Shortest Single Axioms for the Equivalential Calculus with CD and RCD*

Kahlil Hodgson

Department of Mathematics
The University of Auckland, New Zealand
e-mail: khod01@math.auckland.ac.nz

April 15, 1996

Abstract

The automated reasoning assistants OTTER and MACE are applied to the following open problem in equivalential calculus: which equivalential tautologies will serve as shortest single axioms with both condensed detachment and reversed condensed detachment as inference rules. Earlier uses of OTTER are extended and executed systematically using a generic driver program. The application of MACE to this problem required the development of new techniques. A description of the 630 possibilities is given, and seven new shortest single axioms are presented. All but five of the remaining possibilities (up to duality) are shown to be too weak, and it is noted that the above results supersede in some respects the results of a similar unpublished paper.

1 Introduction

Classical equivalential calculus EC refers to the fragment of propositional calculus involving equivalence as its only connective. This is essentially all the well-formed formulae using only the function symbol E and containing an even number of each variable (cf. [**?**], [**?**, pages 1-23]).

A typical model for the equivalential calculus is the normal logical matrix $\mathcal{E} = \langle\{1,0\}; e; \{0\}\rangle$, where e is the operation on $\{1,0\}$ given by the following table:

	e	0	1
$*$	0	0	1
	1	1	0

*This paper was produced while the author was studying at Auckland University, supported in part by a Department of Mathematics Summer Scholarship, and was supervised by Prof. John Kalman.

A formula is satisfied by this matrix, and hence is a tautology of EC, if and only if it evaluates to a designated element under every assignment to its variables. This matrix is *normal* in the sense that if $x = 0$ and $e(x, y) = 0$, then $y = 0$, in other words, it is normal with respect to the inference rule *modus ponens* (cf. [**?**, page 41], [**?**, chapter 2]). The asterisk preceding the 0 in the above table indicates that 0 is designated and that the corresponding matrix is normal with respect to *modus ponens*.

An equivalent description would be to define EC syntactically as the set of all propositional formulas that are derivable by the rules of *substitution* and *modus ponens* from the following three axioms: (a) Epp, (b) $EEpqEqp$, (c) $EEpqEEqrEpr$. These two descriptions are equivalent in the sense that they describe the same set of EC-tautologies (cf. [**?**, page 158], [**?**, page 94]). Many alternate axiomatizations have been presented, and in particular axiomatizations involving only one axiom (a single axiom) have been the focus of much research. Many solutions to the problem of finding single axioms for EC have been derived using resolution-type theorem provers like Otter (cf. [**?**, **?**, **?**, **?**], [**?**, page 351]).

Peterson [**?**] noted Łukasiewicz's [**?**] observation that any single axiom for EC must have at least 11 symbols and must contain exactly two occurrences of each variable. There are 630 such candidates. Many possible single axioms were ruled out by Peterson's paper using matrix counter examples and various syntactic disproofs (cf. [**?**, page 446]). It has been shown by various people that 13 of these serve as single axioms (cf. [**?**, **?**, **?**, **?**, **?**]). To date, XCB (notation described in Appendix **??**) is the only formula whose status has yet to be decided. During the early 1980's, Wos and Winker studied XCB extensively as a continuation of their earlier work reported in [**?**]. More recently, Veroff has studied patterns of deduction for formulae deducible from XCB.

In [**?**] we see that for the purposes of automated deduction the two inference rules, *substitution* and *modus ponens*, can be combined into one, namely, condensed detachment CD. In the language of Otter [**?**] this can be implemented as an instance of the hyper-resolution rule (cf. [**?**, page 200]) with the clause

```
-P(x) | -P(e(x,y)) | P(y).
```

as its nucleus.

Another inference rule, whose validity with respect to EC was alluded to briefly by Meredith [**?**, page 186] and later by Kalman [**?**, page 199] and Peterson [**?**, page], is reversed *modus ponens*. This rule asserts that if q is a theorem and Epq is a theorem, then p must also be a theorem (i.e., it is the dual of *modus ponens*). To see that this is valid, note that in the matrix $\mathcal{E}$ if $y = 0$ and $e(x, y) = 0$, then $x = 0$, that is, this matrix is also normal with respect to reversed *modus ponens*. As in the case of condensed detachment, reversed *modus ponens* can be combined with *substitution* into one inference rule: reversed condensed detachment RCD. In the language of Otter this can be implemented as an instance of the hyper-resolution rule with the clause

```
-P(y) | -P(e(x,y)) | P(x).
```

as its nucleus.

Since the original syntactic description is complete (cf. [?, ?]) and the new inference rule is valid, the syntactic description could be strengthened by the addition of this new rule without changing the set of EC-tautologies. This new deductive system is logically equivalent to the original, that is, every formula that could be proved in this new system could be proved in the old. However, at the first level, twice as many formulae are potentially deducible between any two given formulae. Thus the inference structure of this new system is much more "dense" than that of the older system.

This article investigates the impact that this additional inference rule has on the possible axiomatizations of the system. In particular, we look at the potential for formulae to be shortest single axioms.

2 Initial Investigation

For this new system only one of each pair of dual formulae need be considered as a potential single axiom, since any proof that a particular formula is a single axiom suggests a proof for its dual.[1] Hence only 345 of the 630 length 11 formulae need be considered. The set of formulae considered for investigation was chosen by retaining the element of each dual pair whose bracket type is composed of letters lower in the alphabetic order (see Appendix ??). In particular, any single axiom or dual of a single axiom in the original description of EC must also be a single axiom in the new description of EC. Some of the matrix counterexamples and containments indicated by Peterson [?] can also be extended to this new system, and hence a number of formulae can be ruled out as potential axioms. A number of Peterson-style disproofs were suggested by Elliot Lawes [?]. These proceed by describing the set of formulae derivable from a given formula, then show that the set does not contain a particular EC-Tautology (cf. [?]). However, entirely satisfactory formulations became increasingly difficult to produce. The addition of an inference rule made even the simplest of such disproofs much more complex and introduced even more uncertainty where nonstandard forms of demodulation (cf. [?, page 345]) were employed.[2] Lawes also found a proof that YPG was a single axiom (see Appendix ??) and at the end of his investigation had ruled out all but 137 of the 345 formulae considered.

It was at this point that I became interested in this area of research. At the time I was studying potential methods of increasing the efficiency and scope of our research in automated reasoning. My work dealt with the following problem

[1] The dual of a formula has the arguments of each of its function symbols reversed in order, so if we take a proof that a particular formula is a single axiom and swap appropriate instances of CD or RCD, we get a proof for its dual.

[2] As the generalization of this methodology for EC has yet to be successfully achieved, with some problems coming to light after the publication of [?], I thought it wise to avoid this approach initially. Formulae for which this approach may be applicable have been noted in Section ??.

that is commonly faced in this field. Many applications call for performing a large number of OTTER runs with each run requiring only minor changes to the input file. Often we are aware of the whole sequence of runs at the outset, and so this procedure becomes tedious and time consuming, and also lends itself to error. In response to this I created a series of simple shell programs that automate this process and that can enable much more ambitious projects to be seriously considered (see [?]). Throughout the design and testing of the above system the axiomatization problem mentioned above was a major source of inspiration and application. For example, preliminary runs to decide the potential for each of the 345 formulae to be axioms could be done more easily, and more complex approaches could be attempted systematically, with this system. Over the past year these programs have been significantly improved both in terms of performance and scope of application.

This article makes extensive use of the automated reasoning assistants OTTER and MACE. A number of key input files are included which the reader may wish to experiment with. As noted in [?] both OTTER and MACE are in the public domain and are available by anonymous FTP and through the World Wide Web (WWW). See either of

```
ftp://info.mcs.anl.gov/pub/Otter/README
http://www.mcs.anl.gov/home/mccune/ar
```

for information on obtaining these programs.

3 Inference Database

Any length 11 formula that can be derived from a known nonaxiom clearly cannot be a single axiom. Lawes [?] used this method of syntactic disproof, with the aid of the automated reasoning system OTTER, to rule out a number of formulae. For example, a clause expressing a known nonaxiom, such as YOL (a single axiom for the left-Abelian group calculus), was placed on the `sos`, clauses expressing CD and RCD on the `usable` list, and the Skolemized negations of the 345 formulae on the `passive` list of an appropriate OTTER input file. Answer literals [?, page 42] were also used to facilitate identification of formulae as in the following typical file.

```
set(hyper_res).          set(order_history).
set(ancestor_subsume).   clear(print_given).
clear(print_kept).       clear(print_back_sub).
assign(max_given,250).   assign(max_seconds,1800).
assign(max_proofs,-1).   assign(stats_level,1).

list(usable).
 % Clauses expressing CD and RCD.
  -P(x) | -P(e(x,y)) | P(y).
  -P(y) | -P(e(x,y)) | P(x).
```

```
end_of_list.

list(sos).
 % Clause expressing non-axiom.
  P(e(e(x,y),e(e(e(z,y),x),z))) | $ANS(YOL).
end_of_list.

list(passive).
 % Clauses expressing
 % Skolemised negations of 345 EC formulae.
  -P(e(e(e(e(e(a,a),b),b),c),c)) | $ANS(AXA).
  -P(e(e(e(e(e(a,a),b),c),b),c)) | $ANS(AXB).
                   .
                   .
                   .
  -P(e(e(a,e(b,c)),e(e(c,a),b))) | $ANS(VN).
  -P(e(e(a,e(b,c)),e(e(c,b),a))) | $ANS(VO).
end_of_list.
```

This program completed after 30 minutes, deriving a number of length 11 formulae from YOL, some of which had not been previously eliminated. Clearly this method of syntactic disproof could easily be made systematic using the system of programs mentioned in Section ??. Indeed each of 34 known non-axioms were tested in this fashion (overnight), and a number of formulae were discarded.[3]

One of the obvious shortcomings of this approach was the limitations on the length of the OTTER runs. A particular formula may require more than 30 minutes to be derived from a known nonaxiom and may even require more sophisticated applications of OTTER. The success of the above approach suggested that more results might be obtained indirectly by considering all inferences which could be made between the length 11 formulae in 30 minutes runs.[4] Using the aforementioned system of programs, clauses expressing each of the 345 formulae (from a file already compiled by Lawes) were placed in turn on the `sos` of an appropriate OTTER file which terminates after 30 minutes. A database of 3060 inferences between the 345 length 11 formulae was produced over about 7 days on a SPARCstation 10/40. With the help of various UNIX features a map of the inferential relationships between the length 11 formulae was formed to assist in extending the disproofs. This was later used to identify interesting subsystems (see Section ??). Six new single axioms were found (namely, CXM, DXM, OYJ, UJ, VJ, and VN [see Appendix ?? for proofs]), and

[3]For this and subsequent runs the hot list strategy [?, page 48] was used to focus on the clause expressing the unknown formula, in the hope that more shorter clauses would be generated. This heuristic may not have been the best, since behavior under this strategy is (somewhat) unpredictable.

[4]If a nonaxiom A infers B, and B infers C, in 30-minute runs, then C is refuted.

a number of formulae were ruled out in this fashion; however, these disproofs were later superseded by a much more efficient approach.

4 Matrix Counterexamples

A more simple approach used by Peterson [**?**] was to find a normal logical matrix that satisfies a particular formula but does not satisfy all of the EC-tautologies. The normality of the matrix ensures that every formula derivable (by CD or RCD) from the given formula must also be satisfied. Hence, any formula that does not satisfy the matrix will serve as a counter example. In other words, at least one EC-tautology could not be derived, and so the original formula could not be a single axiom.

A formula may be tested against a particular matrix by a simple application of OTTER. For example, the following input file (cf. [**?**] [**?**]) is used to test YOL against the matrix III [**?**, pages 45-48]).

```
set(hyper_res).
assign(stats_level,1).

list(usable).
% Clauses expressing elements of the domain of III.
  P(0).  P(1).  P(2).
end_of_list.

list(sos).
% Clause expressing assignment to variables of YOL.
 -P(x) | -P(y) | -P(z) | Q(e(e(x,y),e(e(e(z,y),x),z))).
end_of_list.

list(demodulators).
% Table for the matrix III.
  e(0,0) = 0.  e(0,1) = 1.  e(0,2) = 2.
  e(1,0) = 2.  e(1,1) = 0.  e(1,2) = 1.
  e(2,0) = 1.  e(2,1) = 2.  e(2,2) = 0.
end_of_list.

list(passive).
% Clause asserting that YOL is evaluated
% to the designated element of III.
 Q(0).
end_of_list.
```

When OTTER is run with input from this file, clauses expressing each of the possible assignments to the variables of YOL are generated. After demodulation we see that any instance of YOL that is evaluated to the designated element of

III is subsequently subsumed by the clause on the `passive` list. By noting that no clauses are kept, we may deduce that the matrix III satisfies the formula YOL.

The following modification was later introduced as the domain size of the matrices increased. To show that a matrix does not satisfy a formula, it is sufficient to find one assignment that does not reduce the formula to the designated element; all subsequent assignments are redundant. Clearly this testing procedure would be more efficient if this redundancy was removed. By setting the parameter `max_proofs` to 1 (the default) and appending sufficient clauses of the form

```
-Q(1). , -Q(2). , -Q(3). , ...
```

to the `passive` list, OTTER will terminate (by forming a proof) as soon as a nondesignated instance of `P(x)` is generated. Hence, testing matrices against the large set of formulae could be performed more quickly.

This process can easily be automated using the system of programs mentioned in Section **??**. Indeed, each of the finite matrices described in [**?**] which were normal with respect to CD and RCD (namely II, III, VII, IX, X, XI, XIII and XIV) were tested against each of the 345 formulae under consideration. Since all but two of the matrices given by Peterson, namely VII and XII, were accompanied by their duals, the duals of these matrices were also tested.[5] Due to the success of this approach, a large number of matrices suggested by Brian Van Dam were also tested.[6] Most did not satisfy *any* length 11 formulae; however, the matrices BI, BVI, and BX were shown to satisfy 15, 101, and 20 formulae, respectively. Since not 345 length 11 formulae were satisfied, these matrices clearly formed counterexamples for those formulae. Note that this approach was both more efficient and effective: all the previous syntactic disproofs were superseded in a short space of time,[7] leaving only 66 formulae unclassified.

In [**?**] an infinite logical matrix $\langle G, \circ, \langle 5, 1\rangle\rangle$ was shown to be normal with respect to CD and not to satisfy the EC-tautology UJ. This matrix is normal with respect to RCD since the corresponding groupoid $\langle G, \circ\rangle$ satisfies the law $xy = yx$ [**?**, Theorem 1]. Any formula satisfied by this matrix cannot derive UJ, and hence cannot be a single axiom for the new system. Thus all the formulae shown in [**?**] to be satisfied by this matrix could also be ruled out as potential single axioms.[8] This left 44 formulae unclassified.

5 Lattice of Deductive Subsystems

In Section **??** a map of the inferential relationships between the 345 length 11 formulae was described. This map suggested a quasi-ordering "infers" on the set of EC-tautologies.[9] When restricted to the set of length 11 formulae,

[5] The dual of VII was the only matrix that produced any new results.
[6] At the time Van Dam was studying quasi-groups using ANL-DP [**?**].
[7] The matrix III only took 140 seconds to test on a SPARCstation 10/40.
[8] This matrix was later superseded by the matrix found in Section **??**.
[9] This relation is clearly neither symmetric nor antisymmetric.

this could be thought of as a "level surface" through the inferential order, and hence potentially suggestive of the syntactic description's underlying structure. During this mapping process it became apparent that there were a number of distinct deductive subsystems present. These form a lattice in the following natural fashion.[10]

Suppose $\mathcal{B}$ is the class of all subsets of the set of EC-tautologies. Then we may define a lattice $\langle \mathcal{B}, \wedge, \vee \rangle$ by setting

$$S \vee T = \bigcap \{R \mid S \cup T \subseteq R \text{ and } R \text{ is a deductive system}\}$$

$$\text{and } S \wedge T = S \cap T \text{ for all subsets } S \text{ and } T.$$

That this is a lattice follows from the observation that the intersection of any set of deductive systems is also a deductive system. Now suppose E is the set of 345 formulae under consideration, and $\succeq$ is the quasi-order 'infers' on E. If $x, y \in E$, then we may define $x \sim y$ to mean that $x \succeq y$ and $y \succeq x$, $x/{\sim} = \{y \in E \mid y \sim x\}$, and $E/{\sim} = \{x/{\sim} \mid x \in E\}$. Now $\sim$ is an equivalence relation on E, and $\succeq$ determines a partial order $\geq$ on the quotient set by $(x/{\sim}) \geq (y/{\sim})$ if and only if $x \succeq y$, that is, the deductive system generated by y is a subset of the deductive system generated by x.

Comparison with the matrix counterexamples of Section **??** showed that many of the suggested subsystems were modelled by particular matrices, that is, all of the formulae identified as belonging to a particular subsystem were satisfied by that matrix. This suggested that the whole deductive subsystem may be modelled by that matrix and hence suggests a potential semantic description. The following table shows sets of length 11 formulae that appear to be single generators for the deductive subsystem modelled by the given logical matrix. Note that each of the formulae in a particular set can be inferred from one another (hence is a single generator) and have been shown to be strong enough to infer most of the length 11 formulae satisfied by that matrix.

Matrix	Potential Single Generators
II	QYF YQM
III	CXL OYK QYM TO XCK YOL YQH
IX	YPE UK VK
X	UL VL
XIV	TM YOJ

Some of these subsystems were investigated in much greater detail. In Appendix **??**, it is suggested that the subsystem modelled by the matrix II contains a sequence of four deductive subsystems, each apparently contained in the previous, and ending in a particular finite system. This finite system consists of the formulae TJ, QA, QYA, QYD, YA, YQA, YQB, and can be shown (using

[10]The systems are deductively closed under both CD and RCD.

Otter) to be deductively closed.[11] In addition, this system is seen to be contained in each of the subsystems modelled by the matrices II, III, VII, XI, VIIt, XIIt, BI, BVI, and BX.

One motivation for this extensive diversion was the results of a much earlier experiment. This more naive approach was to place 20 clauses expressing known nonaxioms on the `sos` of the Otter input file described in Section **??**. Over 120 thousand clauses were generated and 578 proofs were found, but all these proofs involved more than one nonaxiom and so no syntactic disproofs were found. However, 11 of these proofs gave the following axiom sets for the strengthened description of EC.

Axiom Set								Single Axiom
II	III	BVI	IX	X	XI	XIII	XIV	Inferred
			UK				YOJ	TN
	YOL	YPK	PYL				TM	PYO
	YOL	YPK	UK					YPG
	XCK				FXJ		TM	DXM
YQM			BXM XDC				TM	UJ
				OYM			YOJ TM	VJ
				OYM			TM	VN
			UK				TM	OYJ
			BXM XDC				YOJ	OYJ
			BXM	YOH			TM	CXM
		YPK	BXM XDC			YPH		CXM

Note that each member of an axiom set satisfies only one of the finite matrices described in Section **??**, that is, they are in some sense representative of the subsystem modelled by that matrix. Also, nearly all of these axiom sets contain a suggested single generator for the subsystem corresponding to XIV. This may indicate an important relationship between axiomatizations and this particular subsystem.

6 New Counterexamples

At this point it became apparent that new matrix counterexamples were needed to eliminate the remaining formulae. Peterson's report gave no indication as to how the matrices given there were derived, so initially there was no clear methodology to pursue.[12] To resolve this dilemma generation of matrices by

[11] A potential semantic description is also given at end of Section **??**.

[12] These matrices were originally devised by John Kalman, but no record was kept of their derivation. An alternate description was later found in terms of boolean algebras.

hand was attempted in the hope that it would provide the needed intuition. This resulted in the "perturbation" technique described in Section **??** In addition, examination of literature on related axiomatization problems led to the techniques of Sections **??** and **??**. Ultimately the application of MACE to the problem (Section **??**) proved to be most appropriate approach. Note that the level of success of these techniques increases with the degree of application of automated reasoning.

6.1 Perturbations

Many of the formulae whose status was still unknown were of the form ExA or EAx, where A is a well-formed formulae involving only the function symbol E. A possible matrix counterexample might satisfy ExA but not EAx. If we just consider assignment of designated elements to the variables, then such such a matrix might not be commutative with respect to its designated elements.[13] This is only possible if the matrix has more than one distinguished element.

Thus the addition of distinguished elements could potentially enhance a search for counterexamples. Doing so in an *ad hoc* fashion would be impractical, since the scope of potential models is very large; however, a survey of the smallest possible cases did not appear unreasonable, since the normality condition places a strong restriction on the possible tables defining the given operation. Consider the following schema

		*	*	
	e	0	1	2
*	0	B	B	2
*	1	B	B	2
	2	2	2	A

where 0 and 1 are the only designated elements.[14] It is clear that the positions marked by A and B are the only ones that may be assigned a designated element, and if position A is not, then no EC-tautology can be satisfied by the corresponding matrix. If the positions marked with B are all assigned a designated element, then in the corresponding logical matrix 0 and 1 are congruent, and so this matrix is homomorphic to $\mathcal{E}$ in Section **??**. Similarly if these positions are all assigned a nondesignated element, then the corresponding matrix is homomorphic to a matrix similar to $\mathcal{E}$ but with the operation e defined by the table

	e	0	1
*	0	1	1
	1	1	0

[13] That is, if d_1 and d_2 are designated, then Ed_1p being assigned a designated element might not necessarily imply the same for Epd_2.

[14] The asterisks above the designated elements in the table indicate normality of the matrix with respect to RCD.

Also, given a particular choice of table constrained as above, it can be shown that there is an alternate choice of table (with A assigned the alternate designated element), where the corresponding logical matrices are isomorphic. ¿From the above constraints we see that only 64 possibilities need to be investigated. Each of these was tested, and the following theory appears to describe most of the results.

If we take a particular logical matrix such as $\mathcal{E}$ and make some "small change" or "perturbation" in its structure, then the formulae that are satisfied by this matrix may change. If a perturbation could be achieved such that the normality properties remain unchanged, but the set of formulae satisfied was a subset of the original, then this would suggest a method of finding new matrix counterexamples. Indeed the matrix counterexample described by Diamond and McKinsey [?] and later used by Peterson [?] could be thought of as a subtle change in the equational characteristics of a particular model of EC. For finite models this degree of subtlety is much more difficult to achieve.

Start with the logical matrix $\mathcal{E}_1 = \langle\{0,1,2\}; e; \{0,1\}\rangle$ with the operation e defined by the table

$$\begin{array}{cc|ccc} & & * & * & \\ & e & 0 & 1 & 2 \\ \hline * & 0 & 0 & 0 & 2 \\ * & 1 & 0 & 0 & 2 \\ & 2 & 2 & 2 & 0 \end{array}$$

and which is clearly logically equivalent to $\mathcal{E}$. This matrix could be perturbed by the following alteration to the table.

$$\begin{array}{cc|ccc} & & * & * & \\ & e & 0 & 1 & 2 \\ \hline * & 0 & 0 & 2 & 2 \\ * & 1 & 0 & 0 & 2 \\ & 2 & 2 & 2 & 0 \end{array}$$

Denote this perturbed matrix by $\mathcal{E}_2$. Now assignments that induce the evaluation $e(0,1) = 2$ in a particular formula may no longer yield a designated element. In this case, a formula that was satisfied by $\mathcal{E}_1$ will not be satisfied by $\mathcal{E}_2$. However, most assignments are not affected by this perturbation, and so the effect on number of formulae satisfied should be moderate. Indeed, the number of length 11 formulae satisfied could be used as an approximate measure of the effect of such a perturbation. For example, the above matrix satisfies only 12 of the 345 length 11 formulae considered, and so the perturbation is reasonably strong.

Take the above matrix and replace a zero entry as follows:

$$\begin{array}{cc|ccc} & & * & * & \\ & e & 0 & 1 & 2 \\ \hline * & 0 & 0 & 2 & 2 \\ * & 1 & 1 & 0 & 2 \\ & 2 & 2 & 2 & 0 \end{array}$$

This amplifies the effect of the first perturbation: it increase the likelihood of the evaluation $e(0,1) = 2$ occurring.[15] In this case the number of length 11 formulae satisfied is 14, and so this secondary perturbation has produced a weaker effect; however, the converse is more common.

Obviously a number of alternate perturbations are also possible. If we replace each of the entries x, y and z in the following schema by one of $\{0,1,2\}$ we span all of the possibilities.[16]

		*	*	
	e	0	1	2
*	0	0	x	2
*	1	z	y	2
	2	2	2	0

Hence each perturbation can be represented by the triple (x, y, z).[17] The effect of these possible perturbations is as follows.

x	y	z	Formulae Satisfied
2	0	0	12
2	0	1	14
2	1	0	1
2	1	1	1
0	2	0	26
0	2	1	2
1	2	0	2
1	2	1	0
2	0	2	23
2	1	2	6

x	y	z	Formulae Satisfied
0	0	2	21
1	0	2	20
0	1	2	1
1	1	2	1
2	2	2	6
0	2	2	2
1	2	2	0
0	2	2	0
2	2	0	2
2	2	1	0

Ideally a perturbation has a very small effect. The above effects, however, are very extreme. What we need is a perturbation that is controllably weaker than the ones we have been considering. That this might be possible is evident from the following example. Consider the logical matrices $\mathcal{E}_3$ and $\mathcal{E}_4$ corresponding to the triples (2,1,2) and (2,2,2) in the above table. Both of these matrices satisfy the same set of EC-tautologies of length 11 and 7; however, $\mathcal{E}_3$ satisfies *Epp*, whereas $\mathcal{E}_4$ does not.[18] So with respect to satisfaction, these two matrices are very "close".

Many other aspects of this idea have been investigated; however, experimentation is still incomplete. The following is a survey of current exploration.

[15] Since the evaluation of an arbitrary formula calls for the evaluation of each of its subformulae, in $\mathcal{E}_1$, the evaluation $e(0,1) = 2$ can only occur at the innermost level. This is not so for $\mathcal{E}_2$.

[16] The aforementioned experiment included the varying of $e(0,0)$; however, this perturbation either satisfied no length 11 EC-tautologies or was isomorphic to a perturbation which did not vary $e(0,0)$.

[17] Note that if x, y and z are all designated the resulting matrix will be logically equivalent to the $\mathcal{E}_1$. Such perturbations are trivial and so will not be considered.

[18] Both matrices satisfy TJ, TL, UN, UO, VM, VO, QD, and QG.

1 Increasing the domain size, that is, considering matrices of the form $\langle\{0,1,2,3\};e;\{0,1\}\rangle$. This yielded some interesting data including the following table

		*	*		
	e	0	1	2	3
*	0	0	2	3	2
*	1	0	0	3	2
	2	3	2	0	2
	3	3	2	2	0

which corresponds to a matrix that appears to model the finite subsystem described in Section **??**.

2 Increasing the number of distinguished elements to 3, 4, and 5 produced many different tables one of which

		*	*	*	
	e	0	1	2	3
*	0	0	1	0	3
*	1	0	2	0	3
*	2	0	3	0	3
	3	3	3	3	0

corresponds to a matrix that satisfies XBA - a formula whose status was previously unknown - but does not satisfy the single axiom UJ.[19] Hence XBA was ruled out as a potential single axiom.

3 Moving from one model to another, refining the set formulae satisfied. In particular, the formulae that could be derived from a specific unknown formula (e.g., XCB).

6.2 Matrices Derived from Rings

The following class of logical matrices was suggested by a discussion with John Kalman concerning the matrices presented by Peterson [**?**]. Let $(n, m; k)$ denote the logical matrix $\langle\{0,1,2,...,k-1\};e;\{0\}\rangle$, where the function e is defined by $e(x,y) = nx + my \pmod k$ for positive integers $n, m < k$ (cf. [**?**]). For such a model to be normal with respect to CD and RCD, n and m must both be relatively prime to k. To see this note that

$$e(0,y) = n(0) + m(y) = m(y) = 0 \pmod k$$

has a unique solution if and only if m and k are relatively prime. Note also that if $n + m = 1 \pmod k$,

$$e(x,x) = nx + mx = (n+m)x = x \pmod k,$$

[19]This matrix was later superseded by the matrix MX (see Section **??**).

so the corresponding matrix is idempotent and hence will satisfy no formulae.[20]

To find a matrix counterexample in this class, we need to choose values for n, m and k such that the corresponding matrix satisfies a particular formula but not all EC-tautologies. For example, if each function symbol in the formula YOL is interpreted as the operation e, and terms are collected, then we get the expression

$$(n^2 + nm^2)x + (n^2m^2 + nm)y + (n^3m + m^2)z.$$

For this to be satisfied by the matrix, every assignment to the variables x, y, and z must evaluate to the designated element 0. Hence, finding a matrix of this form that satisfies YOL is equivalent to solving the following set of equations simultaneously.

$$\begin{aligned} n^2 + nm^2 &= 0 \quad \pmod{k} \\ n^2m^2 + nm &= 0 \quad \pmod{k} \\ n^3m + m^2 &= 0 \quad \pmod{k} \end{aligned}$$

The parameters of the matrix (1,1;2) must satisfy these equations since YOL is an EC-tautology and (1,1;2) corresponds to the defining matrix $\mathcal{E}$ in the introduction. Another solution corresponds to the matrix (2,1;3). This is the matrix III from [**?**], which has already been shown to satisfy YOL.[21]

If k is prime, then the algebra underlying the set of equations is a field and hence solutions are more easily found. For example, the equations for YOL reduce to

$$\begin{aligned} n + m^2 &= 0 \quad \pmod{k} \\ nm + 1 &= 0 \quad \pmod{k} \\ n^3 + m &= 0 \quad \pmod{k}. \end{aligned}$$

Otherwise the task is far from obvious and even the existence of solutions is uncertain.

Since the matrices that we are interested in are normal, the set of formulae derivable from a given formula can provide additional information – the corresponding sets of equations must be consistent. For example,

$$n + m = 0 \pmod{k}.$$

is the constraint on matrices $(n, m; k)$ that satisfy *Epp*. Since XCB can derive *Epp*, this constraint must also apply to such matrices satisfying XCB. Further constraints may be applied to show that (1,1;2) is the only matrix of this form that will satisfy XCB (see Appendix **??**). Note that shorter formulae will give more simple constraints and hence will be more useful in determining the solution to the set of equations.

[20] Note that 0 is the only designated element.

[21] This is also the defining matrix for left-Abelian group calculus, of which YOL is a single axiom.

To apply this approach to the remaining formula under consideration, each formula had to be reduced to a set of equations.[22] This evaluation was performed using the symbolic computation package Maple [?]. Some smaller formulae derivable from those under consideration were determined using an appropriate application of OTTER. Appendix ?? contains a table of these sets of equations accompanied by the equations corresponding to the smaller derivable formulae.

Under the assumption that the underlying algebra is a field, each of the sets were shown to have only the trivial solution which corresponds to the matrix (1,1,2). Hence only values of k that are not prime need to be considered. The numeric computation package Matlab [?] was used to test appropriate parameters for $(n, m; k)$ up to k less than 350; however, no new results were found.[23] This suggested that these sets had only the trivial solution. Indeed, proofs of this for all but 11 cases are provided in Appendix ??.

Hence by a minimal use of automated reasoning this approach has been all but discarded.[24]

6.3 A New Use of MACE

MACE is a program that uses the Davis-Putnam procedure to search for models, (of a given domain size), satisfying a set of first-order propositional formulae. Unlike OTTER this program allows for little user intervention except in choice of clauses and the order in which those clauses are represented. In November a new version of MACE was released. The documentation [?] that accompanied it showed an encouraging amount of success with a number of different types of problems.

The successful applications of MACE suggested in [?] concentrated largely on equational characterisations. The axiomatization problem above was not in this form; however, it can be successfully rewritten as follows. The formula Epp would usually be expressed as the following P-formulation:

```
P(e(x,x)).
```

However a clause expressing this formula being evaluated to a designated element 0, more appropriate for use with MACE, would be

```
e(x,x)=0.
```

Similarly the clauses

```
e(0,y)!=0 | y=0.
```

and

```
e(x,0)!=0 | x=0.
```

[22]There were 23 formulae remaining at the time.

[23]Since some of the equations involved high powers, large values for k would not be unexpected.

[24]I have yet to rule out this method for the formulae EXE, FXK, XAK, XBC, XDB, XCL, and XDE. At time of writing I am aware of no matrix counterexamples for these formulae.

could be used to express normality with respect to CD and RCD. Preliminary investigation showed that this formulation requires significantly less time and memory – perhaps arising from fewer isomorphic models being considered.

To search for a matrix satisfying a given formula, but not all EC-tautologies, the following MACE input file was used.

```
list(usable).
 e(0,y)!=0 | y=0. e(x,0)!=0 | x=0.% Assert normality

% Assert that the formula EXE should be satisfied.
 e(e(e(x,e(e(y,x),z)),y),z)=0.

% Assert that the single axiom TN should be refuted.
 e(e(e(a,b),c),e(e(c,a),b))!=0.
end_of_list.

list(passive).
% Assign constants to specific values of the model.
 assign(a, 1).  assign(b, 2).  assign(c, 3).
end_of_list.
```

With this input file MACE searches for normal models (of a particular domain size) which satisfy EXN but not the single axiom TN. The assignment of values to the constants in the refutation allows MACE to terminate more quickly by ruling out a large class of isomorphisms. Another important restriction that was added later is the clause

```
e(x,x)!=x | x=0.
```

This eliminates an important class of unsatisfiable groupoids. That is, if $x^2 = x$ for any nondesignated element x, then uniform assignment to that element will not evaluate to the designated element.

Using MACE with no further strategy, each of the remaining 43 unknown formulae were used to search for models. This produced nine new models and eliminating 20 formulae in the course of a few hours. However, these searches became too restrictive in terms of time and memory. This is explained in [?] by the following observation:

> The space it [MACE] uses is proportional to n^{c+v} where n is the domain size, c is the length of the longest clause and v is the number of variables in the longest clause.

As the clause asserting the formula is long and involves a number of variables, searches are restricted to domain size 4 or 5. In addition, the clause refuting the single axiom is just as long, further compounding the problem

One solution to this was to restrict the search by refuting some shorter EC-tautology. Refutations of each of the shortest EC-tautologies in two or fewer variables were used to restrict the model search for the remaining formulae.[25] One model was found, however, this coincided with BVI - a correction for the formula FXL. Unfortunately, the savings in time and memory were not sufficient to yield any further models.

Another modification was to add the assertion of derivable formulae to the search. For example, since XCB can derive the formula Exx, the clause

$$e(x, x) = 0.$$

can be added to the input file. When mace is run with this input, it completes in 20% less time and requires only half the memory of similar runs. However, only a few formulae had suitably short derivable formula so the impact of this strategy was not great.

A more successful approach was to reduce the size of the refutation of TN. If a matrix $\langle G; e; \{0\}\rangle$, where $0 \in G$, does not satisfy the formula $EEpqr$, then there exists elements a, b, and c in G such that $e(e(a, b), c) \neq 0$. Hence there exists an element k of G such that $e(k, c) \neq 0$, namely, $e(a, b) = k$. So, asserting the formula expressed by the ground negative clause

```
e(e(a,b),c)!=0.
```

is equivalent to asserting the set of clauses

```
e(k,c)!=0. e(a,b)=k.
```

This set has a much shorter effective size.[26] Hence the clause refuting the single axiom TN

```
e(e(e(a,b),c),e(e(c,a),b))!=0.,
```

was split into smaller fragments, by adding constants as above. This produced the following set of clauses:

```
e(k1,k2)!=0. e(k3,c)=k1. e(k3,b)=k2. e(a,b)=k3. e(c,a)=k4.
```

The above the input file was modified to accommodate this new strategy and then applied to the remaining formulae. This resulted in the discovery of the model MX, which is a counterexample for 165 EC-tautologies including the previously unclassified formulae TD and VA. Comparison with the previous strategy showed this approach completing in half the time and requiring 20% less memory.

The success of this approach suggested the extension of this idea to positive clauses. If we interpret the outermost function symbol of an EC-tautology as material equivalence, we get a set of reduced formulae whose satisfaction is antecedent on the satisfaction of the original formula. For example, the clause

```
e(A,B)=0.
```

[25] The aforementioned driver program was used to execute this task systematically.

[26] In terms of the Davis-Putnam procedure I believe this is because 1-tuples of 4 constants are considered, rather than 3-tuples of 3 constants, when instances are being generated.

can be reduced to the set

```
A!:=0 | B=0.  B!:=0 | A=0.
```

If a search based on the reduced formula terminates with a suitably small set of models, then each of these models may be tested in turn using OTTER. However, if a search based on the reduced formulae terminates with no models, then the can be no models, satisfying the full formula at that domain size.

Application of this technique had a large computational effect, enabling domain sizes for a number of formulae to be ruled out. This also allowed reduced assertions (in the sense of the previous paragraph) of larger derivable formulae to be added to the search. At this point only five formulae remained unclassified. The above approach was used to show that XCB and AXL require models with domain size at least 7, BXO and XBC of at least 6, and XDB of at least 5.

6.4 TS-loops

The model G presented by Peterson [?] was derived from the infinite model given in [?]. Peterson used the following properties of G to show that various formulae were satisfied, and that $(xy)z = x(yz)$ (associativity) is not satisfied.

$$
\begin{array}{rl}
\text{(i)} & xy = yx \\
\text{(ii)} & xx = 0. \\
\text{(iii)} & x0 = x. \\
\text{(iv)} & x(xy) = y.
\end{array}
$$

These properties, however, coincide with the defining properties of TS-loops (cf. [?]), which exist on sets of size $6n + 2$ or $6n + 4$ elements, $n \geq 0$. This suggested the existence of a finite model that could be used to supersede the results derived using G.

The task of searching for such a model is an elementary application of MACE. The following MACE input file was used to search for a model satisfying the above laws but not $(xy)z = x(yz)$.

```
list(usable).
 e(0,y)!=0 | y=0.  e(x,0)!=0 | x=0. % Assert normality
 e(x,x)!=x | x=0. % Ensure at least one formula is satisfied

% Refute associativity (UJ) piecewise
 e(k1,c)!=k3. k3=e(a,k2). k1=e(a,b). k2=e(b,c).

% Assert properties of TS-loop.
 e(x,0)=x.  e(0,x)=x.
 e(e(x,y),x)=y.%OD
 e(x,y)=e(y,x).%QG
 e(x,x)=0.%YA
```

```
end_of_list.

list(passive).
 assign(a,1).  assign(b,2).  assign(c,3).
end_of_list.
```

Note that the reduced refutation of associativity was used, enabling much larger domain sizes to be considered. With this input file MACE produced the following normal model in 0.62 seconds of CPU time on a two-processor Axil 320.

```
  | 0 1 2 3 4 5 6 7 8 9
--+--------------------
0 | 0 1 2 3 4 5 6 7 8 9
1 | 1 0 4 5 2 3 7 6 9 8
2 | 2 4 0 6 1 8 3 9 5 7
3 | 3 5 6 0 9 1 2 8 7 4
4 | 4 2 1 9 0 7 8 5 6 3
5 | 5 3 8 1 7 0 9 4 2 6
6 | 6 7 3 2 8 9 0 1 4 5
7 | 7 6 9 8 5 4 1 0 3 2
8 | 8 9 5 7 6 2 4 3 0 1
9 | 9 8 7 4 3 6 5 2 1 0
```

This corresponds to the unique TS-loop on 10 elements. The above model was tested and shown to satisfy 237 of the 345 formulae under consideration. This eliminated a number of formulae that Peterson had ruled out using the model I [?] ,which was not normal with respect to RCD, and superseded many earlier matrices. This left 14 formulae unclassified.

Of the remaining formulae it may be seen that XCB, XCA, XCL, CXH, and CXI can be satisfied by no TS-loop which does not satisfy the single axiom CXM, and XDB and XDE can be satisfied by no TS-loop which does not satisfy the single-axiom XGK. This arises by considering the application of the law $xy = yx$ to each of the formulae, for example,

$$CXH = (((xy)(yz))z)x = (((yz)(xy))z)x = (((yz)(yx))z)x = CXM.$$

Applying this same consideration to the other formulae showed that any TS-loop satisfying one of XAK, XBC, FXK, BXO, AXL, XFE, or EXE, must satisfy them all.

This suggests investigation for TS-loops for the above equivalent forms, and possibly weakening TS-loop properties to find models for the rest.

7 Formulae with Kernels under Inference

One approach applied by Peterson [?] and generalized by Wos et al. [?] relies on characterizing the set of all formulae derivable from a given length 11 EC-tautology. Suppose each element of this set contains some instance of a sub-formula, called a kernel (cf. [**?**, page 329]). Then that EC-tautology is said to have a kernel under inference. If it can be shown that an EC-tautology exists that does not contain that kernel, then the given formula clearly cannot be a single axiom. An example of such kernel is the length 11 formula itself, since the EC-tautology *Epp* is not an instance of any length 11 formula and hence could not be derived.

The following OTTER input file, suggested by John Kalman, examines the possibility of the formulae XCA containing itself as a kernel under inference. The methodology relies heavily on the *ratio* strategy (see [**?**, page 361]).

```
set(hyper_res).                 set(back_demod).
set(print_lists_at_end).        assign(max_given,32).
assign(stats_level,0).          assign(pick_given_ratio,30).

list(usable).
   -P(x*y) | -P(x) | P(y).          % CD
   -P(x*y) | -P(y) | P(x).          % RCD
   -Z | (x* ((x*y)*y*z)*z=XCA).
end_of_list.

weight_list(pick_given).
    weight(Z,1000).
end_of_list.

list(sos).
   P(x* ((x*y)*y*z)*z).   % Clause asserting formula XCA.
   Z.
end_of_list.

list(demodulators).
   x*XCA=XCA.
   XCA*x=XCA.
end_of_list.
```

When OTTER is run with this file, formulae are generated under the *best-first* strategy. In other words, clauses are chosen for inference by weight (the default, see [**?**, page 553]). The formula `Z` does not become the focal clause, since it has been given a very large weight. After 30 clauses have been chosen and inferences made, the strategy changes to *breadth-first*; clauses are chosen for inference by their position on the *set of support* (`sos` list) (see [**?**, page 212]). Since the clause `Z` is first on the `sos`, it becomes focal and clashes with the clause

```
-Z | (x* ((x*y)*y*z)*z=XCA).
```

Thus the demodulator `(x* ((x*y)*y*z)*z=XCA)` is inferred. Each clause in the clause space is then back demodulated with this new demodulator, and any clause in the space which contains an instance of the formula XCA will have that instance replaced by the constant 'XCA'. The two clauses on the demodulator list will then reduce any clause containing the constant 'XCA' to `P(XCA)`. The *best-first* strategy is then resumed.

The program terminates after 32 clauses have become focal, and the clause space is output. If all the clauses derivable from the clause asserting XCA, at this point contained an instance of the formula XCA then they would all be reduced to, and then subsumed by, the clause `P(XCA)`. Hence no further inferences will be made. Indeed, examination of the lists printed at the end of the output file shows that this does occur, suggesting that XCA contains itself as a kernel under inference.

The results of this and similar OTTER runs can be used to suggest kernels or prove that particular kernels do not exist (cf. [**?**]).

This methodology was applied to the 26 formulae unclassified at the time and CXH, CXI, EXE, FXK, XAK, XCA, XCL, XDE, XFE were seen to contain themselves as kernels. Rigorous proofs that these kernels do exist are probably possible; however, the additional inference rule makes them more complicated. I have not attempted this but have instead concentrated on searching for small finite models.

8 Conclusion

Using various applications of automated reasoning a nearly complete classification of the 630 length 11 EC-tautologies, with respect to single axiomatization under CD and RCD, has been achieved. Since the dual of any single axiom must also axiomatize the system, it is sufficient to classify each of the 345 formulae described in Appendix **??**. This classification is as follows:

- The formulae GXL, GXN, QYH, QYJ, QYK, PYM, PYO, TN, UM, XGF, XGK, YQF, YQJ, and YQL are single axioms by considering the duals of the 13 single axioms known from previous investigations.
- Lawes [**?**] showed that YPG is a single axiom, and this report shows that CXM, DXM, OYJ, UJ, VJ, VN are also single axioms.
- The formulae CXH, CXI, EXE, FXK, XAK, XCA, XCL, XDE, and XFE have been tentatively rejected due the apparent presence of a kernel under inference (see Section **??**).
- Matrix counterexamples have been found for all the remaining formulae except XCB, BXO, AXL, XBC, and XDB.

Since the matrices given in this report are normal with respect to both CD and RCD, they also apply to the more usual description of EC with CD as its only inference rule. Hence the counterexamples corresponding to the matrices given here supersede many of the results given by Peterson [**?**]. Further investigation may consider the relationship between these two systems, for instance, finding matrices to show that YPG, VJ, and VN are special cases.

Automated reasoning may still play a large role in completing this classification. The difficulty in eliminating the remaining formulae is perhaps indicative of their potential to be single axioms. More sophisticated uses of OTTER or similar programs may yield the appropriate proofs. Of the formulae eliminated due the apparent presence of a kernel the question of rigorous proof remains open. An extension of the techniques described in [**?**] may resolve this. Alternately it may be possible to find small counterexamples using MACE.

The process of applying automated reasoning to this problem in classical propositional logic has required the development and exploration of many different techniques. These were evolved over a period of time as my understanding of EC and past research increased. Some were ultimately superseded by better approaches but still yielded interesting insights into the calculus and methods of approaching it. In particular the presence of interesting substructures was noted, and conjectures were made concerning their axiomatization.

Perhaps the most significant aspect to note is that I knew very little of the calculus to begin with, but was still able to make significant progress with the aid of OTTER and MACE. Thus, creative use of automated reasoning assistants has enabled difficult problems in equivalential calculus to be solved and has generated conjectures that would probably not have been formed otherwise.

Appendix 1: Classification of Length 11 EC-tautologies

*The following is a modified description of the characterization devised by John Kalman and presented in Peterson's unpublished paper [**?**].*

Each of the 630 length 11 EC-tautologies can be described using a two- or three-letter code. The last letter denotes the order in which the variables occur, and the preceeding letter(s) denote the the arrangement of the connectives (which we will call the bracket type).

Bracket type: Since other important calculi are contained in EC (e.g., left group and right group calculi), we use C to denote the connective and replace with E (L or R) as appropriate. Every formula of length greater than one is then of the form $C\alpha\beta$, where α and β are formulas known as the antecedent and the consequent, respectively. If the length of both the antecedent and the consequent is five, then there are only four possible bracket types:

Name	formula	Dual
T	CCCpqrCCstu	W
U	CCCpqrCsCtu	U
V	CCpCqrCCstu	V
W	CCpCqrCsCtu	T

If the antecedent and consequent have different lengths, then they are each of length 1, 3, 7, or 9. The only possible bracket types of these lengths are

Length	Name	formula	Dual
9	A	CCCCpqrst	N
	B	CCCpCqrst	M
	C	CCCpqCrst	L
	D	CCCpqrCst	I
	E	CCpCCqrst	K
	F	CCpCqCrst	J
	G	CCpCqrCst	H
	H	CCpqCCrst	G
	I	CCpqCrCst	D
	J	CpCCCqrst	F
	K	CpCCqCrst	E
	L	CpCCqrCst	C
	M	CpCqCCrst	B
	N	CpCqCrCst	A
7	O	CCCpqrs	S
	P	CCpCqrs	R
	Q	CCpqCrs	Q
	R	CpCCqrs	P
	S	CpCqCrs	O
3	Y	Cpq	Y
1	X	p	X

The bracket type of a formula whose antecedent and consequent have different lengths is the bracket type of the antecedent followed by that of the consequent.

Variable order: Each formula has three variables each occurring twice. If the first variable to occur is p, the second q, and the third r, then all possible variable orders are

Name	Order	Dual
A	ppqqrr	A
B	ppqrqr	D
C	ppqrrq	G
D	pqpqrr	B
E	pqprqr	E
F	pqprrq	H
G	pqqprr	C
H	pqqrpr	F
I	pqqrrp	I
J	pqrpqr	J
K	pqrprq	L
L	pqrqpr	K
M	pqrqrp	M
N	pqrrpq	N
O	pqrrqp	O

Duality: The dual of a formula has the order of the arguments of each of its connectives reversed. If the formula is represented graphically as a binary tree, the dual of that formula is essentially its image under reflection with variables renamed to preserve the alphabetic order.

The last column of the tables above can be used to find the dual more quickly. To get the dual bracket type, simply replace the letter(s) describing the original bracket type by the dual(s) indicated, and if there is more than one, reverse their order. The dual variable order is indicated in the last table.

Hence the above notation describes the set of 630 length-11 EC-tautologies.
Examples

- The formula TE

has bracket type T	$CCCpqrCCstu$,
and variable order E	$pqp \quad rqr$,
so the formula is	$CCCpqpCCrqr$.

¿From the tables the dual bracket order of T is W, and the variable order E is self dual, so the dual of TE is WE.

- The formula PYH

has bracket type PY	$CCpCqrs \quad Cpq$,
and variable order H	$p \; qqr \quad pr$,
so the formula is	$CCCpCqqrCpr$.

¿From the tables the dual of P is R, and Y is self dual, so the dual bracket order is YR. The dual of the variable order H is F and gives YRF as the dual of PYH.

Clearly any shorter formula can be described using a similar two letter code involving the bracket types O, P, Q, R, S, and Y, and variable orders A, D, and G.

Formulae considered: Since a proof that a formula is a single axiom yields a proof for its dual, this report considers only one of each dual pair. The set of formulae considered for investigation was chosen by retaining the element of each dual pair whose bracket type is composed of letters lower in the alphabetic order. Thus the set of $(7 \times 2 + 3 \times 2 + 3) \times 15 = 345$ formulae considered are those length 11 EC-tautologies whose bracket type is described using only the letters A, B, C, D, E, F, G, O, P, Q, T, U, V, X, and Y.[27]

[27]It was not noticed until the time of writing that the bracket types YQ and QY are also dual, and hence this set could be reduced to 331 formulae.

Appendix 2: Proofs of New Single Axioms Using CD and RCD

The following OTTER proofs [?] were found on a SPARCstation 10/40 in under 30 minutes. All proofs use the *hot list* strategy [?, page 48], and so clauses expressing the formula and the rules of CD and RCD may occur twice. Proofs have been slightly edited for clarity.

Proof that DXM can derive the single axiom TN.

```
Length of proof is 2.  Level of proof is 2.

---------------- PROOF ----------------

2   [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(e(e(x,y),z),e(y,z)),x)).   (DXM)
4   [] -P(e(x,y))| -P(x)|P(y).          (CD)
23  [] -P(e(e(e(a,b),c),e(e(c,a),b))).  (TN)
25 [hyper,2,3,3]
       P(e(e(e(e(e(e(e(x,y),z),e(y,z)),x),u),v),e(u,v))).
29 (heat=1) [hyper,4,25,6] P(e(e(e(x,y),z),e(e(z,x),y))).
30 [binary,29.1,23.1] [].

------------ end of proof -------------
```

Proof that UJ can derive the single axiom TN.

```
Length of proof is 7.  Level of proof is 6.

---------------- PROOF ----------------

1,4 [] -P(e(x,y))| -P(x)|P(y).          (CD)
2,5 [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(e(x,y),z),e(x,e(y,z)))).   (UJ)
23  [] -P(e(e(e(a,b),c),e(e(c,a),b))).  (TN)
25 [hyper,2,3,3] P(e(e(e(e(x,y),z),x),e(y,z))).
28 (heat=1) [hyper,5,25,6]
       P(e(e(e(x,e(e(y,z),u)),e(y,e(z,u))),x)).
32 [hyper,1,25,25] P(e(x,x)).
39 [hyper,2,25,32] P(e(e(e(x,y),y),x)).
42 (heat=1) [hyper,4,6,39] P(e(e(x,y),e(y,x))).
116 [hyper,2,25,42] P(e(e(e(x,e(y,z)),e(z,y)),x)).
4921 [hyper,1,116,28] P(e(e(e(x,y),z),e(e(z,x),y))).
4922 [binary,4921.1,23.1] [].

------------ end of proof -------------
```

Proof that VJ can derive the single axiom QYJ (dual of YQJ).

```
Length of proof is 10.  Level of proof is 7.

---------------- PROOF ----------------

1,4 [] -P(e(x,y))| -P(x)|P(y).          (CD)
2,5 [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(x,e(y,z)),e(e(x,y),z))).   (VJ)
18  [] -P(e(e(e(a,b),e(c,a)),e(b,c))).  (QYJ)
25 [hyper,2,3,3] P(e(x,e(e(y,z),e(e(x,y),z)))).
26 [hyper,1,3,3] P(e(e(e(x,e(y,z)),e(x,y)),z)).
28 (heat=1) [hyper,4,25,6]
   P(e(e(x,y),e(e(e(e(z,e(u,v)),e(e(z,u),v)),x),y))).
36 [hyper,2,26,25]
   P(e(e(x,e(y,e(z,e(e(u,v),e(e(z,u),v))))),e(x,y))).
42 (heat=1) [hyper,5,36,6]
   P(e(e(x,e(y,z)),e(e(e(x,y),z),e(u,e(e(v,w),e(e(u,v),w)))))).
116 [hyper,1,3,28]
   P(e(e(e(x,y),e(e(e(z,e(u,v)),e(e(z,u),v)),x)),y)).
142 (heat=1) [hyper,4,116,6] P(e(e(e(x,y),z),e(x,e(y,z)))).
159 [hyper,2,3,142] P(e(e(x,y),e(z,e(x,e(y,z))))).
22159 [hyper,2,159,42] P(e(e(e(e(x,y),e(z,x)),y),z)).
22340 (heat=1) [hyper,5,6,22159] P(e(e(e(x,y),e(z,x)),e(y,z))).
22341 [binary,22340.1,18.1] [].

------------ end of proof -------------
```

Proof that VN can derive the single axiom UM.

```
Length of proof is 6.  Level of proof is 4.

---------------- PROOF ----------------

1,3 [] -P(e(x,y))| -P(x)|P(y).          (CD)
2,4 [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(x,e(y,z)),e(e(z,x),y))).   (VN)
24  [] -P(e(e(e(a,b),c),e(b,e(c,a)))).  (UM)
25 [hyper,2,3,3] P(e(e(x,y),e(e(e(y,z),x),z))).
26 [hyper,1,3,3] P(e(e(x,e(y,e(x,z))),e(z,y))).
37 [hyper,2,26,25] P(e(x,e(e(e(e(y,z),u),z),e(x,e(u,y))))).
39 [hyper,1,26,25] P(e(x,e(e(y,e(e(z,y),x)),z))).
53 (heat=1) [hyper,4,6,39] P(e(e(x,y),e(z,e(e(x,z),y)))).
7182 [hyper,2,53,37] P(e(e(e(x,y),z),e(y,e(z,x)))).
7183 [binary,7182.1,24.1] [].

------------ end of proof -------------
```

Proof that CXM can derive the single axiom DXM.

```
Length of proof is 12.  Level of proof is 12.

---------------- PROOF ----------------

1,4 [] -P(e(x,y))| -P(x)|P(y).          (CD)
2,5 [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(e(e(x,y),e(z,y)),z),x)).   (CXM)
25  [] -P(e(e(e(e(a,b),c),e(b,c)),a)).  (DXM)
30 [hyper,2,3,3]
   P(e(e(e(e(e(e(e(x,y),e(z,y)),z),x),u),e(v,u)),v))).
33 (heat=1) [hyper,4,30,6] P(e(e(e(x,y),x),y)).
36 [hyper,2,33,3] P(e(e(x,e(e(e(e(y,z),e(u,z)),u),y)),x)).
44 (heat=1) [hyper,5,36,6]
   P(e(e(e(e(e(x,y),e(z,y)),z),x),e(e(e(e(u,v),e(w,v)),w),u))).
2116 [hyper,1,3,44]
   P(e(e(e(e(e(e(e(x,y),e(z,y)),z),x),u),v),e(u,v))).
2119 (heat=1) [hyper,4,2116,6] P(e(x,e(e(e(x,y),z),e(y,z)))).
2150 [hyper,1,2119,2119]
   P(e(e(e(e(x,e(e(e(x,y),z),e(y,z))),u),v),e(u,v))).
2231 (heat=1) [hyper,4,2150,6] P(e(e(e(e(x,e(y,z)),y),z),x)).
2805 [hyper,1,2119,2231]
   P(e(e(e(e(e(e(e(x,e(y,z)),y),z),x),u),v),e(u,v))).
2919 (heat=1) [hyper,4,2805,6] P(e(x,e(e(e(x,y),e(z,y)),z))).
4042 [hyper,1,2119,2919]
   P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v))).
4072 (heat=1) [hyper,4,4042,6] P(e(e(e(e(x,y),z),e(y,z)),x)).
4073 [binary,4072.1,25.1] [].

------------ end of proof -------------
```

Proof that OYJ can derive the single axiom UJ.

```
Length of proof is 5.  Level of proof is 5.

---------------- PROOF ----------------

1   [] -P(e(x,y))| -P(x)|P(y).          (CD)
2,5 [] -P(e(x,y))| -P(y)|P(x).          (RCD)
3,6 [] P(e(e(e(e(x,y),z),x),e(y,z))).   (OYJ)
27  [] -P(e(e(e(a,b),c),e(a,e(b,c)))).  (UJ)
31 [hyper,1,3,3] P(e(x,x)).
36 (heat=1) [hyper,5,6,31] P(e(e(e(x,y),y),x)).
40 [hyper,2,36,3] P(e(e(e(e(e(e(x,y),z),x),e(y,z)),u),u)).
51 (heat=1) [hyper,5,40,6]
   P(e(e(e(e(e(x,y),z),x),e(y,z)),e(e(e(e(u,v),w),u),e(v,w)))).
5779 [hyper,1,3,51] P(e(e(e(x,y),z),e(x,e(y,z)))).
5780 [binary,5779.1,27.1] [].

------------ end of proof -------------
```

The following is a proof that YPG can derive the single axiom XGK. This was first shown by Eliot Lawes [?]; however, I do not have a copy of his proof. Note that this proof does not use the *hot list* strategy.

```
Length of proof is 13.  Level of proof is 11.

---------------- PROOF ----------------

1 [] -P(e(x,y))| -P(x)|P(y).
2 [] -P(e(x,y))| -P(y)|P(x).
3 [] P(e(e(x,y),e(e(y,e(x,z)),z))).    (YPG).
20 [] -P(e(a,e(e(b,e(c,a)),e(c,b)))).  (XGK).
24 [hyper,1,3,3] P(e(e(e(e(x,e(y,z)),z),e(e(y,x),u)),u)).
26 [hyper,2,24,3]
   P(e(e(e(x,e(y,z)),z),e(e(y,x),e(e(u,v),e(e(v,e(u,w)),w))))).
35 [hyper,2,3,26]
   P(e(x,e(y,e(z,e(x,e(e(z,y),e(e(u,v),e(e(v,e(u,w)),w)))))))).
54 [hyper,2,35,26]
   P(e(e(x,e(y,e(e(z,u),e(e(u,e(z,v)),v)))),e(y,x))).
58 [hyper,1,24,35]
P(e(x,e(e(e(y,e(z,u)),u),e(e(x,e(z,y)),e(e(v,w),e(e(w,e(v,v6)),v6)))))).
79 [hyper,1,54,24]
P(e(x,e(e(e(y,e(z,u)),u),e(e(z,y),e(x,e(e(v,w),e(e(w,e(v,v6)),v6))))))).
420 [hyper,2,79,58] P(e(e(e(x,e(e(y,e(z,x)),u)),u),e(z,y))).
462 [hyper,2,3,420] P(e(x,e(y,e(e(z,e(u,y)),e(x,e(u,z)))))).
486 [hyper,2,462,462] P(e(x,e(y,e(e(y,x),e(z,z))))).
512 [hyper,1,54,462] P(e(x,e(e(y,x),e(z,e(y,z))))).
543 [hyper,2,462,486] P(e(e(x,e(x,y)),y)).
957 [hyper,2,512,543] P(e(x,e(y,e(x,y)))).
1020 [hyper,2,58,957] P(e(x,e(e(y,e(z,x)),e(z,y)))).
1021 [binary,1020.1,20.1] [].

------------ end of proof -------------
```

Appendix 3: A Chain of Deductive Subsystems

As described in Section **??** the formulae QYF and YQM appear to be single generators for the deductive subsystem corresponding to the matrix II (described in [**?**]). The following table describes some potential subsystems contained in the one mentioned above. Each of the formulae contained in the table are satisfied by the matrix II and can be derived from QYF or YQM using CD and RCD. The formulae in the first row can derive all the formulae in the column below and so are potential single generators for the corresponding deductive subsystem.

Single Generator	PYN or YPD	XGJ	YPN	QYC or YQO
Formulae Satisfied	EXC			
	EXH			
	EXI			
	GXA			
	GXC	GXC		
	GXG			
	GXI	GXI		
		GXJ		
	PYA	PYA	PYA	
	PYG	PYG		
	QYA	QYA	QYA	QYA
	QYD	QYD	QYD	QYD
	QYN	QYN		
	TC			
	TJ	TJ	TJ	TJ
	XEA			
	XEG			
	XGA	XGA		
	XGG	XGG	XGG	
	XGI	XGI	XGI	
	XGO	XGO		
	YPA	YPA	YPA	
	YPC	YPC		
	YQA	YQA	YQA	YQA
	YQB	YQB	YQB	YQB
	YQD	YQD	YQD	

The subsystem corresponding to the leftmost column is the finite system mentioned in Section **??**. Note that each subsystem, described in the above order, appears to be contained in the previous, suggesting a chain of subsystems.

Appendix 4: Reduction of Formulae to Equation Sets

In Section **??** it was shown that a matrix (n,m;k) satisfies a formula if and only if its parameters satisfy a corresponding set of equations. The following table shows the zeros of the parameters of the matrix corresponding to the indicated formula. Formulae that are known to be derivable from the given formula appear in the adjacent column; the corresponding zeros must all be consistent for a model to exist.

Formula	Derivable	Zeros mod k		
AXL		n^5+nm	n^4m+n^2m	n^3m+m
	OA	n^3+n^2m	$nm+m$	
	OD	n^3+nm	n^2m+m	
	PG	n^2+m	n^2m+nm^2	
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	YA	$n+m$		
AXN		n^5+nm	n^4m+m	n^3m+n^2m
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	YA	$n+m$		
BXO		n^4+m	n^4m+nm	$n^3m^2+n^2m$
	SD	$n+nm^2$	$nm+m^3$	
CXH		n^4+nm	$2n^3m$	n^2m^2+m
CXI		n^4+m	$2n^3m$	n^2m^2+nm
DXD		n^4+n^2m	n^3m+n^2m	nm^2+m
	RD	$n+nm^2$	n^2m+m^2	
EXE		$n^3+n^3m^2$	n^4m+nm	n^2m^2+m
FXK		$n^3+n^2m^3$	n^3m+m	n^3m^2+nm
FXL		n^3+nm	$n^3m+n^2m^3$	n^3m^2+m
GXB		n^3+n^3m	$n^2m^2+nm^2$	n^2m+m
OYF		n^4+n^2m	n^3m+m^2	$2nm$
	OA	n^3+n^2m	$nm+m$	
	OD	n^3+nm	n^2m+m	
	OG	n^3+m	n^2m+nm	
	PG	n^2+m	n^2m+nm^2	
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	YA	$n+m$		
Continued on next page				

Formula	Derivable	Zeros mod k		
TD		n^3+nm	$2n^2m$	nm^2+m^2
	OA	n^3+n^2m	$nm+m$	
	OG	n^3+m	n^2m+nm	
	PG	n^2+m	n^2m+nm^2	
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	YA	$n+m$		
VA		n^2+n^2m	n^2m+nm^2	nm^2+m^2
	OA	n^3+n^2m	$nm+m$	
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	SA	$n+nm$	nm^2+m^3	
	YA	$n+m$		
XAK		$n+n^2m^2$	n^4m+m^2	$n^3m^2+nm^2$
XBC		$n+n^3m$	$n^3m^2+m^2$	$n^2m^3+nm^2$
XCA		$n+n^3m$	$2n^2m^2$	nm^3+m^2
XCB		$n+n^3m$	$n^2m^2+nm^3$	$n^2m^2+m^2$
	OA	n^3+n^2m	$nm+m$	
	PG	n^2+m	n^2m+nm^2	
	QA	$nm+m^2$		
	QD	n^2+nm	$nm+m^2$	
	RG	$n+m^2$	n^2m+nm^2	
	YA	$n+m$		
XCL		$n+nm^3$	$n^3m+n^2m^2$	$n^2m^2+m^2$
XDB		$n+n^3m$	$n^2m^2+nm^2$	nm^2+m^3
XDE		$n+n^2m^2$	n^3m+nm^2	nm^2+m^3
XDM		$n+m^3$	n^3m+nm^2	$n^2m^2+nm^2$
XDO		$n+m^3$	n^3m+nm^2	$n^2m^2+nm^2$
	RA	$n+n^2m$	nm^2+m^2	
XFE		$n+n^2m^2$	n^2m+nm^4	$n^2m^3+m^2$

Appendix 5: Refutations for Models of the Form $(n, m; k)$

Models of the form $(n, m; k)$ satisfying AXL, OYF, TD, and XCB must obey the following equations.

$$\begin{array}{rrl}
\text{(i)} & n+m = & 0 \pmod k \\
\text{(ii)} & n^2+m = & 0 \pmod k \\
\text{(iii)} & nm+m = & 0 \pmod k \\
\text{(iv)} & n^2+nm = & 0 \pmod k
\end{array}$$

Subtracting (i) from (ii) gives $n^2 = n$. Subtracting (iii) from (iv) gives $n^2 = m$. Hence $n = n^2 = m$. But then (i) gives $n + n = 2n = 0$ and hence $k \mid 2n$. But n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying these formulae.

Models of the form $(n, m; k)$ satisfying AXN must obey the following equations.

$$\begin{array}{rrl}
\text{(i)} & n^3m+n^2m = & 0 \pmod k \\
\text{(ii)} & n^4m+m = & 0 \pmod k \\
\text{(iii)} & nm+m^2 = & 0 \pmod k \\
\text{(iv)} & n^2+nm = & 0 \pmod k \\
\text{(v)} & n+m = & 0 \pmod k
\end{array}$$

Multiplying (i) gives $n^4m+n^3m = 0$, and hence subtracting (ii) gives $n^3m = m$. Thus (ii)$= n^4m + m = n(n^3m) + m = nm + m = 0$. Hence (iii) gives $m^2 = m$ and (iv) gives $n^2 = m$. Thus (ii)$= n^4m + m = m^3 + m = m + m = 2m = 0$ and hence $k \mid 2m$. But m and k are relatively prime, so $k \mid 2$. Hence $k = 2$, since $n, m < k$, and then (v) gives $n = m = 1$. Therefore $(1, 1; 2)$ is the only model of this form satisfying AXN.

Models of the form $(n, m; k)$ satisfying VA must obey the following equations.

$$\begin{array}{rrl}
\text{(i)} & n+nm = & 0 \pmod k \\
\text{(ii)} & nm+m = & 0 \pmod k \\
\text{(iii)} & n+m = & 0 \pmod k
\end{array}$$

Subtracting (i) from (ii) gives $n = m$, and hence (iii)$= n+n = 2n = 0$. Therefore $k \mid 2n$. But n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying VA.

Models of the form $(n, m; k)$ satisfying BXO must obey the following equations.

$$\begin{aligned} &\text{(i)} & n^4 + m &= 0 \pmod{k} \\ &\text{(ii)} & n^4 m + nm &= 0 \pmod{k} \\ &\text{(iii)} & n + nm^2 &= 0 \pmod{k} \\ &\text{(iv)} & nm + m^3 &= 0 \pmod{k} \end{aligned}$$

Multiplying (i) gives $n^4 m + m^2$ and hence subtracting (ii) gives $m^2 = nm$. Thus (iii)$= n + nm^2 = n + (nm)m = n + m^3 = 0$. Subtracting (iv) from this gives $n = nm$. But then (iii)$=n + nm^2 = n + (nm)m = n + nm = n + n = 2n = 0$, and hence $k \mid 2n$. But n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = 1$, since $n < k$, and then (i) gives $m = 1$. Therefore $(1, 1; 2)$ is the only model of this form satisfying VA.

Models of the form $(n, m; k)$ satisfying DXD must obey the following equations.

$$\begin{aligned} &\text{(i)} & nm^2 + m &= 0 \pmod{k} \\ &\text{(ii)} & n + nm^2 &= 0 \pmod{k} \\ &\text{(iii)} & n^2 m + m^2 &= 0 \pmod{k} \end{aligned}$$

Subtracting (i) from (ii) gives $n = m$. Hence subtracting (ii)$= n + nm^2 = n + n^3$ from (iii)$= n^2 m + m^2 = n^3 + n^2$ gives $n^2 = n$. Thus (ii)$= n + nm^2 = n + n(n^2) = n + n^2 = n + n = 2n = 0$, and hence $k \mid 2n$. But n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying DXD.

Models of the form $(n, m; k)$ satisfying CXH must obey the following equations.

$$\begin{aligned} &\text{(i)} & 2n^3 m &= 0 \pmod{k} \\ &\text{(ii)} & n^2 m^2 + m &= 0 \pmod{k} \end{aligned}$$

Multiplying (i) by m and (ii) by $2n$, then subtracting, gives $2nm = 0$. Multiplying this by nm and (ii) by 2 then subtracting gives 2m=0. Hence $k \mid 2m$, but m and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying CXH.

Models of the form $(n, m; k)$ satisfying CXI must obey the following equations.

$$\begin{aligned} &\text{(i)} & 2n^3 m &= 0 \pmod{k} \\ &\text{(ii)} & n^2 m^2 + nm &= 0 \pmod{k} \end{aligned}$$

Multiplying (i) by m and (ii) by $2n$, then subtracting, gives $2n^2 m = 0$. Multiplying this by nm and (ii) by 2, then subtracting, gives 2m=0. Hence $k \mid 2m$, but m and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying CXI.

Models of the form $(n, m; k)$ satisfying XCA must obey the following equations.

$$\begin{aligned} \text{(i)} \quad n + n^3 m &= 0 \quad \pmod k \\ \text{(ii)} \quad 2n^2 m^2 &= 0 \quad \pmod k \end{aligned}$$

Multiplying (i) by $2m$ and (ii) by n, then subtracting, gives $2nm = 0$. Multiplying this by n^2 and (i) by 2, then subtracting, gives 2n=0. Hence $k \mid 2n$ but n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying XCA.

Models of the form $(n, m; k)$ satisfying XFE must obey the following equations.

$$\begin{aligned} \text{(i)} \quad n + n^2 m^2 &= 0 \quad \pmod k \\ \text{(ii)} \quad n^2 m + n m^4 &= 0 \quad \pmod k \\ \text{(iii)} \quad n^2 m^3 + m^2 &= 0 \quad \pmod k \end{aligned}$$

Multiplying (i) by m and subtracting (iii) gives the identity $nm = m^2$. Thus (ii)$= n(nm) + (nm)m^3 = nm^2 + m^5 = (nm)m + m^5 = m^3 + m^5$ and (iii)$= (n^2m^2)m + m^2 = m^5 + m^2$. Subtraction then gives $m^2 = m^3$ and thus (iii)$= n^2m^2 + m^2$. Subtracting (i) from this gives $n = m^2$ and hence the first identity gives $n = nm$. Thus (ii)$= n^2 + n^3$ and (iii)$= n^2 m(m^2) + m^2 = n^2(mn) + n = n^3 + n$. Subtraction then gives $n = n^2$ and finally (i)$= n + (nm)^2 = n + n^2 = n + n = 2n = 0$. Hence $k \mid 2n$ but n and k are relatively prime, so $k \mid 2$. Hence $k = 2$ and $n = m = 1$, since $n, m < k$. Therefore $(1, 1; 2)$ is the only model of this form satisfying XFE.

References

[1] Kahlil Hodgson. Program iteration: The development of a generic driving system for OTTER. Summer Scholarship Project, University of Auckland, New Zealand, report for Summer 94/95.

[2] J. A. Kalman. Axiomatisations of logics with values in groups. *Journal of the London Mathematical Society,* (2) 14:193–199, 1976.

[3] J. A. Kalman. Condensed detachment as a rule of inference. *Studia Logica,* 42:443–451,1983.

[4] J. A. Kalman. Testing logical matrices with the help of an automated reasoning program. *Proceedings of the 17th International Symposium on Multiple-Valued Logic*, Boston, Massachusetts, May 26–28, 1987, pp. 276–281.

[5] J. A. Kalman. *Automated Reasoning with* OTTER. Course notes for paper 26.414 at University of Auckland, New Zealand, 1995.

[6] John A. Kalman. A shortest single axiom for the equivalential calculus. *Notre Dame Journal of Formal Logic,* 19:141–144, 1978.

[7] Elliot Lawes. Shortest single axioms for the equivalential calculus using condensed detachment and reversed condensed detachment as inference rules. Summer Scholarship Project, University of Auckland, New Zealand, report for Summer 94/95.

[8] Łukasiewicz, J. Der Äquivalenzenkalkül. *Collectanea Logica,* 1:145–169, 1939. English translation in S. McCall (editor), *Polish Logic 1920–1939*, pp. 88–115. Oxford University Press, Oxford, 1967.

[9] Matlab: User's Guide (Version 4.0). The Math Works, August 1992.

[10] C. A. Meredith and A. N. Prior. Axiomatics of the propositional calculus. *Notre Dame Journal of Formal Logic,* 4:171–187, 1963.

[11] William W. McCune. A Davis-Putnam program and its application to finite first-order model search: Quasi-group existence problems. Technical Memo. ANL/MCS-TM-194, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill.,1994.

[12] William W. McCune. Otter 3.0 Reference Manual and Guide. Technical report ANL-94/6, Argonne National Laboratory, Argonne, Ill., January 1994.

[13] William McCune and R. Padmanabhan. *Automated Deduction in Equational Logic and Geometry.* Springer-Verlag, 1996.

[14] J. C. C. McKinsey and A. H. Diamond. Algebras and their sub-algebras. *Bulletin of the American Mathematical Society,* 53:959–962, 1947.

[15] N.S. Mendelsohn and R. Padmanabhan. Minimal Identities for Boolean Groups. *Journal of Algebra,* 34:451–457, 1975.

[16] A. N. Prior. *Formal Logic,* 2nd ed., Oxford University Press, Oxford, 1962.

[17] J. G. Peterson. The possible shortest single axioms for EC-tautologies. Technical report No. 105, Department of Mathematics, University of Auckland, March 1977.

[18] J. G. Peterson. Shortest single axioms for the equivalential calculus. *Notre Dame Journal of Formal Logic,* 17:267–271, 1976.

[19] Darren Redfern. *The Maple Handbook: Maple V release 3.* New York: Springer-Verlag, 1994.

[20] Alfred Tarski. *Logic, Semantics, Metamathematics: Papers from 1923 to 1938* (translated by J. H. Woodger). Oxford University Press, Oxford, 1956.

[21] Christine Treash. The completion of finite incomplete Steiner triple systems with applications to loop theory. *Journal of Combinatorial Theory,* 10:259–265, 1971.

[22] Ryszard Wojcicki. *Theory of logical calculi: Basic theory of consequence operations.* Dordrecht; Boston: Kluwer Academic Publishers, 1988.

[23] L. Wos, S. Winker, R. Veroff, B. Smith, and L. Henschen. A new use of an automated reasoning assistant: Open questions in equivalential calculus and the study of infinite domains. *Artificial Intelligence,* 22:305–356, 1984.

[24] L. Wos, Ross Overbeek, Ewing Lusk, and James Boyle. *Automated reasoning: Introduction and Applications,* 2nd ed., McGraw-Hill, New York, second edition, 1992.

[25] L. Wos. Searching for circles of pure proofs. *Journal of Automated Reasoning,* 15:279–315, 1995.