Experiments in Automated Deduction with Condensed Detachment*

William McCune and Larry Wos

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439-4801
U.S.A.

Abstract

This paper contains the results of experiments with several search strategies on 112 problems involving condensed detachment. The problems are taken from nine different logic calculi: three versions of the two-valued sentential calculus, the many-valued sentential calculus, the implicational calculus, the equivalential calculus, the R calculus, the left group calculus, and the right group calculus. Each problem was given to the theorem prover OTTER and was run with at least three strategies: (1) a basic strategy, (2) a strategy with a more refined method for selecting clauses on which to focus, and (3) a strategy that uses the refined selection mechanism and deletes deduced formulas according to some simple rules. Two new features of OTTER are also presented: the refined method for selecting the next formula on which to focus, and a method for controlling memory usage.

1 Introduction

The aim of this paper is to examine the role of strategy in the study of logic calculi with condensed detachment. We present results of experiments with the theorem-proving program OTTER on 112 problems, all of which contain the axiom (or, from another point of view, inference rule) condensed detachment.

All of the problems concern axiomatizations of various logic calculi, including the two-valued sentential calculus and two of its variations, the many-valued sentential calculus, the implicational fragment of sentential calculus, equivalential calculus, and three subsystems of the equivalential calculus: the R calculus, the left group calculus, and the right group calculus. The problems should also serve well as test problems for evaluating other search strategies and other theorem-proving programs.

We have experimented extensively with most of the problems, and we have developed specialized strategies for particular logic calculi. For the experiments presented in this paper, however, we sought strategies that perform well on all of the problems.

*This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Aside from a default basic strategy, we experimented with a guidance strategy and a deletion strategy. The guidance strategy, which we call the ratio strategy (Section 1.2.1), combines best-first search with breadth-first search when selecting the next formula on which to focus. The deletion strategy (Section 1.2.2) causes derived formulas that are instances of simple patterns to be deleted.

1.1 Condensed Detachment

All of the problems use C. A. Meredith's *condensed detachment* [6, 12], a rule of inference that combines detachment (modus ponens) and instantiation. Let C be the binary operation of concern. If $C(\alpha, \beta)$ and γ are both theorems (renamed so that they have no variables in common), and if α and γ unify with most general unifier σ, then $\beta\sigma$ is deduced by condensed detachment. The formula $C(\alpha, \beta)$ is the *major premise*, and γ is the *minor premise*. The binary operation is usually interpreted as "implies", "equivalent", or some variation of the group operation, depending on the calculus.

The logic calculi can be studied as first-order theories by a trivial transformation [5]. First, a unary predicate P, interpreted as "is a theorem" or "is the group identity", is introduced. Then, each axiom of the calculus is preceded by P, with its variables universally quantified. Finally, condensed detachment becomes an axiom of the theory.

$$\forall x \forall y (P(C(x, y)) \;\&\; P(x) \;\; \rightarrow \;\; P(y)).$$

An application of hyperresolution with the axiom condensed detachment corresponds directly to an application of the inference rule condensed detachment. Although we used hyperresolution exclusively for the experiments presented in this paper, any inference rule for first-order logic is applicable.

The AN calculus, which is a variation of the two-valued sentential calculus, has a binary operation o, which can be interpreted as disjunction, and a unary operation n, which can be interpreted as negation. For the AN calculus, the following variation of condensed detachment is used:

$$\forall x \forall y (P(o(n(x), y)) \;\&\; P(x) \;\; \rightarrow \;\; P(y)).$$

The study of logic calculi with condensed detachment has been one of the first and most successful applications of automated theorem proving. Original research has been conducted and open questions have been answered by relying heavily on automated theorem proving programs [3, 14, 13, 4, 22, 18, 21, 8, 9].

1.2 Otter and Simple Strategies

Otter [7] is a resolution/paramodulation theorem-proving program for first-order logic with equality. Its basic algorithm, restricted to hyperresolution with condensed detachment, is shown in Figure 1.

1.2.1 Selecting the Given Clause

Our default strategy for selecting the given clause in Step 1 of the basic algorithm has traditionally been to select a clause with the fewest symbols; if there is more

Start with *sos* list containing all axioms and with *usable* list containing the axiom for condensed detachment.

Loop:

1. G = select-given-clause(*sos*);
2. move G from *sos* to *usable*;
3. apply condensed detachment as much as possible, with G as one premise, taking the other premise from *usable*; append to *sos* the results that are not subsumed by anything in *sos* or *usable*;

end loop.

Figure 1: OTTER's Basic Algorithm with Condensed Detachment

than one clause of minimum length, the first of those is selected. We call the default strategy "selecting the smallest clause as given". However, some problems in the logic calculi yield quickly to a breadth-first search, which is accomplished by selecting the first clause in the *sos* list as the given clause. The method we use for most of the experiments presented in this paper combines those two methods. In every fourth iteration of the loop, the first clause is selected, and in the remaining iterations, the first clause of minimum length is selected. We call this refined method "selecting the given clause with ratio 3". The refinement allows large clauses to enter the search while the focus remains mainly on small clauses. It is similar to a selection strategy used by J. Kalman in one of his early programs [3].

1.2.2 Deleting Derived Formulas

In the equivalential calculus, the R calculus, and the left and right group calculi (all of which have binary operator e), we found that formulas containing subformulas that are instances of $e(x,x)$ are generally not as useful or as powerful as formulas without such instances. Searches in which those formulas are deleted are generally more effective, although they can result in longer proofs. The strategy also applies to the implicational calculi by deleting deduced formulas with instances of $i(x,x)$, although it appears to be less effective there. The strategy applies to the AN calculus, in which the binary operation is disjunction, by deleting formulas with instances of $o(n(x),x)$.

In the calculi with unary operation n, meaning negation, we found that deduced formulas containing instances of $n(n(x))$ caused redundancy in the search spaces and that deleting those formulas generally improved the searches. We also ran experiments deleting formulas with instances of $n(n(n(x)))$.

We used demodulation of derived formulas to implement the deletion strategy. When the strategy was in use, demodulation usually accounted for between one third and one half of the CPU time.

1.2.3 Controlling Memory Usage

We limited the OTTER jobs to 12 Mbytes of memory, in which OTTER can store roughly 20,000 formulas. Even with the deletion strategy of the preceding subsection, OTTER quickly fills 12 Mbytes. The list *sos* typically grows much faster than does the number of given clauses that are removed from it. Thus, most formulas in *sos* never enter the search, and memory is wasted.

Our current solution to that problem is the following. When one third of available memory has been filled, we impose a limit on the number of symbols in deduced clauses. The limit, say n, is such that 5% of all formulas in *sos* have $\leq n$ symbols. Every tenth iteration of the main loop after the initial limit has been set, calculate a prospective new limit n' in the same way. If $n' < n$, then the limit is reset to n'. We arrived at the values 1/3 and 5% by trial and error. Although this method is incomplete, its use with condensed detachment problems typically does not have a great effect on the sequence of given clauses, or therefore, on the search. We have not experimented heavily with this method on other problems.

1.3 The Experiments

We ran all of the experiments on SPARCstation 1+ computers with 16 megabytes of main memory. In that environment, OTTER can infer several thousand formulas per second, most of which are deleted because they are subsumed by existing formulas or by the deletion strategy. (Back subsumption, in which newly kept formulas cause the deletion of weaker existing formulas, was not used.)

Here is an example of the way in which problems are presented. Given the equivalential calculus formulas

(EC-1)	$e(e(e(x,y),e(z,x)),e(y,z))$
(EC-4)	$e(e(x,y),e(y,x))$
(EC-5)	$e(e(e(x,y),z),e(x,e(y,z)))$

the problem (EC-4,EC-5 $\Rightarrow$ EC-1) is to find a refutation of the following set of clauses. Symbols x, y, and z are variables, and a, b, and c are Skolem constants.

$\neg P(e(x,y)) \mid \neg P(x) \mid P(y).$	% Condensed Detachment
$P(e(e(x,y),e(y,x))).$	% EC-4
$P(e(e(e(x,y),z),e(x,e(y,z)))).$	% EC-5
$\neg P(e(e(e(a,b),e(c,a)),e(b,c))).$	% Denial of EC-1

Each problem was run with several strategies with a time limit of four hours each. In the tables that follow, "Fail" indicates that no proof was found within four hours, and "*" indicates that no proof is possible, because the goal would be deleted by the deletion strategy. All of the times are given in seconds. The strategies are the following.

Basic. The smallest formula is selected as given.

Ratio. Given clauses are selected with ratio 3.

R-e. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $e(x,x)$ as a subformula are deleted.

R-i. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $i(x,x)$ as a subformula are deleted.

R-nn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $n(n(x))$ are deleted.

R-nnn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $n(n(n(x)))$ are deleted.

R-i-nn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $i(x, x)$ as a subformula or an instance of $n(n(x))$ are deleted.

R-i-nnn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $i(x, x)$ as a subformula or an instance of $n(n(n(x)))$ are deleted.

R-o-nn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $o(n(x), x)$ as a subformula or an instance of $n(n(x))$ are deleted.

R-o-nnn. Given clauses are selected with ratio 3, and deduced clauses containing an instance of $o(n(x), x)$ as a subformula or an instance of $n(n(n(x)))$ are deleted.

We present here a sequence of problems that reflects a wide range of difficulty and that roughly follows the historical development of the individual calculi.

2 Two-Valued Sentential Calculi

We experimented with three versions of two-valued sentential calculus: (1) the CN calculus, with operators intended to mean implication and negation, (2) the C0 calculus, with implication and falsehood, and (3) the AN calculus, with disjunction and negation. If appropriate definitions are added for the missing operators, each version is equivalent to the classical propositional calculus.

2.1 The Implication/Negation Two-Valued Sentential Calculus (CN)

Each of the following formulas holds in the two-valued sentential calculus (CN). The numbering of the formulas is from [16, p. 42–51].

(CN-1)	$i(i(x, y), i(i(y, z), i(x, z)))$
(CN-2)	$i(i(n(x), x), x)$
(CN-3)	$i(x, i(n(x), y))$
(CN-16)	$i(x, x)$
(CN-18)	$i(x, i(y, x))$
(CN-19)	$i(i(i(x, y), z), i(y, z))$
(CN-20)	$i(x, i(i(x, y), y))$
(CN-21)	$i(i(x, i(y, z)), i(y, i(x, z)))$
(CN-22)	$i(i(x, y), i(i(z, x), i(z, y)))$
(CN-24)	$i(i(i(x, y), x), x)$
(CN-30)	$i(i(x, i(x, y)), i(x, y))$
(CN-35)	$i(i(x, i(y, z)), i(i(x, y), i(x, z)))$
(CN-37)	$i(i(i(x, y), z), i(n(x), z))$
(CN-39)	$i(n(n(x)), x)$

(CN-40)	$i(x, n(n(x)))$
(CN-46)	$i(i(x, y), i(n(y), n(x)))$
(CN-49)	$i(i(n(x), n(y)), i(y, x))$
(CN-54)	$i(i(x, y), i(i(n(x), y), y))$
(CN-59)	$i(i(n(x), z), i(i(y, z), i(i(x, y), z)))$
(CN-60)	$i(i(x, i(n(y), z)), i(x, i(i(u, z), i(i(y, u), z))))$
(CN-CAM)	$i(i(i(i(i(x, y), i(n(z), n(u))), z), v), i(i(v, x), i(u, x)))$

According to Łukasiewicz [17, p. 136], the first axiom system for the two-valued sentential calculus was {CN-18,CN-21,CN-35,CN-39,CN-40,CN-46} and was due to Frege. We use that as our starting point. Łukasiewicz showed that CN-21 depends on the remaining axioms of Frege's system (Problem 1, Table 1). Another early axiomatization of CN was due to Hilbert [17, p. 136]: {CN-3,CN-18,CN-21,CN-22,CN-30,CN-54}. Łukasiewicz showed that CN-30 is not necessary (Problem 2, Table 1). Problems 3–6, Table 1, are to derive Frege's simplified system from Hilbert's simplified system.

Table 1: CN Calculus, Frege and Hilbert Systems

#	Theorem	Basic	Ratio	R-i-nn	R-i-nnn	R-nn	R-nnn
1	CN-18,CN-35,CN-39,CN-40,CN-46 ⇒ CN-21	Fail	246	16	176	26	254
2	CN-3,CN-18,CN-21,CN-22,CN-54 ⇒ CN-30	3	5	2	2	6	6
3	CN-3,CN-18,CN-21,CN-22,CN-54 ⇒ CN-35	Fail	Fail	8005	6371	3657	3864
4	CN-3,CN-18,CN-21,CN-22,CN-54 ⇒ CN-39	<1	9	*	8	*	11
5	CN-3,CN-18,CN-21,CN-22,CN-54 ⇒ CN-40	14	10	*	34	*	13
6	CN-3,CN-18,CN-21,CN-22,CN-54 ⇒ CN-46	7366	1534	1467	1509	2857	2401

Łukasiewicz axiomatized CN with {CN-1,CN-2,CN-3} [16]. Other axiom systems for CN are {CN-18,CN-35,CN-49} (Church [1]), {CN-19,CN-37,CN-59} (Łukasiewicz [16]), {CN-19,CN-37,CN-60} (Wos [19]), and {CN-CAM} (C. A. Meredith [10]). Problems 7–24, Table 2, are to start with {CN-1,CN-2,CN-3} and derive formulas in the other axiomatizations. Problems 25–36, Table 3, are to derive Łukasiewicz's system {CN-1,CN-2,CN-3} from the other systems.

2.2 The Implication/Falsehood Two-Valued Sentential Calculus (C0)

Each of the following formulas holds in the C0 calculus:

(C0-1)	$i(i(x, y), i(i(y, z), i(x, z)))$
(C0-2)	$i(x, i(y, x))$
(C0-3)	$i(i(i(x, y), x), x)$
(C0-4)	$i(F, x)$
(C0-5)	$i(i(i(x, F), F), x)$
(C0-6)	$i(i(x, i(y, z)), i(i(x, y), i(x, z)))$
(C0-CAM)	$i(i(i(i(i(x, y), i(z, F)), u), v), i(i(v, x), i(z, x)))$

Each of the sets {C0-1,C0-2,C0-3,C0-4} (Tarski-Bernays, according to [10]), {C0-2,C0-5,C0-6} (Church [1]), and {C0-CAM} (C. A. Meredith [10]) axiomatizes the

Table 2: CN Calculus, Starting with {CN-1,CN-2,CN-3}

#	Theorem	Basic	Ratio	R-i-nn	R-i-nnn	R-nn	R-nnn
7	CN-1,CN-2,CN-3 $\Rightarrow$ CN-16	<1	<1	<1	<1	<1	<1
8	CN-1,CN-2,CN-3 $\Rightarrow$ CN-18	60	7	5	5	10	9
9	CN-1,CN-2,CN-3 $\Rightarrow$ CN-19	60	7	5	5	10	9
10	CN-1,CN-2,CN-3 $\Rightarrow$ CN-20	89	147	23	28	65	157
11	CN-1,CN-2,CN-3 $\Rightarrow$ CN-21	104	148	23	28	66	158
12	CN-1,CN-2,CN-3 $\Rightarrow$ CN-22	105	589	74	184	177	595
13	CN-1,CN-2,CN-3 $\Rightarrow$ CN-24	105	71	31	40	40	86
14	CN-1,CN-2,CN-3 $\Rightarrow$ CN-30	109	71	32	45	40	86
15	CN-1,CN-2,CN-3 $\Rightarrow$ CN-35	Fail	Fail	Fail	Fail	Fail	Fail
16	CN-1,CN-2,CN-3 $\Rightarrow$ CN-37	105	33	31	40	34	40
17	CN-1,CN-2,CN-3 $\Rightarrow$ CN-39	104	31	*	37	*	41
18	CN-1,CN-2,CN-3 $\Rightarrow$ CN-40	106	32	*	38	*	42
19	CN-1,CN-2,CN-3 $\Rightarrow$ CN-46	1021	1262	423	1434	470	1378
20	CN-1,CN-2,CN-3 $\Rightarrow$ CN-49	260	73	31	36	45	88
21	CN-1,CN-2,CN-3 $\Rightarrow$ CN-54	1195	1608	447	1552	509	1763
22	CN-1,CN-2,CN-3 $\Rightarrow$ CN-59	Fail	Fail	Fail	Fail	Fail	Fail
23	CN-1,CN-2,CN-3 $\Rightarrow$ CN-60	Fail	Fail	Fail	Fail	Fail	Fail
24	CN-1,CN-2,CN-3 $\Rightarrow$ CN-CAM	Fail	Fail	Fail	Fail	Fail	Fail

Table 3: CN Calculus, Deriving {CN-1,CN-2,CN-3}

#	Theorem	Basic	Ratio	R-i-nn	R-i-nnn	R-nn	R-nnn
25	CN-18,CN-35,CN-49 $\Rightarrow$ CN-1	Fail	1083	89	531	91	1137
26	CN-18,CN-35,CN-49 $\Rightarrow$ CN-2	4	3	8	11	1	4
27	CN-18,CN-35,CN-49 $\Rightarrow$ CN-3	3	1	3	3	12	2
28	CN-19,CN-37,CN-59 $\Rightarrow$ CN-1	6038	303	89	245	99	286
29	CN-19,CN-37,CN-59 $\Rightarrow$ CN-2	622	359	3107	6800	257	592
30	CN-19,CN-37,CN-59 $\Rightarrow$ CN-3	161	12	5	12	6	15
31	CN-19,CN-37,CN-60 $\Rightarrow$ CN-1	5611	515	493	682	480	702
32	CN-19,CN-37,CN-60 $\Rightarrow$ CN-2	753	546	Fail	Fail	511	755
33	CN-19,CN-37,CN-60 $\Rightarrow$ CN-3	239	224	345	337	329	332
34	CN-CAM $\Rightarrow$ CN-1	Fail	Fail	Fail	Fail	Fail	Fail
35	CN-CAM $\Rightarrow$ CN-2	Fail	Fail	Fail	Fail	Fail	Fail
36	CN-CAM $\Rightarrow$ CN-3	6	10	Fail	Fail	14	14

C0 calculus. Problems 37–49, Table 4, involve deriving each axiom system from the others.

Table 4: The C0 Calculus

#	Theorem	Basic	Ratio	R-i
37	C0-2,C0-5,C0-6 $\Rightarrow$ C0-1	419	72	54
38	C0-2,C0-5,C0-6 $\Rightarrow$ C0-3	337	103	98
39	C0-2,C0-5,C0-6 $\Rightarrow$ C0-4	<1	<1	<1
40	C0-2,C0-5,C0-6 $\Rightarrow$ C0-CAM	Fail	Fail	Fail
41	C0-1,C0-2,C0-3,C0-4 $\Rightarrow$ C0-5	38	7	7
42	C0-1,C0-2,C0-3,C0-4 $\Rightarrow$ C0-6	1251	953	1010
43	C0-1,C0-2,C0-3,C0-4 $\Rightarrow$ C0-CAM	Fail	Fail	Fail
44	C0-CAM $\Rightarrow$ C0-1	Fail	Fail	Fail
45	C0-CAM $\Rightarrow$ C0-2	<1	<1	<1
46	C0-CAM $\Rightarrow$ C0-3	13	24	30
47	C0-CAM $\Rightarrow$ C0-4	<1	<1	<1
48	C0-CAM $\Rightarrow$ C0-5	5	9	12
49	C0-CAM $\Rightarrow$ C0-6	Fail	Fail	Fail

2.3 The Disjunction/Negation Two-Valued Sentential Calculus (AN)

Each of the following formulas holds in the AN calculus:

(AN-1)	$o(n(o(n(y),z)),o(n(o(x,y)),o(x,z)))$
(AN-2)	$o(n(o(x,y)),o(y,x))$
(AN-3)	$o(n(x),o(y,x))$
(AN-4)	$o(n(o(x,x)),x)$
(AN-CAM)	$o(n(o(n(o(n(x),y)),o(z,o(u,v)))),o(n(o(n(u),x)),o(z,o(v,x))))$

Each of the sets {AN-1,AN-2,AN-3,AN-4} (Whitehead-Russell, according to [10]) and {AN-CAM} (C. A. Meredith [10]) axiomatizes the AN calculus. Problems 50–54, Table 5, are to derive each system from the other. Recall that the clause form of condensed detachment for the AN calculus is $\neg P(o(n(x),y)) \mid \neg P(x) \mid P(y)$.

Table 5: The AN Calculus

#	Theorem	Basic	Ratio	R-o-nn	R-o-nnn	R-nn	R-nnn
50	AN-1,AN-2,AN-3,AN-4 $\Rightarrow$ AN-CAM	Fail	Fail	Fail	Fail	Fail	Fail
51	AN-CAM $\Rightarrow$ AN-1	Fail	Fail	Fail	Fail	516	Fail
52	AN-CAM $\Rightarrow$ AN-2	3472	Fail	Fail	Fail	449	5365
53	AN-CAM $\Rightarrow$ AN-3	34	58	133	78	137	78
54	AN-CAM $\Rightarrow$ AN-4	11447	Fail	Fail	Fail	2657	Fail

3 The Many-Valued Sentential Calculus (MV)

Each of the following formulas holds in the many-valued sentential calculus:

(MV-1)	$i(x, i(y, x))$
(MV-2)	$i(i(x, y), i(i(y, z), i(x, z)))$
(MV-3)	$i(i(i(x, y), y), i(i(y, x), x))$
(MV-4)	$i(i(i(x, y), i(y, x)), i(y, x))$
(MV-5)	$i(i(n(x), n(y)), i(y, x))$
(MV-24)	$i(n(n(x)), x)$
(MV-25)	$i(i(x, y), i(i(z, x), i(z, y)))$
(MV-29)	$i(x, n(n(x)))$
(MV-33)	$i(i(n(x), y), i(n(y), x))$
(MV-36)	$i(i(x, y), i(n(y), n(x)))$
(MV-39)	$i(n(i(x, y)), n(y))$
(MV-50)	$i(n(x), i(y, n(i(y, x))))$

Łukasiewicz defined the many-valued sentential calculus $L_{\aleph_0}$ and conjectured that it is axiomatized by {MV-1,MV-2,MV-3,MV-4,MV-5} [17]. Wajsberg proved the conjecture, and C. A. Meredith later proved MV-4 dependent on the remaining axioms [17, p. 144]. Problems 55–62, Table 6, are to prove MV-4 and several other formulas from {MV-1,MV-2,MV-3,MV-5}. (Problem 55 has been called "Luka5" by members of the Argonne group.)

Table 6: The MV Calculus

#	Theorem	Basic	Ratio	R-i-nn	R-i-nnn	R-nn	R-nnn
55	MV-1,MV-2,MV-3,MV-5 ⇒ MV-4	Fail	Fail	Fail	Fail	Fail	Fail
56	MV-1,MV-2,MV-3,MV-5 ⇒ MV-24	3	2	*	8	*	2
57	MV-1,MV-2,MV-3,MV-5 ⇒ MV-25	4475	8	5	5	9	9
58	MV-1,MV-2,MV-3,MV-5 ⇒ MV-29	3	2	*	8	*	2
59	MV-1,MV-2,MV-3,MV-5 ⇒ MV-33	Fail	2036	1468	2665	1827	3955
60	MV-1,MV-2,MV-3,MV-5 ⇒ MV-36	Fail	2035	3138	2664	3812	3955
61	MV-1,MV-2,MV-3,MV-5 ⇒ MV-39	7	17	675	25	628	16
62	MV-1,MV-2,MV-3,MV-5 ⇒ MV-50	Fail	2041	3151	2674	3825	3964

4 The Implicational Propositional Calculus (IC)

The implicational propositional calculus (IC) is the part of the sentential calculus in which the negation operation does not occur. Each of the following formulas holds in IC:

(IC-1)	$i(x, x)$
(IC-2)	$i(x, i(y, x))$
(IC-3)	$i(i(i(x, y), x), x)$
(IC-4)	$i(i(x, y), i(i(y, z), i(x, z)))$
(IC-5)	$i(x, i(i(x, y), y))$
(IC-JL)	$i(i(i(x, y), z), i(i(z, x), i(u, x)))$

Each of the sets {IC-2,IC-3,IC-4} (Tarski-Bernays, according to [17, p. 296]) and {IC-JL} (Łukasiewicz [17, p. 295]) axiomatizes IC. Problems 63–68, Table 7, are to derive each system from the other.

Table 7: The Implicational Propositional Calculus

#	Theorem	Basic	Ratio	R-e
63	IC-2,IC-3,IC-4 $\Rightarrow$ IC-JL	50	101	100
64	IC-JL $\Rightarrow$ IC-1	8	<1	27
65	IC-JL $\Rightarrow$ IC-2	8	<1	<1
66	IC-JL $\Rightarrow$ IC-3	32	47	26
67	IC-JL $\Rightarrow$ IC-4	7933	13985	12224
68	IC-JL $\Rightarrow$ IC-5	2172	3753	2715

5 Equivalential and Group Calculi

The equivalential and group calculi have one binary operator, e. In the equivalential calculus (EC) [17], $e(\alpha, \beta)$ is normally interpreted as equivalence of α and β; however, it can also be interpreted as the group operation $\alpha\beta$ in Boolean groups (groups in which the square of every element is the identity). Under the group interpretation, the theorems of EC are exactly the formulas that are equal to the group identity in Boolean groups.

The theorems of the R calculus [11] are exactly the formulas equal to the identity in Abelian groups when $e(\alpha, \beta)$ is interpreted as $\alpha\beta^{-1}$. There is also an L calculus, whose theorems are equal to the identity when $e(\alpha, \beta)$ is interpreted as $\alpha^{-1}\beta$. We have not experimented with the L calculus, but for completeness, we list here YOL, the shortest single axiom for the L calculus [14]. No other of length 11 exists.

(YOL) $\quad e(e(x,y),e(e(e(z,y),x),z))$

The theorems of the left group (LG) calculus [2] are exactly the formulas equal to the identity in (general) groups when $e(\alpha, \beta)$ is interpreted as $\alpha^{-1}\beta$. Similarly, the theorems of the right group (RG) calculus [2] are exactly the formulas equal to the identity in (general) groups when $e(\alpha, \beta)$ is interpreted as $\alpha\beta^{-1}$.

The following relationships exist between the equivalential and group calculi:

LG theorems $\subset$ L theorems $\subset$ EC theorems.
RG theorems $\subset$ R theorems $\subset$ EC theorems.

5.1 The Equivalential Calculus (EC)

The following formulas hold in the equivalential calculus:

(EC-1) $\quad e(e(e(x,y),e(z,x)),e(y,z))$
(EC-2) $\quad e(e(x,e(y,z)),e(e(x,y),z))$
(EC-4) $\quad e(e(x,y),e(y,x))$
(EC-5) $\quad e(e(e(x,y),z),e(x,e(y,z)))$

According to Łukasiewicz [17, p. 252], the first axiomatization of EC was {EC-1,EC-2}, due to Leśniewski. Soon after, Wajsberg produced others, including {EC-4,EC-5}. Problems 69 and 70, Table 8, are to derive the Leśniewski system from the Wajsberg system.

Each of the following formulas is a single axiom for EC, in roughly the order in which they were discovered. None shorter exists, nor does there exist any other of length 11.

(YQL)	$e(e(x,y),e(e(z,y),e(x,z)))$	Lukasiewicz
(YQF)	$e(e(x,y),e(e(x,z),e(z,y)))$	Lukasiewicz
(YQJ)	$e(e(x,y),e(e(z,x),e(y,z)))$	Lukasiewicz
(UM)	$e(e(e(x,y),z),e(y,e(z,x)))$	Meredith
(XGF)	$e(x,e(e(y,e(x,z)),e(z,y)))$	Meredith
(WN)	$e(e(x,e(y,z)),e(z,e(x,y)))$	Meredith
(YRM)	$e(e(x,y),e(z,e(e(y,z),x)))$	Meredith
(YRO)	$e(e(x,y),e(z,e(e(z,y),x)))$	Meredith
(PYO)	$e(e(e(x,e(y,z)),z),e(y,x))$	Meredith
(PYM)	$e(e(e(x,e(y,z)),y),e(z,x))$	Meredith
(XGK)	$e(x,e(e(y,e(z,x)),e(z,y)))$	Kalman
(XHK)	$e(x,e(e(y,z),e(e(x,z),y)))$	Winker
(XHN)	$e(x,e(e(y,z),e(e(z,x),y)))$	Winker

Problems 71–84, Table 8, are to start with each single axiom and derive the system that precedes it.

Table 8: EC

#	Theorem	Basic	Ratio	R-e
69	EC-4,EC-5 $\Rightarrow$ EC-1	244	366	279
70	EC-4,EC-5 $\Rightarrow$ EC-2	<1	<1	<1
71	YQL $\Rightarrow$ EC-4	<1	<1	<1
72	YQL $\Rightarrow$ EC-5	23	2	2
73	YQF $\Rightarrow$ YQL	2	5	2
74	YQJ $\Rightarrow$ YQF	34	54	33
75	UM $\Rightarrow$ YQJ	558	1074	159
76	XGF $\Rightarrow$ UM	<1	<1	<1
77	WN $\Rightarrow$ XGF	98	164	85
78	YRM $\Rightarrow$ WN	326	474	425
79	YRO $\Rightarrow$ YRM	188	250	151
80	PYO $\Rightarrow$ YRO	281	592	516
81	PYM $\Rightarrow$ PYO	245	449	352
82	XGK $\Rightarrow$ PYM	Fail	Fail	499
83	XHK $\Rightarrow$ XGK	Fail	Fail	886
84	XHN $\Rightarrow$ XHK	750	1690	484

5.2 The R Calculus (R)

Each of the following formulas is a single axiom for the R calculus:

(QYF)	$e(e(e(x,y),e(x,z)),e(z,y))$	Meredith
(YQM)	$e(e(x,y),e(e(z,y),e(z,x)))$	Meredith
(WO)	$e(e(x,e(y,z)),e(z,e(y,x)))$	Meredith
(XGJ)	$e(x,e(e(y,e(z,x)),e(y,z)))$	Winker

Problems 85–88, Table 9, are to show the four formulas equivalent in a circular manner.

Table 9: R Calculus

#	Theorem	Basic	Ratio	R-e
85	YQM $\Rightarrow$ QYF	<1	<1	<1
86	WO $\Rightarrow$ YQM	21	11	5
87	XGJ $\Rightarrow$ WO	Fail	Fail	362
88	QYF $\Rightarrow$ XGJ	41	42	21

5.3 The Left Group Calculus (LG)

Kalman's axiomatization of the LG calculus is {LG-1,LG-2,LG-3,LG-4,LG-5} [2].

(LG-1)	$e(e(e(x,e(e(y,y),x)),z),z)$
(LG-2)	$e(e(e(e(e(x,y),e(x,z)),e(y,z)),u),u)$
(LG-3)	$e(e(e(e(e(e(x,y),e(x,z)),u),e(e(y,z),u)),v),v)$
(LG-4)	$e(e(e(e(x,y),z),u),e(e(e(x,v),z),e(e(y,v),u)))$
(LG-5)	$e(e(e(x,e(e(y,x),z)),e(e(u,x),v)),e(e(e(e(x,y),u),z),v))$
(P-1)	$e(e(e(x,y),z),e(e(u,y),e(e(x,u),z)))$
(P-4)	$e(x,e(e(e(e(y,z),e(y,u)),e(z,u)),x))$
(Q-1)	$e(x,e(e(y,z),e(e(z,y),x)))$
(Q-2)	$e(e(x,y),e(e(z,x),e(z,y)))$
(Q-3)	$e(e(e(x,y),e(e(y,x),z)),z)$
(Q-4)	$e(e(e(x,y),e(x,z)),e(y,z))$
(LG-27-1690)	$e(e(e(e(x,y),z),e(e(u,v),e(e(e(w,v),e(w,u)),s))),e(z,e(e(y,x),s)))$

With great assistance from Otter, McCune later showed that each of the sets {LG-2,LG-3}, {LG-2,P-1}, {LG-2,P-4}, {LG-2,Q-1,Q-2}, {P-1,Q-3}, {P-4,Q-3}, {Q-1,Q-2,Q-3}, {Q-1,Q-3,Q-4}, and {LG-27-1690} also axiomatizes the LG calculus [8, 9]. Problems 89–101, Table 10, roughly parallel the discovery of the new axiom systems for the LG calculus.

5.4 The RG Calculus (RG)

Kalman's axiomatization of the RG calculus is {LG-1′,LG-2′,LG-3′,LG-4′,LG-5′} [2].

(LG-1′)	$e(x,e(x,e(e(y,e(z,z)),y)))$
(LG-2′)	$e(x,e(x,e(e(y,z),e(e(y,u),e(z,u)))))$
(LG-3′)	$e(x,e(x,e(e(y,e(z,u)),e(y,e(e(z,v),e(u,v))))))$
(LG-4′)	$e(e(e(x,e(y,z)),e(u,e(y,v))),e(x,e(u,e(z,v))))$
(LG-5′)	$e(e(x,e(y,e(z,e(u,v)))),e(e(x,e(v,z)),e(e(y,e(v,u)),v)))$
(Q-2′)	$e(e(e(x,y),e(z,y)),e(x,z))$

Table 10: LG Calculus

#	Theorem	Basic	Ratio	R-e
89	LG-2,LG-3,LG-4 $\Rightarrow$ LG-1	115	115	*
90	LG-2,LG-3 $\Rightarrow$ LG-4	222	6	2
91	LG-3 $\Rightarrow$ LG-4	109	8	9
92	LG-2,LG-3 $\Rightarrow$ LG-5	Fail	Fail	845
93	LG-2,P-1 $\Rightarrow$ LG-3	Fail	Fail	536
94	LG-2,P-4 $\Rightarrow$ P-1	<1	<1	<1
95	LG-2,Q-1,Q-2 $\Rightarrow$ P-1	16	30	5
96	P-1,Q-3 $\Rightarrow$ LG-2	211	295	15
97	P-4,Q-3 $\Rightarrow$ P-1	<1	<1	1
98	Q-1,Q-2,Q-3 $\Rightarrow$ LG-2	10905	Fail	7617
99	Q-1,Q-4 $\Rightarrow$ Q-2	<1	<1	<1
100	LG-27-1690 $\Rightarrow$ P-1	<1	<1	<1
101	LG-27-1690 $\Rightarrow$ Q-3	<1	2	3

(Q-3′) $e(x,e(e(x,e(y,z)),e(z,y)))$
(Q-4′) $e(e(x,y),e(e(x,z),e(y,z)))$

With great assistance from Otter, McCune later showed that each of the pairs {Q-2′,Q-3′} and {Q-3′,Q-4′} axiomatizes the RG calculus and that each of the following formulas is a single axiom for the RG calculus [8, 9]:

(LG-2′) $e(x,e(x,e(e(y,z),e(e(y,u),e(z,u)))))$
(S-2′) $e(e(x,e(y,z)),e(x,e(e(y,u),e(z,u))))$
(S-3′) $e(x,e(x,e(e(e(y,z),e(u,z)),e(y,u))))$
(S-4′) $e(e(x,e(y,z)),e(e(x,e(u,z)),e(y,u)))$
(P-4′) $e(e(x,e(e(y,z),e(e(y,u),e(z,u)))),x)$
(S-6′) $e(e(x,e(e(e(y,z),e(u,z)),e(y,u))),x)$

Problems 102–112, Table 11, roughly parallel the discovery of the new axiom systems for the RG calculus.

6 Summary

We have presented 112 condensed detachment problems that offer a large range of difficulty to automated theorem-proving programs, and we have shown how Otter, using several simple strategies (see Section 1.3), performs on those problems.

For the equivalential, R, RG, and LG calculi (the problems with functor e), strategy R-e wins on nearly all problems. We note that the deletion in strategy R-e prevents proofs in problems 89 and 102. For the CN, C0, and MV calculi, no clear overall winner was found. For the AN calculus problems, strategy R-nn performed best. For the IC problems, the basic strategy performed best. Although we did not run experiments using deletion while selecting the smallest clause as given, we can compare the performance of basic and ratio strategies (both without deletion). No

Table 11: RG Calculus

#	Theorem	Basic	Ratio	R-e
102	LG-2′ ⇒ LG-1′	130	133	*
103	LG-2′ ⇒ LG-3′	Fail	Fail	104
104	LG-2′ ⇒ LG-4′	Fail	889	62
105	LG-2′ ⇒ LG-5′	Fail	Fail	809
106	Q-2′,Q-3′ ⇒ LG-2′	9609	Fail	5634
107	Q-3′,Q-4′ ⇒ Q-2′	<1	<1	<1
108	S-2′ ⇒ LG-2′	757	1495	136
109	S-3′ ⇒ LG-2′	91	142	40
110	S-4′ ⇒ LG-2′	5837	Fail	Fail[a]
111	P-4′ ⇒ LG-2′	<1	<1	<1
112	S-6′ ⇒ LG-2′	120	143	19

[a] The deletion strategy eliminates *all* interesting paths.

clear winner was found, but the ratio strategy performed slightly better than the basic strategy overall. The results of the experiments reinforce our long-held position that a single strategy cannot be effective on a wide range of problems.

Several of the problems have been particularly challenging for us. Problem 67, posed as a challenge problem in [15] and called "imp4" by members of the Argonne group, was the first truly difficult condensed detachment theorem proved by OTTER. It has been used extensively as a benchmark for parallel deduction programs. Problem 34, to derive CN-1 from CN-CAM, has resisted all of our attempts at automated proofs. (One attempt generated 1.4 billion formulas and consumed 17 CPU days on a Solbourne 5e/900 computer.) Problem 55, to show the dependence of MV-4 in Lukasiewicz's system for $L_{\aleph_0}$, has also resisted all of our attempts. (One attempt generated 983 million formulas.) We have, however, found many proofs for Problem 55 using OTTER in various proof-checking modes [20]. OTTER's search for a proof for Problem 44, to derive C01 from C0-CAM, is impeded by the memory control feature. The weight limit is lowered, either too much or too soon, which causes key formulas to be discarded. OTTER has found a proof in about seven hours with a strategy similar to the basic strategy but with a constant weight limit of 18 instead of the memory control feature. We have not obtained proofs for problems 24, 40, 43, and 50, which are to derive complicated single axioms, because our strategies are biased towards finding simple formulas. The remaining problems for which the tables list complete failure have yielded to specialized strategies.

References

[1] A. Church. *Introduction to Mathematical Logic*, volume I. Princeton University Press, 1956.

[2] J. A. Kalman. Axiomatizations of logics with values in groups. *Journal of the London Math. Society*, 2(14):193–199, 1975.

[3] J. A. Kalman. Computer studies of $T_{\rightarrow} - W - I$. *Relevance Logic Newsletter*, 1:181–188, 1976.

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

[5] J. A. Kalman. Condensed detachment as a rule of inference. *Studia Logica*, LXII(4):443–451, 1983.

[6] E. J. Lemmon, C. A. Meredith, D. Meredith, A. N. Prior, and I. Thomas. Calculi of pure strict implication. Technical report, Canterbury University College, Christchurch, 1957. Reprinted in *Philosophical Logic*, Reidel, 1970.

[7] W. McCune. OTTER 2.0 Users Guide. Tech. Report ANL-90/9, Argonne National Laboratory, Argonne, IL, March 1990.

[8] W. McCune. Single axioms for the left group and right group calculi. Preprint MCS-P219-0391, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1991. To appear in *Notre Dame J. Formal Logic*.

[9] W. McCune. Automated discovery of new axiomatizations of the left group and right group calculi. *Journal of Automated Reasoning*, 9(1):1–24, 1992.

[10] C. A. Meredith. Single axioms for the systems (C,N), (C,0), and (A,N) of the two-valued propositional calculus. *Journal of Computing Systems*, 1:155–164, 1953.

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

[12] D. Meredith. In memoriam Carew Arthur Meredith (1904–1976). *Notre Dame Journal of Formal Logic*, 18:513–516, 1977.

[13] J. G. Peterson. Shortest single axioms for the classical equivalential calculus. *Notre Dame Journal of Formal Logic*, 17(2):267–271, 1976.

[14] J. G. Peterson. The possible shortest single axioms for EC-tautologies. Report 105, Dept. of Mathematics, University of Auckland, 1977.

[15] F. Pfenning. Single axioms in the implicational propositional calculus. In E. Lusk and R. Overbeek, editors, *Proceedings of the 9th International Conference on Automated Deduction, Lecture Notes in Computer Science, Vol. 310*, pages 710–713, New York, 1988. Springer-Verlag.

[16] J. Łukasiewicz. *Elements of Mathematical Logic*. Pergamon Press, 1963. English translation of the second edition (1958) of *Elementy logiki matematycznej*, PWN, Warsaw.

[17] J. Łukasiewicz. *Selected Works*. North-Holland, 1970. Edited by L. Borkowski.

[18] L. Wos. Meeting the challenge of fifty years of logic. *Journal of Automated Reasoning*, 6(2):213–232, 1990.

[19] L. Wos. Automated reasoning and Bledsoe's dream for the field. In R. S. Boyer, editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, chapter 15, pages 297–342. Kluwer Academic Publishers, 1991.

[20] L. Wos and W. McCune. The application of automated reasoning to proof translation and to finding proofs with specified properties: A case study in many-valued sentential calculus. Tech. Report ANL-91/19, Argonne National Laboratory, Argonne, IL, 1991.

[21] L. Wos, S. Winker, W. McCune, R. Overbeek, E. Lusk, R. Stevens, and R. Butler. Automated reasoning contributes to mathematics and logic. In M. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction, Lecture Notes in Artificial Intelligence, Vol. 449*, pages 485–499, New York, July 1990. Springer-Verlag.

[22] L. Wos, S. Winker, B. Smith, R. Veroff, and L. Henschen. A new use of an automated reasoning assistant: Open questions in equivalential calculus and the study of infinite domains. *Artificial Intelligence*, 22:303–356, 1984.