Short Equational Bases for Ortholattices

W. McCune[*] *R. Padmanabhan*[†] *M. A. Rose*[‡] *R. Veroff*[§]

January 20, 2004

Abstract

Short single axioms for ortholattices, orthomodular lattices, and modular ortholattices are presented, all in terms of the Sheffer stroke. The ortholattice axiom is the shortest possible. Other equational bases in terms of the Sheffer stroke and in terms of join, meet, and complement are presented. Proofs are omitted but are available in an associated technical report. Computers were used extensively to find candidates, reject candidates, and search for proofs that candidates are single axioms. The notion of computer proof is addressed.

1 Introduction

When it comes to mathematics, one of the most creative of human endeavors, computers are often believed to be intrinsically limited compared with humans. But today, when most mathematicians and their students have access to high-speed computers, a more useful approach is to ask "What kinds of mathematics can computers do?" and "How can mathematicians use computers as extensions of their thought processes?" In this paper we show how suitably programmed computers can be of immense help to humans in solving problems in equational logic—problems that might be too difficult for humans to solve by traditional methods. We believe that computers can be programmed to assist in the development of mathematics at many levels, ranging from routine symbolic computation and tedious deduction, through the discovery of useful proofs and countermodels (the focus of this paper), to the formation of interesting concepts.

Consider the problem of expressing equational theories as simply as possible—with the least number of symbols, the least number of equations, the least number of operations, and the least number of variables. The problem for Abelian groups was solved by Tarski in 1938 [19] with the single axiom (i.e., one

[*]Supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

[†]Supported by an operating grant from NSERC of Canada, grant #8215-02.

[‡]Supported by the Division of Educational Programs, Argonne National Laboratory.

[§]Supported by National Science Foundation grant CDA-9503064.

equation from which the theory can be derived) $x/(y/(z/(x/y))) = z$ in terms of the division operation. Other varieties of groups (including other operations) have been addressed, and short single axioms have been found, but minimality has not been proved in most cases.

The problem for Boolean algebra ($\mathcal{BA}$) was solved recently with a shortest single axiom in terms of the Sheffer stroke (i.e., the NAND operation) [12]. Progress has recently been made also with lattices ($\mathcal{L}$), with a reasonably short single axiom in terms of join and meet [10]. Here we look at a chain of varieties between the lattices and the Boolean algebras, namely, ortholattices ($\mathcal{OL}$), orthomodular lattices ($\mathcal{OML}$), and modular ortholattices ($\mathcal{MOL}$).

Although ortholattices and subvarieties are of interest in the study of Hilbert spaces and the logical foundations of quantum computing [16], the search for shortest single axioms is mostly a puzzle. The main results of the work we report here are all in terms of the Sheffer stroke—a shortest single axiom for $\mathcal{OL}$, and reasonably short single axioms for $\mathcal{OML}$ and $\mathcal{MOL}$. Secondary results include simple multiequation axiomatizations for these varieties.

We used several computer programs in our investigations. Otter [6] searches for proofs, Mace [4, 5] searches for (counter)models, and other programs embody decision and enumeration procedures. Otter and Mace are mature, stable, and well-documented systems, available for download from the Web page associated with this paper [9].

Our presentation has elements of a case study, with some details of how computing was used to obtain the results. Computer proofs and countermodels have been omitted because they are long, and also because we wish to focus instead on the methods by which they were discovered. The proofs can be found in a technical report [8] and on a Web page [9]. The Web page contains input files for the programs and full listings of the proofs and countermodels produced by the programs.

2 Equational Bases

First we define a chain of varieties from lattices to Boolean algebra in terms of join, meet, and complement. Then we go from ortholattices to Boolean algebra in terms of the Sheffer stroke.

2.1 In Terms of Join/Meet/Complement

A *lattice* is a nonempty set with two binary operations, join ($\vee$) and meet ($\wedge$), satisfying the following four (independent) laws [1].

AJ:	$x \vee (y \vee z) = y \vee (x \vee z)$	AM:	$x \wedge (y \wedge z) = y \wedge (x \wedge z)$
B1:	$x \vee (x \wedge y) = x$	B2:	$x \wedge (x \vee y) = x$

We say that { AJ, B1, AM, B2 } is a 4-basis for the equational theory of lattices ($\mathcal{L}$) in terms of join and meet. (The commuted forms of associativity allow us to do without the commutativity laws.)

The *ortholattices* ($\mathcal{OL}$) are the lattices with a complement operation ($'$) satisfying the following three laws.

DM: $x \wedge y = (x' \vee y')'$
CC: $x'' = x$
ONE: $x \vee x' = y \vee y'$

The ortholattice laws, however, link join and meet in such a way that the right-hand (or the left-hand) column of the lattice laws become dependent, that is, are derivable from the remaining five laws. In particular, the subset { AJ, B1, DM, CC, ONE } is an independent 5-basis for $\mathcal{OL}$. From this point through the end of this section, we could eliminate the meet operation by using the DM law to rewrite everything; doing so, however, would complicate the other laws. Also note that the law ONE states the existence of a constant, usually referred to as "1", and that law can be written as $x \vee x' = 1$ instead.

The *orthomodular lattices* ($\mathcal{OML}$) are the ortholattices satisfying the law

OM: $x \vee (x' \wedge (x \vee y)) = x \vee y.$

Adding this new law to our $\mathcal{OL}$ 5-basis causes the CC and ONE laws to become dependent; that is, the set { AJ, B1, DM, OM } is an independent 4-basis for $\mathcal{OML}$.

The *modular ortholattices* ($\mathcal{MOL}$) are the ortholattices satisfying the modularity law

MOD: $x \vee (y \wedge (x \vee z)) = x \vee (z \wedge (x \vee y)).$

If we add MOD to our 5-basis for $\mathcal{OL}$, we obtain the independent $\mathcal{MOL}$ 6-basis { AJ, B1, DM, CC, ONE, MOD }. It turns out that the OM law can be easily derived from our $\mathcal{MOL}$ basis, showing that the $\mathcal{MOL}$s are a subvariety of the $\mathcal{OML}$s.

Finally, the *Boolean algebras* ($\mathcal{BA}$) can be defined as the ortholattices satisfying

CUT: $(x \vee y') \wedge (x \vee y) = x.$

If we add CUT to the $\mathcal{OL}$ 5-basis, then B1 and CC become dependent, and we have the 4-basis { AJ, DM, ONE, CUT } for Boolean algebra (independence is open). The law MOD can be derived from our $\mathcal{BA}$ 4-basis, showing that the $\mathcal{BA}$s are a subvariety of the $\mathcal{MOL}$s.

In summary, we have the chain of varieties

$$\mathcal{L} \supset \mathcal{OL} \supset \mathcal{OML} \supset \mathcal{MOL} \supset \mathcal{BA}$$

(Table 1 in Section 3 shows that the inclusions are proper) and the following bases for each variety.

$\mathcal{L}$: { AJ, B1, AM, B2 }
$\mathcal{OL}$: { AJ, B1, DM, CC, ONE }
$\mathcal{OML}$: { AJ, B1, DM, OM }
$\mathcal{MOL}$: { AJ, B1, DM, CC, ONE, MOD }
$\mathcal{BA}$: { AJ, DM, ONE, CUT }

All but the $\mathcal{BA}$ basis are known to be independent.

2.2 In Terms of the Sheffer Stroke

The lattices ($\mathcal{L}$) cannot be defined in terms of a single binary operation [1], but $\mathcal{OL}$ and its subvarieties can be, in particular, in terms of the Sheffer stroke "|".

$\mid$ in terms of $\vee, \wedge, '$	$\vee, \wedge, '$ in terms of $\mid$
$x \mid y = x' \vee y'$	$x \vee y = (x \mid x) \mid (y \mid y)$
	$x \wedge y = (x \mid y) \mid (x \mid y)$
	$x' = x \mid x$

Question: If we rewrite a basis to a different set of operations, do we get a basis in terms of that other set of operations? *Answer*: Sometimes. Translating between Sheffer stroke and join/meet/complement bases is a good illustration. Using the definitions just given to simply rewrite a join/meet/complement basis in terms of the Sheffer stroke *always* gives us a basis in terms of the Sheffer stroke. However, if we rewrite a Sheffer stroke basis in terms of join and complement, we *never* get a basis in terms of join and complement [12].

Even when the translation gives us a basis, it can produce complicated equations; for example, equation AJ in terms of the Sheffer stroke is

$$(x|x)|(((y|y)|(z|z))|((y|y)|(z|z))) = (y|y)|(((x|x)|(z|z))|((x|x)|(z|z))).$$

We can do better than that. In fact, because the Sheffer stroke operation builds in properties of complementation, we can find simpler bases by using the Sheffer stroke rather than join/meet/complement. We list here independent bases for the varieties in question (see [8] or [9] for proofs).

Consider the following equations.

$\widehat{\mathrm{A}}$:	$x \mid ((y \mid z) \mid (y \mid z)) = y \mid ((x \mid z) \mid (x \mid z))$
$\widehat{\mathrm{B}}$:	$(x \mid x) \mid (x \mid y) = x$
$\widehat{\mathrm{ONE}}$:	$x \mid (x \mid x) = y \mid (y \mid y)$
$\widehat{\mathrm{OM}}$:	$x \mid (x \mid (x \mid y)) = x \mid y$
$\widehat{\mathrm{MOD}}$:	$x \mid (y \mid (x \mid (z \mid z))) = x \mid (z \mid (x \mid (y \mid y)))$
$\widehat{\mathrm{CUT}}$:	$(x \mid (y \mid y)) \mid (x \mid y) = x$

Then we have the following independent bases.

$\mathcal{OL}$:	{ $\widehat{\mathrm{A}}$, $\widehat{\mathrm{B}}$, $\widehat{\mathrm{ONE}}$ }
$\mathcal{OML}$:	{ $\widehat{\mathrm{A}}$, $\widehat{\mathrm{B}}$, $\widehat{\mathrm{OM}}$ }
$\mathcal{MOL}$:	{ $\widehat{\mathrm{A}}$, $\widehat{\mathrm{B}}$, $\widehat{\mathrm{ONE}}$, $\widehat{\mathrm{MOD}}$ }
$\mathcal{BA}$:	{ $\widehat{\mathrm{A}}$, $\widehat{\mathrm{CUT}}$ }

2.3 Finding and Proving the Multiequation Bases

Aside from the trick of commuting the associativity laws and the nonstandard modularity axiom, the join/meet/complement bases are fairly straightforward.

The Sheffer stroke bases were found by a combination of human reasoning and computer deduction, including some trial and error. The join/meet/complement bases were rewritten to Sheffer stroke equations then simplified in various heuristic ways, for example, (1) rewrite a basis to Sheffer stroke, (2) use Otter to derive consequences of the Sheffer stroke equations, and then (3) look for simple derived equations that might capture the key properties.

Proofs from Otter and countermodels from Mace (independence proofs) can be found on the Web [9]. These include Otter proofs that the join/meet/complement bases are equivalent to more standard bases for these varieties and Otter proofs that the Sheffer stroke bases are definitionally equivalent to the corresponding join/meet/complement bases.

2.4 Are There Simpler Multiequation Bases?

Our goal in looking for the multiequation bases was to find short, intuitive, and fairly standard bases in terms of join/meet/complement, and then to find similar bases in terms of the Sheffer stroke. We doubt that the preceding bases are the shortest. In fact, for $\mathcal{BA}$ in terms of the Sheffer stroke, the 2-basis with the least number of symbols is known to be $\{\ x \mid y = y \mid x,\ (x \mid y) \mid (x \mid (y \mid z)) = x\ \}$ [20]. We leave open the problem of finding the simplest multiequation bases for the other cases. For $\mathcal{BA}$ in terms of the join and complement, the simplest 2-basis we know of, by C. A. Meredith [13], is $\{\ (x' \vee y)' \vee x = x,\ (x' \vee y)' \vee (z \vee y) = y \vee (z \vee x)\ \}$.

3 Single Axioms

Existence of single equational axioms (1-bases) for $\mathcal{OML}$, $\mathcal{MOL}$, and $\mathcal{BA}$ has been known since 1973 [15, 3], and for $\mathcal{OL}$ has been known since 1977 [14]. By those results, if a variety has particular properties and if there is a basis for that variety with particular syntactic properties, then there exist procedures to construct single axioms for the variety. The varieties in question have the required properties, but the basic procedures have exponential behavior, producing very large axioms, sometimes with millions of symbols. The procedures can be optimized somewhat [7], but they still tend to produce axioms with hundreds of symbols.

Our approach is to start small, considering all possible candidates of a given size, and looking at sizes as large as practical. Candidates that are too strong (not valid in the variety) and those that are too weak (usually determined by finding a finite countermodel) can be eliminated. If all goes well, we can show that a candidate is a shortest single axiom by proving a known basis and by eliminating all shorter candidates. In other cases we can find nice single axioms without being able to eliminate all shorter candidates.

A similar approach led, in previous work, to axiom LT1 (below) for lattice theory [10], axiom BA1 for Boolean algebra [12], and BA2, which is a shortest axiom for Boolean algebra in terms of the Sheffer stroke [12].

LT1: $(((y \vee x) \wedge x) \vee (((z \wedge (x \vee x)) \vee (u \wedge x)) \wedge v)) \wedge (w \vee ((s \vee x) \wedge (x \vee t))) = x$

BA1: $((y \vee z)' \vee x')' \vee ((u' \vee u)' \vee (x' \vee y))' = x$

BA2: $(y \mid ((x \mid y) \mid y)) \mid (x \mid (z \mid y)) = x$

Why start with the Sheffer stroke? In the case of Boolean algebra, the join/complement axiom BA1 was much harder to find than the Sheffer stroke axiom BA2; in addition, BA2 was shown to be a shortest axiom, and BA1 was not.

3.1 Generating and Filtering Candidates

Let the *length* of a term or equation be the number of occurrences of variables and operators (including the equal sign but not parentheses). For example, $x \vee (x \wedge y) = x$ has length 7.

Every Sheffer stroke single axiom for $\mathcal{OL}$, $\mathcal{OML}$, $\mathcal{MOL}$, or $\mathcal{BA}$ has the following properties (see [12] for justifications).

1. The length is odd (this holds for all equations written with just binary operations).
2. The equation must have at least three variables. Otherwise there can be nonassociative models.
3. One side of the equation (say the right-hand side) must be a variable, say x.
4. The leftmost variable of the left-hand side of the equation cannot be x. Such candidates are eliminated by left-projection ($x|y = x$) models. Similarly, the right-most variable of the left side cannot be x.
5. It cannot be of the form $y|\alpha = x$ or $\alpha|y = x$ for any variable y and any term α.
6. If an equation is a single axiom, its mirror image is also a single axiom. Therefore we can restrict our attention to equations $\alpha|\beta = x$ where $\text{length}(\alpha) \leq \text{length}(\beta)$.
7. The equation is true in all Boolean algebras. This follows from the next property, but it is a very fast test, so we include it here.
8. It is true in all models of the variety in question ($\mathcal{OL}$, $\mathcal{OML}$, $\mathcal{MOL}$, or $\mathcal{BA}$).
9. It is false in all nonmodels of the variety in question.

The procedure to generate candidates is roughly as follows.

- Generate all well-formed Sheffer stroke equations of a given length satisfying Properties 1–6.
- Pass the equations through a decision procedure for Boolean algebra identities (Property 7). In practice we simply check the candidate against the 2-element Boolean algebra. A vast majority of the equations are removed by this check.
- Property 8 can be checked correctly if there is a decision procedure for identities of the variety. Without a decision procedure, we can test identities against a set of finite models of the variety; this test admits all identities, but it may also admit some nonidentities, so we have to be prepared to prove later that they are valid. We have a decision procedure for $\mathcal{OL}$, but the equational theories for $\mathcal{OML}$ and $\mathcal{MOL}$ are unsolvable [2, p. 218].
- Property 9 eliminates candidates that are too weak to be single axioms. We do not have a perfect test for this. In practice, we iteratively collect sets of nonmodels by using the program Mace, which searches for finite (counter)models. Consider $\mathcal{OL}$; if a candidate is false in all of the current non-$\mathcal{OL}$s, we use Mace to look for non-$\mathcal{OL}$ models of the candidate. If one is found, we add it to the set and eliminate the candidate. We call this process *filtering the candidates*, and we refer to the nonmodels as *filters*.

3.2 Finite Ortholattices

If we do not have a decision procedure for the variety, we need some of its members to approximate Property 8. Table 1 shows the numbers of members of the varieties up through size 20. These algebras were generated (quickly, except for the $\mathcal{OL}$s of size 14, which took several hours, and the $\mathcal{OML}$s of size 20, which took two weeks) by the programs Mace and Isofilter; details, including the listings of the structures, can be found on the Web [9].

Size	$\mathcal{OL}$	$\mathcal{OML}$	$\mathcal{MOL}$	$\mathcal{BA}$
2	1	1	1	1
4	1	1	1	1
6	2	1	1	0
8	5	2	2	1
10	15	2	1	0
12	60	3	2	0
14	311	4	1	0
16	?	7	3	1
18	?	8	1	0
20	?	14	2	0

Table 1: Numbers of Finite Structures

These algebras can be used as filters (for Property 9) as well. For example, any non-$\mathcal{MOL}$ can be used to eliminate $\mathcal{MOL}$ candidates; the nonmodular $\mathcal{OML}$s and $\mathcal{OL}$s are useful for that purpose.

3.3 Collecting and Applying Filters

When searching for single axioms, we considered the varieties in the order $\mathcal{BA}$ [12], $\mathcal{OL}$, $\mathcal{OML}$, $\mathcal{MOL}$. We were fortunate, because this is also the order of increasing difficulty (with respect to finding good candidates), and each case gave us techniques and filters useful for the next. In the $\mathcal{BA}$ case, all candidates shorter than the axiom that was found (BA2, length 15) can be eliminated by noncommutative structures of size 3 or 4. This can be done automatically in a few seconds; see [9] for details. (Proving that BA2 is a single axiom, however, is difficult [20].)

In the $\mathcal{OL}$ case, we have a decision procedure for identities, which gives us a perfect test for Property 8. For Property 9, all candidates up through length 21 can be eliminated by a set of four non-$\mathcal{OL}$s [9, file non-OL.A-4] of sizes 3, 6, 6, and 8. Many more non-$\mathcal{OL}$s were collected, but those four were sufficient. A single axiom (OL-Sh below) was found among the candidates of length 23.

In the $\mathcal{OML}$ case, we do not have a decision procedure, but the $\mathcal{OML}$s up through size 10 were adequate for Property 8. All candidates up through length 19 can be eliminated with a set of nine non-$\mathcal{OL}$s [9, file non-OL.B-9], all of size ≤ 6. Length 21 was a challenge—we could not eliminate all candidates, and we could not prove any of the survivors to be single axioms. A set of 23 non-$\mathcal{OL}$s was accumulated [9, file non-OL.C-23], eliminating all but 58 candidates. A single axiom (OML-Sh below) was found among the candidates of length 23.

The $\mathcal{MOL}$ case started out like the $\mathcal{OML}$ case, with the elimination of all candidates up through length 19 by using the same filters as in the $\mathcal{OML}$ case. For length 21, 14 more non-$\mathcal{OL}$s were accumulated [9, file non-OL.D-14], and the nine nonmodular $\mathcal{OML}$s up through size 16 (see Table 1) were also used as filters. However, 238 length 21 candidates survived, and none was proved to be a single axiom. As the candidates grow, it becomes more difficult to find countermodels, so we used the existing non-$\mathcal{MOL}$s for lengths 23, 25, and 27. A single axiom (MOL-Sh below) was found among those of length 25.

3.4 Trying to Prove That Candidates Are Single Axioms

Given a set of candidates that had survived all the filters, we tried to prove each to be a single axiom by deriving a known basis, for example, the independent bases given in Section 2.

Automatic proofs were attempted with hundreds of $\mathcal{OL}$ candidates, thousands of $\mathcal{OML}$ candidates, and hundreds of thousands of $\mathcal{MOL}$ candidates before proofs were found for the three cases. The time allocated for each candidate varied from a few minutes to a few seconds, depending on the size of the set. For each proof attempt, we included as goals several important properties

of the variety as well as a known basis. If some interesting properties were derived from the candidate, but not enough for a complete proof, we investigated that candidate later with focused proof attempts.

Length 23 single axioms for $\mathcal{OL}$ and $\mathcal{OML}$ were found without much difficulty. The proofs were not trivial for Otter, but they were found automatically within a few minutes. Finding a $\mathcal{MOL}$ axiom was much more difficult. Many more candidates had to be considered, and proofs with the successful candidates were not found automatically. Promising candidates (those that proved the most interesting properties) were selected from the automatic attempts, and advanced automated deduction techniques involving human guidance (i.e., the method of hints and sketches [21, 22]) were applied, producing a proof for one candidate of length 25.

For the $\mathcal{OML}$ and $\mathcal{MOL}$ cases, which were generated with the imperfect Property 8 test, we also had to prove that the successful candidates are valid in the variety by deriving the candidate from a known basis.

The proofs and more details on the proof searches can be found on the Web [9].

3.5 Single Axioms for $\mathcal{OL}$, $\mathcal{OML}$, and $\mathcal{MOL}$

We give here the main results of the project—single axioms, in terms of the Sheffer stroke, for $\mathcal{OL}$, $\mathcal{OML}$, and $\mathcal{MOL}$.

$$\begin{array}{ll}
\text{OL-Sh:} & (((y \mid x) \mid (x \mid z)) \mid u) \mid (x \mid ((x \mid ((y \mid y)) \mid z)) = x \\
\text{OML-Sh:} & (((y \mid x) \mid (x \mid z)) \mid u) \mid (x \mid ((z \mid ((x \mid x) \mid z)) \mid z)) = x \\
\text{MOL-Sh:} & (y \mid x) \mid (((x \mid x) \mid z) \mid (((((x \mid y) \mid z) \mid z) \mid x) \mid (x \mid u))) = x
\end{array}$$

The $\mathcal{OL}$ axiom (length 23) is the shortest possible. We do not know whether the $\mathcal{OML}$ axiom (length 23) is shortest, because there are 58 open candidates of length 21. We doubt that the $\mathcal{MOL}$ axiom (length 25) is shortest, because the proof that it is a single axiom is very difficult, and there are many open candidates that are shorter.

Each of the axioms has four variables, and the question of short 3-variable axioms is open. In the $\mathcal{OL}$ case, all of the surviving length-23 candidates have 4 variables, so any 3-variable $\mathcal{OL}$ axioms must have length ≥ 25. In the $\mathcal{OML}$ case, four of the 58 length-21 and many of the length-23 candidates have three variables. In the $\mathcal{MOL}$ case, many of the surviving candidates of lengths 21 and 23 have three variables.

4 The Computer Programs

Symbolic computation was used in five ways in this work: (1) to enumerate equations subject to a set of syntactic constraints, (2) to evaluate equations with respect to finite structures, (3) to decide ortholattice identities, (4) to search for equational proofs, and (5) to search for finite structures that satisfy sets of equations and disequations. The first three are relatively straightforward,

although the programs were coded efficiently so that they could handle billions of equations.

Proof search methods and corresponding completeness questions are well covered in the literature of automated deduction, and automated proof search is being applied occasionally to problems in abstract algebra and formal logic. Search methods for finite algebras is less well known and often overlooked. Researchers are starting to apply it, however, in many of the same areas and projects as proof search.

Otter. The program Otter [6] searches for proofs of statements in the first-order predicate calculus with equality. Although it can be applied to any first-order statement, it is usually more effective on problems with fewer operations and simpler statements, and especially on equational problems. Otter has an automatic mode in which the user simply gives the statement of the problem, and standard strategies are applied.

For difficult problems, however, the user usually sets various switches to control the search. For a given area, the user can develop a strategy on easy problems and then apply that strategy to more interesting problems. The strategies that played an important part in this work included (1) limiting the size of derived equations, (2) the relative emphasis of short equations as opposed to breadth-first search, and (3) selecting symbol orderings and goal bases to determine whether the search would be conducted in terms of the Sheffer stroke or in terms of join, meet, and complement. See [9] for detailed examples.

Especially difficult problems (such as the $\mathcal{MOL}$ single axiom candidates) were attacked by iterating searches, that is, examining the output, adjusting the search strategy, and trying again.

Mace. The program Mace [4, 5] searches for finite countermodels of the same class of statement as Otter accepts. In many cases the two programs can use the same input files, and we frequently run the two programs in parallel on the same problem, with Otter searching for a proof and Mace looking for a countermodel. Like Otter, Mace seems to prefer problems with few operations and simple statements, especially equational problems. Unlike Otter, Mace is mostly automatic, with little guidance expected from the user. The size of structure can be specified; otherwise it starts small and iterates.

When Mace is used to look for countermodels, specifying additional constraints can (with the risk of losing completeness) make an enormous difference in the time required to find models. The properties of ortholattices (e.g., commutativity and associativity) are natural examples of additional constraints, but unrelated properties such as quasigroup properties have also been useful. See [9] for detailed examples.

If no goal is given to Mace, it will simply search for structures satisfying the constraints rather than for a countermodel. An example is finding all the ortholattices of size 12. When asked to find more than one model of a given size, Mace is not very smart about isomorphism, and a separate program, *Isofilter*,

can be called to remove the isomorphic models. When looking for all of the $\mathcal{OL}$s of size 12, Mace finds 36,821 models (in a few minutes), and Isofilter shows that all but 60 are isomorphic (also in a few minutes).

Several other powerful theorem provers and finite model generators have been developed by other research groups. Each year the International Conference on Automated Deduction (CADE) hosts a friendly contest (CASC) in which computer programs compete, trying to prove or disprove large numbers of problems of various types. Results of the 2001 competition are reported in [17], and some of the competing programs are available for download and general use.

5 Conclusion

At what point does symbolic computation become a sound, relevant, and interesting computer proof? The would-be proofs in this work fall into several classes: proofs by equational deduction, independence proofs by finite countermodels, and minimality proofs by exhaustive enumeration.

We are quite confident that the first two kinds of proof are sound. Otter and Mace produce results that can be checked by independent programs or by humans. Otter presents detailed line-by-line proofs at a very low level that can be checked by very simple proof-checking programs. Furthermore, program verification techniques have been applied to the proof checkers [11]. Mace presents structures as tables that can be checked in similar ways by independent programs. The Otter proofs and Mace countermodels have been machine checked; they have not, however, been fully checked by humans.

The minimality proofs are fundamentally different from the Otter or Mace proofs. The minimality proofs are similar in spirit (though not in scale or interest) to proofs of the four-color theorem and Thomas Hales's proof of Kepler's conjecture on arrangement of spheres [18]. In short, the problem is reduced to a finite set of cases that are checked by computers. Soundness is especially questionable, with reliance (in our case) on optimized special-purpose code for the equation generation and decision procedures. We doubt that much can be learned from the various components of the minimality proofs.

Otter proofs and Mace counterexamples, on the other hand, are creative in the sense that the users had no idea what the proofs or structures might be. In other projects, Otter has found interesting proofs (e.g., much shorter than previous proofs) and Mace has found structures that are useful in further work; but here, the proofs and structures are secondary to the short equational descriptions that were found. The value of Otter and Mace, in this project, has been reliable and fast deductive support for higher pursuits.

References

[1] G. Grätzer. *General Lattice Theory*. Birkhauser Verlag, 2nd edition, 1998.

[2] G. Kalmbach. *Orthomodular Lattices*. Academic Press, New York, 1983.

[3] D. Kelly and R. Padmanabhan. Orthomodular lattices and congruence permutability. Preprint, 2003.

[4] W. McCune. MACE 2.0 Reference Manual and Guide. Tech. Memo ANL/MCS-TM-249, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, June 2001.

[5] W. McCune. Mace4 Reference Manual and Guide. Tech. Memo ANL/MCS-TM-263, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, August 2003.

[6] W. McCune. Otter 3.3 Reference Manual. Tech. Memo ANL/MCS-TM-263, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, August 2003.

[7] W. McCune and R. Padmanabhan. Single identities for lattice theory and for weakly associative lattices. *Algebra Universalis*, 36(4):436–449, 1996.

[8] W. McCune, R. Padmanabhan, M. A. Rose, and R. Veroff. Short equational bases for ortholattices: Proofs and countermodels. Tech. Report TR-CS-2004-02, Computer Science Department, University of New Mexico, Albuquerque, NM, January, 2004.

[9] W. McCune, R. Padmanabhan, M. A. Rose, and R. Veroff. Short equational bases for ortholattices: Web support. `http://www.mcs.anl.gov/~mccune/papers/olsax/`, 2003.

[10] W. McCune, R. Padmanabhan, and R. Veroff. Yet another single law for lattices. *Algebra Universalis*. To appear.

[11] W. McCune and O. Shumsky. IVY: A preprocessor and proof checker for first-order logic. In M. Kaufmann, P. Manolios, and J Moore, editors, *Computer-Aided Reasoning: ACL2 Case Studies*, chapter 16. Kluwer Academic, 2000.

[12] W. McCune, R. Veroff, B. Fitelson, K. Harris, A. Feist, and L. Wos. Short single axioms for Boolean algebra. *J. Automated Reasoning*, 29(1):1–16, 2002.

[13] C. A. Meredith and A. N. Prior. Equational logic. *Notre Dame J. Formal Logic*, 9:212–226, 1968.

[14] R. Padmanabhan. Equational theory of algebras with a majority polynomial. *Algebra Universalis*, 7(2):273–275, 1977.

[15] R. Padmanabhan and R. W. Quackenbush. Equational theories of algebras with distributive congruences. *Proc. AMS*, 41(2):373–377, 1973.

[16] M. Pavičić and N. Megill. Non-orthomodular models for both standard quantum logic and standard classical logic: Repercussions for quantum computers. *Helv. Phys. Acta*, 72(3):189–210, 1999.

[17] G. Sutcliffe, C. Suttner, and F. J. Pelletier. The IJCAR ATP system competition. *J. Automated Reasoning*, 28(3):307–320, 2002.

[18] G. Szpiro. Mathematics: Does the proof stack up? *Nature*, 242:12–13, July 2003.

[19] A. Tarski. Ein Beitrag zur Axiomatik der Abelschen Gruppen. *Fundamenta Mathematicae*, 30:253–256, 1938.

[20] R. Veroff. A shortest 2-basis for Boolean algebra in terms of the Sheffer stroke. *J. Automated Reasoning*, 31(1):1–9, 2003.

[21] R. Veroff. Using hints to increase the effectiveness of an automated reasoning program: Case studies. *J. Automated Reasoning*, 16(3):223–239, 1996.

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

Authors' Addresses

William McCune
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

R. Padmanabhan
Department of Mathematics
University of Manitoba
Winnipeg R3T 2N2
CANADA

Michael A. Rose
Department of Mathematics
University of Wisconsin-Madison
Madison, WI 53706

Robert Veroff
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131