Milestones for Automated Reasoning*

Larry Wos
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439-4801
e-mail: wos@mcs.anl.gov

May 14, 2004

Abstract

In the beginning (the early 1960s), the long-term goal of automated deduction was the design and implementation of a program whose use would lead to "real" and significant contributions to mathematics by offering sufficient power for the discovery of proofs. The realization of that goal appeared to be at least six decades in the future. However, with amazement and satisfaction, we can report that less than four decades were required. In this article, we present evidence for this claim, thanks to W. McCune's program OTTER. Our focus is on various landmarks, or milestones, of two types. One type concerns the formulation of new strategies and methodologies whose use greatly enhances the power of a reasoning program. A second type focuses on actual contributions to mathematics and (although not initially envisioned) to logic. We give examples of each type of milestone, and, perhaps of equal importance, demonstrate that advances are far more likely to occur if the two classes are indeed intertwined. We draw heavily on material presented in great detail in the new book *Automated Reasoning and the Discovery of Missing and Elegant Proofs,* published by Rinton Press.

1 An Effective Template for Research

In this article, we focus on the pursuit of a single goal: the design and implementation of a general-purpose program whose power is sufficient to prove deep theorems. One obvious template for pursuing such a goal is to consider what might be done in the abstract, with no specific theorem or class of theorems in mind, to increase the power of a reasoning program. Our approach, on the other hand, is to attempt to prove a *specific theorem*—one never before proved with any reasoning program—by devising a *general strategy or methodology* that can be applied to various domains. Indeed, we consider the most effective template for research to be an *experimental* template, where the wellspring for the particular study is a single, apparently-out-of-reach theorem and where the objective is the discovery of a proof of that theorem by means of a new

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

strategy or methodology that can be added to the arsenal offered by an automated reasoning program.

Even at our entrance into the field in the early 1960s, we applied this approach. For example, our attempt to solve a simple classroom exercise—prove that groups in which the square of every x is the identity e are commutative—led to the formulation of the powerful, and general, set of support strategy.

In this article we feature numerous other examples of such successes with W. McCune's automated reasoning program OTTER [McCune2003]. We discuss three strategies and various methodologies whose use greatly enhances the power of a reasoning program. And we relate our experiences with practical applications in mathematics and in logic. Some of the successes featured here have answered questions that had been open and had resisted the minds of the masters for decades.

2 The Resonance Strategy

We begin *in medias res* in the early 1990s when, after a visit to Argonne, Dana Scott sent us 68 theorems to prove. The theorems are numbered by Lukasiewicz theses 04 through 71. Theses 1, 2, and 3 form Lukasiewicz's axiom system for two-valued (classical propositional) calculus, studied in terms of the functions for implication and negation and expressed in the following OTTER notation.

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

Scott challenged us to prove with OTTER these 68 theses, using as inference rule *condensed detachment*, expressed in OTTER notation, with " | " denoting logical **or** and " - " denoting logical **not**.

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

Despite our years of experience with OTTER, we were able to obtain only thirty-three proofs; thirty-five theses remained unproven. Even with ROO, a parallel version of OTTER, we could prove only forty-eight of the theorems, leaving twenty unconquered. Clearly, to meet Scott's challenge—and consistent with our recommended experimental template—we needed to formulate a new technique. The *resonance strategy* was born [Wos1995].

For the problem in hand, we placed correspondents (*resonators*) of all sixty-eight theses in a pick_and_purge weight_list, assigning to each the same small value. A formula or equation placed in that list is assigned a priority, or weight, based on the assigned value, rather than assigning a priority or weight based on symbol count. By placing a formula or equation in that list and assigning it a value strictly greater than that assigned the max_weight, the user of OTTER enables the program to discard any deduced conclusion that matches said formula, where the variables are treated as indistinguishable. On the other hand, by assigning a value is less than or equal to that assigned the max_weight, the user provides guidance for directing the program's reasoning. The smaller the value, the more preferred is a matching retained conclusion, ignoring the individual variables.

A resonator is used to direct a program's reasoning; it does not have a **true** or **false** value. The pattern of the resonator is the key, where all variables are considered as indistinguishable, alike. Any retained conclusion that matches a resonator, ignoring specific variables, is assigned the value that is assigned to the corresponding resonator. With resonators, the user can, by assigning different values, express preference for one class of conclusions over another.

Our intention was to have OTTER prefer over all others any deduced conclusion that matched at the functional level; that is, the preferred conclusion(s) would be used before all others for inference rule initiation. Phrased more generally, we were informing the program that the pattern of any of the sixty-eight formulas was most attractive, that, if one treats all variables as indistinguishable, a deduced conclusion with such a pattern is attractive. The functional shape, ignoring the specific names of the variables, is what counts.

And it worked. Within less than 16 CPU-minutes on a computer of the 1991 type, all 68 theorems of interest were proved.

One could, of course, argue that the set of sixty-eight actually form the outline of a so-called master proof and, therefore, the effort was more in the spirit of proofchecking than of proof finding. Indeed, since the sixty-eight theorems had essentially been proved by Lukasiewicz, one could argue that no new proof of any type had been found. But as the following shows, the resonance strategy does have both the desired generality and the desired power. Not only has it found proofs that resisted full automation for years, but, more important, it has found proofs previously absent from the literature.

3 The Methodology of Lemma Adjunction

We present here a vivid example of a principle that has dominated much of our research since the early 1960s. The principle asserts that advances are likely to occur if one chooses a theorem whose proof is fully detailed and attempt to find a means for a reasoning program to complete a proof of that theorem *without* guidance. If successful, almost certainly the unaided proof will be sharply different from that in hand before the study was undertaken.

The specific theorem to be proved asserts the deducibility of the three-axiom system of Lukasiewicz for classical propositional calculus from the Meredith single axiom, the following.

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

We again used condensed detachment; we also used hyperresolution.

Meredith's proof is (in effect) of length 41. Attempts at discovering a fully automated proof with OTTER were spread over eight years, each, obviously, unsuccessful. Many diverse attacks were tried. Eventually we did formulate a methodology, called *lemma adjunction,* for finally enabling (in 1999) a reasoning program to prove Meredith's theorem without relying on his proof.

Briefly, lemma adjunction has the researcher choose some set of lemmas to be proved, with no certainty that any or all are relevant to the target theorem. Correspondents of those lemmas are placed as resonators in weight_list(pick_and_purge), each with the

same small value assigned to it. (A layered-resonator approach has also proved quite useful, where the set of resonators is partitioned into subsets with members of a given subset assigned a common value.) The denials of the lemmas are placed in list(passive). Members of that list are used mainly to detect the completion of a proof by noting that a unit conflict has been reached, and they are also used for forward subsumption; they do not participate in the reasoning process for drawing conclusions.

The lemma-adjunction methodology is iterative, each experiment (after the first) building on the results of its predecessor. In run $n+1$, one adjoins to list(sos) the lemmas proved in run n. (A powerful variation has one adjoin all of the proof steps of the proved lemmas; a proof step is a line of an OTTER proof, for example, deduced with condensed detachment or with paramodulation.) We note that resonators themselves do not have a **true** or a **false** value; they are merely included to direct a program's reasoning. In other words, resonators are not themselves lemmas. Also, we note that the lemmas that are adjoined in a succeeding run may have no value to completing a proof of the target theorem.

The use of lemma adjunction did succeed in four runs, discovering a 160-step proof of level 74; Meredith's 41-step proof has level 30 [Wos2001]. For the experiments, we used Lukasiewicz theses 4 through 71 as resonators.

Our next step was to test the effectiveness and generality of the new lemma-adjunction methodology. The goal was to find a proof where the literature offered none. We turned to a 23-letter formula (the following), presented in the mid-1930s by Lukasiewicz as the first single axiom for classical propositional calculus.

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

Lukasiewicz offered no guidance regarding the nature of his proof, just noting that three years of study had been required to obtain a proof. We conjectured that we might complete a proof (perhaps his) by deducing a known axiom system. We chose as target various axiom systems, including the Lukasiewicz three-axiom system and a six-axiom system of Hilbert. We again used theses 4 through 71 as resonators and as so-called intermediate lemmas. Four runs sufficed, requiring 4.5 CPU-hours, discovering a 200-step proof of level 68, completing with the deduction of the Lukasiewicz three-axiom system, among others. Hilbert's axiom system was deduced with a proof of length 194 and level 68, before the Lukasiewicz proof was completed. We view our successes as evidence of the generality of the resonance strategy.

4 The Hot List Strategy

Another application domain we have addressed with great success is algebra. In algebra, the proof being examined often requires repeated visits to the special hypothesis. For example, when proving commutativity for rings in which the cube of x is x (for all x), many crucial steps have as a parent the special hypothesis, namely, $xxx = x$. The *hot list strategy* was formulated in response to this observation [Wos1999].

With OTTER, a hot list is provided, in which the researcher places clauses that the program can visit, revisit, and the like. Let us say that *heat* is assigned the value 1; then, when a new clause is retained, immediately it is considered with each of the members

of the initial hot list for drawing additional conclusions for possible retention. If heat is assigned the value 2, then the clauses with heat = 1 (just described) are immediately considered with members of the initial hot list.

McCune generalized the hot list strategy, which was originally formulated in the context of paramodulation only, to apply to all inference rules. He also generalized it to the *dynamic hot list strategy*, whose use permits the program to adjoin during a run new clauses to the hot list.

Our first experiment with McCune's generalization to other inference rules, testing the OTTER version for the first time with hyperresolution employed for condensed detachment, yielded a new result. The field was classical propositional calculus. The theorem to prove asked for a derivation, from the Lukasiewicz three-axiom system, of a different three-axiom system of Lukasiewicz, sometimes referred to as a system of Church. OTTER, using the hot list strategy, found a 21-step proof, whereas the best in hand before the experiment was a 22-step proof. More recently, the hot list strategy was put to most profitable use by Z. Ernst, B. Fitelson, and K. Harris in the study of various areas of logic [Ernst2002]. They succeeded in finding six new single axioms for $C5$, the implicational fragment of the modal logic $S5$. Even more impressive was their success with $C4$, the implicational fragment of the modal logic $S4$. Indeed, where (apparently) Meredith had sought and failed to find a single axiom for this field, the trio did find one.

5 The Strategy of Cramming

The formulation of the *cramming strategy* [Wos2003a] was motivated by a query from one of our colleagues, B. Fitelson. Impressed with many successes with OTTER, he asked about the existence of a proof shorter than Meredith and Prior's 33-step proof for the sufficiency of the Lukasiewicz shortest single axiom for the implicational fragment of propositional calculus. None of the approaches we had devised in the preceding years for proof refinement yielded the prize. However, one of the experiments yielded a 30-step proof of the most complicated Tarski-Bernays axiom, the target three-axiom system, whose denial is the following.

```
-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_BERNAYS).
```

The ANSWER literal can be used to (so to speak) capture a construction found by OTTER but is also useful for identifying what has been proved, especially in the case where the program is instructed to prove a number of theorems.

We therefore hypothesized that, if we could formulate a technique that would enable OTTER to extend the 30-step proof with exactly two additional steps, one deriving each of the two remaining formulas to be proved, we would win: we would surpass Meredith and Prior's marvelous contribution. Put a bit differently, if we could find a way to "cram" the 30-step proof into a 32-step proof of the Tarski-Bernays system, we would succeed in the goal set by Fitelson. And the cramming strategy was born.

One begins with extending the set of support by placing the thirty formulas in it. One next includes as resonators the two formulas to be proved, each assigned a very

small value, and one then assigns that value to max_weight. Then one has OTTER rely on level saturation. The basic idea is to see whether the two formulas can be derived and little or nothing else. The desired 32-step proof was discovered.

Cramming has played an important role in proof refinement with respect to length, a topic we now address.

6 Hilbert's New Problem

For many mathematicians and logicians, the knowledge that some implication holds—is a theorem—suffices; the proof itself offers little interest. Others are satisfied with seeing some proof; its "elegance" is not a concern. Hilbert was a member of neither group.

Indeed, as his recently discovered twenty-fourth problem reveals, Hilbert was greatly interested in actual proofs and in proof simplification. A proof can be simplified by shortening it, by finding a proof with fewer deduced steps than that in hand. Simplification can instead focus on the complexity of the deduced steps, on the avoidance of thought-to-be indispensable lemmas, on the avoidance of some type of term, and more. As it turns out, an automated reasoning program can play a vital role in the discovery of more elegant proofs, of proofs simpler than those offered by the literature. (For a piquant aside, we note that the discovery by R. Thiele in 2000 of Hilbert's twenty-fourth problem delighted us [Thiele2000]; we have been studying proof refinements of various types since perhaps 1992.)

In our pursuit of such proofs, we have formulated various methodologies. For example, we often use demodulation in a nonstandard way, use it to block the retention of one or more unwanted steps of a proof under study. Sometimes we use cramming. We choose a type of term, for example so-called double-negation terms, and use demodulation to avoid retaining any deduced clause that contains such a term. A double-negation term is of the form $n(n(t))$ for some term t. We seek proofs that avoid lemmas the literature suggests are key to finding any proof.

Among our successes, we have—actually, OTTER has—discovered a 38-step proof for the Meredith single axiom, a proof three steps shorter than that of the cited master who himself was clearly interested in finding shorter proofs. Although we cannot make an appropriate comparison because of what is absent from the literature, we have used (OTTER and) cramming to complete a 50-step proof for the Lukasiewicz 23-letter formula. Also in part with that strategy, we now have in hand a 5-variable proof for a 19-letter single axiom of Meredith for propositional calculus in which **false** is part of the language. Meredith's proof includes a deduced formula that relies on eight distinct variables.

We have found double-negation-free proofs showing that one of the Lukasiewicz five axioms for his many-valued sentential calculus is in fact dependent. Meredith was the first to prove the corresponding theorem, but his proof does rely on double negation. Our find coupled with many others of its type led to M. Beeson (with colleagues) proving that the Lukasiewicz three-axiom system for classical propositional calculus has the following charming property. If the target is a theorem free of double negation, then there must exist a double-negation-free proof of the theorem with the Lukasiewicz three-axiom system as hypothesis. Beeson's success answered a question posed by the logician D.

Ulrich. We thus have yet one more example of how automated reasoning is now affecting the research of those whose primary interest is indeed outside of our field.

7 A Bright Future

The applications of OTTER are truly wide ranging. Powerful single axioms have been found for Boolean algebra [McCune2000]. McCune's monograph with R. Padmanabhan provides a most visible and exceedingly satisfying milestone, with its proof after proof and its answers to open questions [McCune1996]. And just two years ago, the last possible shortest single axiom, XCB, was found for equivalential calculus [Wos2002]. the corresponding question had remained open for seven decades. Without automated reasoning, almost certainly the question would still be open.

If this material has inspired some researcher to consider additional challenges and open questions that have been identified, the book *Automated Reasoning and the Discovery of Missing and Elegant Proofs* provides a beginning [Wos2003b]. That book discusses in far more detail than we are able to include here the various strategies, methodologies, and contributions to mathematics and logic that have been witnessed in the past few years. For a question not offered by the cited book, the following might prove of interest. Does there exist the analogue to the double-negation-free theorem for, say, group theory, where negation is replaced by inverse?

In summary, we have mined some treasure—but by no means all of it. Much remains, and we now have some of the means for mining far more.

References

[Ernst2002] Ernst, Z., Fitelson, B., Harris, K., and Wos., L., "Shortest Axiomatizations of Implicational S4 and S5", *Notre Dame J. Formal Logic* **43**, no. 3 (2002) 169–179.

[McCune1996] McCune, W., and Padmanabhan, R., *Automated Deduction in Equational Logic and Cubic Curves,* vol. 1095 in *Lecture Notes in Computer Science*, New York: Springer-Verlag, 1996.

[McCune2000] McCune, W., Veroff, R., Fitelson, B., Harris, K., Feist, A., and Wos, W., "Short Single Axioms for Boolean Algebra", *J. Automated Reasoning* **29**, no. 1 (2002) 1–16.

[McCune2003] McCune, W., *OTTER 3.3 Reference Manual and Guide,* technical memorandum ANL/MCS-TM-263, Argonne National Laboratory, Argonne, Illinois, 2003.

[Thiele2002] Thiele, R., and Wos, L., "Hilbert's Twenty-Fourth Problem", *J. Automated Reasoning* **29**, no. 1 (2002) 67–89.

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

[Wos1999] Wos, L., and Pieper, G. W., "The Hot List Strategy", *J. Automated Reasoning* **22**, no. 1 (January 1999) 1–44.

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

[Wos2002] Wos, L., Ulrich, D., and Fitelson, B., "Vanquishing the XCB Question: The Methodological Discovery of the Last Shortest Single Axiom for the Equivalential Calculus", *J. Automated Reasoning* **29**, no. 2 (2002) 107–124.

[Wos2003] Wos, L., "The Strategy of Cramming", *J. Automated Reasoning* **30**, no. 2 (2003) 179–204.

[Wos2003b] Wos, L., and Pieper, G. W., *Automated Reasoning and the Discovery of Missing and Elegant Proofs,* Paramus, N.J.: Rinton Press, 2003.