The Strategy of Cramming*

Larry Wos
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
e-mail: wos@mcs.anl.gov

Abstract

The problem in focus in this article concerns proof shortening, and featured prominently is a strategy, *cramming*, for addressing that problem. The literature shows that this problem was indeed of interest to some of the masters of logic, masters that included C. A. Meredith, A. Prior, and I. Thomas. The problem of proof shortening (as well as other aspects of simplification) is also germane to the recent discovery by R. Thiele of Hilbert's twenty-fourth problem. The cramming strategy (introduced in this article) was formulated to seek shorter proofs, starting with some known proof, usually the shortest offered by the literature. The most impressive success with the use of this strategy concerns an abridgment of the Meredith-Prior abridging of the Lukasiewicz proof for his shortest single axiom for the implicational fragment of two-valued sentential (or propositional) calculus.

1 Perspective and Wellspring

In this article, the *cramming strategy* is introduced, a strategy designed to aid the researcher in finding shorter proofs than those in hand or those offered by the literature. The events that led to its formulation merit review.

In the early 1990s, my colleague William McCune and I embarked on a study designed to produce a proof relying solely on condensed detachment showing that the fifth of Lukasiewicz's five axioms (which he conjectured to suffice for infinite-valued sentential calculus) is in fact dependent on the other four [Meredith1958]. (The conjecture was proved by Wajsubert; see page 145 in [Lukasiewicz1970].) C. A. Meredith [Lukasiewicz1970] had

*This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

already proved the theorem, but not purely in terms of the cited inference rule; equality-oriented reasoning was involved. What we did not know at the time—and learned only here in 2001—was that, apparently, the literature offers no proof of the type we sought. Our effort was rewarded: Each independently found the desired proof. In other words, we had found a proof absent from the literature, a Hilbert-style axiomatic proof relying solely on condensed detachment.

Each of us then turned to an attempt at finding a means for OTTER to produce a proof of the theorem under discussion, a proof obtained without guidance by the researcher. We each succeeded. A comparison of the two resulting proofs showed they were virtually identical, essentially proofs of length 63 (applications of condensed detachment). And McCune took one research road while I took another.

Indeed, that success quickly led to my study aimed at finding a shorter proof, a study that eventually yielded a 32-step proof in contrast to the 63-step proof that initiated this phase of my research. (The reasons for my study: First, it simply struck my fancy, perhaps influenced by my master's program in mathematics at the University of Chicago decades earlier; second, I delight in competition, even when the opponent is me.) If memory serves, at the time I had no methodology in hand and, certainly, no notion of where this road would take me almost a decade later. In particular, and especially during the 1990s, I have formulated various approaches to proof refinement with respect to length (as well as to other properties such as formula structure).

One additional occurrence completes the perspective for the formulation of the cramming strategy introduced in this article. In the late 1990s, Branden Fitelson contacted me by e-mail in the context of his finding a new proof (with McCune's OTTER [McCune1994]) for a result of Frege. And we have sought new proofs ever since. As part of our effort, Fitelson posed the problem of finding a proof better than that published by Meredith and Prior. They had found "an abridgment" [Meredith1963] of a proof that Lukasiewicz provided for his shortest single axiom for the implicational fragment of two-valued sentential (or propositional) calculus [Lukasiewicz1970]. In other words, Fitelson suggested I seek an abridgment (shorter proof) of the Meredith-Prior abridgment—which was the wellspring for the formulation of the *cramming strategy* featured here.

Indeed, a single theorem was the force for the birth of this new strategy. As detailed later, the goal was reached: The desired abridgment was found by OTTER. As required, the offering of this strategy would be premature were it not for its successful use in other contexts, also to be detailed. Its existence has perhaps added import. Specifically, Rudiger Thiele [Thiele2001] discovered that Hilbert in fact offered twenty-four problems—not twenty-three as was believed for almost a century. That twenty-fourth problem is concerned with proof simplification.

Everything being equal, the most obvious aspect of proof simplification concerns proof length. In that regard, use of the cramming strategy provides one avenue to pursue. (For those less familiar with the literature of mathematics, a proof can be shortened, at the price of a sharp decrease in simplicity to the point of being obscure, either by omitting many steps or by relying on far more complex formulas or equations than needed.) Proofs produced by an automated reasoning program may also suffer from unnecessary formula or equation complexity. The reduction in such complexity addresses a second aspect of proof simplification. Other aspects are featured later in this article.

The rules of the game here are that all of the deduced steps are within the theory under consideration, one or more specific reasoning mechanisms account for deductions, and steps are not omitted—rules that Hilbert might indeed have found appropriate.

2 The Nature and Object of Cramming

This new strategy derives its name from its nature. In effect, the strategy focuses on a chosen subproof of the total proof in hand of some conjunction and attempts to cram steps of that subproof into other subproofs needed to complete a new and shorter total proof. Alternatively, the cramming strategy derives its name from emphasizing the role of the subproof of one member and attempting to find subproofs of the remaining members of a conjunction such that their lengths can be (so-to-speak) crammed into a chosen length j. Typically, j is chosen to be less than the length of the shortest proof known for the theorem under study. Because of the basic mechanism that is employed by the strategy, level saturation, it seems almost certain that none of the masters applied such an approach, whereas cramming is indeed well suited to the actions of an automated reasoning program.

As my colleague Robert Veroff phrases the problem, the seeking of a shortest proof for a conjunction asks for the minimization of the set-theoretic union of the subproofs of its members. That problem offers unexpected obstacles, the main one captured by the following maxim. *Shorter subproofs do not necessarily a shorter total proof make.* This maxim dispels a myth that could easily be believed: Simply find ever shorter subproofs of the members of a conjunction, and progress will occur when the goal is that of finding a shorter or shortest proof of the entire conjunction. A simple example is in order.

Since the plum of this article concerns the Lukasiewicz shortest single axiom for the implicational fragment of two-valued (classical) logic, it (expressed as a clause) will be the focus of the example. .(l P(i(i(i(x,y),z),i(i(z,x),i(u,x)))). .)l The following Tarski-Bernays 3-basis provides a target for proving that the preceding formula is in fact a complete axiom system. .(l P(i(x,i(y,x))). P(i(i(i(x,y),x),x)). P(i(i(x,y),i(i(y,z),i(x,z)))). .)l For pedagogical reasons, I shall begin with a synthetic bit of data, turning to actual data once the point has been made patently clear.

First, note that the first step of any condensed-detachment proof relying on the Lukasiewicz formula is forced, must be present in any proof in which the formula serves as the only hypothesis. Next, assume that one has separately attacked the problem of finding a subproof for each of the three target axioms. Then, assume that the three attacks have been unusually successful, yielding, respectively, proofs of lengths 10, 20, and 30 for the first through the third target axioms. Finally, assume that, other than the first deduced step of each proof, the three proofs are disjoint. In the hypothetical example under discussion, the length of the proof of the conjunction is 10 + 20 + 30 - 2 (duplicate first steps) = 58.

At the other end of the spectrum is the (hypothetical) example in which one is able to find a 40-step proof of the third Tarski-Bernays axiom that is also a proof of the conjunction of the three axioms such that it contains a 15-step subproof of the first target axiom and a 25-step subproof of the second. In the cited case the presence of the proofs of the first and second axioms does not lengthen the proof of the conjunction. The preceding shorter subproofs do indeed not a shorter total proof make.

For the promised actual data—and a glimpse of the end of the story focusing on the abridgment of the abridgment—OTTER did find a 35-step proof of the desired conjunction. The respective subproofs (for the three target axioms) have in order length 10, 23, and 30. The better proof (given later) was discovered by OTTER's application of the cramming strategy. It has length 32, containing subproofs of respective length 10, 28, and 30.

3 The First Success

This section illustrates the first use of the cramming strategy and (as promised) presents a most satisfying proof obtained with it. A comparison of the new proof with the Meredith-Prior proof and with the Lukasiewicz proof provides further insight into the nature of proof refinement with respect to length and evidence of the value of cramming.

My attempt to reach the objective (posed by Fitelson) of finding a proof shorter than the Meredith-Prior proof began with focusing on their proof and to a much lesser degree on the Lukasiewicz proof for his shortest single axiom. Their 33-step proof was used with the *resonance strategy* [Wos1995] to direct OTTER's search, and, occasionally, the Lukasiewicz 34-step proof was also used in this manner. If one uses Veroff's *hints strategy* [Veroff1996] rather than the resonance strategy, the program has a strong tendency to simply reproduce the proof in hand, especially when the max_weight is set, say, to 1. (The cited property is put to good use, however, as discussed later in this section.) A partial explanation rests with the fact that the hints strategy focuses on the specific formulas or equations supplied and, if desired, clauses that subsume or are subsumed by a hint. In contrast, because a resonator is treated as if its variables are indistinguishable, guidance from the resonance strategy encourages a program to focus on equivalence classes (rather than on subsumption) defined by a resonator. In other words, the resonance strategy compared with the hints strategy gives the program a different type of latitude.

That phase of my research eventually yielded a 33-step proof of the Tarski-Bernays system, a proof a few steps different from the Meredith-Prior 33-step proof. Next, to each 33-step proof, I applied the methodology whose main component is that of blocking proof steps one at a time to force the program to yield (if possible) a different proof. Sometimes a given proof has the most unexpected property of containing in a not rigorous sense a proper subset that is in fact also a proof of the theorem under study. In particular, sometimes a proper subset of the deduced steps of the proof in hand is such that, with a different history and a different ordering, it itself is a shorter proof of the theorem to be proved. Failure was the result: No 32-step proof, or shorter, was found.

However, a small diamond was mined. Specifically, the new 33-step proof contained a 30-step proof of the third Tarski-Bernays axiom (TB-3), that axiom that (intuitively) is the most difficult to deduce. The Meredith-Prior 33-step proof proves TB-3 in 31 steps, and the Lukasiewicz 34-step proof proves TB-3 in 32 steps. In all three cases—the old Meredith-Prior 33-step proof, the Lukasiewicz 34-step proof, and, most important here, the new 33-step proof—the proof of TB-3 fails to include a subproof of either TB-1 or TB-2.

Perhaps—and I thought almost certainly not the case—a means could be devised to take the 30-step proof of TB-3 and *cram* many, many of its steps into proofs of TB-1 and TB-2. To be of interest, the cramming must have the property that the only needed steps to complete the desired subproofs (of 1 and 2) would be, respectively, 1 and 2 themselves. Stated a slightly different way, the cramming must have the property that 30 steps (proving TB-3) and TB-1 and TB-2 must (in effect) form a 32-step proof of the target conjunction. To be the case, the 30-step proof must be such that one pair of its formulas yields with condensed detachment TB-1 and another pair yields TB-2.

To test whether I had indeed struck gold, OTTER offers just what is needed. Indeed, one places the thirty formulas in the set of support list and instructs the program to conduct a level-saturation search; see [Wos1999] for details regarding the set of support strategy and the function of various lists in the context of using OTTER. If the hard-to-accept fact that cramming may succeed is indeed the case, then two level-1 proofs will be found, respectively, of TB-1 and TB-2. Success: Condensed detachment applied to the ninth step of the 30-step proof with itself yields TB-1; TB-2 is obtained by applying condensed detachment to the twenty-ninth and twenty-seventh steps. OTTER had indeed discovered a proof shorter than the Meredith-Prior abridgment, a proof of length 32.

Although that proof could be presented by merely appending to the 30-step proof the two cited applications of condensed detachment, perhaps an approach could be devised that would yield the 32-step proof directly, more in the spirit of automated reasoning and full automation. More to the point—and for use in other contexts—perhaps a strategy could be formulated that would yield the 32-step proof from the 30-step proof, would appropriately apply cramming in a manner that sufficed.

One obvious way to proceed calls for the inclusion of thirty-two resonators corresponding to the desired 32-step proof, each assigned a small value, say, 2. Then one assigns a small value to max_weight, equal to or not much larger than that assigned to the resonators. The clear intent is to force the program to focus on the thirty-two steps and not much else. If the assigned value to max_weight is 4, OTTER finds a 35-step proof; if the value is 2, OTTER finds a 33-step proof. Conclusion: the resonance strategy permits more than is desired. This observation holds in this case with or without the inclusion of the option ancestor_subsumption, a procedure offered by OTTER to automatically attempt to seek shorter proofs by comparing derivation lengths to the same conclusion.

In contrast to the resonance strategy, as noted earlier in this section, the hints strategy gives a different type of latitude; its use is more likely to have a program focus on the proof in hand. Indeed, when the thirty-two formulas of concern were included as hints and not as resonators and a small max_weight (2) was assigned, OTTER discovered the expected 32-step proof that deduces the Tarski-Bernays three-axiom system from the Łukasiewicz shortest single axiom for the implicational fragment of two-valued sentential calculus. For a fine point that might eventually provide needed insight in the future for some researcher, an assignment of the value 4 to max_weight did permit OTTER to retain the clause P(i(x,x)) when either the resonance strategy or the hints strategy was employed as described. This weight-4 clause was not used in any proof when the hints strategy was relied upon; it was present in proofs when resonance was used instead.

The 32-Step Proof

```
----- Otter 3.0.5b, March 1998 -----
The process was started by wos on soot.mcs.anl.gov, Wed May 24 16:24:24 2000
The command was "otter".  The process ID is 5369.


-----> EMPTY CLAUSE at   0.23 sec ----> 82 [hyper,34,53,77,79] $ANS(TARSKI_BERNAYS).


Length of proof is 32.  Level of proof is 29.


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


33 [] -P(i(x,y)) | -P(x) | P(y).
34 [] -P(i(p,i(q,p))) | -P(i(i(i(p,q),p),p)) | -P(i(i(p,q),i(i(q,r),i(p,r)))) | $ANS(TARSKI
35 [] P(i(i(i(x,y),z),i(i(z,x),i(u,x)))).
44 [hyper,33,35,35] P(i(i(i(i(x,y),i(z,y)),i(y,u)),i(v,i(y,u)))).
45 [hyper,33,35,44] P(i(i(i(x,i(y,z)),i(i(u,y),i(v,y))),i(w,i(i(u,y),i(v,y))))).
```

```
46 [hyper,33,45,35] P(i(x,i(i(i(y,z),y),i(u,y)))).
47 [hyper,33,46,46] P(i(i(i(x,y),x),i(z,x))).
48 [hyper,33,35,47] P(i(i(i(x,y),i(y,z)),i(u,i(y,z)))).
49 [hyper,33,35,48] P(i(i(i(x,i(y,z)),i(u,y)),i(v,i(u,y)))).
50 [hyper,33,35,49] P(i(i(i(x,i(y,z)),i(u,i(z,v))),i(w,i(u,i(z,v))))).
51 [hyper,33,50,35] P(i(x,i(i(i(y,z),u),i(z,u)))).
52 [hyper,33,51,51] P(i(i(i(x,y),z),i(y,z))).
53 [hyper,33,52,52] P(i(x,i(y,x))).
55 [hyper,33,52,35] P(i(x,i(i(x,y),i(z,y)))).
56 [hyper,33,35,55] P(i(i(i(i(i(x,y),z),i(u,z)),x),i(v,x))).
57 [hyper,33,35,56] P(i(i(i(x,y),i(i(i(y,z),u),i(v,u))),i(w,i(i(i(y,z),u),i(v,u))))).
58 [hyper,33,35,57] P(i(i(i(x,i(i(i(y,z),u),i(v,u))),i(w,y)),i(v6,i(w,y)))).
59 [hyper,33,35,58] P(i(i(i(x,i(y,z)),i(u,i(i(i(z,v),w),i(v6,w)))),i(v7,i(u,i(i(i(z,v),w),i
60 [hyper,33,59,35] P(i(x,i(i(i(y,z),i(u,v)),i(i(i(z,w),v),i(u,v))))).
61 [hyper,33,60,60] P(i(i(i(x,y),i(z,u)),i(i(i(y,v),u),i(z,u)))).
62 [hyper,33,61,35] P(i(i(i(x,y),i(z,u)),i(i(x,u),i(z,u)))).
63 [hyper,33,62,55] P(i(i(x,i(y,z)),i(i(i(x,u),z),i(y,z)))).
64 [hyper,33,63,35] P(i(i(i(i(i(x,y),z),u),i(v,x)),i(i(z,x),i(v,x)))).
65 [hyper,33,35,64] P(i(i(i(i(x,y),i(z,y)),i(i(i(y,u),x),v)),i(w,i(i(i(y,u),x),v)))).
66 [hyper,33,65,65] P(i(x,i(i(i(i(y,z),u),i(v,z)),i(i(i(z,w),v),i(y,z))))).
67 [hyper,33,66,66] P(i(i(i(i(x,y),z),i(u,y)),i(i(i(y,v),u),i(x,y)))).
68 [hyper,33,61,67] P(i(i(i(i(x,y),z),i(u,y)),i(i(i(y,v),x),i(u,y)))).
69 [hyper,33,67,68] P(i(i(i(i(x,y),z),i(i(y,u),v)),i(i(v,y),i(x,y)))).
70 [hyper,33,68,62] P(i(i(i(i(x,y),z),u),i(i(u,y),i(x,y)))).
71 [hyper,33,69,64] P(i(i(i(x,y),z),i(i(i(y,u),z),z))).
73 [hyper,33,64,71] P(i(i(x,y),i(i(i(x,z),y),y))).
75 [hyper,33,73,73] P(i(i(i(i(x,y),z),i(i(i(x,u),y),y)),i(i(i(x,u),y),y))).
76 [hyper,33,70,75] P(i(i(i(i(i(x,y),z),z),u),i(i(x,z),u))).
77 [hyper,33,75,71] P(i(i(i(x,y),x),x)).
79 [hyper,33,76,70] P(i(i(x,y),i(i(y,z),i(x,z)))).
```

As promised, comparison and analysis are in order. Crucial (apparently) is the 30-step proof of TB-3. It contains two steps not in the Meredith-Prior 31-step proof of TB-3 and three not in the 32-step Łukasiewicz proof of TB-3, the latter two proofs being subproofs, respectively, of the cited 33-step and 34-step proofs. In the context of shorter subproofs do not necessarily a shorter total proof make, the 34-step proof contains subproofs of lengths 10, 25, and 32 respectively of TB-1, TB-2, and TB-3. The Meredith-Prior 33-step proof contain subproofs of lengths 10, 26, and 31. The given 32-step proof contains subproofs of lengths 10, 28, and 30. When resonance was used in the obvious attempt to produce the desired 32-step proof, the resulting 35-step proof contained subproofs of lengths 10, 23, and 30; in other words, the use of hints rather than resonance in effect traded a 28-step proof of TB-2 for a 23-step proof, again showing how shorter subproofs can interfere with finding a shorter proof of the whole. The 31-step proof and the 30-step proof of TB-3 begin with the same twenty formulas, which might suggest that Meredith and Prior were proceeding along a most profitable path, a path that culminates (if appropriate action is taken) with the given 32-step proof. What won the game was (1) the finding of a 30-step proof deducing TB-3 and (2) cramming many of its steps into needed proofs of TB-1 and TB-2, constrained by a limit of 32 on length.

By way of clarification, the length of the subproof on which cramming is occurring guarantees nothing. In particular, a different 30-step proof might fail to offer the needed pairs for condensed detachment to produce the desired 32-step proof. Later experiments in fact found 30-step proofs of TB-3 that proved inadequate, not offering appropriate pairs.

4 Incarnations of Cramming

The cramming strategy admits different incarnations, depending on the intent. For example, the intent may be to block the use of all or almost all new conclusions that fail to match an included resonator or hint. Instead, the intent may be to encourage the program to consider new conclusions, especially those that do not match an included resonator or hint. Then, within the total proof of the conjunction under attack, there is the question of which member's proof to use to cram, which of the subproofs of the total proof to choose to cram as many of its steps into proofs of the remaining members.

The first formulated incarnation of the cramming strategy has one choose from among the subproofs of the members of a conjunction that proof that is the longest provided its length is not equal to the length of the conjunction. (The exception focuses on what are called *compact* proofs, a proof of a conjunction that is also the proof of one of its members; in such a case, the next longest subproof takes center stage.) Its proof steps are added as (so-to-speak) lemmas in the set of support, and a level-saturation approach is conducted. The steps, as resonators or as hints, are (in effect) treated as being simple, short in nature, which can be done with an appropriate assigned value. Then max_weight is assigned the same small value or a value not much larger. The lemma-reliance phase coupled with level saturation is intended to find the fewest additional steps needed to complete proofs of the members not directly in focus (through the use of resonance or hints). (For those familiar with OTTER, note that setting input_sos_first will not always produce the effects of sos_queue, level saturation, in particular, in the case where a number of levels are to be searched.) Note that, when level saturation is the choice, the value assigned to a resonator or a hint is not consulted for directing the program's attack; it is relevant only to purging newly deduced conclusions. The small max_weight is chosen to prevent the program from straying.

A pause or interruption at this point is merited to briefly discuss the difference between lemma reliance (as used in cramming in its various incarnations) and lemma adjunction [Wos2001] (as used when searching for a first proof). In the former, the goal is proof refinement with respect to length, where a proof is typically in hand. In the latter, typically no proof is available. As the name itself suggests, in the former the intention is to force the program to rely on the included proof steps, for the proof as a whole as well as for the various target subproofs. In the latter, the lemmas are adjoined temporarily with the expectation that many will not be used in the completed proof but used mainly to facilitate a path to the desired proof.

In the case featured in the preceding section, the object was to show (if possible) that just two steps, in addition to the thirty that correspond to a 30-step proof of TB-3, sufficed. For resonators, thirty-two were chosen, one for each of TB-1 and TB-2 and thirty for the proof of TB-3. By using level saturation and (as lemmas) the deduced steps of the proof of TB-3, the plan was to force the program to apply condensed detachment to *all* pairs of the steps of the TB-3 proof together with the Lukasiewicz shortest single axiom. The small max_weight blocks the consideration of virtually all other pairs, which has (among others) the effect of avoiding the examination of possibly shorter proofs of TB-1 and TB-2. Without the actions taken, the program might indeed either fail to apply condensed detachment to some pair of formulas of the TB-3 proof or, more likely, reject the results because of having already in hand a shorter proof of, say, TB-2. As noted, the game would be won if two perfect pairs could be found from among the steps of the thirty-step proof of TB-3 (possibly including the Lukasiewicz axiom), where it was not required that the elements of a pair be distinct. A perfect pair, in this case, is one that yields with condensed detachment either TB-1 or TB-2.

The last phase of the first formulated incarnation of cramming consists of having the program focus on the selected subproof together with the new deduced steps its use yielded. Hints served well for having the program concentrate almost exclusively on the designated steps. In fact, as discussed in Section 3, the just-described first phase failed with the use of resonators and succeeded only when the 32 desired steps were used as hints. This original incarnation was formulated specifically to thwart the program from seeking shorter proofs for individual members of the conjunction in focus.

As predictable, the given incarnation may fail to yield the steps needed to complete proofs of the remaining members of the conjunction under study. After all, little room is being given for steps other than those related to the included resonators or included hints. Further, the needed steps may be many, perhaps eleven or more. In such a case—and in a certain sense contrary to the original spirit of cramming—a second incarnation was born. One proceeds as in the first incarnation but assigns a rather generous value to max_weight. For example, if the resonators or hints are treated as having a weight of 2, one might assign the value 16, or 20, or even 28 to max_weight. The intent is to find a few steps, some of which are definitely not in the set of subproofs in hand, such that a new total proof (of the conjunction) is available. The resonance strategy appears to be of more use in this context than does the hints strategy.

In this second incarnation, if the proof is not compact, one begins by relying on the deduced steps of the longest subproof of the set of subproofs within the proof in hand of the conjunction to be proved. However, rather than including the remaining target elements of the conjunction as resonators or hints, one simply has the program find its own way to their respective proofs. This second incarnation, ordinarily accompanied by the use of ancestor subsumption, encourages the program to seek shorter proofs of the members other than that driving the reasoning. Of course, the length is not true length, for proof steps of the driving member are present as lemmas in the set of support. Indeed, the intent is to cram many of those steps into the needed proofs of the other members.

A fairly good indicator of the likelihood of success rests with a rather small difference, say five or less, between the level of the subproof driving the attack and the level of the total proof in hand. If the difference is much greater, then the level-saturation search might indeed be impractical, especially with a max_weight of 24 or larger. If one has not had the opportunity to run many experiments, the essentially exponential growth of level saturation is often an unpleasant surprise.

The following symbolic example illustrates the second incarnation of the cramming strategy. Consider the case in which one is asked to find a shorter proof than that in hand showing that some given formula is a single axiom. Assume one has in hand a proof of some known axiom system consisting of A, B, C, and D. Within that proof, by assumption, the longest proof is that of B, and the proof of the conjunction is assumed not to be compact. With the second incarnation of the cramming strategy, one places in the initial set of support, in addition to the single axiom, the deduced steps of the proof of B. One places in list(usable) (with OTTER) the disjunction produced by negating the conjunction of A through D. The deduced steps of the proof of B are used as resonators, each assigned a small value, and the max_weight is assigned the value 28. Level saturation is the choice for the search strategy.

The intent is that after some time proofs of each of A, C, and D will be found such that the lengths of the proofs suggests an advance has occurred with respect to proof length of the conjunction. If so, almost certainly, the proofs will depend on the use of proof steps from the proof of B. The so-called new proofs are then placed in a succeeding run as resonators with a small assigned value, say, 1. The proof steps of B are placed next, each assigned a value, say, 2. Level saturation is replaced with the ratio strategy. The max_weight is typically assigned the value 2.

If all goes as planned, a shorter proof of the conjunction is found. Quite likely the length of any or all of the subproofs of A, C, and D will be longer than their correspondents in the proof that prompted the use of the cramming strategy. The new and shorter proof will contain the cited proof of B, and an analysis of the proofs of the other members will show that a sometimes rather large number of the proof steps of the proof of B have been crammed into proofs of the other members.

A third incarnation of the cramming strategy has one focus on the next to the longest proof among the proofs of the members of the conjunction, provided that at least three items need to be proved. Its proof steps are (as in the other two incarnations) placed in the set of support to act as lemmas, and a level-saturation approach is taken. For some success to occur, usually, the proof length of the selected member best not be more than fifteen less than the length of the total proof. In addition, the proof length of the chosen member ordinarily had best be at least twelve. The suggested lengths are merely guesses based on experimentation. The idea is (1) not to have too far to travel to complete a proof of the most-difficult-to-prove member (based on proof length) and (2) to give the program a fair amount to work with in the form of lemmas.

A pleasing feature of the cramming strategy is its performance when iterated. I have, for example, taken a proof of a theorem, used the cramming strategy (in one of its incarnations) to produce a shorter proof, and then again used the cramming strategy on this new and shorter proof to gain even more ground. Occasionally, three or more such iterations have yielded one shorter proof after another.

5 Evaluating Cramming with Additional Successes

One naturally wonders whether the strategy just happened to succeed when the Lukasiewicz shortest single axiom for the implicational fragment of two-valued logic was the focus. Any new strategy demands application to other areas to gain some insight into its generality and power. Here I touch on some of those applications.

For an example taken from recent research, the right group calculus serves nicely. J. Kalman originally offered a five-axiom system (the following) for this area of logic [Kalman1976]. .(l P(d(z,d(z,d(d(x,d(y,y)),x)))). P(d(u,d(u,d(d(z,y),d(d(z,x),d(y,x)))))). P(d(v,d(v,d(d(u,d(z,y)),d(u,d(d(z,x),d(y,x))))))). P(d(d(d(u,d(v,y)),d(z,d(v,x))),d(u,d(z,d(y,x))))). P(d(d(v,d(z,d(u,d(y,x)))),d(d(v,d(x,u)),d(d(z,d(x,y)),x)))). .)l With OTTER, McCune showed that the second of the five is in fact sufficient to axiomatize the area [McCune1992].

When I chose to test the cramming strategy by studying this logic, McCune first told me that I should try on my own to determine which of the five axioms sufficed, rather than simply beginning with his success. In other words, I did not at the time recall that the second of the Kalman axioms can be used to deduce the remaining four. As it turned out, I began by focusing on the third axiom, choosing to rely on the second incarnation of cramming.

Early in my study of this third formula, I asked McCune about its status with regard to being a single axiom and learned that that question was in fact open. When I answered the question in the affirmative, a natural extension of the research was to focus on seeking a shorter proof showing that the conjunction of K-1, K-2, K-4, and K-5 could be proved from K-3. What a fine test of the cramming strategy!

One can treat the problem of showing K-3 to be a single axiom as the limiting case of an axiom-dependence problem. The object is to prove that each of the other four (of the five Kalman axioms) is dependent on K-3 alone. A promising approach that appealed to me was to focus on the four axioms one at a time, starting, for example, by trying to prove that K-1 is dependent on the remaining four. If successful, the plan was to use the resonators of the proof and seek to add to the list of dependent axioms, say, by trying to next prove K-2 dependent on 3 through 5. Eventually, only K-3 remained, with K-5 having been the most difficult to prove dependent. The reason may rest with its length.

When the four proofs were found, their steps as resonators led to the completion of a proof of the conjunction of 1, 2, 4, and 5 from 3. The proof of K-5 did not contain as a subproof a proof of any of 1, 2, and 4.

The proof of K-5 is generally longer than that of 1, 2, or 4. Therefore, for the cramming strategy, this proof offers more meat, more for the program to key upon as lemmas adjoined to the set of support for a level-saturation run. The goal is to find proofs of 1, 2, and 4 in which (no doubt) steps of K-5 will be present, crammed into their proofs. When an iterative cramming attack was initiated, I had in hand a proof of length 39 of the conjunction of 1, 2, 4, and 5 and a 29-step proof of 5. The first phase proved 1, 2, and 4 in a total of eight steps, in addition to the twenty-nine from a proof of K-5. Therefore, most likely cramming led to finding a proof of length strictly less than 39 for the target conjunction. It turned out that a proof of length 36 was yielded.

Occasionally with recourse to other methodologies to find a slightly shorter proof at various stages, the final iteration with cramming led to OTTER's discovery of a 27-step proof of the conjunction of 1, 2, 4, and 5, with K-3 as the sole hypothesis. The last iteration relied upon an 18-step proof of K-5, finding nine steps that together prove 1, 2, and 4. Currently, I know of no shorter proof that deduces the conjunction of 1, 2, 4, and 5 for K-3. In summary, the open question concerning the axiomatic status of K-3 was settled, and it was settled quite well in the sense of finding through cramming a rather short proof.

Next in order was the application of the cramming strategy to K-2. McCune (as noted) had proved it to be a single axiom, but not by directly deducing the conjunction of 1, 3, 4, and 5. Again, iteration with cramming was the choice, with a succession of proofs of K-5 playing the starring role. The last phase of the iteration yielded a 29-step proof of the desired conjunction, relying upon a 21-step proof of K-5 and on nine steps proving 1, 3, and 4 through cramming steps of the K-5 proof into proofs of 1, 3, and 4. In other words, although a length of 30 was expected, a slightly more satisfying result was obtained.

Finally, in order was the open question of the axiomatic status of K-4; 1 and 5 are insufficient. That question was settled in the affirmative. A 49-step proof was the first I found. Through cramming with iteration, OTTER eventually discovered a 31-step proof of the conjunction of 1, 2, 3, and 5. Rather than a succession of proofs of K-5 playing the key role, those for K-3 did; they were longer and more difficult in general to find.

In contrast to the preceding example that rested on the second incarnation of the cramming strategy and to the prize of this article (the 32-step proof given earlier) that was discovered with the first incarnation, an example of a success with the third incarnation is provided by focusing on two-valued sentential calculus studied in terms of the Sheffer stroke. Various single axioms have been found for this area of logic, each of length twenty-three (not counting a predicate symbol). My colleague Fitelson has most profitably attacked this area and found many other such axioms, including finding the following new single axiom. .(l P((D(D(x,D(y,z)),D(D(x,D(y,z)),D(D(u,z),D(D(z,u),D(x,u))))))). .)l To prove a candidate formula to be a single axiom in the Sheffer stroke, one can aim at proving the following Nicod 1-basis. .(l P(D(D(x,D(y,z)),D(D(u,D(u,u)),D(D(v,y),D(D(x,v),D(x,v)))))). .)l I instead, one can seek a proof of the following Łukasiewicz 3-basis. .(l P(D(D(x,x),x)). P(D(D(x,D(y,z)),D(D(D(y,u),D(D(x,u),D(x,u))),D(D(y,u),D(D(x,u),D(x,u)))))). P(D(x,D(D(D(x,x),y),D(D(
.)l

Fitelson asked for a short proof for his axiom—he had found a 45-step proof—and I considered both targets, with a bit more emphasis on the Łukasiewicz 3-basis. I first found a 47-step proof and, with cramming, was able to then obtain a 39-step proof. Within that proof is a 28-step proof of the third member of the Łukasiewicz axiom system that in turn contains a proof of the first member. The second member was proved in thirty-six applications of condensed detachment. Because little room existed between the longer of the three subproofs (length 36) and the total proof (length 39), I turned to the third incarnation of the cramming strategy, focusing on the 28-step proof as lemmas with level saturation.

The problem offered substantial resistance. Indeed, after fourteen CPU-hours (on a 296 MHz computer), an 11-step proof of the second member was found, which was not of much use. However, after twenty-nine CPU-hours, a 9-step proof of the desired type was found. As a sign of the robustness of OTTER and also of the difficulty of the problem, the cited proof was found after 25,999 (given) clauses had been chosen to direct the search.

The results of this long run strongly suggested that a 37-step proof of the 3-basis could be produced, which in fact is what occurred. One of the aspects of cramming that makes it so effective is its consideration of formulas whose complexity might otherwise cause the program to ignore them for participating in the reasoning. As evidence, in the cited experiment, three of the steps of the 37-step proof do not appear in the 39-step proof, each of weight twenty-four. Additional cramming then led to a 36-step proof. For the curious—by relying on other methodologies—the shortest proof found in my experiments with the Fitelson axiom that completes with the deduction of the Łukasiewicz basis has length 34.

If, instead, Nicod's 1-basis is used as the target, I can offer a 31-step proof (not obtained with the cramming strategy).

Yet one more example of the effects of cramming much of one proof into another is in order. The focus is Meredith's second single axiom (the following) for the ¡C,O¿ calculus, where O denotes **false** [Meredith1953]. .(l P(i(i(i(x,y),i(i(O,z),u)),i(i(u,x),i(v,i(w,x))))). .)l The target in this study was (what appears to be) the shorest basis for this area of logic, a 2-basis from Łukasiewicz, the following. .(l P(i(i(i(x,y),z),i(i(z,x),i(u,x)))). P(i(O,x)). .)l The first member of this 2-basis is an old friend (from Section 2), namely, Łukasiewicz's shortest single axiom for the implicational fragment of two-valued logic.

Through the use of various methodologies, I had found a 61-step proof of the 2-basis from the Meredith axiom. Within that proof, one finds a 54-step proof that deduces the first member of the 2-basis and a 10-step proof of the second member. With cramming and the focus on the 54-step proof as temporary lemmas, OTTER discovered a 56-step proof of the 2-basis. The newer proof is obtained by (in effect) trading the cited 10-step proof for a 35-step proof of the second member of the 2-basis, a proof of course sharing all but two of its steps with the cited 54-step proof. In other words, thirty-three of the fifty-four steps of the proof of the first member were crammed into a 35-step proof of the second member. So much cramming occurred in fact that the second needed proof was nearly absorbed.

Regarding another example of success with the original incarnation, my study of $C5$ serves well. That area of logic is the implicational fragment of $S5$, a modal logic that was formulated in part to capture the more commonly accepted notion of implication. Meredith offered the following single axiom for $C5$ [Lemmon1957,Meredith1964]. .(l P(i(i(i(i(i(x,x),y),z),i(u,v)),i(i(v,y),i(w,i(u,y))))). .)l After various experiments, my colleague Fitelson and I had found a 28-step proof, completing with the deduction of a Meredith 3-basis. That proof contains a 24-step subproof of one of the members which, when used with cramming suggested that a 27-step proof of the 3-basis exited. With the max_weight assigned a value of 4 and the inclusion of resonators for the 24-step proof together with three obtained with cramming, the first incarnation did yield a 27-step proof. Again a piquant trade was (in effect) made: An 8-step proof of the first member of the 3-basis was traded for a 25-step proof. In contrast to relying on the resonance strategy or the hints strategy when cramming, often progress occurs when no resonators or hints are included, thus giving the program even more freedom and independence from the proof in hand.

At a later point in the study, I had obtained a 21-step proof that was compact, meaning that the longest subproof of the three subproofs of the members of the 3-basis was in fact a proof of the entire basis. Put another way, I had found a proof of one of the members of the 3-basis such that it contained as subproofs proofs of the other two members. Still intent on more progress, I turned to the third incarnation, focusing on a 12-step proof of the first member as lemmas for cramming. That action produced eight steps of proof such that a 20-step proof was found, but, instead of deducing the 3-basis, a new and shorter 2-basis (due to Z. Ernst) was the target for completion. In particular, cramming had succeeded in coping with a compact proof by using the proof steps obtained from cramming with one target to reach in a shorter distance a different target axiom system.

6 Broadening the Use of Cramming

A natural question to ask at this point focuses on the use of the cramming strategy when the conclusion of the theorem to be proved takes the form of a unit clause, rather than a conjunction of two or more formulas or equations. In particular, imagine that one has expended a substantial amount of effort and time with the result of obtaining a 40-step proof of a theorem of the form P implies Q, where Q consists of a single formula or equation. If the 40-step proof is viewed as a compact proof of Q and, say, its thirtieth step and its twenty-fifth step, then one might wonder about the value of cramming in pursuit of a shorter proof of Q. For example, one might be rewarded for cramming on the first thirty steps, an apparent proof of step 30, used as lemmas with level saturation. I was so disposed and conducted appropriate experiments focusing on diverse areas of logic. None of them proved profitable.

After perhaps two weeks and a more careful review of the cramming strategy, I realized that the preceding does not precisely emulate any of the incarnations of cramming. The error was in treating the first thirty steps as indeed a proof of step 30. In particular, those thirty steps might contain steps *not* present in a proof of step 30, steps used later in the 40-step proof. What was required was to take the chosen step of the proof under study and, using its negation, have OTTER find its proof within the total proof. When the step number and length of the corresponding subproof are the same, then one could expect cramming to fail. For a chosen step to be of interest for cramming on its subproof within the total proof in hand, its proof length must be less than its number. In general, the use of the first k steps of the total proof, rather than a proof of step k, ties the hands of the program and typically forces it to simply reproduce the proof in hand.

Some experiments that adhered to the given constraint did produce results that were rewarding. For example, through cramming, OTTER discovered somewhat shorter proofs for single axioms (found by Fitelson) in the Sheffer stroke, completing with the deduction of Nicod's single axiom.

Those small successes caused me to revisit an area of research of more than one year ago. The focus of that study was the Lukasiewicz 23-letter single axiom (the following) for two-valued sentential (or propositional) calculus [Lukasiewicz1970].

```
P(i(i(i(x,y),i(i(i(n(z),n(u)),v),z)),i(w,i(i(z,x),i(u,x))))).
```

With many experiments and various methodologies spread over many months OTTER had discovered a 56-step proof of the sufficiency of this formula by deducing the following well-known three-axiom system of Lukasiewicz [Lukasiewicz1963].

```
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(x,i(n(x),y))).
P(i(i(n(x),x),x)).
```

My notion was to use cramming but not in a manner discussed in the earlier sections.

Specifically, rather than cramming on one of the subproofs of one of the three Lukasiewicz axioms within the 56-step proof, I instead asked for proofs of its 53rd, 54th, and 55th steps. OTTER produced a 50-step proof of the 53rd step of the 56-step proof. And I decided to cram on it, in a manner similar to that discussed when the conclusion is a unit formula or equation.

Although the corresponding experiment suggested that no reduction in proof length of the target conjunction was obtained, a proof of the first of the three Lukasiewicz axioms was discovered, two of whose steps are not present in the 56-step proof that initiated the study. When hints were relied upon corresponding to what was expected to be a 56-step proof, and a max_weight of 1 was assigned—instead of a new 56-step proof—OTTER produced a 50-step proof. The discovery of this 50-step proof was more than astounding to me, especially in view of the effort I had devoted to finding a proof of length strictly less than 56.

Summarizing, cramming did not immediately and directly produce a proof shorter than 56 steps; but, in fact, its use did result in such a discovery.

Given the preceding uses of cramming covered in this section and in earlier sections, one might naturally wonder about its use in the context of a set of hypotheses consisting of two or more formulas or equations. One experiment suffices to illustrate that this strategy can be useful in such a context. The area of logic is $C5$. The axiom system of concern is a 3-basis due to Meredith, one called 56_2, the following [Prior1962].

```
P(i(x,x)).
P(i(i(i(i(x,y),z),y),i(x,y))).
P(i(i(x,y),i(i(y,z),i(x,z)))).
```

The target was a axiom system due to Prior, denoted by 58_1, the following [Prior1962].

```
P(i(x,i(y,y))).
P(i(i(i(i(x,y),z),u),i(i(y,u),i(x,u)))).
```

I had in hand a 23-step proof of the Prior 2-basis within which was a 10-step subproof of the first of its two members. The experiment had OTTER cram on the 10-step proof with the object of finding a proof of the second member such that a proof shorter than length 23 would result. Cramming found a 12-step proof of the second member. When the 10-step proof that drove the attack and the 12-step proof that was found were used as hints with a max_weight assigned the value 1, OTTER produced a 22-step proof. The 22-step proof contains ten steps not present in the 23-step proof. As for the difficulty of finding the 12-step subproof (dependent on the 10-step proof used with cramming as lemmas), just under 86,000 CPU-seconds were required, 36,074 clauses were chosen as given clauses to direct the program's reasoning, and the proof completed with retention of clause (415578).

7 Highlights and Summary

The cramming strategy attempts to take a proof **P** of a member of a conjunction and find proofs of the remaining members by forcing or cramming many of the steps of **P** into the needed proofs. The objective is to discover a proof of the entire conjunction that is shorter than that in hand. The proof **P** is a subproof of the proof in hand of the conjunction, often the longest of the subproofs of the members of the conjunction. The cramming is by means of a type of lemma reliance coupled with level saturation. Specifically, the steps of **P** are added to the hypothesis or hypotheses already present in the set of support. The introduction in this article of the cramming strategy marks a somewhat different use of level saturation.

One of the piquant properties of the strategy is its encouragement of trading shorter proofs for longer proofs when the target is a member of a conjunction, where the objective is to find a shorter proof of the whole. Although not guaranteed, shorter proofs are in general simpler proofs, which makes them relevant to Hilbert's twenty-fourth problem (discovered only recently). Of course, proofs can be shortened (as occurs in literature) by simply omitting steps and treating them as implicitly present. Explicitly required for the type of result reported in this article is that all steps of a proof be present and that one or more inference rules be specified.

If the use of the cramming strategy leads to advances in the direction of proof refinement with respect to length, they can usually be traced to the introduction of new proof steps. (This observation may seem patently obvious until one notes that a shorter proof can sometimes be found such that all of its steps are among the formulas or equations of the longer proof.) Such new steps often arise from the application of an inference rule to a set of hypotheses that might ordinarily be skipped because of complexity considerations. For example, condensed detachment might be applied to a pair of formulas one or both of which have a length that causes the program to fail to consider the pair.

In the context of cramming, the prize to this date in early 2001 is an abridgment of the Meredith-Prior abridgment of a Łukasiewicz proof. The proof concerns the Łukasiewicz shortest single axiom for the implicational fragment of two-valued sentential (or propositional) calculus. Added satisfaction is derived from the knowledge that Meredith in particular was quite concerned with proof length. Other successes are given in Section 5, some derived from each of the three incarnations of the cramming strategy offered in Section 4.

The prized 32-step proof (given in Section 3) was discovered by cramming so much of a 30-step proof into the proofs of two other members of a Tarski-Bernays axiom system that only two additional steps were needed to prove the conjunction, that for TB-1 and that for TB-2. As it turned out, the 30-step proof (of TB-3) that starred in the success with cramming itself offered unique properties. Indeed, subsequent experimentation led to finding two other 30-step proof of TB-3, neither of which enable the completion of a 32-step proof. In contrast, the 30-step proof that appears as a subproof of the 32-proof given in Section 3 is such that (clearly) two of its pairs admit the deduction, respectively, of TB-1 and TB-2 with a single application of condensed detachment.

One has yet another wondrous example of the beauty of logic and mathematics, to the point of being startling. Indeed, as I myself was so vividly aware before running the crucial experiment, the probability was near zero of cramming exactly two steps and the 30-step proof in hand into a 32-step proof with the desired properties; in fact, as noted, later experiments did yield other 30-step proofs, none of which led with cramming to a 32-step proof. Especially when one has access to a powerful reasoning program such as OTTER, one might be strongly tempted to emphasize the study of proofs and their properties rather than placing the emphasis almost exclusively on which assertions are in fact theorems. More generally, (as my colleague Ted Ulrich has observed) an entire field might emerge devoted to the study of the space of proofs, in contrast to the study of the space of deducible conclusions. For but one question, does there exist a small set of resonators or hints (say, less than 100) that provides powerful guidance for studying disparate fields of logic or of mathematics?

The cramming strategy opens a new window to the huge set of proofs for any given theorem, a set far larger than one might ordinarily imagine. Its use can focus on sets of hypotheses that an unaided researcher is prohibited from considering. The result might be the discovery of a proof some of whose steps would have proved most elusive. Cramming gives one a weapon for attacking the proof of a conjunction as a whole in a manner that avoids traps that take the form of encountering a very short proof of one of the members of the conjunction. Section 6 shows that this strategy is also useful in the context of a target that consists of a single formula or equation and in the case where the axiom set under study contains two or more members. How effective this strategy will prove in domains where equality dominates is yet to be determined. I can report one success from lattice theory in which a 108-step proof had been found and no shorter one was in hand. When cramming was applied to a proof of the 103rd step (whose proof has length 99), a five-step proof of the goal was found, of course based on the 99-step subproof. In other words, one can produce a 104-step proof by merely appending (as is often done in mathematics or logic) the five-step proof to the 99-step subproof. Unfortunately, an obstacle exists if the goal is to produce in a single run a fully automated 104-step proof. The difficulty rests with the fact that the entire study (including that yielding the 108-step proof) was based on Kunth-Bendix. Because of the complications one can encounter with such an approach—in particular, the order in which demodulators are applied often presents a problem—OTTER did not succeed in producing the desired 104-step proof.

An example of the profitable interplay of cramming with other methodologies designed to find shorter proofs is provided by focusing on the use of demodulation to block steps of a proof in hand. In particular, where difficulty concerns proof length alone, if blocking a step of a proof of a conjunction produces another proof of the same length as that in hand, but if such blocking produces a shorter proof of the most difficult member to prove, then one can sometimes profitably apply cramming to the new and shorter subproof. The increase in the distance between the length of the most difficult-to-prove member and the entire conjunction is encouraging. In a somewhat related way, cramming is more likely to yield results of interest if the focus is on the longest subproof of the members of a conjunction rather than on the next to the longest. The explanation rests with a larger supply of (so-to-speak) lemmas for the program to rely upon.

As for the use of cramming in the context of answering open questions of various types, my studies of the right-group calculus provide some insight. Specifically, imagine that one is attempting to deduce three formulas or equations from a set of hypotheses, and that one has in hand proofs of two of the three. One takes the steps of the longer of the two proofs, uses them as lemmas in the set of support, chooses a level-saturation approach, and attempts to cram some or all of those steps into a proof of the third member of the conjunction.

Cramming also admits an odd type of inversion. In particular, one can focus on the most-difficult-to-prove (in terms of length) member to cram many of its steps into a proof of the next to the most difficult. Then one can turn instead to the next-most-difficult-to-prove in the context of the newer and shorter proof and attempt to cram many of its steps into a new shorter proof of the most-difficult-to-prove member.

When certain incarnations of cramming are relied upon, one finds that the space of proofs for a given theorem is far, far larger than one might intuitively suspect. Some of those proofs are found by forcing the program to apply inference rules to sets of hypotheses that might otherwise not be considered. Indeed, an examination of some of the results produced with cramming shows that a breakthrough occurred because condensed detachment was applied to a pair of formulas each of which was quite complex, a pair that would most likely have never been considered because of the complexity. Such inference rule application can complete a proof in which are present formulas of more complexity than was the case with the proof that initiated the study. In other words, from the viewpoint of proof simplification, a tradeoff may occur, a tradeoff of more proof complexity for less proof length. In contrast, some of the cramming successes that yielded a shorter proof also yielded a proof with less variable richness, a proof requiring fewer distinct variables. A reasonable conjecture asserts that some of the proofs found with cramming might have remained hidden from all who attacked the problem unaided.

The introduction of cramming and the evidence of its effectiveness add more support to the assertion that an automated reasoning program can indeed play an indispensable role in research.

References

[Kalman1976] Kalman, J. A., "Axiomatizations of logics with values in groups", *J. London Math. Society* **2**, no. 14 (1976) 193–199.

[Lemmon1957] Lemmon, E. J., Meredith, C. A., Meredith, D., Prior, A. N., and Thomas, I., "Calculi of pure strict implication", Philosophy Department, Canterbury University College, Christchurch, New Zealand, 1957. i+22 pp. (mimeographed).

[Lukasiewicz1970] Lukasiewicz, J., *Selected Works,* edited by L. Borokowski, North Holland, Amsterdam, 1970.

[Lukasiewicz1963] Lukasiewicz, J., *Elements of Mathamtical Logic,* Macmillan, New York, 1963.

[McCune1992] McCune, W., "Automated discovery of new axiomatizations of the left group and right group calculi", *J. Automated Reasoning* **9**, no. 1 (1992) 1–24.

[McCune1994] McCune, W., *OTTER 3.0 Reference Manual and Guide,* Tech. Report ANL-94/6, Argonne National Laboratory, Argonne, IL, 1994.

[Meredith1953] Meredith, C. A., "Single axioms for the systems ¡C,N¿, ¡C,O¿, and ¡A,N¿ of the two-valued propositional calculus", *J. Computing Systems* **1** (1953) 155–164.

[Meredith1958] Meredith, C. A., "The dependence of an axiom of Lukasiewicz", *Trans. AMS* **87**, no. 1 (1958) 54.

[Meredith1963] Meredith, C. A., and Prior, A., "Notes on the axiomatics of the propositional calculus", *Notre Dame J. Formal Logic* **4**, no. 3 (1963) 171–187.

[Meredith1964] Meredith, C. A., and Prior, A. N., "Investigations into implicational S5", *Z. Math. Logik Grundlagen Math.,* 10 (1964) 203–220.

[Prior1962] Prior, A. N., *Formal Logic,* 2nd ed., Clarendon Press, Oxford, 1962.

[Thiele2001] Thiele, R., and Wos, L., "Hilbert's twenty-fourth problem", preprint, ANL/MCS-P899-0891, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2001.

[Veroff1996] Veroff, R., "Using hints to increase the effectiveness of an automated reasoning program: Case studies", *J. Automated Reasoning* **16**, no. 3 (1996) 223–239.

[Wos1995] Wos, L., "The resonance strategy", *Computers and Mathematics with Applications* **29**, no. 2 (1995) 133–178.

[Wos1999] Wos, L., and Pieper, G. W., *A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning*, Singapore, World Scientific, 1999.

[Wos2001] Wos, L., "Conquering the Meredith single axiom", *J. Automated Reasoning* **27**, no. 2 (August 2001) 175–199.