Axiomatic Proofs through Automated Reasoning*

Branden Fitelson[1,2]

Larry Wos[2]

[1]University of Wisconsin
Department of Philosophy
Madison, WI 53706
fitelson@facstaff.wisc.edu

[2]Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439-4844
wos@mcs.anl.gov

Abstract

A search of the seminal papers and significant books devoted to the study of various types of logic reveals that many proofs are missing. Indeed, if one seeks an axiomatic proof (of the type that Hilbert enjoyed) relying solely on, say, condensed detachment, in many cases one finds that none is offered by the literature. In part prompted by this discovery, we embarked on an intense study designed to find the missing proofs. In this article, we list many of our successes and discuss the methodologies used to obtain them. We also show how OTTER has proved valuable in finding proofs that avoid both the use of perhaps-thought-to-be indispensable lemmas and the use of an unexpected class of terms. Because crucial to our attack was the use of the automated reasoning program OTTER, we include an intuitive introduction to the subject. OTTER may well prove useful as an assistant in the research of others.

1. Nature, Incentive, and Perspective

Papers in logic and in mathematics abound with lists of lemmas, corollaries, and theorems, none, some, or all of which are accompanied by a proof. The nature and type of proof vary widely. At one end of the spectrum, in both logic and mathematics, one finds proofs based on induction, on the use of Zorn's lemma, or on some other type of metaargument. At the other end of the spectrum, (as preferred by Hilbert) one finds axiomatic proofs.

In contrast to the cited shared properties, frequently in logic proofs rely on a specific and explicit inference rule (such as condensed detachment), which is seldom the case in mathematics. Also common in a proof supplied in a paper on logic, the precise history of each deduced conclusion is given; again, such is rare in mathematics. Often, in the mid-1950s in the spirit of a joke, logicians were quoted as maintaining that mathematicians never gave proofs, only outlines of proofs.

In view of the given observations, automated reasoning offers unusual appeal for the logician. Indeed, the proofs one obtains with the reasoning program OTTER [McCune1994] rely on specific and well-stated inference rules. In addition, the precise list of hypotheses for each deduced conclusion is given explicitly. Furthermore, the proofs OTTER produces are axiomatic. From our perspective, the proofs found by McCune's program are far more appealing than other types of proof. The nature of those proofs and the various successes we cite in Section 3 suggest (to us) that much incentive exists

*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.

for the researcher to learn about automated reasoning. If one prefers to immediately browse among our cited successes (in Section 3), no need exists for examining the material on the elements of automated reasoning (Section 2). On the other hand, if one wishes a far more detailed treatment of the subject and access to a CD-ROM that some view as a gold mine, we suggest the recent book entitled *A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning* [Wos1999]. That book offers open questions to consider, a guide to the use of OTTER, and OTTER itself in various formats.

For the curious and for those who have had virtually no exposure to automated reasoning, this article includes in Section 2 an informal introduction to the field. We briefly review the underlying language, some inference rules, and, most important for success, diverse strategies that control the program's reasoning. If such reasoning is not effectively restricted and appropriately directed, the size of the space of deducible conclusions becomes (in most cases) an overwhelming obstacle. Fortunately, OTTER offers a veritable arsenal of strategies from which to choose. Our recent research demonstrates that we typically make effective use of many of the strategies.

Featured in this article are the results of that research. Although space does not permit the inclusion of proofs, we do list with needed detail the various theorems we have proved. The theorems are taken from various areas of logic that include two-valued sentential (or propositional) calculus and infinite-valued sentential calculus. Put another way, we offer here a discussion of proofs, each axiomatic, that have been missing from the literature, sometimes for more than six decades. (Many, many proofs that were missing will be offered in the planned book *Automated Reasoning and the Finding of Missing and Elegant Proofs in Formal Logic*, a book in which we shall detail various methodologies that have proved most useful.) We have in hand a condensed detachment proof for each of the theorems we discuss.

One of the more satisfying examples of the use of an automated reasoning program (in particular, OTTER) focuses on the following Lukasiewicz 23-letter single axiom for two-valued sentential calculus. (The function *i* denotes implication; the function *n* denotes negation; *x*, *y*, *z*, *u*, *v*, and *w* are variables; and the predictate *P* denotes ‘‘provable’’.)

```
% Following is Lukasiewicz's 23-letter single axiom.
P(i(i(i(x,y),i(i(i(n(z),n(u)),v),z)),i(w,i(i(z,x),i(u,x))))).
```

When Lukasiewicz announced his result in 1936, he provided no proof nor any clue concerning the nature of the proof. Further, from what we can ascertain, the literature offers no proof of that fine theorem. The proof OTTER found relies solely on condensed detachment, and (a fact we find piquant) the proof contains no formula in which double negation occurs, no term of the form $n(n(t))$ for any term t.

Consistent with the research of Meredith, Thomas, and Ulrich, we have made an intense and successful study of finding proofs shorter than those offered by the literature. (In [Meredith1963, page 171], Meredith and Prior report what they call a ‘‘very slight abridgment’’ of Lukasiewicz's [Lukasiewicz1970, page 300] proof of the sufficiency of his shortest single axiom for the implicational fragment of two-valued sentential logic. Their proof saves a step of condensed detachment, and it contains four steps not present in the 34-step Lukasiewicz proof. (The results of our study of this theorem are given in Section 3.) Later in the same paper [Meredith1963, page 180], it is reported that Thomas was able to shorten Meredith's proof of the sufficiency of his single axiom for the implicational fragment of intuitionistic sentential calculus by a single detachment step. Recently, Ulrich [unpublished] has been able to further shorten Thomas's proof by one more detachment step.) We cite a number of our results and successes—some in the context of proof shortening—and touch on the manner in which OTTER played the key role. Somewhat related, we also focus on how such a program can succeed in avoiding the use of various lemmas ordinarily relied upon, and we show how certain types of term can be dispensed with. Indeed, although contrary to what is found in print, (in almost all cases in which we have made an attempt) we were able to find proofs that totally avoid the use of double negation within a deduced conclusion, terms of the form $n(n(t))$ for any term t.

To test the methodologies we have developed, we are eager to learn of other theorems whose proof is missing. Although of a quite different nature, we also welcome requests whose objective is to find a proof shorter than currently available. We shall briefly touch on the approaches we employ for finding missing proofs, finding shorter proofs, and finding proofs satisfying various constraints.

2. An Informal Introduction to Automated Reasoning

The nature of this article and the intent to keep this article short dictate an informal treatment, free of definitions (in the main), and reliance instead on examples. Nevertheless, a reading of this section will provide one with some intuition concerning the elements of automated reasoning, how it works, and why use of the program OTTER has yielded so many new results. Although we shall favor the conventions and notation appropriate to the use of OTTER, the discussion applies in general to much of the field.

One has two choices when submitting a question or problem to OTTER, standard first-order predictate calculus or a closely related language, the *clause language*. If one chooses the former, the program translates the input to the latter. In the latter, only two connectives are permitted, logical **or** and logical **not**, with logical **and** present implicitly between each pair of clauses. For example, where *i* denotes impication and *n* negation and *P* ‘‘provable’’, the three-axiom system of Lukasiewicz [Lukasiewicz1963] for two-valued sentential (or propositional) calculus is represented with the following three clauses.

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))).

The following clause is included (with an appropriate inference rule called *hyperresolution*) when condensed detachment is in use, where ‘‘-’’ denotes **not** and ‘‘ | ’’ denotes **or**.

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

Variables, expressions beginning with *u* through *z,* are implicitly treated as universally quantified. Existentially quantified variables are replaced by appropriate Skolem functions.

The program knows essentially nothing, except for an understanding of equality (which is denoted in various ways). It simply accepts what one gives it (even if the information is inconsistent, which it can detect) and draws one conclusion after another. All conclusions drawn are *sound*, follow inevitably from the hypotheses used to obtain them. Typically, the program accrues new information at an astounding rate until a proof is complete, detected by finding a contradiction. To set the stage for such to occur, one assumes the theorem to be proved to be false (by denying its conclusion). Syntactically, if the theorem to be proved is of the form *R* implies *S*, one gives the program *R* and also gives it **not** *S*. Quite often, a proof completes by deducing *S*, which provides a contradiction.

One might suspect that the speed of today’s computer allows a simple accrual of a vast number of new conclusions until a proof is completed. Such is indeed not the case. The space of deducible conclusions is too huge for such an innocent approach to succeed, except for the most trivial theorems. Vital (and overlooked even by some of those in automated reasoning) is access to a variety of powerful strategies to control the program’s reasoning, some to restrict it, some to direct it, and (although less crucial) some for other purposes.

Among the strategies that restrict reasoning effectively are the following.

1. The *set of support strategy* typically restricts a program from exploring the space of conclusions that follow from the axioms.

2. The *expression complexity strategy* restricts a program from considering terms, formulas, or equations conjectured to interfere with effectiveness because of their complexity.

3. The *variable richness strategy* restricts the program from considering formulas or equations whose number of distinct variables appears to make them unattractive.

4. The *term-avoidance strategy* prevents the program from retaining any newly deduced conclusion that contains a term in the class of those designated as unwanted.

In contrast to the preceding four strategies that restrict a program’s reasoning, the following useful strategies direct its reasoning.

5. The *resonance strategy* instructs a program to focus on conclusions that resemble any of a set designated by the researcher as appealing, in preference to all other available conclusions.

6. The *ratio strategy* instructs a program to choose *k* conclusions by complexity, 1 by first come first serve, then *k*, then 1, and the like, where *k* is assigned by the researcher.
7. The *weighting strategy* directs a program to focus on items whose complexity is smallest, where the complexity is in part determined by user-assigned values.

As for types of inference rule, one type, called hyperresolution, requires that all conclusions that are drawn be free of **not**. Another type, called *UR-resolution*, requires that the drawn conclusion be nonempty and free of **or**. Among the other inference rules offered by OTTER—one that presents sharp contrast to the cited two—is *paramodulation*, which generalizes Church's rule of equality substitution, and enables the program to ‘‘understand’’ the notion of equality.

In addition to the elements of automated reasoning already briskly addressed, of the others, one focusing on the removal of redundant information demands attention. Indeed, unless prevented from doing so, a reasoning program will deduce and retain various types of redundant information. The fault does not lie with the nature of automated reasoning; instead, it lies with the nature of information. For one example, ordinarily an item of information expressed as an equation can be presented in a number of equivalent forms, differing in the association of the terms or in the commuting of the terms. To avoid such, a reasoning program typically offers a means, *demodulation*, for automatically left associating, right associating, commuting, or other. More generally, demodulation enables a program to automatically simplify and canonicalize deduced information.

For a second type of redundancy, two items of information may be such that the second is merely an instance of the first, obtainable by substituting in the first appropriate terms for variables. A means exists, *subsumption*, for avoiding this type of redundancy as well.

3. Successes and Methodologies

OTTER, through the use of strategy, permits one to give advice based on knowledge, conjecture, or intuition. What adds (for us) satisfaction in the context of the successes we shortly detail is the fact that we are far from expert in the areas of logic that were studied. Nevertheless, we were able to find proofs of theorems considered important by logicians that include Lukasiewicz, Meredith, and Rose and Rosser. Those studies in turn led to our development of a number of methodologies for finding first proofs, shorter proofs, and proofs satisfying various constraints (such as the avoidance of double-negation terms). To test the power of our approaches, we began working through the encyclopedic Appendix in Prior's text on formal logic [Prior1962, Appendix I]. Prior's Appendix contains over 80 axiom systems for various logical calculi. Our goal was to find axiomatic proofs for the theorems given in Prior's Appendix. So far, having worked through well over half of the Appendix, we have proved one theorem after another, with no exceptions. All of the proofs we have found rely solely on condensed detachment.

Our proofs are often unlike those found in the literature. For example, (as noted earlier) almost always we avoid terms of the form $n(n(t))$ for any term t, where the function n denotes negation. We were (at first) prompted to seek such proofs purely out of curiosity. However, we have learned that the addition of the cited constraint increases the likelihood of success. We have not yet ascertained whether, without such constraints, OTTER would have been unable to complete some of the long proofs that were found, sometimes as long as 200 applications of condensed detachment. Also, we note that the proofs we find occasionally rely on a retained clause whose number is greater than 994,000, meaning that the program indeed kept many conclusions before the proof was complete.

More generally, the program is not confined to seeking proofs that emulate those offered by the literature. Indeed, sometimes we avoid the use of some lemma that had appeared to be vital to completing a proof. The ability to avoid (at the request of the researcher to do so) the use of one or more specified lemmas is yet one more aspect of automated reasoning that adds to its appeal. Therefore, one should not be surprised to find that many of the proofs OTTER offers are sharply unlike those found by the masters. Rather than disappointment, some of the differences may prove quite enlightening.

At this point, without proof because of space limitations, we turn to a number of theorems whose proof was (in some sense) missing but found with the use of OTTER.

(Almost all of the proofs we have found and discuss here are free of the use of double negation.) Meredith and Prior [Meredith1963, page 171] present a ‘‘very slight abridgement’’ of Lukasiewicz’s proof [Lukasiewicz1970, pages 299-300] of the sufficiency of his shortest single axiom for the implicational fragment of two-valued sentential (or propositional) calculus. Lukasiewicz’s proof (when reasonably reconstructed from his detachment and substitution proof) requires 34 condensed detachment steps. Meredith and Prior were able to eliminate one step of Lukasiewicz’s proof, yielding a 33-step condensed detachment proof. Using OTTER, we have been able to find a 32-step proof, two of whose steps are not present in the 33-step proof of Meredith and Prior.

Lukasiewicz [Lukasiewicz1970, page 225, footnote 10] reports (without a sufficiency proof) a 23-letter single axiom for two-valued sentential (or propositional) calculus (in terms of implication *i* and negation *n*). Using OTTER, we have found the needed and missing proof, a 58-step condensed detachment proof of Lukasiewicz’s 3-axiom basis for two-valued sentential logic [Lukasiewicz1963]; this system is typically taken as the reference axiomatization in this area for establishing sufficiency.

Wajsberg [Wajsberg1977, Theorem 37, page 209] gives a generalization of the Tarski-Bernays axiomatization of the implicational fragment of two-valued sentential logic. Wajsberg’s proof relies on mathematical induction. Using OTTER, we have found a 17-step, pure condensed detachment proof of Wajsberg’s theorem.

Meredith [Meredith1953] gives two 19-letter single axioms for the system <C,O> of two-valued sentential (or propositional) calculus. Meredith [Meredith1953, pages 160-163] reports a 31-step detachment plus substitution sufficiency proof for the first of these axioms. His proof is not a condensed detachment proof; it instead relies on unneeded identifications of variables in several steps. Using OTTER, we have determined that the shortest condensed detachment proof that contains Meredith’s 31 reported steps is in fact a 37-step, 8-variable condensed detachment proof. Using OTTER, we have found a 26-step, 6-variable condensed detachment proof. Our OTTER proof is significantly more elegant than Meredith’s, in terms of its length, and in terms of the complexity of the formulas that appear in the proof.

As for the second <C,O> single axiom, Meredith and Prior later report a proof in [Meredith1963, pages 183-184]. This proof requires 60 condensed detachment steps. Using OTTER, we have found a 54-step condensed detachment proof.

Meredith and Prior [Meredith1963, page 182] also report a single axiom for the two formulas P(i(i(i(x,x),y),y)) and P(i(i(x,y),i(i(y,z),i(x,z)))) (known in the modern literature as ‘‘specialized assertion’’ and ‘‘suffixing’’, respectively). Meredith gives a proof of the two desired formulas from his single axiom. But, Meredith’s proof uses combinators and lambda conversion, not condensed detachment on propositional formulas. As such, an explicit condensed detachment proof was missing. Using OTTER, we have found a 7-step condensed detachment proof. Since this proof was found by using an exhaustive (breadth-first) search, most likely no shorter condensed detachment proof exists. We have also used OTTER to prove the other direction of the equivalence. We have a 6-step condensed detachment proof of Meredith’s axiom from specialized assertion and suffixing.

Rose and Rosser [Rose1958, page 12] mention that they are unable to prove several distributivity laws from the axioms of Lukasiewicz’s infinite-valued sentential logic. It is easy to show that these distributivity laws are valid in the semantics of infinite-valued logic. And, since Rose and Rosser were able to prove the completeness of the axioms of infinite-valued logic, they knew that these distributive laws must be provable from the axioms. However, condensed detachment proofs of these laws from the axioms of infinite-valued logic eluded Rose and Rosser. Recently, Harris and Fitelson [Harris2000] have used OTTER to find condensed detachment proofs of these elusive distributivity laws.

Rose and Rosser [Rose1958] use metatheoretic equalities (together with metatheorems about substitutivity involving classes of such equalities) to prove various theorems, including the associativity of logical **or** in infinite-valued logic. Using OTTER, we have found condensed detachment proofs of all of the reported theorems in Sections 2 and 3 of Rose and Rosser’s paper. Moreover, we have used OTTER to prove several generalizations of the associative laws for **or**.

Lukasiewicz [Lukasiewicz1970, page 144] gives a 5-axiom basis for infinite-valued sentential logic. Meredith [Meredith1958] and Chang [Chang1958] independently found detachment plus substitution proofs of the dependence of Lukasiewicz's fifth axiom. Their proofs are at least 37 condensed detachment steps long (when they are reconstructed reasonably from their original detachment plus substitution form), and both rely heavily on the use of double negation terms as well as Rose and Rosser's lemmas 2.22, 3.5, and 3.51 [Rose1958]. Using OTTER, we have found a 32-step condensed detachment proof that makes use of neither double-negation terms nor lemmas 3.5 or 3.51 from [Rose1958]. It remains open whether a double-negation free dependence proof exists that uses none of Rose and Rosser's three lemmas.

Lemmon et al. [Lemmon1969] report a single axiom of Meredith for the system C5 of strict implication. Meredith's 21-step condensed detachment sufficiency proof is reported. Using OTTER, we have found a 24-step proof, which is somewhat disappointing. No proof of the soundness of Meredith's C5 single axiom is given in the literature. Using OTTER, we have found a 26-step proof of Meredith's C5 single axiom from axiom set (ii) in [Lemmon1969].

The sufficiency of the single axioms XHN [e(x,e(e(y,z),e(e(z,x),y)))] and XHK [e(x,e(e(y,z),e(e(x,z),y)))] for equivalential calculus has been established using OTTER. Indeed, OTTER furnished the discovery of very elegant proofs of the sufficiency of these two single axioms. It is still an open question whether the formula XCB [e(x,e(e(e(x,y),e(z,y)),z))] is a single axiom for equivalential calculus. We have used OTTER to show that XCB implies reflexivity [e(x,x)], and that XCB plus symmetry [e(e(x,y),e(y,x))] is sufficient for equivalential calculus. This reduces the open question to determining whether symmetry is deducible from XCB.

Meredith [Meredith1953, pages 157-160] gives a 41-step condensed detachment proof of the sufficiency of his 21-letter single axiom for the <C,N> system of two-valued sentential logic. Meredith's proof uses 7 distinct variables and relies heavily upon the use of double-negation terms. Using OTTER, Deepak Kapur [unpublished] was able to find a 6-variable proof of length 63. We now have a 50-step, 6-variable proof (containing double-negation terms). We have also used OTTER to find a 51-step, 7-variable proof that is free of double negation terms. It remains open whether there exists a 6-variable proof that is also free of double-negation terms. In that Meredith's 7-variable proof has length 41, do the cited results imply that a price must be paid for variable reduction or for term constraint? Obviously, such is not always the case.

Meredith [Meredith1963, page 172] reports two 19-letter single axioms for the <C,I> system of the implicational fragment of two-valued sentential logic. He proves the sufficiency of the first of these, using 38 condensed detachment steps. Using OTTER, we have found a 31-step condensed detachment proof. Meredith reports no proof of the sufficiency of his second <C,I> single axiom. Using OTTER, we have found a 61-step condensed detachment sufficiency proof of this second <C,I> single axiom.

References

[Chang1958] Chang, C. C., ‘‘Proof of an Axiom of Lukasiewicz’’, *Trans. AMS* 87, no. 1 (1958), 55-56.

[Harris2000] Harris, K., and Fitelson, B., ‘‘Distributivity in Lw and Other Sentential Logics’’, submitted to *J. Automated Reasoning* (special issue on logical calculi), 2000.

[Lemmon1969] Lemmon, E. J., Meredith, C. A., Meredith, D., Prior, A. N., and Thomas, I., ‘‘Calculi of Pure Strict Implication’’, in J. W. Davis, D. J. Hockney, and W. K. Wilson, (eds.), *Philosophical Logic*, D. Reidel, Dordrecht, 1969.

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

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

[McCune1994] McCune, W., *OTTER: 3.0 Reference Manual and Guide*, Tech. Report ANL-94/6, Argonne National Laboratory, Argonne, IL, January 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, no. 3 (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.

[Prior1962] Prior, A., *Formal Logic*, second edition, Clarendon, Oxford, 1962.

[Rose1958] Rose, A., and Rosser, J. B., ‘‘Fragments of Many-Valued Statement Calculi’’, *Trans. AMS* 87 (1958), 1-53.

[Wajsberg1977] Wajsberg, M., *Logical Works*, Polish Academy of Sciences, Warsaw, 1977.

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