Journal of Automated Reasoning **6**: 213–232, 1990.

Problem Corner:

Meeting the Challenge of Fifty Years of Logic*

LARRY WOS
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439–4801, U.S.A.

(Received: 22 January 1990)

Abstract. In this article, we tell a story of good fortune. The good fortune concerns the discovery of a systematic approach to compress 50 years of excellent research in logic into a single day's use of an automated reasoning program. The discovery resulted from a colleague's experiment with a new representation and a new use of the weighting strategy. The experiment focused on an attempt – which I knew would fail – to prove one of the benchmark theorems that had eluded us for years. Fortunately, I was wrong; my colleague's attempt was successful, and a proof was found. The proof led to proving in one day 13 theorems, theorems that resulted from 50 years of excellent research in logic. We present these theorems as intriguing problems to test the power of a reasoning program or to evaluate the effectiveness of a new idea. In addition to the challenge problems, we discuss a possible approach to finding short proofs and the results achieved with it

Key words. Equivalential calculus, theorem proving.

1. Setting the Stage

In the early 1920s, the eminent logician Lukasiewicz began a study of equivalence in which Tarski also took an interest. By the end of that decade, other logicians had joined the investigation, with the aim of finding axioms for the field that grew out of that study, a field of logic (discussed in some detail almost immediately) known as equivalential calculus. By 1933, they had succeeded; they had found nine formulas, each of which is sufficiently powerful that it alone provides a complete axiomatization. Of the nine, eight consist of 15 symbols, and one (found by Lukasiewicz in 1933) consists of 11. Lukasiewicz also proved that no shorter formula can serve as a single axiom for this area of logic [2].

EQUIVALENTIAL CALCULUS, AN INTUITIVE DESCRIPTION

Before telling the rest of the story (in Section 2) and before presenting the theorems we offer (in Section 5) as challenges for automated reasoning programs and for tests of new ideas, let us discuss equivalential calculus in an intuitive manner. Indeed, let us provide essentially all that is needed for one to feel at home in this area of logic.

*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.

The elements to be studied are formulas that one can produce with the two-place function e (for equivalent) and the variables $x, y, z, \ldots$. Among such formulas, we have

(1) $e(x, x)$,
(2) $e(e(x, y), e(y, x))$,

and

(3) $e(e(x, y), e(e(y, z), e(x, z)))$.

It is no coincidence that we selected these three formulas; they were chosen because they will remind one, respectively, of reflexivity, symmetry, and transitivity, properties naturally associated with 'equivalence'. In fact, these three formulas taken together provide a complete axiomatization for equivalential calculus, for which in Section 3 we supply proofs obtained with our newest program OTTER [4]. If one prefers a smaller axiom set, although hardly intuitive, one discovers that there exist 13 formulas (given in this article) each of which is sufficiently powerful by itself to provide a complete axiomatization for the calculus; each of the 13 consists of 11 symbols, excluding commas and parentheses, and no shorter formula by itself axiomatizes the calculus. As it turns out – although not obviously, but easily proved – the first of the three given formulas (reflexivity) can be proved from the other two; we shall give that proof in Theorem 1 of Section 3. (We were unprepared for the fact that reflexivity is a dependent axiom, for such a fact is for us counterintuitive. Indeed, for equality, reflexivity cannot be proved from symmetry and transitivity. As my colleague R. Veroff observes, the formula thought of as transitivity in equivalential calculus is far more powerful than is its counterpart in equality, for the former acts like an if and only if statement, in contrast to the latter which has the form of a one-way implication.) Conversely, these three formulas can each be proved as a theorem of the calculus – not to be confused with a theorem about the calculus, which is the main study in this article – if one starts with any of the 13 shortest single axioms. For this study, theorems (of the calculus) are those formulas in which variables occur exactly twice. An immediate question to ask focuses on what means of proof is available for making deductions in equivalential calculus.

Indeed, one way in which areas of logic often differ from areas of mathematics concerns the use of a specific inference rule. Equivalential calculus is no exception, for it can be studied with the inference rule called *condensed detachment*. Condensed detachment considers two formulas, $e(A, B)$ and C, and, if C unifies with A, yields the formula D, where D is obtained by applying to B the unifier of C and A. For example, if we apply condensed detachment to

$e(x, e(x, e(y, y)))$ and $e(z, z)$

with the second formula playing the role of C, we obtain

$e(e(z, z), e(y, y))$.

If we reverse the roles of the two formulas and apply condensed detachment, we obtain a copy of the first formula. To gain a fuller appreciation of the intricacy of using condensed detachment, one might by hand attempt to produce the conclusion obtainable from applying the inference rule to two copies of

(4) $e(e(e(e(x, e(y, z)), e(z, u)), u)$.

If one succeeds in the suggested attempt, one finds that the conclusion is simply the appropriate instance of the second occurrence of the variable u. One can easily see why this is so by considering the clause

$\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$

which one can use to enable a reasoning program to apply condensed detachment. In particular, the unification of the clause equivalent of (4) with the first literal of the given three-literal clause causes the second occurrence of u to become the argument of the positive literal. To obtain the actual conclusion, one completes a hyperresolution step by unifying (a second copy of) the clause equivalent of (4) with the instantiated second literal of the three-literal clause to obtain $e(x, x)$. Obviously, unification is not a radically new idea to the logicians who have studied equivalential calculus.

The property just witnessed – that of deducing a short formula from two rather longer ones – and the propensity for deducing a formula longer than either of its two parents clearly plays havoc with an attempt to design an approach for showing that some given set of formulas can serve as a complete axiomatization of equivalential calculus. Nevertheless, as we shall discuss in Section 2, it can be done – at least for each of the 13 formulas presented in Section 5. In Theorem 2 of Section 3, we shall select one of those formulas and give a proof obtained with our newest program OTTER that the selected formula can serve as a complete axiomatization. In Theorem 1 of Section 3, as promised, we shall prove that reflexivity can be deduced from the set consisting of symmetry and transitivity. However, before turning to those proofs and to others given later, let us complete the stage setting by telling the rest of the story leading to the challenge presented here.

2. A Story of Good Fortune

By the late 1970s, Meredith [5], Kalman [1], Peterson [6, 7], and other logicians succeeded in finding (in addition to the formula found by Lukasiewicz in 1933 [2]) 10 additional formulas of length 11, each of which can serve as a single axiom. At that point, seven formulas remained – of the 630 possible candidates – of length 11 that had resisted classification. (From here on, because the theorems we do not consider are obtainable by substituting terms for variables, it is sufficient to be concerned only with formulas in which variables that occur in them occur exactly twice.) A most believable conjecture existed that all of the shortest single axioms had indeed been found. In other words, the conjecture asserted that each of the seven unclassified formulas was too weak to serve by itself as a complete axiomatization of equivalential calculus.

At Kalman's suggestion – no doubt motivated by his success with his own theorem-proving program – Wos and Winker took an interest in the study. Using the automated reasoning program AURA [8] designed and implemented by researchers at Argonne National Laboratory, Wos and Winker devised a method to study the seven remaining unclassified formulas, each consisting of 11 symbols. The method [10] relies on the use of schemata to attempt to obtain a complete characterization of all deducible theorems. (For this study, the theorems of the calculus are precisely those formulas in which variables occur twice. Rather than these theorems, the focus here is on theorems about the calculus, for example, the assertion that the formula XHK which is given shortly is a shortest single axiom.) With the method relying on the use of schemata, Wos and Winker showed that five of the seven formulas are indeed each too weak to serve as a single axiom, a fact that is consistent with the conjecture made by the various logicians. However, rather than showing that the other two formulas (known as XHK and XHN, given shortly) are each inadequate for serving as a single axiom, Winker [personal communication] – with many weeks of study and many runs with AURA – refuted the conjecture by discovering that each is in fact a shortest single axiom for the calculus.

The Argonne study led to the establishment of two benchmark problems for the field of automated reasoning. The object of the benchmark problems is to have an automated reasoning program prove, in single and separate runs, that each of XHK and XHN provides a complete axiomatization for equivalential calculus.

(XHK) $e(x, e(e(y, z), e(e(x, z), y)))$,
(XHN) $e(x, e(e(y, z), e(e(z, x), y)))$.

The emphasis on single runs is in contrast to the many runs coupled with the excellent guidance provided by Winker when he succeeded in the autumn of 1980.

As an indication of the difficulty offered by the two benchmark theorems, Winker's 83-step proof for XHK relies on the use of a formula of length 71, and his 159-step proof for XHN relies on the use of a formula of length 103. Since there exist more than 100 000 000 000 expressions consisting of 27 symbols – expressions that are called theorems, where each variable in an expression occurs exactly twice – one immediately sees why it might be extremely difficult for a program to prove, unaided and in a single run, either of the theorems that focus on the axioms XHK and XHN.

Indeed, for more than eight years, all attempts to obtain a proof of either theorem in a single run with one of the Argonne programs failed, and failed under a variety of attacks with different strategies and diverse procedures. From the viewpoint of automated reasoning, it appeared that, without something like Winker's insight and study of the results of many computer runs, the two theorems were simply too tough for an unaided reasoning program to prove.

At least, that was our view in the autumn of 1980 when Winker – with great assistance from AURA – produced the two proofs, and that was still our view in

February of 1989. In fact, were it not for one fortuitous occurrence, we would still hold that view; we would still predict that to obtain – with an automated reasoning program – a proof that XHK or that XHN is a single axiom for equivalential calculus would require substantial intervention, guidance, and assistance from some bright and insightful researcher.

The fortuitous occurrence concerns the conjunction of three forces, namely, (a) the study of a particular approach to applying the weighting strategy [3, 9, 11], (b) the use of a new representation (which turned out to be irrelevant to obtaining the results), and (c) access to the recently completed automated reasoning program OTTER [4]. (The full story will be presented in a future paper, 'A measure of success for automated reasoning', featuring the history of the successful study of XHK and XHN.) The use of weighting turned out to be the key to obtaining the results.

The weighting strategy provides a means for assigning priorities to clauses to enable a program to choose where next to focus its attention, and also provides a means for deciding that a conclusion is too complex to be of interest. At the simplest end of the spectrum, the priorities (weights) can be computed solely in terms of the number of symbols present. In contrast, one can use weights that emphasize certain chosen subexpressions to be the sole basis of priority assignment. For example, one can have a program prefer expressions in which the function e occurs three times in succession, or prefer some specific formula, or base its preference on a chosen argument of a formula. The conclusions that are kept are placed on a list from which our program OTTER chooses for the focus of attention, choosing the clause with the smallest weight. One can also use weighting to cause a program to discard a specific formula if ever encountered or discard formulas whose weight exceeds some given input value.

McCune decided to have OTTER prefer to focus on formulas with a short tail, where the tail of a formula is the second argument of the leftmost occurrence of the function e. For example, in the formulas

(PYM) $e(e(e(x, e(y, z)), y), e(z, x))$

and

(XGK) $e(x, e(e(y, e(z, x)), e(z, y)))$,

the first formula has as its tail $e(z, x)$, and the second has all but the leading x as a tail. McCune's reason for emphasizing the role of the tail of a formula rests with the fact that – as we learned in Section 1 when we focused on the inference rule condensed detachment that can be used in equivalential calculus – the conclusions yielded by applying that inference rule to a pair of formulas are obtained from the appropriate instantiation of the tail of one of the formulas.

OTTER, using the new approach to weighting and the new representation, won the first round. In a single run of approximately 8 CPU minutes on a Sun 3/60 workstation, OTTER found a proof that XHK is a single axiom for equivalential calculus. In contrast to the 83-step proof found by Winker with AURA (where the input clauses do not contribute to the count), OTTER's proof consists of 40 steps. Where the Winker proof requires the use of formulas containing as many as 71 symbols, the OTTER proof requires formulas containing no more than 47.

Correctly recognizing the importance of his success, McCune next asked OTTER to again use his approach in an attempt to prove that XHN is also a single axiom. Unfortunately – if we keep in mind the precise nature of the objective – in the strictest sense, OTTER did not succeed. Rather than finding a proof in a single run, OTTER required two runs, the first designed to accumulate some long formulas that could be used by the second under the condition that the second would not keep any additional long formulas it found.

Even though McCune was forced to slightly broaden the rules to permit two runs – because OTTER does not yet offer dynamic weighting adjustment – we still give the second round to OTTER. After all, the program did find a proof for XHN in just over 90 CPU minutes on an Encore Multimax using an NS32332 processor, and a proof quite unlike Winker's. (The switch to the Encore was caused by the need for additional memory.) The OTTER proof consists of 84 steps; the Winker/AURA proof consists of 159 steps. The OTTER proof requires formulas no longer than 47 symbols in length; the Winker/AURA proof relies on the use of a formula of length 103.

One of the two brenchmark problems had fallen – the other almost had – and fallen much sooner than we would have predicted. In addition, OTTER had found shorter and simpler proofs. The fact that a program finds proofs that are both shorter and simpler than those found earlier does not imply that the CPU time to obtain the newer proofs must have been less than that for the longer and more complex ones, as one can see from the examples we give in Section 4. Indeed, were one to predict a sharp increase in effectiveness – finding the proofs in 8 and 90 CPU minutes, respectively – merely because the new proofs are shorter and simpler, then one might find the results of various experiments rather startling. Where the level of an input clause is 0, the level of a clause is one greater than the levels of the immediate parents of that clause, and the level of a proof is the maximum of the levels of all of the clauses in the proof, the explanation rests with the fact that the number of steps in a proof and the complexity of those steps do not tell the full story, for the level of a proof also materially affects the CPU time that may be required to obtain it. A study of the interplay of these three aspects of proof should prove most intriguing and challenging.

Finding proofs far shorter and far simpler than Winker's does not detract from his achievement in any way. After all, we had the advantage of knowing that each of XHK and XHN is an axiom; in contrast, Winker was faced with the prospect that the conjecture of the logicians was probably correct and, therefore, that each of the formulas XHK and XHN was too weak to be proved an axiom. Of substantial importance, in no way did McCune use either of the Winker/AURA proofs. On the other hand, from a broader perspective, his success is another example of what the group theorist Reinhold Baer once said: "It is far easier to prove a theorem when one knows that a proof exists than when one is uncertain." The full story of how Winker succeeded must wait for the promised long paper.

By chance – in the sense that I considered McCune's experiment with XHK mainly to be a test of ideas, fully expecting it to fail – I found that an automated reasoning program could obtain unaided proofs of the corresponding two benchmark theorems. I had expected failure because all of the attempts made over the preceding eight years had been unsuccessful. Nevertheless, McCune's decision to have OTTER seek a proof was an excellent one, for – among other reasons – one occasionally makes a useful discovery even when the immediate goal is not reached. Since, in this case, the goal was reached, we were ready to attempt to meet the challenge of fifty years of excellent research in logic.

THE CHALLENGE

The challenge for us – and the one we offer in Section 5 for those researchers who enjoy challenges and who wish to test their own programs and their own ideas for adding to the power of automated reasoning programs – is to prove the 13 theorems accumulated over more than 50 years, beginning with the contributions made in the early 1930s. These theorems respectively establish that each of 13 formulas is so powerful that it can serve by itself as a complete axiomatization for equivalential calculus. (In Section 1, we provided an intuitive understanding of that calculus, and in Section 5 we give the thirteen theorems to be proved.)

By accepting the challenge, we decided that we must answer the following specific questions.

(1) How many of the 13 theorems could OTTER prove without substantial guidance of a researcher?
(2) How long in CPU time and real time would it take?
(3) How much memory would be required?
(4) How many runs must we make?
(5) Perhaps most important, how much must we modify the approach we used to prove that each of XHK and XHN is a single axiom?

Here is what occurred.

In a single day, OTTER succeeded in finding proofs of all 13 theorems that establish that each of the appropriate 13 formulas of length 11 is a (shortest) single axiom. Further, the method of attack – although it is a broadening of the approach McCune used for XHK and XHN – is systematic and general, not tuned to each of the corresponding 13 theorems. Specifically, the approach consists of using McCune's weighting of formulas to prefer those with shorter tails, and then, if not successful in 15 CPU minutes on a Sun 3/60 workstation, using a weighting strategy based on symbol count, and finally, if that also fails in 15 CPU minutes, then using weighting to prefer formulas with short tails or short heads. In other words, except for the use of a general weighting strategy, the 13 proofs were obtained by OTTER without substantial intervention, guidance, and assistance from a researcher.

To produce the 13 proofs required approximately 320 CPU minutes of the computer's time – mostly on a Sun 3/60 workstation, but partly on an Encore Multimax using an NS32332 processor – spread over 12 hours of our time. The number of steps in the 13 proofs is 403. The required memory varies from 0.1 to 20 megabytes. Almost 5 000 000 clauses were generated during the successful searches, and we were forced to abandon as useless nine of the 22 runs we made with OTTER.

If we quickly revisit the results of our recent study of equivalential calculus, we cannot help but sense a sharp increase in the value of using an automated reasoning program. Indeed, in just over eight years after Winker and AURA's impressive completion of the search for shortest single axioms – a search that began in the late 1920s – we showed that an automated reasoning program could start and finish the search for the corresponding 13 proofs, and in a single day. In no way did our effort rely on knowledge of the structure of the existing proofs. In fact, we had seen only two such proofs, Winker's proofs for XHK and XHN.

On the other hand, we did have the advantage over Winker and over the logicians who had studied equivalential calculus – and clearly an important advantage – of knowing which 13 formulas, from among the 630 to be considered, have the property of being so powerful that each provides a complete axiomatization of equivalential calculus. Unfortunately, one cannot simply try to prove that a randomly selected formula – from among the 630 that are too weak – is a single axiom, for such an action will result in the deduction of an ever-growing set of conclusions with virtually no clue that failure awaits one.

Incidentally, during the writing of this article, we succeeded in obtaining a proof in a single run of the XHN theorem, but we did intervene substantially by choosing a somewhat complex weighting strategy strongly influenced by McCune's 84-step proof just mentioned. However – rather than reproducing the 84-step proof, as one might predict – we instead found a 65-step proof; more details are given when we discuss in Section 4 methods for obtaining shorter proofs. In addition, rather than requiring the use of a formula of length 47, the new proof relies on formulas no longer than 39 symbols in length. Without access to McCune's success in studying both XHK and XHN, we would not have known where to begin.

As further evidence of the depth of this study of shortest single axioms, we note that Lukasiewicz appeared to believe in the late 1940s that only one such axiom existed, rather than the thirteen that do. We say this because he referred to 'the shortest single axiom' [2]. In addition, Meredith [5] in 1963 incorrectly claimed that the formula XGJ can serve as a complete axiomatization for equivalential calculus.

(XGJ) $e(x, e(e(y, e(z, x)), e(y, z)))$

Therefore, if one has less success than desired when one tackles the theorems we present in Section 5, one might keep in mind the difficulties experienced by eminent logicians during the comparable study.

3. Four Theorems Proved by OTTER

THEOREM 1. *In equivalential calculus, the axiom of reflexivity is dependent on the axioms of symmetry and transitivity.*

Proof. Let us assume by way of contradiction that reflexivity cannot be deduced from symmetry and transitivity. We therefore have four input clauses – the clause to capture condensed detachment, the clause to deny the conclusion of the theorem, and the clauses for symmetry and transitivity. Using hyperresolution as the inference rule, we obtain the following proof from OTTER in which the number of a clause reflects the order in which it was input or deduced. In the proofs below, the input clauses occur above the horizontal line; derived clauses below. The only clauses used are those that appear in the proof. The clause numbers give some indication of the density of the proof within the set of clauses that are derived and not subsumed. The triple of numbers in brackets gives in order the clause driving the inference, the nucleus of the step, and the other clause used to complete the deduction.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
2 $\neg P(e(a, a))$.
3 $P(e(e(x, y), e(y, x)))$.
4 $P(e(e(x, y), e(e(y, z), e(x, z))))$.
- - - - - - - - - - - - - - -
5 [hyper, 4, 1, 3]$P(e(e(e(x, y), e(z, y)), e(z, x)))$.
9 [hyper, 5, 1, 3]$P(e(x, x))$.

Clause (9) contradicts clause (2), and the proof is complete.

THEOREM 2. *The formula* YQL

(YQL) $e(e(x, y), e(e(z, y), e(x, z)))$

can serve as a complete axiomatization for equivalential calculus.

Proof. We assume by way of contradiction that YQL is not a single axiom for the calculus. This assumption asserts that none of the known single axioms for equivalential calculus is implied by YQL; indeed, if any formula that is a single axiom for the calculus were implied by YQL, then YQL would itself be a single axiom also. Therefore, we assume that none of the following eight formulas, proved by various logicians to be single axioms, is implied by YQL.

$e(e(e(x, e(y, z)), e(e(z, u), u)), e(x, y))$
$e(e(e(e(x, y), z), u), e(u, e(x, e(y, z))))$
$e(e(x, e(y, z)), e(e(y, e(u, z)), e(u, x)))$
$e(e(x, e(y, z)), e(e(y, e(z, u)), e(u, x)))$
$e(e(u, e(x, e(y, z))), e(e(x, y), e(z, u)))$
$e(e(x, e(y, z)), e(e(x, e(z, u)), e(u, y)))$
$e(e(x, e(y, z)), e(e(x, e(u, z)), e(u, y)))$
$e(e(x, e(y, z)), e(e(x, e(z, u)), e(y, u)))$

The following proof that YQL implies the last of the eight given single axioms was found by OTTER.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
2 $\neg P(e(e(a, e(b, c)), e(e(a, e(c, d)), e(b, d))))$.
3 $P(e(e(x, y), e(e(z, y), e(x, z))))$.
- - - - - - - - - - - - - - -
4 [hyper, 3, 1, 3]$P(e(e(x, e(e(y, z), e(u, y))), e(e(u, z), x)))$.
6 [hyper, 4, 1, 3]$P(e(e(x, y), e(x, y)))$.
7 [hyper, 6, 1, 3]$P(e(e(x, e(y, z)), e(e(y, z), x)))$.
12 [hyper, 7, 1, 3]$P(e(e(e(x, y), e(z, x)), e(z, y)))$.
28 [hyper, 12, 1, 3]$P(e(e(x, e(y, z)), e(e(e(u, z), e(y, u)), x)))$.
22336 [hyper, 28, 1, 4]$P(e(e(x, e(y, z)), e(e(x, e(z, u)), e(y, u))))$.

Clause (22336) contradicts clause (2), and the proof is complete.

For our third example of OTTER's usefulness and power, let us prove that the set of formulas consisting of symmetry and transitivity provides a complete axiomatization for equivalential calculus. Our method of proof is similar to that for Theorem 2; namely, we prove that the use of the clauses for symmetry and transitivity leads to a deduction of a formula that itself is a complete axiomatization. We give two proofs obtained with OTTER, a short one and a long one. They were found by using OTTER's capacity to find more than one proof in a single run. We included among the input clauses the negations of the thirteen shortest single axioms and the negations of the eight axioms of length 15 given earlier.

THEOREM 3. *The axioms of symmetry and transitivity provide a complete axiomatization for equivalential calculus.*

Proof. We proceed as in Theorem 2 by assuming by way of contradiction that the theorem is false. The first proof shows that the single axiom YQL is deducible, and the second that the single axom YRO is deducible.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
2 $\neg P(e(e(a, b), e(e(c, b), e(a, c))))$.
23 $P(e(e(x, y), e(y, x)))$.
24 $P(e(e(x, y), e(e(y, z), e(x, z))))$.
- - - - - - - - - - - - - - -
25 [hyper, 24, 1, 24]$P(e(e(e(e(x, y), e(z, y)), u), e(e(z, x), u)))$.
27 [hyper, 24, 1, 23]$P(e(e(e(x, y), z), e(e(y, x), z)))$.
9792 [hyper, 25, 1, 27]$P(e(e(x, y), e(e(z, y), e(x, z))))$.

Clause (9792) contradicts clause (2), and the first proof is complete.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
9 $\neg P(e(e(a, b), e(c, e(e(c, b), a))))$.
23 $P(e(e(x, y), e(y, x)))$.
24 $P(e(e(x, y), e(e(y, z), e(x, z))))$.

25 [hyper, 24, 1, 24]$P(e(e(e(e(x, y), e(z, y)), u), e(e(z, x), u)))$.
26 [hyper, 24, 1, 23]$P(e(e(e(x, y), e(z, y)), e(z, x)))$.
27 [hyper, 24, 1, 23]$P(e(e(e(x, y), z), e(e(y, x), z)))$.
28 [hyper, 26, 1, 26]$P(e(x, e(e(x, y), y)))$.
30 [hyper, 26, 1, 24]$P(e(e(e(x, y), y), x))$.
33 [hyper, 28, 1, 24]$P(e(e(e(e(x, y), y), z), e(x, z)))$.
39 [hyper, 30, 1, 24]$P(e(e(x, y), e(e(e(x, z), z), y)))$.
58 [hyper, 27, 1, 27]$P(e(e(x, e(y, z)), e(e(z, y), x)))$.
124 [hyper, 33, 1, 27]$P(e(x, e(e(y, x), y)))$.
133 [hyper, 124, 1, 24]$P(e(e(e(e(x, y), x), z), e(y, z)))$.
215 [hyper, 39, 1, 27]$P(e(e(x, y), e(e(e(y, z), z), x)))$.
377 [hyper, 58, 1, 58]$P(e(e(x, e(y, z)), e(x, e(z, y))))$.
1342 [hyper, 133, 1, 27]$P(e(e(x, e(e(y, z), y)), e(z, x)))$.
2278 [hyper, 215, 1, 58]$P(e(e(x, e(e(y, z), z)), e(x, y)))$.
3059 [hyper, 377, 1, 133]$P(e(e(e(e(x, y), x), z), e(z, y)))$.
9725 [hyper, 25, 1, 3059]$P(e(e(e(x, y), e(y, z)), e(x, z)))$.
9767 [hyper, 25, 1, 1342]$P(e(e(e(x, y), z), e(y, e(z, x))))$.
9812 [hyper, 9725, 1, 2278]$P(e(e(e(e(x, y), z), e(z, y)), x))$.
15190 [hyper, 9812, 1, 9767]$P(e(e(x, y), e(z, e(e(z, y), x))))$.

Clause (15190) contradicts clause (9), and the second proof is complete.

We close this section by turning our attention to one of the earliest axiomatizations of equivalential calculus. In 1929, Lesniewski [2] presented an axiomatization of equivalential calculus consisting of the formulas

$$e(e(e(x, y), e(z, x)), e(y, z))$$

and

$$e(e(x, e(y, z)), e(e(x, y), z)),$$

the first of the two being a form of transitivity and the second the commuted form of associativity. The proof that this set axiomatizes the calculus rests with an appeal to natural language. In 1932, Wajsberg and Lesniewski [2] presented four simpler sets of axioms, one of which consists of symmetry and associativity.

$$e(e(x, y), e(y, x)),$$

$$e(e(e(x, y), z), e(x, e(y, z)))$$

Let us immediately give a proof, obtained with OTTER, that one can derive one of the shortest single axioms, YQL, from symmetry and associativity, and give in Section 4 a proof that this pair of formulas implies the initial set of axioms given by Lesniewski. Let us also give OTTER's proofs of YRM and transitivity, using symmetry and associativity, respectively.

Regarding the three proofs we are about to give, a comparison of the first two with their correspondents given earlier – where the axioms are symmetry and transitivity – suggests that establishing that symmetry and transitivity axiomatize equivalential calculus is somewhat easier than establishing that symmetry and associativity do. We find that evaluation rather pleasing, if accurate, for the set of formulas consisting of reflexivity, symmetry, and transitivity should – in some sense of naturalness – be a good choice for that area of logic focusing on equivalence. Of course, as we proved earlier, the axiom of reflexivity is derivable from symmetry and transitivity. As for the third proof we shall give, its appeal rests with the fact that symmetry and transitivity – with reflexivity – are the most obvious choices for an axiomatization of this calculus. Therefore, one might indeed prefer to establish that symmetry and associativity provide a complete axiomatization by deriving transitivity, rather than deriving some shortest single axiom.

If one examines all three proofs together, one finds that all derived steps consist of exactly 11 symbols – a fact that, although we cannot explain our reaction, we find intriguing. In contrast, if one replaces associativity with transitivity and then prevents OTTER from keeping any formula of length greater than 11 – but allows it to use formulas of length less than or equal to 11 – one cannot derive any of the 13 shortest single axioms from symmetry and transitivity alone. OTTER derives $e(x, x)$, 14 of the 15 formulas of length seven – skipping only $e(e(x, y), e(x, y))$ – and 432 of the 630 formulas of length 11 (in which each variable occurs exactly twice). At that point, the set of support becomes empty, which in effect asserts that nothing more is accessible and all 13 shortest single axioms are out of reach when one prevents the use of all formulas of length greater than 11. Here are the three promised proofs.

THEOREM 4. *The axioms of symmetry and associativity provide a complete axiomatization for equivalential calculus.*

Proof. We proceed as in Theorem 3 by assuming that the theorem is false. We therefore add to the input, in addition to clauses for symmetry and associativity, clauses that correspond to the negation of the thirteen shortest single axioms and the negation of transitivity. The clause for the negation of transitivity permits OTTER to seek to obtain the three promised proofs in a single run. We also include clauses corresponding to the negation of eight single axioms, each of length 15, because that is our standard approach in these studies. We extract the following three proofs of YQL, YRM, and transitivity, respectively.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
2 $\neg P(e(e(a, b), e(e(c, b), e(a, c))))$.
24 $P(e(e(x, y), e(y, x)))$.
25 $P(e(e(e(x, y), z), e(x, e(y, z))))$.
\- - - - - - - - - - -
26 [hyper, 25, 1, 25]$P(e(e(x, y), e(z, e(x, e(y, z)))))$.
27 [hyper, 25, 1, 24]$P(e(e(x, e(y, z)), e(e(x, y), z)))$.
28 [hyper, 26, 1, 25]$P(e(x, e(y, e(z, e(x, e(y, z))))))$.
29 [hyper, 26, 1, 24]$P(e(e(x, e(y, e(z, x))), e(y, z)))$.
30 [hyper, 26, 1, 24]$P(e(x, e(e(y, z), e(e(z, y), x))))$.

34 [hyper, 28, 1, 24]$P(e(e(x, e(y, e(z, e(x, y)))), z))$.
39 [hyper, 30, 1, 27]$P(e(e(x, e(y, z)), e(e(z, y), x)))$.
48 [hyper, 34, 1, 25]$P(e(x, e(e(y, e(z, e(x, y))), z)))$.
63 [hyper, 39, 1, 39]$P(e(e(x, e(y, z)), e(x, e(z, y))))$.
75 [hyper, 39, 1, 25]$P(e(e(e(x, y), z), e(e(z, x), y)))$.
155 [hyper, 48, 1, 29]$P(e(e(x, e(e(y, z), e(z, x))), y))$.
455 [hyper, 155, 1, 75]$P(e(e(x, y), e(e(x, z), e(z, y))))$.
650 [hyper, 455, 1, 63]$P(e(e(x, y), e(e(z, y), e(x, z))))$.

Clause (650) contradicts clause (2), and the proof of YQL is complete.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
9 $\neg P(e(e(a, b), e(c, e(e(c, b), a))))$.
24 $P(e(e(x, y), e(y, x)))$.
25 $P(e(e(e(x, y), z), e(x, e(y, z))))$.
- - - - - - - - - - -
26 [hyper, 25, 1, 25]$P(e(e(x, y), e(z, e(x, e(y, z)))))$.
27 [hyper, 25, 1, 24]$P(e(e(x, e(y, z)), e(e(x, y), z)))$.
30 [hyper, 26, 1, 24]$P(e(x, e(e(y, z), e(e(z, y), x))))$.
33 [hyper, 27, 1, 25]$P(e(e(e(e(x, y), z), x), e(y, z)))$.
39 [hyper, 30, 1, 27]$P(e(e(x, e(y, z)), e(e(z, y), x)))$.
63 [hyper, 39, 1, 39]$P(e(e(x, e(y, z)), e(x, e(z, y))))$.
68 [hyper, 39, 1, 33]$P(e(e(x, y), e(e(e(z, y), x), z)))$.
220 [hyper, 68, 1, 63]$P(e(e(x, y), e(z, e(e(z, y), x))))$.

Clause (220) contradicts clause (9), and the proof of YRM is complete.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
23 $\neg P(e(e(a, b), e(e(b, c), e(a, c))))$.
24 $P(e(e(x, y), e(y, x)))$.
25 $P(e(e(e(x, y), z), e(x, e(y, z))))$.
- - - - - - - - - - -
26 [hyper, 25, 1, 25]$P(e(e(x, y), e(z, e(x, e(y, z)))))$.
27 [hyper, 25, 1, 24]$P(e(e(x, e(y, z)), e(e(x, y), z)))$.
28 [hyper, 26, 1, 25]$P(e(x, e(y, e(z, e(x, e(y, z))))))$.
30 [hyper, 26, 1, 24]$P(e(x, e(e(y, z), e(e(z, y), x))))$.
33 [hyper, 27, 1, 25]$P(e(e(e(e(x, y), z), x), e(y, z)))$.
34 [hyper, 28, 1, 24]$P(e(e(x, e(y, e(z, e(x, y)))), z))$.
39 [hyper, 30, 1, 27]$P(e(e(x, e(y, z)), e(e(z, y), x)))$.
47 [hyper, 34, 1, 33]$P(e(x, e(y, e(z, e(e(z, x), y)))))$.
74 [hyper, 39, 1, 26]$P(e(e(e(x, e(y, z)), z), e(x, y)))$.
75 [hyper, 39, 1, 25]$P(e(e(e(x, y), z), e(e(z, x), y)))$.
150 [hyper, 47, 1, 39]$P(e(e(e(x, e(e(x, y), z)), z), y))$.
453 [hyper, 150, 1, 74]$P(e(e(x, e(e(x, y), e(z, y))), z))$.
641 [hyper, 453, 1, 75]$P(e(e(x, y), e(e(y, z), e(x, z))))$.

Clause (641) contradicts clause (23), and the proof of transitivity is complete.

4. Seeking Shorter Proofs

In this section, we focus on one of the questions occasionally asked by mathematicians and logicians, that of finding a shorter proof than has been found so far. The frequency of such a question being posed appears to increase sharply when one is presented with a proof obtained with an automated reasoning program. No doubt the questioner, recognizing the arduousness of conducting the corresponding search by hand but encouraged by the success of a computer program, conjectures that some way must exist to effectively assign the task to a computer.

In principle the conjecture is correct, for such a way does exist. One can simply have one's reasoning program conduct a level-saturation run through level k, where k is the length of the shortest known proof. We recall that the level of a clause is one greater than the levels of the immediate parents of that clause, and the level of a proof is the maximum of the levels of all of the clauses in the proof. Unfortunately, for most studies, a level-saturation run is virtually untenable, for the size (the number of clauses) of succeeding levels grows very rapidly. Nevertheless, consideration of using an automated reasoning program in the attempt to find shorter proofs should not simply be rejected.

In fact, our entrance into the field of equivalential calculus was mostly prompted by a suggestion from the logician Kalman that we seek a proof that the formula XGK implies the formula PYO – each of which is a shortest single axiom – but attempt to find a proof shorter than the one he found [1]. As we reported in an earlier paper [10], we did succeed – actually, Brian Smith deserves full credit for this success. Smith, using the automated reasoning program AURA, found a 23-step proof [10]; Kalman's proof consists of 43 steps. (We very recently found a 10-step proof with OTTER using an approach based on level saturation; the result duplicates a proof first obtained by Marien using one of Stickel's programs.)

Despite this success – until now – we have had little to contribute when asked about how one might systematically use an automated reasoning program for seeking shorter proofs. In fact, although we provide some hints in this section about how one might proceed, we must point out that we have gained little understanding of the precise nature of the mechanisms we discuss, and we consider our discoveries to be mostly good fortune.

The change that has occurred, during the writing of this article, resulted simply from curiosity. We examined McCune's proof that XHN is a single axiom, noting that – after the deduction of $e(x, x)$ – almost all of the steps in the proof have a short tail *or* a short head. Obviously, his emphasis of short tails was an excellent choice, leading to the superior proofs for XHK and XHN. Therefore, we decided to go even further in that direction. In particular, with weighting, we instructed OTTER to emphasize the use of formulas whose head or tail was no longer than three symbols. Our goal was to obtain McCune's proof for XHN, but obtain it in a single run.

After 4 CPU hours on an Encore, the attempt was judged a failure and terminated. However, an examination of the output revealed a most unexpected find – a far shorter proof of $e(x, x)$. Where the McCune/OTTER proof takes 29 derived steps, the new proof takes 13; where the McCune/OTTER proof relies on a formula of 47 symbols, the new proof requires formulas containing no more than 39 symbols. The immediate question we asked focuses on using some weighting strategy that would produce in a single run a (new) proof of XHN, where the proof would begin by producing the 13-step deduction of $e(x, x)$. We had already changed the rules; we were still in pursuit of a single-run proof of XHN, but we were also after a proof shorter than that found by McCune.

The attempt failed, failed because OTTER used unwanted steps on its way to deducing $e(x, x)$, which made that proof longer than we intended. However, because weighting permits one to assign a weight to a clause – whose pattern of e's is unwanted – in such a way that clauses of the unwanted type are discarded, we were able to add to the weighting templates and avoid the unwanted clauses. Our attempt succeeded in reaching both goals. OTTER found a proof that XHN is a single axiom, and the proof is 65 steps in length as compared with McCune's proof of 84 steps. In addition, the shorter proof is less complex – the longest formula used in it consists of 39 symbols, compared to the use of formulas consisting of 47 symbols for the longer proof.

Other experiments that led to finding shorter proofs rest on instructing OTTER to block the use of formulas with weight greater than some chosen bound, and also block the use of formulas with weight less than another chosen bound. Since the latter is not directly available in OTTER, to achieve the desired restriction, we were forced to employ an indirect means – to add templates that purge certain types of formulas. We also succeeded in finding shorter proofs with an entirely different approach, that of interchanging the order of the two negative literals in the nucleus for condensed detachment. This approach was discovered purely by chance, for we were simply trying for a uniform set of proofs, each placing the role of the $e(x, y)$ literal first. The result was the discovery by OTTER of a proof for XHN consisting of 63 steps instead of 65. Other experiments are planned.

Summarizing, we have discovered by chance the beginning of an approach for systematically seeking shorter proofs with the assistance of an automated reasoning program. One aspect of the approach rests on choosing some type or types of formulas to be avoided, and then using weighting to purge unwanted clauses. The clauses to be avoided are typically present in a known proof, and the object is to prevent the program from finding that proof. The other aspect of the approach focuses on the use of various orderings of the literals in the nucleus or nuclei from which conclusions are drawn – drawn by using hyperresolution, UR-resolution, or some other inference rule. The object of this aspect is to perturb the search space enough to cause the program to concentrate on conclusions in sharply different sequences. Again, as in the results reported in earlier sections, we are indebted to McCune for providing the needed beginning.

We close this section with some promised observations and two proofs relevant to the relation between finding shorter proofs and the corresponding CPU time that is required. The experiments and proofs focus on proving that the combination of symmetry and associativity axiomatize equivalential calculus, where the proof depends on deducing the commuted form of associativity and an odd form of transitivity. In particular, we prove that the axioms first given by Lesniewski in 1929 are derivable from a second set given in 1932 [2]. Since the commuted form of associativity is easily derivable from symmetry and associativity, all that remains is the two proofs of Theorem 5 – two are needed to illustrate that shorter proofs can indeed take far longer to find.

THEOREM 5. *The axioms of symmetry and associativity completely axiomatize equivalential calculus; in particular, their use leads to the deduction of*

$$e(e(e(x, y), e(z, x)), e(y, z)),$$

which, with symmetry, forms a complete axiomatization of the calculus.

Proof. As usual, we assume by way of contradiction that the theorem is false, which explains the presence of the negative unit clause in the following two proofs.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
3 $\neg P(e(e(e(a, b), e(c, a)), e(b, c)))$.
4 $P(e(e(x, y), e(y, x)))$.
5 $P(e(e(e(x, y), z), e(x, e(y, z))))$.

\- - - - - - - - - - -

6 [hyper, 5, 1, 4]$P(e(e(x, e(y, z)), e(e(x, y), z)))$.
7 [hyper, 5, 1, 4]$P(e(x, e(y, e(y, x))))$.
9 [hyper, 7, 1, 4]$P(e(e(x, e(x, y)), y))$.
10 [hyper, 9, 1, 5]$P(e(x, e(e(x, y), y)))$.
11 [hyper, 9, 1, 7]$P(e(x, x))$.
12 [hyper, 11, 1, 7]$P(e(x, e(x, e(y, y))))$.
14 [hyper, 10, 1, 10]$P(e(e(e(x, e(e(x, y), y)), z), z))$.
18 [hyper, 10, 1, 7]$P(e(e(e(x, e(y, e(y, x))), z), z))$.
28 [hyper, 12, 1, 4]$P(e(e(x, e(y, y)), x))$.
38 [hyper, 28, 1, 5]$P(e(e(e(x, y), x), y))$.
41 [hyper, 38, 1, 4]$P(e(x, e(e(y, x), y)))$.
59 [hyper, 41, 1, 41]$P(e(e(x, e(y, e(e(z, y), z))), x))$.
67 [hyper, 41, 1, 18]$P(e(e(x, e(y, e(z, e(z, y)))), x))$.
69 [hyper, 41, 1, 14]$P(e(e(x, e(y, e(e(y, z), z))), x))$.
405 [hyper, 6, 1, 5]$P(e(e(e(e(x, y), z), x), e(y, z)))$.
431 [hyper, 405, 1, 69]$P(e(e(e(x, y), e(e(y, z), z)), x))$.
433 [hyper, 405, 1, 67]$P(e(e(e(x, y), e(z, e(z, y))), x))$.
440 [hyper, 405, 1, 59]$P(e(e(e(x, y), e(e(z, y), z)), x))$.
617 [hyper, 440, 1, 433]$P(e(e(e(x, e(y, e(y, z))), x), z))$.
619 [hyper, 440, 1, 431]$P(e(e(e(x, e(e(y, z), z)), x), y))$.
777 [hyper, 617, 1, 5]$P(e(e(x, e(y, e(y, z))), e(x, z)))$.
784 [hyper, 619, 1, 5]$P(e(e(x, e(e(y, z), z)), e(x, y)))$.

1472 [hyper, 777, 1, 405]$P(e(e(e(e(x, y), e(y, z)), x), z))$.
1536 [hyper, 1472, 1, 5]$P(e(e(e(x, y), e(y, z)), e(x, z)))$.
1637 [hyper, 784, 1, 405]$P(e(e(e(e(x, e(y, z)), z), x), y))$.
1707 [hyper, 1637, 1, 5]$P(e(e(e(x, e(y, z)), z), e(x, y)))$.
2229 [hyper, 1536, 1, 784]$P(e(e(e(e(x, y), z), e(z, y)), x))$.
2248 [hyper, 2229, 1, 1536]$P(e(e(e(e(e(x, y), z), y), x), z))$.
2498 [hyper, 1707, 1, 2248]$P(e(e(e(e(e(x, y), z), y), z), x))$.
2569 [hyper, 2498, 1, 1707]$P(e(e(e(e(x, y), e(z, x)), y), z))$.
2727 [hyper, 2569, 1, 5]$P(e(e(e(x, y), e(z, x)), e(y, z)))$.

Clause (2727) contradicts clause (3), and the first proof is complete.

1 $\neg P(e(x, y)) \mid \neg P(x) \mid P(y)$.
3 $\neg P(e(e(e(a, b), e(c, a)), e(b, c)))$.
4 $P(e(e(x, y), e(y, x)))$.
5 $P(e(e(e(x, y), z), e(x, e(y, z))))$.

\- - - - - - - - - - -

6 [hyper, 5, 1, 5]$P(e(e(x, y), e(z, e(x, e(y, z)))))$.
7 [hyper, 5, 1, 4]$P(e(e(x, e(y, z)), e(e(x, y), z)))$.
8 [hyper, 5, 1, 4]$P(e(x, e(y, e(y, x))))$.
12 [hyper, 8, 1, 4]$P(e(e(x, e(x, y)), y))$.
16 [hyper, 12, 1, 5]$P(e(x, e(e(x, y), y)))$.
127 [hyper, 6, 1, 4]$P(e(e(x, e(y, e(z, x))), e(y, z)))$.
138 [hyper, 6, 1, 16]$P(e(x, e(y, e(e(e(y, z), z), x))))$.
142 [hyper, 6, 1, 4]$P(e(x, e(e(y, z), e(e(z, y), x))))$.
159 [hyper, 7, 1, 5]$P(e(e(e(e(x, y), z), x), e(y, z)))$.
1421 [hyper, 138, 1, 7]$P(e(e(x, y), e(e(e(y, z), z), x)))$.
1493 [hyper, 142, 1, 7]$P(e(e(x, e(y, z)), e(e(z, y), x)))$.
1704 [hyper, 159, 1, 127]$P(e(x, e(y, e(z, e(e(y, z), x)))))$.
3118 [hyper, 1421, 1, 4]$P(e(e(e(e(x, y), y), z), e(z, x)))$.
3604 [hyper, 1493, 1, 127]$P(e(e(x, y), e(z, e(y, e(x, z)))))$.
4589 [hyper, 1704, 1, 7]$P(e(e(x, y), e(z, e(e(y, z), x))))$.
9613 [hyper, 3604, 1, 7]$P(e(e(e(x, y), z), e(y, e(x, z))))$.
10395 [hyper, 4589, 1, 3118]$P(e(e(x, e(e(y, x), e(z, y))), z))$.
17286 [hyper, 10395, 1, 9613]$P(e(e(e(x, y), e(z, x)), e(y, z)))$.

Clause (17286) contradicts clause (3), and the second proof is complete.

To obtain the second and shorter proof required approximately seven times as much CPU time as was required to obtain the first. Both proofs were obtained on a Sun 3/60 workstation.

5. Challenge Problems

The object of the challenge problems we offer is to prove with an automated reasoning program that each of the following 13 formulas is by itself a complete axiomatization of equivalential calculus.

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

For the clauses that are ordinarily needed to produce an unsatisfiable set, one might begin with the negations of the eight axioms of length 15 given in Section 3. In the promised long paper mentioned in Section 2, we shall include proofs – each obtained with OTTER – that each of the 13 formulas that are the object of the challenge problems is indeed a single axiom for equivalential calculus. In that paper, to produce a sound argument and avoid the circularity of simply proving that, A implies B implies C implies A, we present a tree of implications rooted in the first set of two axioms in turn proved (in 1929 by Lesniewski [2]) to axiomatize equivalential calculus, where Lesniewski's argument rests on the already-mentioned appeal to natural language.

For additional test problems, one might seek a proof with an automated reasoning program of any or all of the theorems proved in this article – perhaps a different proof from that given here. For the final test problem, one might try to obtain a proof with an automated reasoning program that symmetry and transitivity

$$e(e(x, y), e(y, x))$$

$$e(e(x, y), e(e(y, z), e(x, z)))$$

together imply associativity

$$e(e(e(x, y), z), e(x, e(y, z))).$$

This theorem provides the missing piece in the discussion of reflexivity, symmetry, transitivity, and associativity.

6. Conclusions

We have discovered – in part by chance, as so often occurs in science – a systematic approach that an automated reasoning program can use for proving that each of thirteen formulas is by itself a single axiom for all of equivalential calculus. The 13 proofs are of particular interest since no other formula from among the 630 of length 11 has sufficient power to axiomatize, by itself, this area of logic. The approach for obtaining these proofs and others like them relies heavily on the use of the weighting strategy, a strategy that permits the user to assign priorities to formulas and subformulas; those priorities are used to direct a reasoning program in its choice of where next to focus attention, and are also used by the program to discard unwanted conclusions.

To put into perspective the research reported here, we first note that proofs of the corresponding thirteen theorems were originally obtained with approximately 50 years of excellent research by logicians and other scientists in a period beginning in the late 1920s and ending in 1980. In contrast, OTTER was able to prove the 13 theorems in a single day of use, which – rather than in any way suggesting that all of the theorems are easy to prove – may mark a significant advance for automated reasoning. Although we did have the distinct advantage of knowing which 13 – of the possible 630 formulas – to study, in no way did our effort rely on or benefit from knowledge about the specific existing proofs. In fact, when we began the study, we had seen only two such proofs, the two impressive results Winker had found for the formulas known as XHK and XHN; neither proof was used in any way during the investigation. On the other hand, when we began our search for shorter proofs than those found by McCune in his use of OTTER, we did gain important insight about what to try from reading his proofs.

In addition to discovering a general approach to proving the 13 tightly coupled theorems, the proofs we found by using OTTER exhibit unexpected improvements. Specifically, although McCune's proofs – establishing that XHK and XHN are each a single axiom for equivalential calculus – are, respectively, far shorter and far less complex than those found by Winker in 1980, the proofs we then found exhibit a further reduction in length and complexity. Our success suggests that we may have discovered some mechanisms for finding shorter proofs. In particular, to seek a new proof, one can select a step of a known proof and assign such a high weight to the corresponding clause that the reasoning program is prevented from ever focusing on that clause; if, in addition, the assigned weight does not exceed the maximum weight permitted for retained conclusions, then all instances of the chosen clause will be purged with subsumption, preventing the program from using them in a proof. In contrast to this action which blocks the exploration of a given proof, one can encourage the exploration of a path that appears promising by taking one or more steps on that path and assigning such low weights to the corresponding clauses that the program will choose them as the focus of attention before any other retained conclusion.

As well as providing evidence of the progress occurring in automated reasoning and of the power of our newest program OTTER, the material presented in this article contains a number of challenge problems. These problems offer a wide spectrum of difficulty and can be used to test new programs and new approaches.

References

1. Kalman, J., 'A shortest single axiom for the classical equivalential calculus', *Notre Dame J. Formal Logic* **19**, 141–144 (1978).
2. Lukasiewicz, J., 'The equivalential calculus', in L. Borkowski (ed.), *Jan Lukasiewicz: Selected Works*, North-Holland, Amsterdam (1970).
3. McCharen, J., Overbeek, R., and Wos, L., 'Complexity and related enhancements for automated theorem-proving programs', *Computers and Mathematics with Applications* **2**, 1–16 (1976).
4. McCune, W., 'OTTER 2.0 users guide', Technical Report ANL-90-9, Argonne National Laboratory, Argonne, Illinois (1990).
5. Meredith, C. and Prior, B., 'Notes on the axiomatics of the propositional calculus', *Notre Dame J. Formal Logic* **4**, 171–187 (1963).
6. Peterson, J., 'Shortest single axioms for the equivalential calculus', *Notre Dame J. Formal Logic* **17**, 267–271 (1976).
7. Peterson, J., 'An automatic theorem prover for substitution and detachment systems', *Notre Dame J. Formal Logic* **19**, 119–122 (1978).
8. Smith, B., 'Reference manual for the environmental theorem prover, an incarnation of AURA', Technical Report ANL-88-2, Argonne National Laboratory, Argonne, Illinois (1988).
9. Wos, L., Overbeek, R., Lusk, E., and Boyle, J., *Automated Reasoning: Introduction and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey (1984).
10. Wos, L., Winker, S., Veroff, R., Smith, B., and Henschen, L., '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).
11. Wos, L., *Automated Reasoning: 33 Basic Research Problems*, Prentice-Hall, Englewood Cliffs, New Jersey (1987).