regarded as "weak associativity". For instance, in addition to the Bol and Moufang identities, examples of identities of Bol-Moufang type include the *extra law*

$$x(y \cdot zx) = (xy \cdot z)x, \tag{extra}$$

and the *C-law*

$$x(y \cdot yz) = (xy \cdot y)z. \tag{C}$$

There are many others, as we shall see. Some varieties of Bol-Moufang type are presented in Figure 2 (for a complete picture, see [PV05a]).

For loops in which each element has a 2-sided inverse, we use x^{-1} to denote this 2-sided inverse of x. In other words,

$$x^{-1}x = xx^{-1} = 1.$$

In Bol loops (hence, also in Moufang loops), all elements have 2-sided inverses. In Moufang loops, inverses are especially well behaved; they satisfy the *anti-automorphic inverse property*

$$(xy)^{-1} = y^{-1}x^{-1}, \tag{AAIP}$$

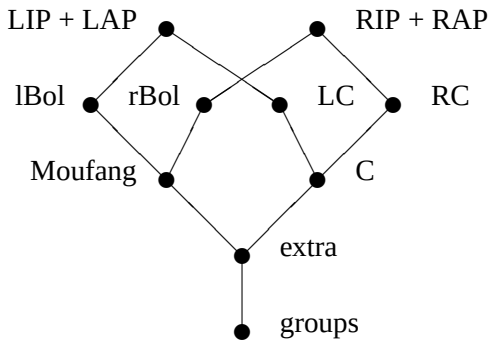

Figure 2: Some varieties of weakly associative loops.

a familiar law from the theory of groups. Bol loops don't necessarily satisfy the AAIP; in fact, the ones that do (left or right), are Moufang. Dual to the AAIP is the *automorphic inverse property*

$$(xy)^{-1} = x^{-1}y^{-1}. \tag{AIP}$$

Not every Bol loop satisfies the AIP, but those that do are called *Bruck loops*. Bruck loops are thus dual to Moufang loops, with respect to these two inverse properties, in the class of Bol loops.

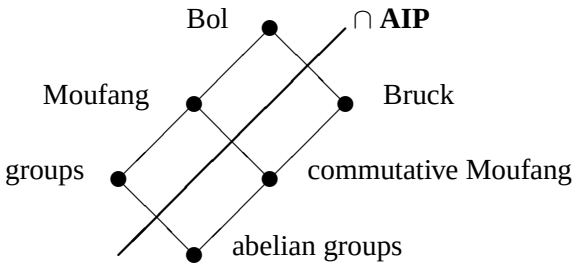

Figure 3: The role of AIP.

A loop is *power associative* if each singleton generates an associative subloop. Bol loops are power associative. A loop is *diassociative* if each pair of elements generates an associative subloop. Moufang loops are diassociative. Thus, Moufang loops satisfy the *flexible* law

$$x \cdot yx = xy \cdot x. \tag{flex}$$

Flexible Bol loops, either left or right, are Moufang. Left Bol loops satisfy both the *left inverse property*

$$x^{-1} \cdot xy = y \tag{LIP}$$

and the *left alternative property*

$$x \cdot xy = xx \cdot y. \tag{LAP}$$

The *right inverse property* (RIP) and the *right alternative property* (RAP) are defined in the obvious ways. The *inverse property* (IP) thus means both the RIP and the LIP, and a loop is called *alternative* if it is both RAP and LAP. Moufang loops and C-loops are alternative and have the inverse property. The *weak inverse property* is given by

$$(yx)\backslash 1 = x\backslash(y\backslash 1). \tag{WIP}$$

2.2 Translations

In a loop Q, the left and right translations by $x \in Q$ are defined by

$$L(x) : y \mapsto xy, \qquad R(y) : x \mapsto xy.$$

The *multiplication group*, $\mathrm{Mlt}(Q)$, of a loop Q is the subgroup of the group of all bijections on Q generated by right and left translations:

$$\mathrm{Mlt}(Q) = \langle R(x), L(x) : x \in Q \rangle.$$

The *inner mapping group* is the subgroup $\mathrm{Mlt}_1(Q)$ fixing the unit element 1. $\mathrm{Mlt}_1(Q)$ is generated by the following three families of mappings, thus rendering the definition equational, and fit for automated theorem provers:

$$T(x) = L(x)^{-1}R(x),$$
$$R(x,y) = R(xy)^{-1}R(y)R(x),$$
$$L(x,y) = L(yx)^{-1}L(y)L(x).$$

If Q is a group, then $\mathrm{Mlt}_1(Q)$ is the group of inner automorphisms of Q. In general, though, $\mathrm{Mlt}_1(Q)$ need not consist of automorphisms. But in those cases in which it does, the loop is called an *A-loop*. Groups and commutative Moufang loops are examples of A-loops.

A subloop invariant to the action of $\mathrm{Mlt}_1(Q)$ (or, equivalently, closed under $T(x)$, $R(x,y)$, $L(x,y)$) is called *normal*. Normal subloops are kernels of homomorphisms, and are thus analogous to normal subgroups in group theory. (In loops, there is no counterpart of the coset definition of a normal subgroup.)

A loop is called *left conjugacy closed* if the conjugate of each left translation by a left translation is again a left translation. This can be expressed equationally as

$$z \cdot yx = ((zy)/z) \cdot zx. \tag{LCC}$$

The definition of *right conjugacy closed* is now obvious, and is given equationally as

$$xy \cdot z = xz \cdot (z\backslash(yz)). \tag{RCC}$$

A *conjugacy closed loop* (CC-loop) is a loop that is both LCC and RCC.

We end this section by defining two classes of loops that are closely related to both Moufang loops and A-loops. *RIF loops* are inverse property loops that satisfy

$$xy \cdot (z \cdot xy) = (x \cdot yz)x \cdot y. \tag{RIF}$$

ARIF loops are flexible loops that satisfy

$$zx \cdot (yx \cdot y) = z(xy \cdot x) \cdot y. \tag{ARIF}$$

2.3 Important subsets and subloops

The *commutant*, $C(Q)$, of a loop Q is the set of those elements which commute with each element in the loop. That is,

$$C(Q) = \{c : \forall x \in Q, cx = xc\}.$$

The commutant of a loop need not be a subloop. Even in those cases when the commutant is a subloop (for instance, in Moufang loops), it need not be normal (of course, the commutant in a group is normal, and in group theory it is called the center, as we shall see).

The *left nucleus* of a loop Q is the subloop given by

$$N_\lambda(Q) = \{a : a \cdot xy = ax \cdot y, \forall x,y \in Q\}.$$

The middle nucleus and the right nucleus, $N_\mu(Q)$ and $N_\rho(Q)$ respectively, are defined analogously; both are subloops. The *nucleus*, then, is the subloop given by

$$N(Q) = N_\lambda(Q) \cap N_\mu(Q) \cap N_\rho(Q).$$

The *center* is the normal subloop given by

$$Z(Q) = N(Q) \cap C(Q),$$

thus coinciding with the language from groups. $C(Q)$ need not have any relationship with $N(Q)$; that is, $C(Q) \cap N(Q) = Z(Q)$ can be trivial. The situation in Bol loops is strikingly different. In a (left) Bol loop Q, $N_\lambda(Q) = N_\mu(Q)$, and this subloop need not have any relationship with $N_\rho(Q)$, i.e., the intersection can be trivial. Thus, in a Moufang loop, all nuclei coincide, and $N(Q)$ is a normal subloop. Moreover, if Q is Bruck, then $N_\lambda(Q) \leq C(Q)$.

A loop Q is called *centrally nilpotent* of class n, if it possesses a sequence of subloops $\{1\} = Q_0 \leq Q_1 \leq \ldots \leq Q_n = Q$ such that the successive quotients are central, in the sense that $Q_{i+1}/Q_i \leq Z(Q/Q_i)$.

The *commutator*, $[x,y]$ of x and y, in a loop Q is given by

$$xy = yx \cdot [x,y].$$

The *associator*, $[x,y,z]$ of x, y, and z, is given by

$$xy \cdot z = (x \cdot yz) \cdot [x,y,z].$$

The point is that the lack of associativity in loops provides a structural richness, part of which can be captured equationally, thus rendering loops excellent algebraic objects to investigate with automated theorem provers.

2.4 Quasigroups

Quasigroups are loops without an identity element. Thus, quasigroups are to loops as semigroups are to monoids. Formally, a *quasigroup* is a set with a single binary operation such that $x \cdot y = z$ has a unique solution whenever two of the elements x,y,z are specified. Or in the language of universal algebra, the variety of quasigroups is axiomatized by the following four equations:

$$x\backslash(xy) = y, \quad x(x\backslash y) = y, \quad (yx)/x = y, \quad (y/x)x = y.$$

The lack of an identity element makes the theory much more subtle than is the theory of loops. Automated theorem provers have been used successfully in the theory of quasigroups, but to a lesser extent than they have in loop theory.

The algebraic part of the theory usually investigates particular subvarieties of quasigroups. A very important subclass, for many reasons, is that of *medial* quasigroups, defined by the identity

$$xy \cdot uv = xu \cdot yv,$$

and its many generalizations. A quasigroup is said to be *trimedial* if each subquasigroup generated by three (or fewer) elements is medial. Here, for example, are several consquences of the medial law in three variables: (1) *left semimediality*: $xx \cdot yz = xy \cdot xz$; (2) *right semimediality*: $zy \cdot xx = zx \cdot yx$; (3) *left*

F-law: $x \cdot yz = xy \cdot (x\backslash x)z$; (4) *right F-law*: $zy \cdot x = z(x/x) \cdot yx$. Quasigroups satisfying both F-laws are called shortly *F-quasigroups*. Another interesting identities are *left* and *right distributivity*:

$$x \cdot yz = xy \cdot xz \quad \text{and} \quad zy \cdot x = zx \cdot yx,$$

providing an equational description for the fact that all left and right translations are automorphisms.

A triple of bijections (f,g,h) from a quasigroup Q_1 to a quasigroup Q_2 is called an *isotopism* if

$$f(x) \cdot g(y) = h(x \cdot y)$$

for every x,y in Q_1. Combinatorially, isotopism describes shuffling the rows, the columns and renaming the entries of the multiplication table. Indeed, f is an *isomorphism* iff (f,f,f) is an isotopism. We can thus talk about isomorphic, or isotopic quasigroups. In group theory, isotopic groups are isomorphic. But in the theory of quasigroups, isotopism is more general than isomorphism, that is isomorphic quasigroups are isotopic, but isotopic quasigroups need not be isomorphic.

Isotopy is a particularly interesting concept when Q_2 is a loop. It is easy to see that every quasigroup is isotopic to a loop, take $x \circ y = x/a \cdot b\backslash y$ for the loop operation (then, ba is the unit). The classic Toyoda-Bruck theorem (1941), for instance, asserts that every medial quasigroup is isotopic to an abelian group. Kepka (1979) proved that every trimedial quasigroup is isotopic to a commutative Moufang loops. And one of the papers in Section 4 asserts that F-quasigroups are isotopic to (general) Moufang loops.

Quasigroups can also be viewed as combinatorial objects: the multiplication table of a finite quasigroup is a Latin square; the converse is true as well (even in the infinite case). This perspective proves useful in many settings, for instance, Steiner triple systems (classic combinatorial objects) can be viewed as quasigroups satisfying the *symmetric* laws $x \cdot xy = y = yx \cdot x$.

3 Techniques

We shortly describe how algebraists usually use automated reasoning tools. In general, one can recognize, for example, the following types of computation:

- direct proofs of difficult open problems (very rarely successful),
- proving tedious technical steps,
- quick experimentation, checking out (often false) conjectures,
- exhaustive search.

Most hard problems are not attacked directly. In most cases, the proof of the main result is assisted by theorem provers only partly. Very often a prover handles only several technical steps (which can be still quite difficult) in a long classical proof. Sometimes, only a particular case of a theorem can be proven automatically, and a general result is sussed out from partial proofs. For concrete examples, see Section 4, e.g. the description of [AKP06].

Interesting and important problems are almost never stated in a form that can be directly "fed" into a first order theorem prover. One can, in general, recognize the following phases:

1. formalization in first order logic,
2. finding a proof,
3. reading and understanding the proof.

3.1 Formalization

So, one particular skill involved is *first order formalization*, and often simplification, of the original goal. Most problems, indeed, cannot be formalized in a way accessible to a theorem prover. Sometimes, the formalization is straightforward, but in some cases a good formalization may require as many as several pages of correctness proof. For instance, statements about inner mapping groups can be formalized using the description of their generators (not an entirely easy result). Statements about central nilpotence can be formalized using the associator-commutator calculus (instead of central series). Let's take an example from [PSxx]. The neat English statement

Bruck loops with abelian inner mapping group are centrally nilpotent of class 2.

can be formalized in TPTP language in the following way (see the file `PSxx_2`):

```
cnf(sos,axiom,mult(unit,A) = A).
cnf(sos,axiom,mult(A,unit) = A).
cnf(sos,axiom,mult(A,i(A)) = unit).
cnf(sos,axiom,mult(i(A),A) = unit).
cnf(sos,axiom,i(mult(A,B)) = mult(i(A),i(B))).
cnf(sos,axiom,mult(i(A),mult(A,B)) = B).
cnf(sos,axiom,rd(mult(A,B),B) = A).
cnf(sos,axiom,mult(rd(A,B),B) = A).
cnf(sos,axiom,mult(mult(A,mult(B,A)),C) = mult(A,mult(B,mult(A,C)))).
cnf(sos,axiom,mult(mult(A,B),C) = mult(mult(A,mult(B,C)),asoc(A,B,C))).
cnf(sos,axiom,op_l(A,B,C) = mult(i(mult(C,B)),mult(C,mult(B,A)))).
cnf(sos,axiom,op_r(A,B,C) = rd(mult(mult(A,B),C),mult(B,C))).
cnf(sos,axiom,op_t(A,B) = mult(i(B),mult(A,B))).
cnf(sos,axiom,op_r(op_r(A,B,C),D,E) = op_r(op_r(A,D,E),B,C)).
cnf(sos,axiom,op_l(op_r(A,B,C),D,E) = op_r(op_l(A,D,E),B,C)).
cnf(sos,axiom,op_l(op_l(A,B,C),D,E) = op_l(op_l(A,D,E),B,C)).
cnf(sos,axiom,op_t(op_r(A,B,C),D) = op_r(op_t(A,D),B,C)).
cnf(sos,axiom,op_t(op_l(A,B,C),D) = op_l(op_t(A,D),B,C)).
cnf(sos,axiom,op_t(op_t(A,B),C) = op_t(op_t(A,C),B)).
cnf(goals,negated_conjecture,asoc(asoc(a,b,c),d,e) != unit).
```

The first 9 lines define a Bruck loop, the next 4 lines define the associator and the generators of the inner mapping group, the remaining 6 lines say that the generating mappings commute. The goal is one of the six identities describing nilpotence of class 2; in this particular example, the other five identities follow quite easily from the present one.

Another interesting example of first order formalization can be found in [Sta08], see Section 6.

There is a related question: which formalization is optimal — a short one or a one with redundant but nontrivial information added? One with less symbols but longer formulas? Or one with many additional concepts and compact statements, etc. The answer is ambiguous and a particular solution very much depends on both experience and the problem at hand, and indeed the prover in use.

Perhaps the easiest to understand example is proving the existence of a unit element. The naive description is

$$\exists z\, \forall x\, (x \cdot z = x \,\&\, z \cdot x = x).$$

However, in quasigroups, this formula is equivalent to each of the following two pairs of identities:

$$x \cdot (y/y) = x \,\&\, (y/y) \cdot x = x,$$
$$x \cdot (y\backslash y) = x \,\&\, (y\backslash y) \cdot x = x.$$

The equations indeed advice the prover what the z shall be. The advice may help, but it also may be misleading — see the results in Figure 5, problems `Kun96a_1`, `Kun96a_2` and `Kun96_b3`.

Another example of ambiguity in QPTP is proving that a loop is Moufang. There are four equivalent identities defining the property, one can thus ask to prove either a disjunction of them (standard formalization), or a particularly chosen one (`alt1` to `alt4`) — see the results in Figure 5, problems `CGKxx_4`, `KKP02a_1` and `KKP02b_1`.

3.2 Proving

Now, assume, a formalization is given. Few interesting problems can be proven directly by any prover in a few minutes, and often not even in a few days. It is always worth trying various combinations of *parameters* for proof search (the most important one indeed is the ordering); again, there is no general rule. Many open problems were solved only by using the *hints strategy*, or sketches [Ver01], implemented in Prover9.

To date, all but two computer solutions in loop theory have been obtained by Prover9 [McC05] or its predecessor Otter [McC03]; the sole exception are two theorems from [PSxx], proved by Waldmeister in 2008 (see Section 4). One of the main goals of the present paper is to give guidelines on which prover shall be selected for which problem (see the discussion in Section 5).

For *model finding*, algebraists are using mostly Mace4 [McC05], probably due to that fact that it comes in a package with Prover9. Our informal experiments show that Paradox [CS03], considered the top model finder, doesn't behave significantly better on quasigroup problems. Recently, several important examples of loops were found with assistance of the Loops package for GAP [NV07], including the first example of a non-Moufang, finite simple Bol loop by G. Nagy [Nag08], whose existence was the most prominent open problem in loop theory for many years. Our study focuses on theorem proving, rather than model building, since there is no remarkable open problem solved by a model builder without major human involvement. This is probably due to most interesting problems about finding finite quasigroups either include properties that cannot be easily formalized in first order theory (such as being simple), or are known to have a large lower bound on the number of elements.

3.3 Understanding the proof

We wish to stress that mathematicians (usually) want to *understand the proof*. Almost all the papers in our survey contain a human oriented 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.

There is no universal methodology, how to simplify and understand a computer generated proof. There have been several attempts (such as [Pud07]), but none of them yields satisfactory results for algebraic purposes. The area deserves future research.

4 The Theorems

The present section catalogues all papers in quasigroup and loop theory to date whose results were obtained with the assistance of an automated theorem prover. As noted above, all but one papers were assisted by Prover9 or Otter. Published proofs were always translated to human language and usually simplified (none of the papers presents a raw output from a prover), hence none of the results relies on soundness of Otter/Prover9. As far as we know, no automatically generated proof was found to be incorrect during translation.

To summarize the achievments: the story started in 1996, when Kenneth Kunen used Otter to show that quasigroups satisfying any of the four Moufang identities possesses a unit element. Since then, 28 papers containing results obtained with assistance of ATP appeared, and the number is growing each year. They include solutions to several longstanding open problems and significant new results in various projects in loop theory. Theorem provers often help to lay a groundwork in a particular class of loops, on top of which mathematicians build an elegant theory (such as decomposition theorems etc.).

We list the papers in chronological order. From each paper, we choose up to five theorems for the QPTP library.

[Kun96a]. This is an important paper, because it was the first to use automated theorem provers in loop theory and, in fact, one of the first noneasy results in mathematics obtained by computer. The theorem says that a quasigroup satisfying any one of the four Moufang laws is, in fact, a loop, i.e., has a unit element. We analyze this result for each of the four Moufang identites. Note that the proof for the third and the fourth Moufang identities can by done relatively easily by hand, while the proof for the first and the second one was only discovered by Otter.

[Kun96b]. This is a sequel to the previous paper. The main result is the determination of which of the Bol-Moufang identities, implies, in a quasigroup, the existence of a unit element. We analyze three of these identities.

[Kun98]. If R is an associative commutative ring and Q a loop, one can define a loop ring RQ in a similar way group rings are constructed. The main result of [Kun98] is the following: if R has characteristic $\neq 2$ and RQ is right alternative, then it is also left alternative. First, rings were eliminated from the problem: one can prove that, in characteristic $\neq 2$, (1) RQ is right alternative, iff for all $x,y,z \in Q$, both conditions ($x \cdot yz = xy \cdot z$ or $x \cdot yz = xz \cdot y$) and ($x \cdot yz = xy \cdot z$ or $x \cdot zy = xy \cdot z$) are satisfied in Q; (2) if RQ is right alternative, then it is also left alternative, provided the inverse mapping $i(x) = x\backslash 1$ defined on Q satisfies $i(xy) = i(y)i(x)$ for all $x,y \in Q$. Hence, we are left with two first order conditions in loop theory. The proof was found with help of Otter, and we include the problem in our library as it stands.

[Kun00]. This is a groundwork on conjugacy closed loops. Many useful properties of CC-loops were obtained by Otter and later used to prove structure theorems. We analyze the following two: (1) If Q is conjugacy closed, $a,b \in Q$ and $ab = 1$, then ba is in the nucleus of Q. (2) If Q is conjugacy closed, the commutant of Q is contained in the nucleus.

[KKP02a]. The main result is that diassociative A-loops are Moufang. Diassociativity, in general, is not finitely axiomatizable poperty. However, in A-loops, it is known to be equivalent to the inverse property. Hence, we include the problem stating that IP A-loops are Moufang. This was one of the major longstanding open problems in loop theory, and perhaps the most important automated theorem proving success in loop theory. And it marks the point at which the number of loop theorists using automated theorem provers in their work jumped from one to three.

[KKP02b]. There are many results in this paper proved by automated theorem provers; we include the following four: (1) 2-divisible ARIF loops are Moufang, (2) flexible C-loops are ARIF, (3) Moufang loops are RIF, (4) RIF loops are ARIF.

[KP02]. T. Kepka showed in 1978 that a quasigroup is trimedial if and only if it is left semimedial, right semimedial, and satisfies the identity $(x \cdot xx) \cdot uv = xu \cdot (xx \cdot v)$. The present paper sharpens this result by showing that, in fact, only one of the two semimedial laws is sufficient in the basis (either of them). We analyze this result, here.

[KK04]. This is a groundwork on extra loops. Many useful properties were obtained by Otter, e.g., finite nonassociative extra loops have nontrivial centers. We analyze the following result: in an extra loop, z commutes with $[x,y,t]$ if and only if t commutes with $[x,y,z]$ if and only if $[x,y,z][x,y,t] = [x,y,zt]$.

[KKP04]. There are many results in this paper proved by automated theorem provers. We include the following one: in CC-loops, associators are in the center of the nucleus.

[KP04a]. The main result in this paper is that commutants of Bol loops of odd order are, in fact, subloops. Obviously, this is not a first order statement, however its proof relies on several lemmas proved by a theorem prover. We analyze the following one: if Q is a Bol loop, and if $a, b \in C(Q)$, then so too are a^2, b^{-1} and $a^2 b$.

[KP04b]. Yet another equational basis for trimedial quasigroups is found: $x \cdot yz = (x/x)y \cdot xz$ and $zy \cdot y = zx \cdot y(x\backslash x)$. We include the equivalence of the new basis with the basis from [KP02].

[KP05]. The main purpose of this paper is to give a basis for the variety of rectangular loops which consists of 7 identities, thus improving Krapež's pre-existing basis of 12 axioms [Kra00]. A *rectangular loop* is a direct product of a loop and a rectangular band. A *rectangular band* is a semigroup which is a direct product of a left zero semigroup and right zero semigroup. A *left* (*right*, resp.) *zero semigroup* is a semigroup satisfying $x \cdot y = x$ ($x \cdot y = y$, resp.). We analyze part of this result by showing that the identities

$$x\backslash(xx) = x,\ (xx)/x = x,\ x \cdot (x\backslash y) = x\backslash(xy),\ (x/y) \cdot y = (xy)/y,\ x\backslash(x(y\backslash y)) = ((x/x)y)/y,$$

$$(x\backslash y)\backslash((x\backslash y) \cdot (zu)) = (x\backslash(xz)) \cdot u,\ ((xy) \cdot (z/u))/(z/u) = x \cdot ((yu)/u)$$

imply each of the following identities (in algebras with three binary operations $\cdot, \backslash$, and $/$):

$$(x\backslash y)\backslash((x\backslash y)z) = x\backslash(xz),\ (x/y)\backslash((x/y)z) = x\backslash(xz),\ x(y\backslash(yz)) = xz,\ ((xy)/y)z = xz,$$

$$(x \cdot yz)/(yz) = (xz)/z,\ (x(y\backslash z))/(y\backslash z) = (xz)/z,\ (x(y/z))/(y/z) = (xz)/z.$$

[PV05a]. The main result of this paper is the systematic classification of all varieties of loops axiomatized by a single identity of Bol-Moufang type, achieved to a large extent automatically. We include a typical result: in loops, the following two identities are equivalent (and thus both axiomatize the so-called variety of LC-loops): $x(y \cdot yz) = (x \cdot yy)z$ and $xx \cdot yz = (x \cdot xy)z$.

[PV05b]. The purpose of this paper is to do for quasigroups what [PV05a] did for loops: i.e., to offer a systematic classification of all varieties of quasigroups axiomatized by a single identity of Bol-Moufang type. The results were achieved automatically (and due to much larger number of cases, the proofs were presented as raw Otter outputs). We include a typical result: in quasigroups, the identity $x(yy \cdot z) = xy \cdot yz$ implies associativity.

[AKP06]. One of the main results in this paper is that in a Bruck loop, elements of order a power of two commute with elements of odd order. Obviously, automated theorem provers can't prove this result directly, as it is a result about infinitely many positive integers. On the other hand, one may use automated theorem provers to generate proofs about *specific* integers, and then use these proofs to help construct the proof of the general result. The three specific cases we analyze here: in a (left) Bruck loop, elements of order 2^2 commute with elements of oder 3, elements of order 2^2 commute with elements of order 3^2, and elements of order 2^4 commute with elements of order 3^2. The three different cases give rise to clear performance differences between the automated theorem provers, see Figure 5. We note that this property was used in [AKP06] to prove a deep decomposition theorem for Bruck loops. That is, this also was an important success for automated theorem provers in loop theory.

[KK06]. There are many results in this paper proved by automated theorem provers. We analyze the following results: for each c in a power associative conjugacy closed loop, c^3 is WIP (i.e., $c^3(xc)^{-1} = x^{-1}$ for every x), c^6 is extra (i.e., $c^6(x \cdot yc^6) = (c^6x \cdot y)c^6$ for every x,y) and c^{12} is in the nucleus. (Initially, the last property wasn't obtained directly by Otter. Interestingly, other provers can do it.)

[Phi06]. The main result in this paper is that the variety of power associative, WIP conjugacy closed loops is axiomatized, in loops, by the identities $(xy \cdot x) \cdot xz = x \cdot ((yx \cdot x)z)$ and $zx \cdot (x \cdot yx) = (z(x \cdot xy)) \cdot x$. We analyze this result.

[PV06]. This is a groundwork on C-loops, and, as usually, there are many properties proved by automated theorem provers. We analyze the following two: (1) in C-loops, the nucleus is a normal subloop, and (2) in a commutative C-loop, if a has order 4 and b has order 9, then $a \cdot bx = ab \cdot x$ (this is one of the cases that led to a proof of the decomposition theorem for commutative torsion C-loops).

[KKP07]. The main result of the paper is that every F-quasigroup is isotopic to a Moufang loop. This was a longstanding open problem—it was the first open problem listed in Belousov's 1967 book [Bel67]. We include this result as it stands, although in the original paper, Otter was used to prove only one particular step.

[KPV07]. There are many results in this paper proved by automated theorem provers. We analyze the following one: a C-loop of exponent four with central squares is flexible.

[KPV08]. There are many results in this paper proved by automated theorem provers. We analyze the following one: in a Bol loop, if c is a commutant element, then c^2 is in the left nucleus if and only if c is in the right nucleus.

[PV08]. The purpose of this paper is to find group-like axiomatizations for the varieties of loops of Bol-Moufang type. We include the following typical result: a magma with 2-sided inverses satisfying the C-law is a loop.

[CDKxx]. This is a groundwork on *Buchsteiner loops*, i.e. loops that satisfy the identity $x\backslash(xy \cdot z) = (y \cdot zx)/x$. Buchsteiner loops arise from a study of loops of Bol-Moufang type [DJxx] and are closely related to conjugacy closed loops. Again, some properties in the paper were proved automatically. The result we analyze here is that in Buchsteiner loops, fourth powers are nuclear, i.e., $x^4 \in N(Q)$ for every $x \in Q$.

[CGKxx]. The authors investigate some connections between loops whose loop rings, in characteristic 2, satisfy the Moufang identities and loops whose loop rings, in characteristic 2, satisfy the right Bol identities. Similarly to [Kun98], rings are eliminated from the problem, and an automated theorem prover is used for reasoning about the resulting first order conditions. We analyze the first order translation of the main theorem, and also the following technical lemma: if Q is a right Bol loop with the property that, for all $x,y \in Q$, $xy = yx$ or $x^{-1}(xy) = y$, then Q is Moufang.

[JKVxx]. The structure and properties of commutative A-loops is revealed in the paper. Some technical steps were carried out automatically. For our study, we chose several problems related to the fact that the product of two squares is again a square.

[KKPxx]. The main result in this paper is that in a strongly right alternative ring (with a unit element), the set of invertible elements is a Bol loop under ring multiplication, and the set of quasiregular elements is a Bol loop under "circle" multiplication. A *right alternative ring* is a set R with two binary operations, $+$ and $\cdot$, such that under $+$, R is an abelian group, under $\cdot$, R is a right alternative magma, and such that $\cdot$ distributes over $+$. A right alternative ring is *strongly right alternative* if $\cdot$ is a right Bol loop. The circle operation is given by $x \circ y = x + y + xy$ and an element is called *quasiregular* if it has a two-sided inverse under $\circ$. We analyze the following technical result: if a is invertible (it has a 2-sided inverse under $\cdot$), then $R(a^{-1}) = R(a)^{-1}$ and $L(a)^{-1} = R(a)L(a^{-1})R(a^{-1})$.

[KVxx]. There are many results in this paper proved by automated theorem provers. We analyze the following one: in a commutative RIF loop, all squares are Moufang elements and all cubes are C-elements. An element a is a *Moufang element* if for all x and y, $a(xy \cdot a) = ax \cdot ya$. And it is a *C-element* if for all x and y, $x(a \cdot ay) = (xa \cdot a)y$.

[PSxx]. Two results on loops with commuting inner mappings were obtained: (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, (1) is a natural complement to a recent result by Nagy and Vojěchovský claiming the same property when "Bruck" is replaced by "Moufang of odd order". Both theorems were obtained by Waldmeister from scratch, generating perhaps the most complicated proof ever obtained by a computer in loop theory. The final proof of the Bruck case 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. It's simplification and understanding is under current development.

5 Benchmark

5.1 QPTP library

The problems described in Section 4 are collected in the QPTP library. As such, QPTP is a representative collection of results in quasigroup and loop theory obtained by a computer. In its 12/08 distribution, reflecting the state at the end of 2008, it contains 109 problems (of which 88 unit equality). We intend to keep the QPTP library updated.

The problems are stored in an internal format, making an extensive use of a list of basic definitions. For example, the [Kun96a] problem saying that a quasigroup satisfying the first Moufang identity has a unit, can be stated

```
#assumptions:
```

	E 0.999-006	Gandalf c-2.6	Prover9 2008-09A	Spass 3.0	Vampire 9.0	Waldmeister 806
proofs in 300s	68	42	62	38	61	58
proofs in 3600s	79	69	70	48	76	69
proofs in 86400s	84	80	76	57	79	76
timeouts	25	29	33	52	30	12

Figure 4: Summary.

```
<<quasigroup
<<Moufang1
#goals:
x*(y/y)=x.
(y/y)*x=x.
```

We provide a tool for a translation to the TPTP format. QPTP files are split into several TPTP files whenever they contain multiple goals. For technical details, see the Readme file.

Problems are named after the code of the paper in the bibliography followed by the underline and the number of the selected result; multiple goals are then distinguished by letters (hence, e.g., `Phi06_2a` refers to the second problem selected from [Phi06], the first goal). If there are more reasonable ways to formalize the statement in first order theory, alternative axiomatizations are given.

5.2 Analysis

We have done a simple analysis of the problems in the QPTP library by running selected automated theorem provers. Based on the results of the CASC competition in recent years [SS06], we chose the following six provers: E [Sch02], Gandalf [Tam97], Prover9 [McC05], Spass [S], Vampire [RV02] and Waldmeister [Hil03]. (We experimented with iProver, too, but, with little surprize, its performance on unit equality problems was very poor.)

Each prover was running on each problem with 24 hours time limit. The input files were provided by translating the corresponding TPTP files using the tptp2X tool (coming with the distribution of the TPTP library). We ran the provers with their default settings, and we didn't tune any of the input files for a particular prover, thus obtaining conditions similar to the CASC competition.

The overall performance of the provers is summarized in Figure 4. Out of 109 problems, 99 were solved by at least one prover, 52 by all eligible ones. The 300s bound marks the time limit of the last CASC competition.

The detailed results are presented in Figure 5. Running time (i.e., the time it took to find a proof) is displayed in rounded seconds; a blank space means timeout, cross means that the problem is not equational and thus ineligible for Waldmeister. Running times below 300s are displayed in bold, running times over 1 hour in italic.

In our study, Waldmeister performed significantly better than the other five provers on equational problems. The performances of E, Gandalf, Prover9 and Vampire look similar (incomparable in the strict sense). Spass seems to be somewhat outdated nowadays.

Our results may seem somewhat surprizing: why Prover9 proved so useful in mathematical research, while it did relatively badly in the benchmark? This is partly due to how we organized the test: default settings, autonomous mode. On the other hand, focusing solely on one prover in the past was a mistake.

Indeed, parameter setting deserves much greater attention, but this is beyond the scope of the present study. Let's just note the perhaps the most influential parameter is the term ordering. For instance,

	E 0.999-006	Gandalf c-2.6	Prover9 2008-09A	Spass 3.0	Vampire 9.0	Waldmeister 806
AKP06_1	**0**	**0**	**9**	302	**6**	**0**
AKP06_2	**16**		**199**			**80**
AKP06_3	*69492*					
CDKxx_1a						
CDKxx_1b						
CDKxx_1c						
CGKxx_1	416	*13672*	1519			x
CGKxx_2	**35**		449			x
CGKxx_3	**0**	*4320*	**69**	1230	**4**	x
CGKxx_4	**0**	*7778*	**23**	**0**	**43**	x
CGKxx_4alt1	**0**	*7792*	**235**	2069	**266**	x
CGKxx_4alt2	**0**	*4333*	**25**	**0**	**10**	x
CGKxx_4alt3	**0**	*7780*	**90**	621	**3**	x
CGKxx_4alt4	**0**	*7781*	**98**	795	**3**	x
JKVxx_1		*9539*	**68**			**228**
JKVxx_1alt1		*9705*	**72**		**141**	**228**
JKVxx_2						351
JKVxx_3						
JKVxx_4						*28545*
KK04_1						*28020*
KK04_2						*31096*
KK04_3						*9778*
KK06_1a	**64**	**147**			508	**94**
KK06_1b	1031	1490				739
KK06_1c	*46267*	1043				725
KK06_1d	*53792*	1043				723
KK06_1e	*55126*	1037				723
KKP02a_1	1327	860	*32427*			x
KKP02a_1alt1	844	605			488	**224**
KKP02a_1alt2	854	610			493	**224**
KKP02a_1alt3	830	623			493	**224**
KKP02a_1alt4	842	622			502	**222**
KKP02b_1	**3**	**35**	**223**	**211**	398	x
KKP02b_1alt1	**8**	**47**	**230**	*82650*	495	**9**
KKP02b_1alt2	**3**	**35**	**170**	**244**	489	**9**
KKP02b_1alt3	**9**	**37**	**226**	*36184*	491	**10**
KKP02b_1alt4	**8**	**37**	**211**	*23118*	491	**10**
KKP02b_2	**0**	**145**	**0**	**184**	**10**	**1**
KKP02b_3	**0**	**78**	**0**	**0**	**0**	**2**
KKP02b_4a	**26**	315	1452		473	**5**
KKP02b_4b	**0**	**0**	**0**	**0**	**0**	**0**
KKP04_1a						
KKP04_1b						
KKP04_1c						
KKP04_2		*13059*			2998	517
KKP07_1						2079
KKPxx_1	**2**	592	**0**	**2**	**0**	**0**
KKPxx_2a						
KKPxx_2b						
KP02_1						
KP02_2	**1**	**1**	**58**	613	**16**	**87**
KP04a_1	**0**	**0**	**0**	**0**	**0**	**0**
KP04a_2	**0**	**0**	**0**	**0**	**0**	**0**
KP04a_3	**6**	**13**	**92**	*27531*	**137**	**3**
KP04b_1a	**3**	**1**	**182**	*9028*	**34**	**4**
KP04b_1b	**2**	**5**	**258**	*10703*	**25**	**5**
KP04b_2a	**3**	2094	**122**		464	
KP04b_2b	**0**	**10**	**38**	**205**	**7**	**79**
KP05_1a	**0**	1122	**0**	**0**	**0**	**0**
KP05_1b	**0**	1122	**0**	**0**	**0**	**0**
KP05_1c	**0**	**0**	**0**	**0**	**0**	**0**
KP05_1d	**0**	1121	**0**	**0**	**0**	**0**
KPV07_1	**0**	**0**	**0**	**0**	**0**	**0**
KPV08_1	**0**	**0**	**0**	**0**	**0**	**0**
KPV08_2	**0**	**0**	**0**	**0**	**0**	**0**
Kun00_1a	304		*8345*		*15659*	802
Kun00_1b	304		*8389*		1713	805
Kun00_1c	360		*7055*			799
Kun00_1alt1			*9435*			815
Kun00_2	**0**	**0**	**0**	**0**	**0**	**0**
Kun96a_1	**56**	336	**81**		**271**	x
Kun96a_1alt1	**151**	1090	**108**		**220**	**3**
Kun96a_1alt2	**9**	334	**189**		**238**	**3**
Kun96a_2	**58**	**13**	1889		**290**	x
Kun96a_2alt1	**8**	2053	1432		**247**	**4**
Kun96a_2alt2	**52**	**13**	1541		**85**	**4**
Kun96a_3	**0**	**0**	**0**	**0**	**0**	x
Kun96a_4	**0**	3241	**0**	**0**	**0**	x
Kun96b_1	**0**	**0**	**0**	**0**	**0**	x
Kun96b_2	**0**	**0**	**2**	**5**	**0**	x
Kun96b_3	**0**	**0**	**17**	**93**	**0**	x
Kun96b_3alt1	**0**	1080	**6**	**77**	**20**	**0**
Kun96b_3alt2	**0**	**0**	**5**	**91**	**42**	**0**
Kun98_1	**5**	*5431*	**151**	**158**	**2**	x
KVxx_1		1380	**254**		*3677*	**52**
KVxx_2			*6362*		*70723*	**104**
Phi06_1a	**61**		**28**		**14**	**23**
Phi06_1b	**39**		**2**	*4110*	**6**	**17**
Phi06_2a	**0**	**0**	**109**	**32**	**1**	**0**
Phi06_2b	**0**	**1**	**1**	**0**	379	**0**
Phi06_2c	**0**	**0**	**0**	**0**	**0**	x
Phi06_3	**0**	**3**	**41**	412	**9**	**0**
PSxx_1a						*10146*
PSxx_1b	*6227*				**229**	*8692*
PSxx_2						*48430*
PSxx_3	815	**60**		*18936*	**24**	**98**
PSxx_4a	**0**	**0**		2946	**6**	**0**
PSxx_4b	**0**	**0**	692	3388	**10**	**0**
PSxx_4c	**0**	**0**		*3962*	**10**	**0**
PSxx_4d	**0**	**1**	951	1545	**6**	**0**
PV05a_1	**0**	**0**	**1**	**8**	**6**	**0**
PV05a_2		**0**	**9**	**1**	**1**	**0**
PV05b_1	**0**	**0**	**0**	**0**	**0**	**0**
PV06_1a	**0**	1082	**0**	**0**	**0**	**0**
PV06_1b	**0**	1083	**0**	**0**	**0**	**0**
PV06_1c	**0**	1084	**0**	**0**	**0**	**0**
PV06_2	**35**	**1**	**28**		**5**	**0**
PV08_1a	**0**	1080	**0**	**0**	**0**	x
PV08_1b	**0**	1621	**0**	**0**	**0**	x

Figure 5: Detailed results.

Prover9's default ordering is LPO. If Prover9 is manually reset to KBO (which algebraists usually do), it proves a bit larger number of problems.

Finally, the reader may wonder about those theorems on which all provers were unsuccessful (these are indicated by blank entries in Figure 5). After all, these are theorems that were first proved with the assistance of an automated theorem prover (which was the sole criterion for inclusion in our study). Why were none of the provers able to find proofs in our study? The answer is threefold. Firstly, we did not use any advanced techniques in our study, such as the hints strategy (which is often the only way to obtain a new result). Secondly, we didn't tune the provers for each particular problem. And last but not least, we imposed a relatively short time limit.

6 Other areas of algebra

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, achieved mostly by the Argonne group and their collaborators since early 1990's (see [MP96] for references, or [MPV03], [MVFHFW02] for more recent results).

Several open problems were solved by automated theorem provers in the domain of lattices with operators (such as Boolean algebras and their many generalizations), the most prominent one being the Robbins problem [McC97]. Recently, many interesting questions that can be approached automatically are coming from algebraic logic, e.g. [VS06], [SV08].

We shall also make a reference to the book [MP96], an early attempt on using automated theorem provers in general algebra in a large scale.

In this paper, we focus on non-associative algebra. In addition to the many significant results in quasigroups and loops, there are several other attempts to use ATP in this field. In fact, we believe that this is a perfect playground for ATP, as the problems approached are often technical and unintuitive. We survey all related papers we know about.

[PV05c]. A term is called *linear* if each variable occurs at most once in it. An identity is said to be linear, if both sides are linear and contain the same variables. Otter and Mace4 were used in this paper to classify all varieties of groupoids defined by a single linear identity in three variables (there are exactly 14 nontrivial ones). Hentzel et. al. (1993) showed that the linear identity $(xy)z = y(zx)$ implies commutativity and associativity in all products of at least 5 factors. The present paper completes their project by showing that no other linear identity of any length behaves this way, and by showing how the identity $(xy)z = y(zx)$ affects products of fewer than 5 factors.

[DJMKS07]. This is an interesting example of exhaustive search. 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. For the computer solution, 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 all possible 2-generated free algebras (in fact, they appeared earlier in the literature). It took several days by hand and about 2 hours by computer to find all 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.

[Phi06b]. Prover9 helped to sharpen a result of D. A. Bredikhin (1992) by finding short equational bases for two varieties of groupoids associated with involuted restrictive bisemigroups of binary relations.

[APSxx]. Otter was used to prove some of the partial cases for a general conjecture that, in idempotent groupoids, certain term condition implies mediality.

[VM]. Automated theorem provers 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 reproved—much more compactly—a result by Kolibiar and Marcisová (1974) on median algebras, certain ternary algebras coming from modular lattices.

[Sta08]. This is another example, providing a direct proof of a decompostion result for distributive groupoids by Ježek and Kepka (1982). The theorem says that on every idempotent distributive groupoid G (i.e., G satisfies $xx = x$ and both left and right distributivity), there exists a congruence α of G such that G/α is symmetric and all blocks of α are medial. This is a nice example of a second order statement with a pretty simple but highly nonobvious first order formalization: existence of such a congruence on G is equivalent to the fact that G satisfies the identities

$$\begin{aligned}(xy\cdot zu)\cdot((xy\cdot zu)\cdot(xz\cdot yu)) &= xz\cdot yu\\ ((xy\cdot zu)\cdot(xz\cdot yu))\cdot(xz\cdot yu) &= xy\cdot zu\\ (xy\cdot zu)\cdot(xz\cdot yu) &= (xz\cdot yu)\cdot(xy\cdot zu).\end{aligned}$$

Several problems extracted from these papers can be found in the `nq` folder (for non-quasigroup problems) of the QPTP library. The problems did not participate in our benchmark, but they also were submitted for the TPTP library. Running theorem provers on these problems, we realized one remarkable case: Waldmeister fails on the distributive groupoid problems, even on those considered easy for other provers (see also the discussion in [Sta08]).

7 Conclusions

While we hope our results are interesting to automated reasoning researchers (especially since they involve problems from an active area of mathematical research), they may not be *surprising* to these same researchers, informed as these researchers are by the CASC results over the past ten years. Our results, though, might surprise loop theorists, who are less familiar with most of the provers in our study. But again, we stress that some of these loop theory results were originally obtained using advanced Otter/Prover9 techniques such as the hints strategy. Could these be implemented in other provers?

Since the various automated theorem provers have different strengths and weaknesses, loop theorists could profit by using a suite of theorem provers in their investigations. For instance, the result in [KKP07] was originally derived as a series of results, a number of steps eventually leading to the main theorem. In our study, Waldmeister proved it from scratch in 35 minutes. To state the obvious: some theorems will be

missed if one uses only one automated theorem prover. On the other hand, the actual proofs themselves are, of course, of great importance, and the various automated theorem provers differ greatly in this regard. A fruitful area for future research is simplification and interpretation of computer generated proofs.

The QPTP library itself offers many opportunities for future work. It is a relatively large collection of nontrivial but doable first order problems, in a single domain, yet of different nature. We believe QPTP can be exploited for further research, for instance, on optimization of first order formalizations, or on parameter setting.

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 (at least some of us). They can help us with some tedious work and, occasionally, even prove difficult theorems. 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. In order to attract even more mathematicians today, we suggest the following:

- Make the provers work in more developed areas. (This will probably require using large libraries of known results.)
- 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!

References

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

[AKP06] M. Aschbacher, M.K. Kinyon, and J.D. Phillips, *Finite Bruck loops*, Transactions of the American Mathematical Society, **358** (2006), 3061–3075.

[Bel67] V.D. Belousov, *Foundations of the Theory of Quasigroups and Loops*, Nauka, Moscow, 1967 (in Russian).

[Bru71] R. H. Bruck, *A Survey of Binary Systems*, third printing, corrected, Ergebnisse der Mathematik und ihrer Grenzgebiete 20, Springer-Verlag, 1971.

[CDKxx] P. Csorgö, A. Drápal, M.K. Kinyon, *Buchsteiner loops*, submitted.

[CGKxx] O. Chein, E. Goodaire, M.K. Kinyon, *When is a right alternative loop ring that is also right Bol actually Moufang?*, submitted.

[CS03] K. Claessen, N. Sörensson, *New Techniques that Improve MACE-style Model Finding* Proc. of Workshop on Model Computation (MODEL), 2003.

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

[DJxx] A. Drápal, P. Jedlička, On loop identities that can be obtained by nuclear identification, submitted.

[Hil03] T. Hillenbrand, Citius altius fortius: Lessons Learned from the Theorem Prover Waldmeister, in Dahn I., Vigneron L., Proceedings of the 4th International Workshop on First-Order Theorem Proving (Valencia, Spain), Electronic Notes in Theoretical Computer Science 86.1, Elsevier Science, 2003.

[JKVxx] P. Jedlička, M.K. Kinyon and P. Vojtěchovský, *The structure of commutative automorphic loops*, submitted.

[Kra00] A. Krapež, Rectangular loops, *Publ. Inst. Math. (Beograd)*, **68(82)** (2000), 59–66.

[Kun96a] K. Kunen, Moufang quasigroups *Journal of Algebra*, **183** (1996) no. 1, 231–234.

[Kun96b] K. Kunen, Quasigroups, loops, and associative laws *Journal of Algebra*, **185** (1996) no. 1, 194–204.

[Kun98] K. Kunen, Alternative loop rings. Comm. Algebra 26 (1998), no. 2, 557–564.

[Kun00] K. Kunen, The structure of conjugacy closed loops, *Transactions of the American Mathematical Society*, **352** (2000) no. 6, 2889–2911.

[KK04] M.K. Kinyon and K. Kunen, The structure of extra loops, *Quasigroups Related Systems*, **12** (2004), 39–60.

[KK06] M.K. Kinyon and K. Kunen, Power-associative, conjugacy closed loops, *Journal of Algebra*, **304** (2006), no. 2, 679–711.

[KKP02a] M.K. Kinyon, K. Kunen, and J.D. Phillips, Every diassociative A-loop is Moufang, *Proceedings pf the American Mathematical Society*, **17** 130 (2002), 619–624.

[KKP02b] M.K. Kinyon, K. Kunen, and J.D. Phillips, A generalization of Moufang and Steiner loops, *Algebra Universalis*, **48** (2002), 81–101.

[KKP04] M.K. Kinyon, K. Kunen, and J.D. Phillips, Diassociativity in conjugacy closed loops, *Communications in Algebra*, **32** (2004), 767–786.

[KKP07] T. Kepka, M.K. Kinyon, and J.D. Phillips, The structure of F-quasigroups, *Journal of Algebra*, 317 (2007), 435–461.

[KKPxx] M.K. Kinyon, K. Kunen, and J.D. Phillips, Strongly right alternative rings and Bol loops, *Publicationes Mathematicae Debrecen*, submittted.

[KP02] M.K. Kinyon and J.D. Phillips A note on trimedial quasigroups, *Quasigroups and Related Systems*, **9** (2002), 65–66.

[KP04a] M.K. Kinyon and J.D. Phillips, Commutants of Bol loops of odd order, *Proceedings of the American Mathematical Society*, **132** (2004), 617–619.

[KP04b] M.K. Kinyon and J.D. Phillips Axioms for trimedial quasigroups, *Commentationes Mathematicae Universitatis Carolinae*, **45** (2004), 287–294.

[KP05] M.K. Kinyon and J.D. Phillips, Rectangular quasigroups and rectangular loops, *Computers and Mathematics with Applications* **49** (2005), **11–12**, 1679–1685.

[KPV07] M.K. Kinyon, J.D. Phillips, and P. Vojtěchovský, C-loops: extensions and constructions, *Journal of Algebra and its Applications*, **6 (1)**, (2007), 1–20.

[KPV08] M.K. Kinyon, J.D. Phillips, and P. Vojtěchovský, When is the commutant of a Bol loop a subloop? *Transactions of the American Mathematical Society*, 360 (2008), no. 5, 2393–2408.

[KVxx] M.K. Kinyon and P. Vojtěchovský, Primary decompositions in varieties of commutative diassociative loops, to appear in Commun. Algebra.

[McC97] W. McCune, *Solution of the Robbins problem*, J. Autom. Reasoning 19, No.3, 263-276 (1997).

[McC03] W. W. McCune, *OTTER 3.3 Reference Manual and Guide*, Argonne National Laboratory Technical Memorandum ANL/MCS-TM-263, 2003;
`http://www.mcs.anl.gov/AR/otter/`

[McC05] W. W. McCune, *Prover9*, automated reasoning software, and *Mace4*, finite model builder, Argonne National Laboratory, 2005.
`http://www.prover9.org`

[MP96] W.W. McCune and R. Padmanabhan, *Automated Deduction in Equational Logic and Cubic Curves*, Springer-Verlag, 1996.

[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.

[Nag08] G. P. Nagy. *A class of finite simple Bol loops of exponent 2*, to appear in Trans. Amer. Math. Soc., 2008.

[NV07] G. P. Nagy and P. Vojtěchovský, *Computing with small quasigroups and loops*, Quasigroups and Related Systems 15 (2007), 77–94.

[Pfl90] H. O. Pflugfelder, Quasigroups and Loops: Introduction, *Sigma Series in Pure Mathematics* **7**, Heldermann Verlag Berlin, 1990.

[Pfl00] H. O. Pflugfelder, *Historical notes on loop theory*, Comment. Math. Univ. Carolin. 41/2 (2000), 359–370.

[Phi03] J.D. Phillips, See Otter digging for algebraic pearls, *Quasigroups and Related Systems*, **10** (2003), 95–

114.

[Phi06] J.D. Phillips, A short basis for the variety of WIP PACC-loops, *Quasigroups and Related Systems*, **14** (2006), 73–80.

[Phi06b] 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).

[Pud07] P. Pudlák, *Semantic Selection of Premisses for Automated Theorem Proving*, proceedings of the ESARLT Workshop, Bremen, 2007.

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

[PSxx] J.D. Phillips, D. Stanovský, *Loops with abelian inner mapping groups*, work in progress.

[PV05a] J.D. Phillips and P. Vojtěchovský, The varieties of loops of Bol-Moufang type, *Algebra Universalis*, **54 (3)** (2005), 259–271.

[PV05b] J.D. Phillips and P. Vojtěchovský, The varieties of quasigroups of Bol-Moufang type: an equational reasoning approach, *Journal of Algebra*, 293 (2005), 17–33.

[PV05c] J.D. Phillips and P. Vojtěchovský, Linear groupoids and the associated wreath products, *Journal of Symbolic Computation*, **40 (3)**, (2005), 1106–1125.

[PV06] J.D. Phillips and P. Vojtěchovský, C-loops: an introduction, *Publicationes Mathematicae Debrecen*, **68/1–2** (2006), p. 115–137.

[PV08] J.D. Phillips and P. Vojtěchovský, A scoop from groups: new equational foundations for loops, *Commentationes Mathematicae Universitatis Carolinae*, 49/2 (2008), 279–290.

[RV02] A. Riazanov, A. Voronkov, The Design and Implementation of Vampire, *AI Communications* 15(2-3) (2002), 91–110.

[S] `http://www.spass-prover.org/`

[Sch02] S. Schulz, E: A Brainiac Theorem Prover, *AI Communications* 15(2-3) (2002), 111–126.

[SV08] M. Spinks, R. Veroff, *Constructive logic with strong negation is a substructural logic*, Studia Logica 88/3 (2008), 325–348.

[Sta08] D. Stanovský, *Distributive groupoids are symmetric-by-medial: An elementary proof*, Comment. Math. Univ. Carolinae 49/4 (2008), 541–546.

[SS98] G. Sutcliffe, C. Suttner, The TPTP Problem Library: CNF Release v1.2.1, *Journal of Automated Reasoning*, 21/2 (1998), 177–203.

[SS06] G. Sutcliffe, C. Suttner, The State of CASC, *AI Communications* 19/1 (2006), 35–48.

[Tam97] T. Tammet, *Gandalf*, J. of Automated Reasoning 18/2 (1997), 199–204.

[Ver01] 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).