The Legacy of a Great Researcher

Larry Wos*

Argonne National Laboratory

Abstract. This article presents examples of the monumental contributions of Bill McCune to mathematics, logic, computer science, and especially automated reasoning. The examples are presented in the form of short stories and recollections of the author during his long association with Bill. In addition to Bill's accomplishments as a researcher, the author provides personal memories giving glimpses into Bill's complex personality and his generosity as a collaborator.

1 Perspective and Genesis

Perhaps you have wondered what would result if you had the opportunity to spend thousands of hours with a great mind. For more than two decades, I had that opportunity as I shared research time with my esteemed colleague William (Bill) McCune. We shared many ideas, conjectures, and, yes, guesses. Each of us had two main goals. The first goal was to formulate enhancements for an automated reasoning program, enhancements that would substantially add to its power. The second goal was to employ the program in a way that would contribute to various areas of mathematics and logic.

From the viewpoint of making contributions to mathematics and logic, Bill and I had a marvelous automated assistant; indeed, in 1988, if memory serves, he designed and implemented the automated reasoning program called OTTER. (We did have access to a program designed at Argonne before OTTER was produced.) In but four months, even though Bill was also involved in research of different aspects, he wrote more than 30,000 lines of code, producing a program that, from then until now, has exhibited the smallest number of bugs. Yes, his effort was and is monumental; indeed, when you obtain a conclusion, a set of conclusions, a proof, you can assume with almost total certainty that all is in order. Also important is the robustness of OTTER, permitting you to have it search for desired objects for weeks, if needed, without stopping.

In this article, I shall tell a number of short stories that provide ample evidence of Bill's inventive mind, his accurate insights, and his impeccable professionalism. His successes in the context of enhancements have played a key role in much of what has occurred in the past ten years. As I shall highlight here, Bill answered (with one of his programs) open questions in areas that include group theory, lattice theory, Boolean algebra, combinatory logic, and—so impressive—Robbins algebra; for various open questions, see Chapter 7 of [14].

* This work was supported by the U.S. Department of Energy, under Contract DE-AC02-06CH11357.

As many of you may know, Bill provided at Argonne National Laboratory a means for researchers to easily copy OTTER. And, possibly because that program was correctly viewed as extremely powerful, many, many copies were taken. As but one bit of evidence of his professionalism, he enabled me to place (on disc) a copy of OTTER, with a manual, in the back of one of my books, thus materially adding to the usefulness of my books. Even his users manual is well written [5]. An examination of history would reveal that OTTER provided the basis for a large number of programs that followed its birth.

This narrative will almost certainly not follow the chronology of history. Nor will it echo my view of the significance of Bill's achievements. Instead, the order of the topics will reflect, though probably somewhat hidden, some form of intensity of my recollections, recollections coupled with memories of excitement, curiosity, and, yes, surprising results in many cases. The field of automated reasoning is deeply indebted to Bill for all that he provided.

2 Combinatory Logic

Our foray into combinatory logic began with our colleague Ross Overbeek, who had read a charming book on combinatory logic by Raymond Smullyan, *To Mock a Mockingbird* [12]. Ross began the study of two combinators, B and W, that respectively satisfy the following, where expressions by convention are assumed to be left associated unless otherwise indicated.

```
Bxyz = x(yz)
Wxy = xyy
```

The object of Ross's study was to determine whether the fragment with basis consisting just of B and W satisfied the strong fixed point property, in the context of the following definition: A combinator $\mathbf{F}$ is a fixed point combinator if and only if $\mathbf{F}x = x(\mathbf{F}x)$ for all x. (As I recall, Smullyan called such combinators "sages".) The $\mathbf{F}$ in this case must be expressed solely in terms of B and W. To his disappointment during his study, Ross found that the open question concerning B and W had been answered by R. Statman, who found (in February 1986) a fixed point combinator for B and W expressed in eight letters.

```
B(WW)(BW(BBB))  (Statman's fixed point combinator)
```

Upon learning of Statman's result, Ross asked me to find a way for an automated reasoning program to find a fixed point combinator $\mathbf{F}$ expressed solely in terms of B and W. In other words, you could view his request as amounting to answering, with a program of ours and no knowledge about whether the answer was yes or no, the following open question: Does the fragment based on B and W alone satisfy the strong fixed point property?

I began thinking of how paramodulation could be used. (Paramodulation, as many of you probably know, is an inference rule that generalizes equality substitution; its use builds in a powerful treatment of equality.) Shortly after I initiated my effort, Bill entered my office. Upon learning of my activity, he asked if he could join me, a request I gladly agreed to. What Bill did was have the reasoning program ITP make two searches, one forward and one backward, believing that insufficient memory was present to attack the question directly. (ITP, designed and implemented by Ross and Ewing Lusk, and contributed to heavily by Bill, preceded the birth of OTTER [3].)

Bill accumulated two sets of deductions. Then, on a weekend, he assigned the program the task of comparing pairs of equations, one from the forward search and one from the backward search, to see whether unit conflict could be found. In particular, the forward search yielded (positive) equalities, and the backward negative equalities.

On the following Monday, to the amazement of both Bill and me, five pairs were discovered by ITP. That result showed that, rather than one fixed point combinator for the B-and-W fragment, there were five. When Bill and I wrote to Smullyan, he was indeed surprised and impressed at our discovery—mostly Bill's, to be precise [11].

For the curious reader, the discovery of the five fixed point combinators for the B-and-W fragment eventually led to the discovery of an infinite set of fixed point combinators for this fragment. Not too long after that discovery, an infinite class of infinite sets of such fixed point combinators was proved (by me) to exist. Yes, these discoveries can be traced directly to Bill's use of ITP and that profitable weekend. And this episode led to a second, one that illustrates Bill's insight.

Indeed, some time later, Bill called me at home and made the following observation: "Did you know that, if you took the five fixed point combinators, each of length eight, and demodulated each with B, you get the same expression?" The magical expression is the following.

```
(W(B(Bx))) cubed
```

Eventually, this expression would be called a kernel for the B-and-W fragment, after the *kernel strategy* was formulated [13]. I would never have thought of this strategy were it not for Bill's insightful observation. That strategy proved to be most powerful when seeking fixed point combinators.

Sometime during the study with Bill—I cannot pinpoint exactly when that was—I proved that the B-and-L fragment was too weak to satisfy the strong fixed point property. I am fairly certain that result also answered a question that had been open. Bill, not too much later, generalized what I had done and showed that various other fragments also failed to satisfy the fixed point property. That bit of research provides yet more evidence of Bill's research strength, nicely illustrating his ability to think as a mathematician.

3 Boolean Algebra

During Bill's position as a staff member at Argonne, I believe 1983 to 2006, I was certainly not the only researcher who benefited from collaboration with him. For example, with Ranganathan Padmanabhan he answered a number of open questions, many of which were published in a marvelous monograph (in 1996) by the two of them [8]. One of the questions that was answered with OTTER concerns a theorem called by Bill DUAL-BA-3. The following equations (which rely on the notation used in that monograph) capture that theorem, where $x@$ denotes the complement of x and where the two inequalities arise from, respectively, negating the theorem to be proved and negating its dual.

```
x = x.
x * (y + z) = (x * y) + (x * z).
x + (y * z) = (x + y) * (x + z).
x + x@ = 1.
x * x@ = 0.
(x * y@) + ((x * x) + (y@ * x)) = x.
(x * x@) + ((x * z) + (x@ * z)) = z.
(x * y@) + ((x * y) + (y@ * y)) = x.
(x + y@) * ((x + x) * (y@ + x)) = x.
(x + x@) * ((x + z) * (x@ + z)) = z.
(x + y@) * ((x + y) * (y@ + y)) = x.
(A * B) + B != B | $ANS(A2).
(A + B) * B != B | $ANS(A4).
```

Bill called me at home, having in mind my interest in improving on given proofs. First, he told me that he had a proof of the theorem; then he informed me that the proof consisted of 816 equations, the longest proof I had ever heard of produced by OTTER. The proof relied on a Knuth-Bendix approach and, therefore, featured the use of demodulation. After I expressed amazement, he asked whether I could "elegantize" the proof.

I in turn asked whether there was one or more equations in his proof that he wished to avoid. That was not his goal. Instead, being well aware of my interest in proof shortening, he asked me to produce a shorter proof, far shorter; I agreed to try. But before I hung up, he interrupted to refine his request, saying that he wished me to find a proof of length 100 or less—which I felt was either a joke or essentially absurd. Bill, however, knew I enjoyed challenges, especially with such a nice number as the goal. (He told me that he intended to place the proof in a monograph, I believe written with Padmanabhan, but wished to avoid the cited long proof, which took more than nineteen pages to display.)

Indeed, over the next couple of weeks—inspired by Bill, no doubt—with Bill's goal in mind, I developed methodology, much of which I still use, for proof shortening.

I had a fine start because of one of Bill's enhancements to OTTER, namely, *ancestor subsumption.* Ancestor subsumption is a procedure that compares derivation lengths to the same conclusion and prefers the strictly shorter. Yes, Bill's professionalism is exhibited: he designed and implemented this procedure solely because of my interest in finding "short" proofs.

Of course, you have anticipated what is now to be said. With that superb procedure and the methodology I was able to formulate, Bill did get his 100-step proof. (Bill was gratified in that the proof required only a bit more than three pages to display.) Quite a while later, I found a 94-step proof, which, as was true of the 100-step proof, relied on demodulation. Therefore, you could properly object to measuring its length when demodulation steps are not counted. So, I note that I did find a proof of length 147 in which demodulation is not used. (If you enjoy a challenge, from what I can recall, I know of no proof of length strictly less than 147 that relies on forward reasoning, with paramodulation, avoids demodulation, and proves DUAL-BA-3.) For those who enjoy history, along the way OTTER produced a proof of level 107. Bill's request was made, I believe, in May 2002; the 100-step proof was found perhaps in June; in November of that year, OTTER produced a 94-step proof.

At this point, some of you may be wondering why I have not yet cited one of Bill's greatest successes. I shall shortly. For now, I turn to another study Bill conducted in Boolean algebra. Specifically, he had formulated a technique for generating thousands of candidates, when seeking a single axiom, with the set filtered to avoid those that could not possibly succeed. Although I am not certain, I believe he employed this technique in his search in 2002 for single axioms for Boolean algebra in terms of **not** and **or**. Whether such is the case or not, Bill found the following ten (given in the notation he used), each of length 22.

```
~ (~ (x + y) + ~ z) + ~ (~ (~ u + u) + (~ z + x)) = z.
  % 13345 685 sec
~ (~ (~ (x + y) + z) + ~ (x + ~ (~ z + ~ (z + u)))) = z.
  % 20615   6 sec
~ (~ (x + y) + ~ z) + ~ (x + ~ (z + ~ (~ z + u))) = z.
  % 20629  19 sec
~ (~ (x + y) + ~ (~ (x + z) + ~ (~ y + ~ (y + u)))) = y
  % 20775  18 sec
~ (x + ~ y) + ~ (~ (x + z) + ~ (y + ~ (~ y + u))) = y.
  % 20787  80 sec
~ (~ (x + y) + ~ (~ (~ y + ~ (z + y)) + ~ (x + u))) = y.
  % 24070  28 sec
~ (x + ~ y) + ~ (~ (y + ~ (z + ~ y)) + ~ (x + u)) = y
  % 24086  44 sec
~ (~ (x + y) + ~ (~ (~ y + ~ (z + y)) + ~ (u + x))) = y.
  % 24412  40 sec
~ (x + ~ y) + ~ (~ (y + ~ (z + ~ y)) + ~ (u + x)) = y.
  % 24429  36 sec
~ (~ (~ (x + y) + z) + ~ (~ (~ z + ~ (u + z)) + y)) = z.
  % 24970  47 sec
```

Open question: For a far greater challenge that, if met, would merit a publication, one might study the open question concerning the possible existence of a single axiom for Boolean algebra in terms of disjunction and negation that has length strictly less than 22, the length of Bill's single axioms.

And now for the long-awaited highlight of Bill's study of Boolean algebra; even the New York Times was impressed, enough to write an article on the success under the byline of Gina Kolata [2]. Bill designed another automated reasoning program he called EQP, a program with built-in commutative/associative unification. Perhaps one reason he did so, perhaps the main one, was his intention of answering the decades-old Robbins algebra problem. A Robbins algebra is axiomatized with the following three axioms, where (in the notation Bill used in this study) + denotes union and the function n denotes complement.

```
x + y = y + x.                    % commutativity
(x + y) + z = x + (y + z).        % associativity
n(n(n(x) + y) + n(x + y)) = y.    % Robbins
```

Whether a Robbins algebra is a Boolean algebra was unknown for decades. S. Winker, I believe in the late 1970s, brought the problem to the Argonne researchers in automated reasoning. Not too long afterwards, the problem was attacked by various people throughout the world without success. Winker did supply a number of conditions that, if satisfied along with the three given axioms, sufficed to enable a deduction of the properties of a Boolean algebra. Stated a bit differently, if one of Winker's conditions was adjoined to the three given axioms, then you could prove that the resulting algebra is a Boolean algebra.

Bill's approach, with EQP, was to try to prove, from the three Robbins algebra axioms, one of Winker's conditions. Now, as far as I know, all the people attempting to answer the open question focusing on Robbins algebra believed that the key axiom (known as the Robbins axiom) was a misprint. Nevertheless, the question proved fascinating.

In almost eight days of computing, EQP deduced one of Winker's conditions, and, therefore, the question was no longer open: Indeed, every Robbins algebra is a Boolean algebra. Bill found the proof in late 1996, I believe [7]. Adding piquancy to the story was what occurred after Bill's monumental success. Specifically, Bill called Robbins to inform him of the result, commenting that the third axiom, as we thought we knew, had been published with an error. Robbins replied that, no, it was not an error, that he had conjectured that the three axioms—commutativity of union, associativity of union, and the third axiom (which focuses on a complicated expression involving complement and union)—might axiomatize Boolean algebra. Robbins was elated at Bill's information—how nice for the originator, then 81 years old, to learn of this result.

By way of what might be termed a post mortem to the story, Bill followed the tradition of the solid mathematician. Indeed, to find the earlier-cited ten single axioms for Boolean algebra in terms of **or** and **not**, he needed a target (as is typical) to show that each totally axiomatized the algebra. He did not choose to attempt to deduce the usual set of axioms for Boolean algebra. Instead, building on his success with Robbins algebra, he chose as target to deduce the Robbins basis, the set of three equations (given earlier) that characterize a Boolean algebra. A most charming action to take.

4 Logic Calculi and More Enhancements

Various areas of logic are often referred to as calculi. Bill and I spent some time with equivalential calculus. A possible obstacle that Bill noted was that, early in a run, depending on the hypothesis, deduced formulas could be complex (long in symbol count). If you permit your program to retain many complex formulas, the program can drown in deduced conclusions. On the other hand, if you place a tight bound on all deduced and retained information, then some key item might not be retained. In particular, if you are using a program to study, say, XHK or XHN, captured respectively by the following two clauses, and if you assign a large value to retained items, you will find early in the output rather lengthy deductions. (Predicates such as P can be thought of as meaning "is provable".)

```
P(e(x,e(e(y,z),e(e(x,z),y)))).  % XHK
P(e(x,e(e(y,z),e(e(z,x),y)))).  % XHN
```

With OTTER, if you were to assign the value, say, 35 to max_weight, you would find that too many conclusions were being retained. But if instead you assigned the value, say, 15, then you might be prevented from reaching your goal, perhaps of deducing some known single axiom for this area of logic. Yes, XHK and XHN each are single axioms.

Bill formulated and then encoded a feature that permits the program to retain, early in a run, very complex conclusions, but, shortly afterwards, discard such new conclusions. The following two commands show how it works.

```
assign(change_limit_after,100).
assign(new_max_weight,10).
```

The first of the two commands has OTTER, after 100 clauses have been chosen to initiate application of inference rules, change the assigned value to max_weight. The second command assigns the new value of 10 to max_weight. Summarizing, Bill designed and implemented a feature that allows the program to both have and eat its cake, to attack problems in which the retention of a so-called messy formula was required (early in the study) and yet not drown before the assignment was completed.

Bill came through again in a totally different context, the following. In the mid-1980s, I suggested that we at Argonne have access to a new strategy, namely, the *hot list strategy* [15]. You may have witnessed many, many times in a textbook on mathematics the phenomenon in which some assumption for the theorem in question is frequently cited. (A glance at the literature reveals that, in various proofs in logic, researchers often do this. Branden Fitelson pointed this out, noting he sometimes assigned the value 8 to the "heat" parameter. C. A. Meredith, in effect, used the hot list strategy.) The value assigned to heat denotes how much so-to-speak recursion is being asked for. Members of the hot list are used to *complete* applications of inference rules and not to *initiate* applications. The use of the hot list strategy thus enables an automated reasoning program to briefly consider a newly retained conclusion whose complexity might otherwise prevent its use for perhaps many CPU-hours. For example, if the textbook is proving that commutativity holds for rings in which the cube of x is x for all x, that property, $xxx = x$, may be used in many of the steps of the proof. With this theorem, you might well include $xxx = x$ in the hot list and assign the value of 1, or greater, to heat. I suggested for paramodulation—and this particularization is crucial to this story—that one of our programs offer to the user access to the hot list strategy. The researcher would choose, at the beginning, one or more items to be placed in the "hot list", a list that is often consulted in making additional deductions. The hot list strategy was shortly thereafter added, by Overbeek and Lusk, to the repertoire of ITP.

Years passed. Then I said to Bill it would be nice to have OTTER offer the hot list strategy, of course, in the context of paramodulation. Not too long after our discussion, Bill called me and asked that I test, in his so-called presence, his implementation of the strategy. However—and here again you witness his inventiveness—he informed me that he had implemented the strategy not just for paramodulation but for whatever inference rules were in use. Among such rules was hyperresolution. I was, at the time, conducting various studies in the use of condensed detachment (of course, relying on hyperresolution), an inference rule frequently used in the study of some logical calculus. The following clause captures that rule when studying, say, two-valued sentential or (classical propositional) calculus. (For OTTER, "-" denotes logical **not** and "|" denotes logical **or**.)

```
% condensed detachment
-P(i(x,y)) | -P(x) | P(y).
```

Immediately, I made a run to test Bill's version of the hot list strategy, in the context of deducing one 3-axiom system from another. I had at the time a 22-step proof. Astounding: With the hot list strategy, OTTER found a 21-step proof, a proof I had been seeking for a long time.

Again, I note that Bill had generalized what I had in mind in the mid-1980s by implementing the strategy for all inference rules in use. Further, he had added substantially to my idea by implementing the means to, in effect, apply the strategy recursively; you simply assign a value to heat that is 2 or greater, depending on how much recursion you wish. Even more, he implemented an incarnation that, if you chose to use it, had the program adjoin new elements to the hot list during a run—the dynamic hot list strategy.

If you wonder whether researchers outside Argonne have found Bill's various generalizations useful, I mention as one example Fitelson, who used the strategy heavily in various incarnations.

5 Group Theory

I do not know what motivated Bill, but he completed research in group theory that had been begun by the logician John Kalman, who studied the area in a manner quite different from what you might be familiar with. Kalman's study relied on condensed detachment and on the Sheffer stroke, denoted by the function d, captured with the following clause and the use of hyperresolution.

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

Technically, Kalman focused on the right-group calculus. (For the researcher who enjoys relationships, I note that equivalential calculus corresponds to Boolean groups, the R-calculus and the L-calculus to Abelian groups, and the right-group calculus to ordinary groups.) Kalman proved the following axiom system for the right-group calculus.

```
P(d(z,d(z,d(d(x,d(y,y)),x)))).                                   % R1
P(d(u,d(u,d(d(z,y),d(d(z,x),d(y,x)))))).                         % R2
P(d(v,d(v,d(d(u,d(z,y)),d(u,d(d(z,x),d(y,x))))))).               % R3
P(d(d(d(u,d(v,y)),d(z,d(v,x))),d(u,d(z,d(y,x))))).               % R4
P(d(d(v,d(z,d(u,d(y,x)))),d(d(v,d(x,u)),d(d(z,d(x,y)),x))))      % R5
```

As for interpretation, $d(x, y)$ can be thought of as the Sheffer stroke (the **nand** of x and y, and, when preceded by the predicate P, the formula is equivalent to the identity. The theorems of the right-group calculus are a proper subset of those of the R-calculus, which in turn are a proper subset of the theorems of equivalential calculus.

Bill initiated a study of the Kalman 5-axiom system—out of simple curiosity, perhaps. His study produced charming results. In particular, Bill proved that each of the second, third, and fourth of Kalman's five axioms provides a complete axiomatization (by itself) for the calculus [6]. With his model-generation program MACE [4], you can prove that the first of Kalman's axioms is too weak to serve as a single axiom; you can find a 3-element model to yield this result. Again, I offer you a challenge, an open question, actually. Indeed, the status of the fifth remains undetermined—at least, such was the case in late 2003. (Bill's study was conducted, I believe, in 2001.)

Now, if you wonder about Bill and so-called standard group theory, yes, he did study aspects of that area of abstract algebra. Indeed, relying (I am almost certain) on his method for generating thousands of promising candidates, he sought (I believe in 1991) single axioms of Boolean groups, groups of exponent 2. A group has exponent 2 if and only if the square of every element x is the identity e. He was successful.

Upon learning of his achievement, I suggested he seek single axioms for groups of exponent 3, groups in which the cube of every element x is the identity e. Again, he succeeded, presenting to me four interesting single axioms. One important aspect of research is that it leads to further discoveries. Bill's certainly did. I used one of his four single axioms for groups of exponent 3 to find single axioms for groups of exponent 5, 7, 9, ..., 19. To permit you, if you choose, to attempt to produce the pattern that generalizes to all odd exponents, I give you single axioms for exponents 7 and 9.

```
(f(x,f(x,f(x,f(f(x,f(x,f(x,f(f(x,y),z)))),f(e,f(z,f(z,f(z,
  f(z,f(z,z)))))))))) = y).
```

```
(f(x,f(x,f(x,f(x,f(f(x,f(x,f(x,f(x,f(f(x,y),z))))),f(e,
  f(z,f(z,f(z,f(z,f(z,f(z,f(z,z))))))))))))) = y).
```

If you were to seek a proof that the given equation for exponent 7 is in fact a single axiom for the variety of groups of exponent 7, you could seek four proofs, for each of the following given in negated form.

```
(f(f(a,b),c) != f(a,f(b,c))) | $ANS(assoc).
(f(a,f(a,f(a,f(a,f(a,f(a,a)))))) != e) | $ANS(exp7).
(f(e,a) != a) | $ANS(lid).
(f(a,e) != a) | $ANS(rid).
```

My citing of "all odd exponents" is appropriate; indeed, Ken Kunen, with his student Joan Hart, proved [1] that my generalization through exponent 19 continues for all odd n with n greater than or equal to 3. Bill, not to be outdone so-to-speak, produced his own generalization for 3, 5, 7, ..., a set of single axioms in which the identity e is not explicitly present. (Tarski noted without proof that no single axiom exists in which product, inverse, and identity are all explicitly present; Neumann supplied a proof.) For the curious, and for an example of OTTER's going where no researcher has gone before, its occasional application of an inference rule to a set of hypotheses one or more of which is most complex has led to breakthroughs such as a detailed proof focusing on groups of exponent 19, a proof an unaided researcher would have found most burdensome to produce in view of relying on equations with more than 700 symbols.

6 Other Areas of Abstract Algebra

In the early part of this twenty-first century, Bill collaborated in studies of abstract algebra with Robert Veroff, Padmanahban, and (for a little while) his student Michael Rose. These collaborations proved indeed profitable, as the following evidences.

I never asked Bill about his choice of variety to study. For example, did he deliberately study a variety and then study subvarieties? For a nice example, commutative groups form a subvariety of groups. Did he begin by explicitly considering the following chain of algebraic varieties: Boolean algebra (BA), modular ortholattices (MOL), orthomodular lattices (OML), ortholattices (OL), complemented lattices (CL), lattice theory (LT), and quasilattice theory (QLT)? If you begin with an equational basis for quasilattice theory in terms of meet and join, with the addition of axioms, (in steps) you obtain bases for LT, CL, OL, OML, MOL, then BA. Mainly he and Veroff did find single axioms for Boolean algebra in terms of the Sheffer stroke; but I leave that topic for another's paper. And, as cited, he made other contributions to Boolean algebra, most notably (from the world of long-standing open questions in mathematics) the splendid result focusing on Robbins algebra.

From Bill's many successes in algebra, I shall highlight a theorem from quasilattices and some results from lattice theory. The challenge offered by a theorem (denoted by Bill as QLT-3) in quasilattices was strikingly different from the search for new single axioms. Specifically, only model-theoretic proofs of the theorem existed—before OTTER, and Bill, entered the game to search for the missing axiomatic proof. The theorem asserts that the following self-dual equation can be used to specify distributivity, where "v" denotes join (union) and "^" denotes meet (intersection).

```
(((x ^ y) v z) ^ y) v (z ^ x) = (((x v y) ^ z) v y) ^ (z v x)
```

The first proof OTTER discovered has length 183. Access to the 183-step proof in turn prompted the search for yet another missing proof, still axiomatic, but simpler. With the various methodologies, a proof of length 108 was completed.

As for lattice theory, Bill did indeed make a monumental search. Bill's goal was not just some single axiom for lattice theory; after all, R. McKenzie had already devised a method that produces single axioms. The use of that method typically produces gigantic (in length) single axioms. Bill sought a single axiom of (undefined) "reasonable" length. Through a variety of techniques that keyed on the cited algorithm but incorporated the assistance of OTTER, he began with a single axiom of more than 1,000,000 symbols and eventually found a 79-symbol single axiom. The nature of his approach guaranteed that the result was sufficient, a theorem; no proof was needed. But, after a gap in time, Bill decided upon a new approach that would filter candidates, yielding equations that were promising. The goal was a far shorter single axiom.

Among the candidates, after one year, he found the following promising 29-symbol equation.

```
(((y v x)^x) v (((z^ (x v x)) v (u^x))^v))^ (w v ((v6 v x)^
  (x v v7))) = x.
```

Success would be his when and if a proof of some basis could be found. One of the standard bases (axiom systems) for lattice theory consists of the following set of six equations.

```
y ^ x = x ^ y
(x ^ y) ^ z = x ^ (y ^ z)
y v x = x v y
(x v y) v z = x v (y v z)
x ^ (x v y) = x
x v (x ^ y) = x
```

Bill, however, preferred a 4-basis as he sought single axioms. The (nonstandard) 4-basis he chose as target was the following.

```
y v (x^ (y^z)) = y
y^ (x v (y v z)) = y
((x^y) v (y^z)) v y = y
((x v y)^ (y v z))^y = y
```

And he won [9].

But still he was not finished; indeed, as I learned, Bill had a 135-step proof that the cited equation sufficed to axiomatize lattice theory. He called me and asked if I would seek a shorter proof—a proof, I concluded from knowing him so well, that would be far shorter in length than 135 applications of paramodulation. After not too long, I did find what I thought he would like, and called and told him so.

He asked about its length. I told him the proof was of length 50, which brought from Bill a surprising response. "Can I buy it?" "No", I said; "you can have it". And he published it, from what I know. (For the curious or for one who might decide to seek a shorter proof, the following might be of interest. Specifically, in July 2007, I returned to the study of that single axiom and found a 42-step proof.)

Bill also found a second 29-letter single axiom, the following.

```
(((y v x) ^ x) v (((z ^ (x v x)) v (u ^ x)) ^ v)) ^ (((w v x) ^
  (v6 v x)) v v7) = x.
```

From what I know, open questions still exist concerning Bill's two single axioms for lattice theory. In particular, is there a shorter single axiom, shorter than length 29? Are there other single axioms of length 29? What is the shortest single axiom for LT in terms of meet and join?

Bill contributed to other areas of mathematics, in geometry, with Padmanabhan, Rose, and Veroff, by finding single axioms for OL and OML [10].

7 Introducing Bill McCune

Who was Bill, really? Yes, of course, he was a fine researcher, attacking and answering various open questions from different fields of mathematics and logic. He designed, from my perspective, the most powerful automated reasoning program, OTTER, a program that I still use today, though it was designed and implemented mainly in 1988. But, what about the so-called nonprofessional side of Bill?

He played the piano; however, having never heard him play, I know not how well nor what type of music he played. He cooked, rather fancy sometimes; for example, he made his own mayonnaise. He immensely enjoyed eating; indeed, we shared many lunches, especially Thai food and Chinese food—not Americanized. He had a golden retriever he was deeply fond of. In the winter, they would go for long walks, the dog emerging from cold water with icicles on its coat. As evidence of Bill's deep ethical concerns, when the dog was diagnosed with cancer and Bill was given the choice of treatment or putting the dog down, he chose the latter. Yes, he did not want the dog to experience pain, and, instead, Bill made the supreme sacrifice of giving up his friend.

Had you worked at Argonne National Laboratory with Bill, sharing research experiments with him, you still would have known little about him. I would not say he was shy. Rather, his personal life was kept to himself. I did learn, after many, many years, that he was delighted with blueberries, wandering the trails in Maine, picking different varieties of wild blueberry. Did he enjoy dessert? Well, I had told him about my refrigerator, that it contained twenty-eight pints of ice cream. One late afternoon, Bill drove me home. Upon arriving, he brought up my claim of having twenty-eight pints of ice cream and expressed strong doubt about its accuracy. At my invitation, we entered my apartment, and I beckoned him to the refrigerator, indicating that he should investigate, which he did. After counting out loud, reaching the figure I had cited, he expressed amusement and surprise—and asked if he could try some. Of course, I told him to help himself. And he did, sampling, in one dish, three types of that frozen concoction.

Bill, as I said, was kind. Upon learning of my interests and also learning of how I worked, he wrote special programs for me. For example, he wrote a "subtract" program that takes as input two files and produces as output the set-theoretic difference. Another program he wrote for me interrogates an output file (from OTTER) containing numerous proofs, many of the same theorem, and returns in another file the shortest proof for each of the theorems proved in the experiment. Also, to enable the researcher to run without intervention a series of experiments, Bill wrote otter-loop and super-loop.

And, as many of you might know, Bill was aware of the chore some experienced when confronted with the array of choices OTTER offers. Perhaps because of his knowledge, he added to OTTER the "autonomous mode", a mode that removes from the user the need to make choices. In that mode, his program still proved to be of great assistance, often finding the proof(s) being sought.

Then there is the example of Bill's kindness combined with his thoroughness and professionalism. In particular, Kalman was writing a most detailed book about OTTER. That book promised to provide at the most formal level essentially most of what you would wish to know about OTTER. Before it was completed, however, Kalman notified me that he could not complete it because of a serious illness, one that eventually took his life. I promised him it would get finished. Indeed, I knew, or was almost certain, that I could count on Bill. And he did come through. After my informing him of the situation, Bill completed the book, enabling it to be delivered into Kalman's hands. A magnanimous gesture!

His sense of humor? The question is not whether Bill had one, but rather in what way it was expressed. Sometimes you gain insight into a person's view of life by having some information about that person's enjoyment of humor. I can say that, if only occasionally, some of my remarks did cause Bill to laugh heartily, to explode thunderously with enjoyment. For a different side of him, was he making a joke in one sense when he added to OTTER the command set(very_verbose)? That command has the program return copious, copious detail that enables you to, if you wish, check each inference, each application of demodulation, and such.

So long, Bill; you were unique; we do miss you. Mathematics, logic, computer science, and, even more, automated reasoning are each indebted to you, forever.

References

1. Hart, J., Kunen, K.: Single axioms for odd exponent groups. J. Automated Reasoning **14** (1995) 383–412
2. Kolata, G.: With major math proof, brute computers show flash of reasoning power. New York Times, December 10 (1996)
3. Lusk, E., McCune, W., Overbeek, R.: ITP at Argonne National Laboratory. In J. Siekmann, editor, Proceedings of the 8th International Conference on Automated Deduction, volume 230 of Lecture Notes in Computer Science, pages 697-698, Springer-Verlag, Berlin (1986)
4. McCune, W. Mace4 reference manual and guide. Technical Memorandum ANL/MCS-TM-264, August (2003)
5. McCune, W. Otter 1.0 users' guide. Tech. report ANL-88/44, Argonne National Laboratory, January (1989)
6. McCune, W. Single axioms for the left group and right group calculi. Notre Dame J. Formal Logic **34**(1) (1993) 132–139
7. McCune, W. Solution of the Robbins problem. J. Automated Reasoning **19**(3) (1997) 263–276
8. McCune, W., Padmanabhan, R. Automated deduction in equational logic and cubic curves. Volume 1095 of Lecture Notes in Computer Science (AI subseries). Springer-Verlag, Berlin (1996)
9. McCune, W., Padmanabhan, R. Single identities for lattice theory and for weakly associative lattices. Algebra Universalis **36**(4) (1996) 436–449.
10. McCune, W., Padmanabhan, R., Rose, M. A., Veroff, R. Automated discovery of single axioms for ortholattices. Algebra Universalis **52** (2005) 541–549
11. McCune, W., Wos, L. The absence and the presence of fixed point combinators. Theoretical Computer Science **87** (1991) 221–228
12. Smullyan, R. To mock a mockingbird, and other logic puzzles: Including an amazing adventure in combinatory logic. Knopf (1985)
13. Wos, L. The kernel strategy and its use for the study of combinatory logic. J. Automated Reasoning **10**(3) (1993) 287–343
14. Wos, L., Pieper, G. W. Automated reasoning and the discovery of missing and elegant proofs. Rinton Press (2003)
15. Wos, L., Pieper, G. W. The hot list strategy. J. Automated Reasoning **22**(1) (1999) 1–44