



Larry Wos: Visions of Automated Reasoning

Michael Beeson¹ · Maria Paola Bonacina² · Michael Kinyon³ ·
Geoff Sutcliffe⁴

Received: 14 June 2021 / Accepted: 24 January 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

This paper celebrates the scientific discoveries and the service to the automated reasoning community of Lawrence (Larry) T. Wos, who passed away in August 2020. The narrative covers Larry's most long-lasting ideas about inference rules and search strategies for theorem proving, his work on applications of theorem proving, and a collection of personal memories and anecdotes that let readers appreciate Larry's personality and enthusiasm for automated reasoning.

Keywords Larry Wos · Automated reasoning

1 Introduction

by Maria Paola Bonacina and Geoff Sutcliffe

Larry Wos was born on July 13, 1930, in Chicago, USA. His parents were Polish immigrants, a background that he remembered with pride. Wos was an accomplished mathematician and computer scientist, with an exceptional vision for theorem proving, an inexhaustible enthusiasm for its application in mathematics and logic, and a vivid curiosity for computer experiments. There was a certain playfulness about him, as if research was a bit like a game or puzzle, where he found joy in the discovery of patterns.

✉ Geoff Sutcliffe
geoff@cs.miami.edu

Michael Beeson
profbeeson@gmail.com

Maria Paola Bonacina
mariapaola.bonacina@univr.it

Michael Kinyon
mkinyon@math.du.edu

¹ San José State University, San José, USA

² Università degli Studi di Verona, Verona, Italy

³ University of Denver, Denver, USA

⁴ University of Miami, Coral Gables, USA

After studying at the University of Chicago (BS and MS) and at the University of Illinois at Urbana-Champaign (PhD), Wos spent his entire career at the Mathematics and Computer Science (MCS) Division of Argonne National Laboratory, in Argonne near Chicago. Argonne was the birth place of *resolution*: (John) Alan Robinson alternated summer jobs at Argonne and at Stanford University in the years 1961–1966, with an implementation of the Davis–Putnam procedure as his initial task in the summers at Argonne. It was at Argonne that Robinson worked on resolution and *unification* in 1962–1964, writing his milestone paper on “A Machine-Oriented Logic Based on the Resolution Principle” in 1963 [61]. Larry Wos, at Argonne since 1957, seized the moment and developed Argonne into the cradle of automated theorem proving.

Larry Wos understood right away the power of resolution, as well as the need to control it by appropriate strategies. Control strategies remained central in his research, and in his relentless experimentation with theorem provers. This was the reason why he preferred the name automated reasoning over automated deduction, feeling that “reasoning” better captures the mix of inference and search that constitutes theorem proving by machine. This was also a motivation for the name of the Association for Automated Reasoning. When the International Joint Conference on Automated Reasoning (IJCAR) conference series was being planned around 2000, Larry was delighted that the new event would use the phrase “Automated Reasoning.”

Larry Wos was also a pioneer in recognizing how crucial it was to start building equality into resolution. To this end, he developed the concept of *demodulation*, a first attempt at mechanizing the replacement of equals by equals. Even more importantly, George A. Robinson and Wos were the first to propose the *paramodulation* inference rule, marking the beginning of most of the ensuing research on paramodulation and superposition.

Not only was Larry Wos a discoverer of strategies and inference rules, he was also exceptional in leading the theorem proving group at Argonne, including Larry J. Henschen of Northwestern University, Ewing (Rusty) Lusk, the late William W. (Bill) McCune, Ross Overbeek, Robert (Bob) Veroff of (later) the University of New Mexico, Steven K. Winker, and in connecting with other scientists invited to Argonne, including Deepak Kapur, Hantao Zhang, and Maria Paola Bonacina. Wos also collaborated with numerous mathematicians, especially from the University of Chicago. He was fond of applying the Argonne provers, mainly Bill McCune’s OTTER, to search for proofs of mathematical theorems, constantly seeking shorter or more elegant proofs, as if he wanted to show that automated theorem proving can display not only brute force but also insightfulness and beauty. Many of his quests were reported in his books [82, 87, 93, 94], the latter two written with Gail W. Pieper, coordinator of writing and editing at MCS. These books show how keen he was on scientific writing, directed also to the non-specialists. He also maintained an online page¹ to share his theorem proving experiments. Under Larry’s leadership, the Argonne group played a key role in the establishment of the Conference on Automated Deduction (CADE), the Journal of Automated Reasoning (JAR), and the Association for Automated Reasoning (AAR).

Conference on Automated Deduction is the oldest conference on automated reasoning. Prior to the inception of CADE, papers on theorem proving and related topics could appear at only theory or artificial intelligence conferences. CADE provided a dedicated venue for presentation of research in this area. After a Symposium on Automatic Demonstration held at Rocquencourt, France, in 1968, an IEEE Workshop on Automated Theorem Proving was organized by Wos and Henschen at Argonne in 1975. According to Gérard Huet, use of the term “Automated Deduction” for the event began in 1977, and indeed a Workshop on

¹ <http://www.automatedreasoning.net>.

Automated Deduction met later that year in Cambridge, USA. The acronym “CADE” was suggested by Wolfgang Bibel and used for the first time in 1980 for the 5th Conference on Automated Deduction held in Les Arcs, France. (The prior events were retrospectively named CADE when the history of the conference was reconstructed in 2012.) CADE returned to Argonne in 1988, when CADE-9 was organized by Rusty Lusk and Ross Overbeek. It was at CADE-9 that Bill McCune released the first publicly available version of OTTER. Larry gave a memorable banquet speech at the Natural History Museum in Chicago, delving into a comparison of theorem proving with gambling, which he was fond of.

The Journal of Automated Reasoning, founded in 1985 with Wos as editor-in-chief, and Gail Pieper as managing editor, played the same role at the journal level as CADE did at the conference level. Prior to the creation of the JAR, journal papers on automated reasoning could be submitted to only theory, artificial intelligence, or symbolic computation journals. The foundation of a dedicated journal witnessed and furthered the growth of the field. The first issue of JAR in March 1985 included a foreword by Wos and an overview paper with several co-authors [92].

The Association for Automated Reasoning was established in 1983, with Larry Wos as president, Bill McCune as secretary, and Larry Henschen as treasurer. The first issue of the AAR Newsletter appeared in March 1983, with Gail Pieper as technical editor. Larry was a warm supporter of the AAR as a way of connecting people beyond conference attendance. He collaborated enthusiastically with all AAR secretaries (after Bill: Bob Veroff, Maria Paola Bonacina, Amy Felty, Wolfgang Ahrendt, Martin Giese, and currently Philipp Rümmer), and AAR Newsletter editors (after Gail: Jasmin Blanchette, and currently Sophie Tournet), relentlessly contributing reports of theorem proving experiments and theorem proving challenges to the newsletter.

Larry Wos was honored with the first Automated Theorem Proving Prize of the American Mathematical Society (with Steve Winker) in 1982 and with a *liber amicorum* in 1997 [75]. Wos won the first Herbrand Award in 1992, in recognition of his research and his role as a founder of automated reasoning. His acceptance speech in the CADE-11 proceedings is still inspiring today [83]. The Herbrand Award to Bill McCune in 2000 sealed the pioneering season when Argonne was a beacon for theorem proving, a season that would not have been possible without Larry Wos’ leadership. He will be remembered by many in automated reasoning and beyond.

2 The Inference Rules

by Maria Paola Bonacina

Larry Wos’ research program was centered on the design of *inference rules* and *control strategies*, the two components of a *theorem proving strategy*, and their refinement through experiments. In Wos’ work, the inspiration for new approaches came from a mix of theoretical quests and experimental trials. This style of conducting research emerges vividly from the written records of Wos’ greatest inventions that are surveyed in this section: the *set-of-support strategy* for resolution theorem proving [95], the *unit-resulting resolution* refinement of resolution [96], the *demodulation* inference rule for equational replacement [96], and the *paramodulation* inference rule for equality reasoning [60]. This section is subdivided into four subsections, each devoted to one of these major contributions to the foundations of automated theorem proving.

Precisely because in Wos' research the conception of new ideas was constantly interleaved with their experimental evaluation with theorem provers, it is important to mention the systems that Wos used and contributed to, before delving into his theoretical work. Wos used several theorem provers based on Robinson's *resolution* inference rule [61], contributing to their development (see [43] for a detailed account). The first one was P1, written by Wos with Dan Carson: Five variants of P1, called PG1-5, were used in the experiments described in Wos' seminal papers on the set of support strategy [95] and unit-resulting resolution [96].

The second prover, called RW1 from the initials of George A. Robinson and Wos, offered the first implementation of Wos' own inference rules, demodulation [96] and paramodulation [60]. Wos' collaboration with Ross Overbeek led to a third system named WOS1 [54], which implemented also *hyperresolution* [62], and formed the basis for the NIUTP1-7 series of Argonne provers. With this series the Argonne group pioneered the notion of building theorem provers from a toolkit of subroutines, leading to the development of the LMA library. Subsequently Wos used ITP [44], built from the LMA library, and AURA [68], which was a substantial step forward largely driven by Wos himself. The system that Wos used the most is undoubtedly Bill McCune's OTTER [49, 52], which remained Larry's favorite even after Bill had built the equational prover EQP [47, 48] from the *Otter Parts Store* library, which was LMA's successor, and later PROVER9 [46].

2.1 Resolution with Set of Support

A main motivation for the *set-of-support strategy* was the idea that a theorem proving strategy should minimize irrelevant inferences, that is, those that appear in the *derivation* but not in the *proof* [95]. The derivation is the sequence of all inferences performed by the strategy during the search, whereas the proof is the resolution proof tree with the empty clause \square at the root and the input clauses at the leaves. The idea was that a way to reduce irrelevant inferences is to make the strategy sensitive to the *goal* [95], where the goal is the negation of the conjecture φ to be proved from a set H of assumptions. Since resolution works *refutationally* with *clauses*, the formulae in H and $\neg\varphi$ are transformed into clausal form. The resulting set of clauses is the input to the theorem prover, which searches for a contradiction. At this point, however, the distinction between clauses that come from the assumptions and the clauses that come from the goal is lost. The set-of-support strategy addresses this by partitioning the input set of clauses into two sets A and SOS , where A contains the clauses in the clausal form of H and SOS contains the clauses in the clausal form of $\neg\varphi$. The set SOS is the *set of support*, and the clauses in the clausal form of $\neg\varphi$ are called *goal clauses*.

The key element in the set-of-support strategy came from Larry Wos' keen interest in the application of automated theorem proving to mathematics. If H is the axiomatization of a mathematical theory, it is known that H is consistent, and so is A , as transformation to clausal form preserves satisfiability. Wos' insight was that there is little point in generating resolvents from clauses in A , because that will not lead to a contradiction. Therefore, after a preprocessing step in which all factors of clauses in A are added to A , the set-of-support strategy applies resolution with the restriction that at least one of the two parents is in SOS . In other words, resolution steps where both parents are in A are forbidden. All generated resolvents are added to SOS , so that the SOS is expanded and A remains fixed. Clauses in SOS are said to be *supported*, and a resolution inference is supported if at least one parent is. The set-of-support strategy is goal-sensitive, in the sense that all clauses added to SOS have at least one goal clause as an ancestor. The set-of-support strategy inspired the definition

of *goal-sensitive strategies* [58]: A theorem proving strategy is *goal-sensitive* if it generates only clauses connected via inferences to goal clauses.

Wos proved that the set-of-support strategy preserves the refutational completeness of resolution and also suggested two incomplete refinements. The first one forbids generating a clause if the depth of its resolution proof tree is higher than a given bound. The second one forbids generating a clause if the number of its literals is higher than a given bound [95]. Indeed, Wos knew all too well the difficulty of simultaneously achieving completeness and efficiency in automated theorem proving. Throughout his amazingly long activity as an experimenter with OTTER, he was interested in trying incomplete strategies. Proving the refutational completeness of an inference system is indispensable, but it is only a beginning. Incomplete strategies can enable the theorem prover to prove more theorems, and can be a starting point for developing new complete strategies that are more efficient.

The impact of the set-of-support strategy has been vast and long-lasting. The *given-clause algorithm*, also known as the *closure algorithm*, was originally designed as a way to implement the set-of-support strategy in several Argonne provers, and then evolved into the central algorithm of automated theorem provers (e.g., OTTER, E [65], SPASS [79], VAMPIRE [42], WALDMEISTER [35], PROVER9, and ZIPPERPOSITION [27]), and as such it is still being investigated, e.g., [34, 66].

Resolution with set of support inspired *semantic resolution* [67], where A is the set of clauses satisfied by a fixed guiding interpretation. Semantic resolution is a *hyperinference* rule, as it combines multiple resolution inferences into one without generating intermediate resolvents, in order to generate only resolvents that are false in the guiding interpretation. Semantic resolution generalizes *hyperresolution* [62], where the guiding interpretation is either *all positive* (all positive literals are satisfied) or *all negative* (all negative literals are satisfied). The set-of-support strategy opened the way to goal-sensitive strategies, semantically guided strategies, and supported strategies with different definitions of set of support (see, e.g., [18, 20, 58] for surveys and [24, 25] for recent developments).

2.2 A Practical Hyperinference Rule: Unit-Resulting Resolution

When discussing features of inference rules Larry Wos mentioned *generality*—the use of *most general unifiers* as in resolution, *immediacy*—avoiding the generation of intermediate clauses, and *convergence*—preventing the generation of consequences of intermediate clauses [60]. While hyperresolution has these properties, and resolution with set of support is goal-sensitive, Wos considered these refinements of resolution still not powerful enough to conquer many interesting theorem proving problems [96].

A well-known disadvantage of (binary) resolution is that a resolvent inherits all the literals of its parents except the two literals resolved upon. Inferred clauses grow longer and longer, and hence more expensive to process. The only exception to this shortcoming, which was later studied as *duplication by combination* [58], is represented by *unit resolution*, which originated in the Davis–Putnam procedure [28]. In unit resolution one of the two parents is a *unit clause* (a clause made of a single literal), and the resolvent is one literal shorter than its non-unit parent.

Wos proposed a *unit-preference strategy* [89], where unit resolution steps have priority over other resolution steps, a very natural choice also when executing inferences manually. Also, Wos understood that it would be beneficial to extend as much as possible the *unit sections* of a derivation, as he called the series of unit resolution steps [96]. This requires in turn the availability of unit clauses. Thus, Wos devised *unit-resulting (UR) resolution* [45,

[96] as a hyperinference rule designed to produce unit clauses. Given a *nucleus* clause with $k + 1$ literals, where $k \geq 1$, and k unit *satellite* clauses, if the k literals of the satellites simultaneously unify with k literals of opposite sign in the nucleus, UR resolution generates the unit resolvent obtained by performing these resolution steps as one. If the nucleus is allowed to have only k literals, UR resolution can establish a contradiction.²

While semantic resolution performs hyperinferences in order to get a resolvent that is false in the guiding interpretation, and in hyperresolution the guiding interpretation is based on syntax (the sign of literals), UR resolution is an outright syntactic hyperinference rule, as the feature that a resolvent must have is purely syntactic, having only one literal. UR resolution represents well the style of Wos' research, his interest in what is practical, besides being refutationally complete. Indeed, UR resolution is not refutationally complete on its own, but in practice it is a useful addition to a refutationally complete inference system (e.g., [71]).

2.3 A First Foray into Equational Reasoning: Demodulation

Many theorem proving problems involve equality, and this is especially true for problems in mathematics, which was Larry Wos' favorite domain of application. A fundamental meta-theorem in mathematics known as *Birkhoff's theorem* says that replacing equals by equals is complete for equational reasoning. In other words, given a set of equations E and an equational conjecture $s \simeq t$, where \simeq is equality and all variables are implicitly universally quantified, $s \simeq t$ is a theorem of E if and only if it is possible to transform s into t , or t into s , by applying the equations in E as replacement rules. However, Birkhoff's theorem does not provide a mechanical procedure because it requires that the equations can be applied in either direction, which means that a mechanical procedure would loop. The quest to automate Birkhoff's theorem is one of the most fascinating stories in automated theorem proving. Wos broke new ground in this pursuit with the *demodulation* inference rule [96].

Wos was not specifically interested in purely equational problems as described above. His objective was to introduce equational replacement in the more general setting of clausal theorem proving by resolution. Given an *equality unit clause* $l \simeq r$, and a clause C with a subterm t , denoted $C[t]$, such that l matches t (i.e., $t = l\sigma$ for a substitution σ), the new clause $C[r\sigma]$ obtained by replacing t with $r\sigma$ is an *immediate modulant* of C . A k -*modulant*, for $k > 0$, is the outcome of k such replacement steps, and a *modulant* is a k -modulant for some k . The non-termination of replacing equals by equals in a mechanical rendering of Birkhoff's theorem arises also in the context of demodulation: A clause can have infinitely many modulants in general. However, for a fixed k , there are only finitely many k -modulants. Thus, Wos defined *k-modulation* as the generation of a resolvent of parents C_k and D_k , where C_k and D_k are k -modulants of clauses C and D . In order to capture equational replacement independent of resolution, and without the parameter k , he defined *demodulation* as replacement by a modulant, where each immediate modulant has strictly fewer symbols than its predecessor, and the final modulant has no immediate modulant with fewer symbols.

In theorem proving, the signature (the set of predicate, function, and constant symbols) is finite, but an infinite supply of variable symbols is assumed to be available. Every clause has its own variables and every clause has infinitely many *variants* (clauses differing only by a renaming of variables). Thus, there are infinitely many clauses with the same number of symbols. It follows that demodulation in its original definition cannot be considered, strictly speaking, a well-founded replacement rule.

² According to [16], the concept of UR resolution appeared independently in the NEU strategy [74], in the same year as [96].

Clearly, Wos was interested in a practical inference rule: A clause to be demodulated can be considered a *ground clause*, since a matching substitution and not a unifier is applied. A ground clause can be measured by its *size* (the number of occurrences of predicate, function, and constant symbols), and the sizes of ground clauses can thus be compared. Nevertheless, the size ordering does not allow the prover to apply demodulation when the two sides of the applied equation have the same number of symbols, because such an application does not decrease size. Furthermore, the size ordering may not allow the system to demodulate in the desired direction. For example, equations used to define a function symbol are meant to be applied so as to unfold the definition by replacing a term with a term of usually larger size, and demodulation driven by symbol count cannot do that. Thus, the problem of *well-founded demodulation* remained open. Moreover, well-founded demodulation alone cannot implement Birkhoff's theorem, because once the application of equations in E is restricted in order to be well-founded, it is necessary to *complete* the set of equations E by generating equations from equations in order to be able to prove any equational conjecture. Both these issues were solved by the *Knuth–Bendix completion procedure* [39, 41] for rewrite rules (equations that can be oriented by a well-founded ordering), and the *unfailing* or *ordered* completion procedure for equations [3–5, 37]. The fundamental ingredient of completion is a *well-founded ordering* on terms that is used to orient demodulation, called *simplification* or *rewriting*, and to define *superposition*, the more general inference rule that generates new equations from existing equations.

Independently of completion, and in the more general setting of clausal theorem proving by resolution, Wos approached the problem of designing a more general inference rule that would use equations to generate clauses from clauses: the *paramodulation* inference rule.

2.4 The True Beginning of Equational Reasoning: Paramodulation

Since resolution is refutationally complete for first-order logic, if the theorem proving problem involves equality, then the axioms of equality can be added to the input set: reflexivity, symmetry, transitivity, and substitutivity axioms for every function and predicate symbol in the signature of the problem. Unfortunately, these equality axioms are so general that their presence causes resolution to generate so many clauses that any resolution-based strategy becomes too inefficient to be practical in most cases. In order to address this problem, Larry Wos proposed *paramodulation* [60] as a generalization of resolution where equality is built-in: Given a clause $l \simeq r \vee C$ where one of the literals is an equation $l \simeq r$, and a clause $D[t]$ with a subterm t such that l and t unify with most general unifier σ , *paramodulation* generates the *paramodulant* $(C \vee D[r])\sigma$. The clause $l \simeq r \vee C$ and the literal $l \simeq r$ are called the clause and literal *paramodulated from*, whereas the clause D and the literal containing t are called the clause and the literal *paramodulated into*.

The appearance of paramodulation was nothing short of revolutionary. However, the proof of refutational completeness required paramodulation into variable terms, and also augmenting the input set with the reflexivity axiom ($x \simeq x$) and *functionally reflexive axioms* (the instances of reflexivity of the form $f(\bar{x}) \simeq f(\bar{x})$, for all function symbols f in the signature of the input clause set). Wos and Robinson conjectured that both requirements could be dropped, a conjecture that became known as the *Wos–Robinson conjecture*. Since a variable unifies with any term, paramodulation into variables makes the inference rule exceedingly prolific, as do the functionally reflexive axioms. The presence of these axioms meant that equality was still only partially built into the inference system. Several researchers endeavored for years to settle the Wos–Robinson conjecture and the related problem of merging resolution and

paramodulation on the one hand, with completion by superposition and simplification on the other, to obtain refutationally complete inference systems for first-order logic with equality [26, 36, 38, 56, 63, 64]. The resulting inference systems combine resolution, paramodulation, and superposition, with well-founded demodulation and subsumption—subsumption is another inference rule that Was deemed fundamental [91].

These inference systems have been called *completion-based*, *rewrite-based*, *saturation-based*, or *ordering-based*, given the key role played by well-founded orderings on terms, literals, and clauses [7, 18, 19, 53, 57]. They were implemented first in OTTER and then in most subsequent theorem provers for first-order logic with equality, up to those that represent the current state of the art (e.g., E, SPASS, VAMPIRE, WALDMEISTER, and ZIPPERPOSITION). These inference systems specialize to unailing completion in the purely equational case [6, 22, 64]. The growth of the inference system beyond resolution contributed to the evolution of the given-clause algorithm from an implementation of the set of support strategy into a general algorithm for implementing multiple strategies. Indeed, the set of support strategy is not complete in general in the presence of equality, unless the complement of the set of support is saturated with respect to the inference system in a preprocessing phase [6], which defeats the spirit of the strategy.

As additional evidence of the amazing impact of the theorem proving approach that Was pioneered with paramodulation and demodulation, it suffices to mention that these inference systems yield decision procedures (e.g., [2, 31, 33]), get integrated with other reasoning paradigms (e.g., [23, 29, 59]), form the basis of approaches to parallel theorem proving (e.g., [17, 21] and [19] for a recent survey), and are generalized to higher-order logic as in *lambda-superposition* [11, 12, 77, 78] and *combinatory superposition* [13–15].

3 The Applications

by Michael Kinyon

This section discusses some of the applications of automated reasoning that Larry Was worked on. The presentation gives a representative sample of his many interests, in topical order, rather than an exhaustive survey in chronological order. Particular attention is paid to Was' ongoing interest in proof simplification.

3.1 Logic

Much of Larry Was' direct work in the application of automated reasoning (especially with OTTER) was within the realm of logic. This subsection gives an overview of his and his collaborators' work.

Starting in the area of propositional calculus, Was showed that for Meredith's single axiom there is a proof (in fact several) of Łukasiewicz's three-axiom system that uses only condensed detachment [88]. Meredith's original 38-step proof uses condensed detachment but also uses substitution and (regular) detachment. Was originally found a 41-step proof using only condensed detachment by checking Meredith's proof and then went on to find "fully automated" proofs (meaning independent of other proofs, particularly Meredith's). A notable observation was that blocking double negation from occurring led to more efficient searches. At about the same time Fitelson and Was were able to find, among other results, condensed detachment proofs that Łukasiewicz's 23-symbol single axiom axiomatizes propositional calculus [32, 90]. (Alluding to the discussion in Sect. 3.3, some of this work is devoted to

consideration of how to find simple proofs.) Later, Beeson, Veroff, and Wos returned their attention to the goal of blocking double negations and looked for conditions guaranteeing the existence of double-negation-free proofs when the conclusion to be proved is also free of double negations. They found conditions that address this in general, and then focused on the Łukasiewicz three-axiom system as one whose use for theorems with double-negation-free conclusions guarantees the existence of a double-negation-free proof [8]. In the broader context of Boolean algebra, McCune, Veroff, Fitelson, Harris, Feist, and Wos found shortest single axioms, including identities of length 22 in terms of conjunction and negation, and identities of length 15 in terms of the Sheffer stroke [50].

Turning to classical equivalential calculus (the calculus of equivalence relations), Wos, Winker, Veroff, Smith, and Henschen showed that four of seven candidate shortest single axioms do not have sufficient strength to serve as single axioms [100]. Later, Wos, Ulrich, and Fitelson settled in the affirmative the open question of whether the formula $XCB = e(x, e(e(e(x, y), e(z, y)), z))$ is a single axiom under the inference rules of detachment and substitution [98, 99]. This was the only remaining unresolved case in the list of shortest possible axioms.

In combinatory logic, Wos used a unique strategy, which he dubbed the kernel strategy, to solve several open problems regarding whether or not certain combinators have fixed point properties [84]. In the same paper, he posed many problems which remain open.

In the area of non-classical logics, Ernst, Fitelson, Harris, and Wos found a concise three-clause axiomatization of the implicational fragment RM_{\rightarrow} of the Dunn–McCall system RM , a cousin of relevance logic. Ernst, Fitelson, Harris, and Wos also found several shortest possible axiomatizations for the strict implicational fragments of the modal logics $S4$ and $S5$ [30].

3.2 Tarskian Geometry

Tarskian geometry is a single-sorted axiom system that deals with only congruence and betweenness relations [9]. This is in contrast to Hilbert-style axiom systems that are less economical. Tarskian geometry is of more interest in metamathematical or formal (especially automated) investigations. Michael Beeson and Larry Wos used OTTER to prove theorems in Tarskian geometry. The papers were not the first to apply automated reasoning to Tarskian geometry, but of particular note were their solutions to four challenge problems to prove the following properties without any parallel axiom or line–circle continuity: Every line segment has a midpoint; every segment is the base of some isosceles triangle; the outer Pasch axiom (assuming inner Pasch as an axiom); and the first outer connectivity property of betweenness. In addition, Beeson and Wos found OTTER proofs of all of Hilbert’s axioms starting from Tarski’s axioms [10].

3.3 Proof Simplification

Anyone who encountered Larry Wos in a professional context very quickly learned of his long-standing interest in finding elegant and simple proofs. There are two likely reasons for this interest. First, Wos was a mathematician before he began his seminal work in automated reasoning. The culture of mathematics, especially pure mathematics of the sort in which he originally did research, has always placed a high aesthetic value on elegant proofs. A well-known example of this aesthetic was expressed by the late Paul Erdős, who always valued what he called “proofs from the Book,” meaning the Book in which God kept the best proof of

every theorem [1]. (Erdős, incidentally, doubted God's existence but believed in the existence of the Book.) Although Wos apparently never expressed his views on elegant proofs quite so colorfully, he was certainly steeped in the same mathematical milieu. Second, as noted in [40], an automated theorem prover will report the *first* proof it finds, but such a proof is rarely optimal or elegant by any measure. These two observations, Wos' background as a mathematician and the standard behavior of automated theorem provers, suggest why Wos maintained an interest in finding elegant and simple proofs.

The quest for elegant proofs leads to the question of how to measure the simplicity of a proof (generated by an automated theorem prover). The measure that Wos used most often was the number of inference steps: Shorter proofs are simpler. There are a few tacit assumptions being made here when comparing proofs for relative simplicity. First, the axioms of the proved theorems are presumed to be the same. Second, the rules of inference are assumed to be fixed. The latter point also means, for example, that it does not make sense to compare a proof using hyperresolution with a proof that replaces each hyperresolution step with multiple binary resolution steps. As another possible measure, forward proofs might be considered to be simpler than proofs that contain steps reasoning backwards from the denial of the conjecture. It is certainly conceivable that proofs that mix forward and backward reasoning could be shorter than strictly forward proofs, and thus this prescription reveals that sometimes other aesthetic considerations can override the desire for short proofs. Note that strictly forward and mixed proofs are never compared, since the latter usually include inference rules such as UR resolution that are not allowed in the former. Third, no secondary inferences, rewriting in particular, are allowed in a simple proof. This tacit rule mostly applies to problems with equality. In OTTER/PROVER9 jargon, back demodulation must be turned off so that rewrites have to be replaced with "top-level" paramodulations. This rule may seem somewhat arbitrarily restrictive, and it is generally true that a proof with rewrites can have a more structured appearance than a proof without rewrites. However, it is almost always the case that proofs with rewrites are longer than similar proofs where all inferences are primary. Thus the restriction is somewhat natural if the primary measure of simplicity is proof length.

None of this is to suggest that Wos was not aware of the difficulty of coming up with a robust measure of the simplicity of a proof. An exploration of other possible factors was discussed in his collaboration with Rüdiger Thiele [73, 97]. The following is a summary of a similar discussion in [40]: If a proof is visualized as a directed graph with clauses as vertices and an edge connects clause *A* to clause *B* if *A* is an immediate parent of the second, then the complexity of that graph is an interesting measure of proof simplicity. One would intuitively expect a proof with fewer edges emanating from the axioms to be simpler than a proof with more such edges. Experimentally, this measure of simplicity does seem to be rather well correlated with proof length, because when the proof clauses are arranged in a sequence, fewer paths emanating from the axioms generally means there are fewer inference steps.

Other measures of proof simplicity could be based on clause size, for example, the maximum or average clause size. This measure can, however, conflict with proof length. Wos did many experiments in which he used OTTER to find shorter proofs of the Robbins problem originally solved by McCune. The clauses in the shorter proofs are very large (according to a talk given by Wos at the 2004 Argonne Workshop on Automated Deduction and Mathematics). Thus, there is a simplicity trade-off: Is a shorter proof in which the clauses themselves are more complex really a simpler proof? Wos was certainly aware of this issue and others like it, but was just as aware that the overall problem of determining criteria for simplicity and how such criteria interact with each other requires much more investigation.

One of OTTER's "fringe features," ancestor subsumption, seems to have been implemented by McCune in response to Wos' desire to search for short proofs (cf. [93], p. 299). Clause A ancestor subsumes clause B if either (1) A properly subsumes B in the usual sense, or (2) A and B are variants and the derivation length of A is shorter than that of B . There seems to be little to no evidence that ancestor subsumption helps in basic proof search. It does, however, help quite a bit in searching for short proofs.

It is probably fair to say that Wos' interests in proof simplification have not generally been shared by many in the automated reasoning community. The reasons for this are quite practical: Developing automated theorem provers that can find proofs faster and more efficiently seems to yield a better return on invested time than developing provers that can find elegant proofs (e.g., the annual CADE ATP system competition does not have an award for shortest or simplest proofs).

Wos' interest in proof simplification seems to have been partially historically justified for him by similar interests among earlier logicians, such as Meredith and Prior. Wos found even more historical justification in the well-known cancelled 24th Problem of David Hilbert. This led to Wos' collaboration with Rüdiger Thiele, mentioned above. In 2000 Thiele discovered from Hilbert's unpublished notebooks that he had planned to include a 24th problem in addition to the famous 23 problems that he discussed in his Paris lecture of 1900. The lecture was already long, and apparently Hilbert was unable to formulate the problem quite to his liking, so he decided to omit it. The wording of the cancelled problem exactly matches Wos' interests [72]: "Criteria of simplicity, or proof of the greatest simplicity of certain proofs. Develop a theory of the method of proof in mathematics in general. Under a given set of conditions there can be but one simplest proof."

3.4 Pure Proofs

As part of his general interest in the elegance of proofs, which was mostly expressed in terms of proof simplification, Larry Wos also worked on other possible ways that elegance could be expressed. For instance, Wos was very interested in the notion of what he called a "pure" proof [85, 86]. This specifically arises when proving a theorem of the following form:

Theorem 1 *Assume [various hypotheses]. Then, the following are equivalent.*

1. *Statement 1*
2. *Statement 2*
3. *Statement 3*
4. *etc.*

In practice it might be proved, for examples, that Statements 1 and 2 are equivalent, that together they imply Statement 3, and so on, perhaps via a very tortured path, eventually coming back to some Statement implying Statement 1 (or 2). To Wos, a *pure* proof of one of the implications, say Statement 2 implies Statement 3, is one that does not prove any other Statement along the way. The challenge in automated theorem proving is to steer the prover in such a way that in the course of proving an implication between Statements it does not prove one of the other Statements.

Wos discussed the specific example of the thirteen shortest single axioms for equivalential calculus, subject to condensed detachment being used as the sole rule of inference [85]. A single application of condensed detachment to the (shortest single) axiom known as $P4$ with itself yields the (shortest single) axiom $P5$, and two applications of condensed detachment beginning with $P5$ yields $P4$. This means that starting with $P4$ cannot yield a pure proof

of any of the axioms other than $P5$ (because one application of condensed detachment must yield $P5$). Thus, a derivation of $P5$ must be present in the proofs of all the other axioms.

Wos also considered pure proofs for the four Moufang identities from loop theory [86]. Any one of the identities can be used to define the variety of Moufang loops; that is, the identities are all equivalent. With four identities, this means there are 12 pure proofs to seek out, and Wos found them all with OTTER. See also Veroff's solution of Wos' pure proof challenge [76].

It is interesting to note that Wos' dissertation advisor, Reinhold Baer, had a strong preference for "the following are equivalent" types of theorems; they can be found throughout his work. It is amusing to speculate that Wos might have inherited some of his interest in such theorems from his advisor.

3.5 Miscellaneous

This subsection collects some of Larry Wos' "one-shot" work that doesn't seem to fit in any of the other subsections.

- In [81], Winker, Wos, and Lusk settled an open problem first proposed by Kaplansky in the theory of semigroups: Does there exist a finite semigroup (a set with an associative binary operation) with an antiautomorphism (a permutation f such that $f(xy) = f(y)f(x)$ for all x, y) but no involutions (antiautomorphisms satisfying $f^2 = \text{id}$)? What made the problem particularly challenging was not the prescription of the given antiautomorphism of order higher than 2, but rather specifying the conditions that guarantee that there are no involutions at all. The unnamed Argonne ATP system used in the paper did not solve the problem directly, but rather assisted in the generation of models that eventually led to a "by hand" solution.
- In [69], Smith and Wos solved some problems in the theory of Jordan rings. The mathematician who proposed the problems to them thought that automated reasoning would be helpful, but amusingly, it turned out that the solved problems were handled simply by some clever algorithms and intense computation. See Wos' reminiscence on pp. 395-396 of [93].
- In [80], Winker and Wos found models and counterexamples to conjectures regarding the independence of axioms in ternary Boolean algebras, which are essentially Boolean algebras axiomatized by a single ternary operation. Their approach used then existing automated deduction tools and an iterative procedure to add properties to a potential model until it was completed.
- In [55], Parrello, Kabat, and Wos studied the job-shop scheduling problem for production of cars. The problem is known to be NP-complete, so rather than seek optimal solutions the authors used the ITP system to study "good" sequences of cars, as measured by economic considerations.
- In [51], McCune and Wos studied a known theorem in combinatorial logic: The strong fixed point property is satisfied in a system that contains the B and W combinators. They used the colorful formulation due to R.M. Smullyan [70]: A sage exists in a forest that contains both a bluebird and a warbler. One sage was known to exist already; McCune and Wos found four more with the assistance of ITP and some of its extensions.

4 The Persona

by Michael Beeson

with contributions from Jim Boyle, Branden Fitelson, Deepak Kapur, Ranganathan Padmanabhan, Gail W. Pieper, and Brian Smith

Larry knew a lot of people. He did not make much distinction between personal and scientific relations; he treated everyone as a whole person. He could entertain a table (or two) at dinner, a scientific lecture audience, or his bowling companions. He used the telephone a lot, carrying on both scientific and personal conversations with great intensity. He was never in a hurry to finish these conversations. He was always in a hurry to make progress on the next proof. I sent Larry a challenge problem in October 2000. I did not know him then, so I was surprised to get a phone call. These phone calls went on for the next twenty years. When we were working on a project together, they were often daily. This was the way Larry worked. Listen, for example, to Branden Fitelson's story:

I was a graduate student at UW-Madison, and (through Ken Kunen – who also sadly passed recently) I heard about Otter. I used it to solve a historical problem in sentential logic (the dependence of one of Frege's *Begriffsschrift* axioms for sentential logic, which was first pointed out by Łukasiewicz). I cold-emailed Larry with my solution of the problem. He immediately replied with his phone number. We talked more than three hours that first day, and we had many, many long phone conversations regularly for the next eight years or so. He was so encouraging and energetic. We published several papers on open problems in sentential logics during those years. I will cherish those memories. Larry was one of a kind.

Larry did not like to waste time. He would start talking as soon as I picked up the phone—no wasting time with “Hello, this is Larry.” He said that if I identified myself I would be insulting his ability to recognize voices. If my wife picked up the phone, he would say, “Is he there?” On the other hand, he was always willing to spend time on personal issues. But he would never discuss politics or the news: “why should I waste my time thinking about something I can't do anything about?” He also wasn't much interested in discussing theoretical aspects of automated reasoning, even though he had invented several fundamental methods. If you wanted to get his attention, you had to send him an OTTER input file! Nothing else would work. Send him a file that should get a proof, but didn't; or better yet, one that did get a proof, but should get a shorter proof. You would hear from him very soon.

Another person with whom Larry worked was “RP,” that is, Ranganathan Padmanabhan. Here is his story of his work with Larry:

Thanks to the suggestions of Dr. Stanley Burris (University of Waterloo) and Dr. David Kelly (University of Manitoba), I wrote my first e-mail to Larry Wos in 1993, proposing some problems of equational nature. I first met Larry in 1993 during an Argonne Workshop. After that, I visited Argonne almost every summer. These workshops were very informal. Each participant would describe the problem he or she was currently working on, and others participated in the discussion. I was perhaps the only non-computer scientist in the group. In the beginning, I never actually used Otter. Using all sorts of mathematical tools, I manufactured several theorems of first-order logic relevant to my research (lattices, quasigroups, group theory, geometry) and passed on these theorems to Bill (McCune) during these workshops and subsequently through emails. It was Bill (and later Bob Veroff with Prover9) who fine-tuned the various settings, proved the theorems and emailed me back the proofs in the form I liked. This

went on for some three years. All my conjectures turned out to be true. Towards the end of one such workshop, Larry exclaimed with a gleaming smile, “I know how RP gets these conjectures. He already has proofs using some kind of maximal principle, Zorn’s Lemma, axiom of choice or some such fancy second order tool and he wants to get a real down-to-earth equational proof, am I right, RP?”. Well, he was 100 percent right. Larry had a clear perception of the subtle distinction between a first-order proof and a second-order proof and what kind of hypotheses were ideal candidates for Otter or Prover9. On several occasions, Larry would “elegatize” the equational proofs by simplifying or reducing the length of the proof etc. My monograph with McCune and subsequent papers with Bill and Bob Veroff may be described as corollaries to our interactions and discussions with Larry Vos.

Larry always liked Indian food, especially when we went together for lunch or dinner during the Argonne workshops. His favorite choice would be Saag Paneer, a vegetarian dish. Was it because of his epicurean taste or was it because of his innate desire to accommodate my food habits?

I can testify that Larry liked Indian food even when RP was not there. In April 2004, his wife Nancy died, and I offered to fly out to Chicago and keep him company for a few days. He accepted that offer, and I cooked several good Indian meals in his kitchen. During that visit, I prodded Larry repeatedly to get out of the house and go for a walk. Reluctantly he did so, but he took his revenge on me. He managed to time his conversation on the elevator so that as we passed through the apartment house lobby (with several others and the doorman present) he could make a loud reference to my “criminal past” and my upcoming appointment with my parole officer! He was so amused at my supposed discomfiture. When I congratulated him on his timing, he was amused and said he had tried to make it appear an accident. On that walk, I told him, “Larry, life is like white-water rafting: sometimes you have to paddle like crazy, but other times, you should just take it easy and float downstream.” “This is one of those times,” I said. I will never forget his reply: “Michael, I’ve been paddling as hard as I can all my life, and I’m not about to stop now!”. This was very true; over the years, I heard how Larry had struggled to overcome his lack of vision, something he only talked about after knowing me very well.

In the days before I knew him Larry had played a lot of poker, sometimes for very large stakes. He played regularly with some rather shady characters. He described these games for me in some detail. He taught me that poker is a mental game, it’s about understanding how the other players think, how they make decisions. Of course you had to master the probabilities, but that was child’s play compared to the real game, which was all psychological. Although Larry could not see, he was very good at interpreting tones of voice, small sounds of clothing rustling, etc. Sometimes his fellow players would insist on writing down their bets, and having someone else read them, because “Vos gets too much information” if they speak. These people grew to respect Larry. He told me, “If you ever need someone bumped off, or just kneecapped, I can help you with that.” By the time he said that to me, I had realized that Vos loved to say things that were just at the borderline of truth—if I couldn’t decide if what he said was true or just fiction, that was what he was aiming for.

Poker was not the only thing on which Vos gambled. He bet on sports (at least baseball and football) and on horse races. For example, in November 2006, he asked me to help him find streaming audio for the Breeder’s Cup race, on which he “had some bets.” He would typically bet on the “spread” rather than just the raw win-or-loss outcome of games, and bookies would call him sometimes to help estimate the spread, unless of course, that was just something he told me to see if I would believe it or not.

When it came time to write up some of our work for publication, I found that Larry had some fixed ideas about style. For example, it had to be “There exists” and never just “There is.” He was also very particular about the correct use of “which” and “that.” He was a very careful proofreader. His favorite story about editing was one in which Tarski had quotation marks inside the period at the end of a sentence, throughout his paper, and the editor changed them, citing numerous authorities. Tarski changed them all back, telling the editor, “I am Tarski!”. Tarski was Larry’s ideal in regard to editors. The only person whose writing advice he would accept was Gail Pieper. Here is her story:

Larry and I were colleagues and friends for over 40 years. When I first was introduced to him as the Mathematics and Computer Science Division editor, we started discussing a paper of his I had just looked at. I suggested that he change the em dash to a simple comma, maintaining in my most “professional” voice that the em dash should be reserved for emphasis. “Exactly!” he replied. (I lost that round.) But Larry did respect style and grammar and word choice, and he would look carefully at each of my editing comments. One day I remember he called me and groaned, “Do you know how many changes you are suggesting I consider? One hundred and seventy!”. “Exactly!” I replied. (I won that round.) Counting needed changes was typical of Larry’s interest in numbers and perhaps explains his many decades of research focusing on finding shorter proofs. He also liked challenges, and he frequently included one or more challenge problems in his “President’s Column” in the AAR Newsletter. I once took advantage of these two interests when he called in delight one day to tell me he had reduced a proof of 136 steps to 103 (I am not sure of the exact number). My response was to point out that the number 100 would be much more satisfying. He agreed, and a few days later I received another call. He had succeeded, and we both shared in his success in meeting my challenge.

Larry could not see. Of course that was a central fact in his life experience, but he hated to be characterized by that limitation. The word “blind” was to him like the “N-word” is to those with dark skin. It took time (on my part) and effort (on his part) for me to understand the discrimination that the unsighted (his preferred word) face. Here is what his long-time colleague (since 1974–1975) at Argonne National Laboratory, Brian Smith, has to say about how Larry dealt with his lack of vision:

Working with Larry was a challenging but most enjoyable experience. Although he was blind, Larry never let you mention that about himself. He would use words like “look” and “see” that surprised you when he uttered them. On the other hand, you would be writing something on the blackboard to explain something, and he would correct what you had written as if he could see it. He would walk the halls of our office and turn at the correct places to get to a meeting or a seminar in the building without assistance. The only time he would accept assistance was to walk outside to the cafeteria roughly a quarter of a mile from our offices. But he was always quick to tell you about a curb or obstruction as if instead he was leading you rather than the other way around.

The only context in which Larry ever used the word “blind” was in the phrase “blind bowling.” I was astonished to learn that there is such a sport, but he set aside one night a week for it and was proud that he was the best in the league. Brian Smith has a story about that:

One of the most enjoyable experiences I had was to have him invite me to be on his bowling team of blind bowlers. It was a blind bowling league in which they would

request a sighted person to join the team to write down the scores and in addition tell them what pins were left standing after they bowled the first ball. They bowled by using a portable guide rail that the blind bowlers would hold onto with their left hand while bowling the ball with their right hand. Larry's ten pin bowling average was in the mid 140s if I recall correctly, but most other blind bowlers would bowl on average in the 120s. So I was there to keep score and call the standing pins after the first ball. I would announce 8 pins down on the first ball with pins 1 and 2 left standing and then on the bench behind me I would hear the correction that only 7 pins were down and pin 5 was also back there, still standing. If I said mistakenly the 1 and 2 pin was still standing, I would hear the correction that it was the 1 and 3 pin. And of course he was always right. He could tell from the sound of the ball hitting the pins what was likely left standing, and he was always right. And of course that experience convinced him I was more blind than he was and he would always challenge me playfully about my eyesight.

In his professional work at the laboratory, Larry had to find a way to interface with a computer; when he first started, the usual way was via paper tape! Brian Smith describes what happened:

In the late 1960s the engineers at Argonne modified a paper tape writer so that it embossed the tape rather than punched holes in the paper tape. The tape was wide enough to have up to 8 lumps embossed across the tape, instead of the usual 8 holes. I learned the tape code. This is how he would read computer output, and how he would read programs he wrote. Later in the 1980s, the lab bought a Braille printer that would print output for him. He would carry rolls of paper tape around with him in his pocket, and walk the halls of building reading the tapes by rubbing his fingers across the tape.

By the time I stayed in his apartment in 2004, he had a teletype with Braille printer in his bedroom. The lights were never turned on, unless I turned them on; I watched in awe as Larry sat in the dark, running OTTER on the lab's computers.

Jim Boyle was another colleague of Larry at Argonne, who knew him even before Brian Smith did. Here is part of his story about working with Larry, starting from his first day at Argonne in 1967. Jim wrote it in the form of a "letter to Larry":

I was given an office one down the hall from yours, and I was duly ensconced in it with the door closed, trying to work. I remember hearing loud talk and laughter in the hallway outside my door and wondering how I could get any thinking done while that was going on. Finally, I stepped out into the hall to see if I could tone it down, and there you were, talking and joking with another member of the Division. Before I could finish saying something (I should say, complaining) about the noise, you had included me in the conversation and were telling a story that left me in stitches! A bit later I went into your office, and I found you, sitting at a teletype machine, complete with a paper tape reader and punch attached. Soon the machine began to chatter, spewing both paper and tape, and I watched you grab the end of the tape and run your fingers over it as it came out of the punch. I asked what you were doing, and you told me you were reading it, reading the output from the latest run of your program.

Larry loved to play with people, telling them things just to see how they would react. Eventually, I learned to play this game too, reacting in unexpected ways, or telling him

things “on the edge of believability.” Here is one of Jim Boyle’s favorite stories illustrating how Larry did things like this:

We all knew Larry as Lawrence T. Wos, but when I asked what the ‘T’ stood for, he answered with this story, about playing poker with a group of friends when he was a student at the University of Chicago. One of the players was Bernie, who was eighteen – younger than the other players and a tad naive. At some point, the game broke for dinner, and Bernie sat down next to Larry. After a bit, he asked, “Hey, Larry, I hear your middle initial is ‘T’, what does that stand for?”. “Just ‘T’, Bernie, it just stands for ‘T’”. “Awww, c’mon Larry, it must stand for something!”. “No, Bernie, it’s just ‘T’”. Not one to take a hint, Bernie began to bounce in his chair, and repeated, “Please tell me what it stands for!”. After Bernie had demanded this several times, Larry finally said, with a small smile, “Well, you know my mother is a Latin scholar, and she chose a Latin word for my middle name. But, it’s embarrassing – I never tell anyone what it is”. Bernie continued to wheedle, “Please, pretty please, Larry, tell me what it is!”. Finally, Larry said, “ALL RIGHT! She named me Lawrence Terraponesta Wos”. Amidst general laughter at the table, Bernie exclaimed, “Terraponesta! That is embarrassing, is that REALLY your middle name? I don’t think I believe that”. To which Larry replied, “Well, then, I bet you five dollars that it is Terraponesta”. “You want me to bet five dollars that you don’t know your own middle name, to bet that it ISN’T Terraponesta? You must be crazy!”. “Well, you said you didn’t believe it – put your money where your mouth is!”. “No, man, I’m not gonna waste five dollars!”. At this point, Bernie’s brother Allen, who was 26, an ex-Marine, and worldly wise, said, “BET HIM, BERNIE! BET HIM! If you don’t, you’ll come back to the room to find that I’ve locked you out”. Finally, Bernie said despondently, “OK, I bet five dollars that your middle name isn’t Terraponesta”. At which point, Larry handed him a five-dollar bill!

For the record, Larry’s middle name was “Thomas.”

Here are some memories of Larry by Deepak Kapur:

I first met Wos at CADE at NYU in New York city in 1982 where I was essentially ignored. My second encounter was in 1987/88, when Hantao Zhang and I got invited to Argonne National Lab by Overbeek and Lusk, after they heard that we had been able to prove many ring identities fast using associative-commutative (AC) Knuth–Bendix completion procedure implemented in our Rewrite Rule Laboratory (RRL). I recall being taken to a Thai restaurant for lunch by Overbeek, one of Wos’ favorite places in the neighborhood where Wos lived in Chicago. Larry did not remember meeting me in NYC, but soon started mimicking my Indian accent despite Overbeek politely suggesting that he should not do so given that he had just met me. Wos asked me to explain what we had done in RRL and immediately challenged me to prove the Robbins algebra problem, which was open at that time. Wos was skeptical about using Knuth–Bendix over paramodulation and demodulation, but immediately grasped the power of the use of AC unification. This visit led McCune to integrate AC unification in his theorem prover EQP, leading to settling the long standing conjecture that Robbins’ algebras are indeed Boolean.

Wos was the key note speaker of CADE-11 in Saratoga Springs in 1992, where he also got the first Herbrand award. When he arrived at the Rensselaer/Albany train station, he asked me to take his wife and him to the front of the train to feel its engine as he was very fond of hearing sounds generated by rail engines. My family invited him

to our home for dinner where he met my then 4-year old daughter. She was amazed to learn that a blind person could be such an accomplished scientist.

In 1993 *Wos* started contemplating to step down from the editor-in-chief of *JAR*. He called one night and told me he had a proposition that I would not and could not refuse: “How would you like to be the next editor-in-chief of *JAR*?”. He graciously agreed to mentor me, which meant calls late nights and over the weekends. Whenever he called, he never bothered to find out whether I was busy or doing anything else, and instead just dived into whatever issue he wanted to talk about. When I decided to become the chair of the CS department at the University of New Mexico in late 1998, *Wos* told me that my career as an automated reasoning researcher was over. However, he still insisted on me continuing being the editor-in-chief of *JAR*. Our interactions became less frequent even though I would occasionally get late night and weekend calls, and the conversation would start with whether I was still interested in theorem proving and whether I was doing research. Sadly, I did not get a chance to visit Argonne for over the last 20 years, consequently missed my interaction with *Wos*.

Larry was a faithful and generous person; he helped various people that he encountered in the course of life, for example various unsighted persons whom he advised or helped financially. If you were Larry’s friend, you could count on him absolutely; he would be there. Larry had strong opinions about people; some of them earned his respect, and others his contempt. Of course, those with whom he worked were in the first group. All the people that I know who worked with him agree that he made them feel good about working with him. As Jim Boyle put it, he “had a talent for identifying the individual strengths of the people with whom he worked. He said it was his way of getting what he needed from each of us, but it made us feel enabled and competent.” He was quick to praise and very slow to criticize. We will always remember what it was like in the days when we could pick up the phone and hear Larry begin to speak. Let this paper record that we return the respect that Larry paid to us all.

5 Conclusion

by Geoff Sutcliffe

Larry *Wos*’ interests in automated reasoning were broad and influential: Larry valued and developed underlying theory, but he did not stop there. Larry pushed the theory into effective system implementations, but he did not stop there. Larry used the implementations in a range of applications, but he did not stop there. Larry inspired many people to join his efforts, through publications and conversations, thus contributing to the development of the thriving automated reasoning community that we have today.

This paper has surveyed the highlights of Larry *Wos*’ deep contributions to the discipline of automated reasoning:

- the how - his inference rules;
- the what - his applications;
- and the style - his persona.

Each section has been appropriately and adequately laudatory, so that more is unnecessary. Let it simply be said, he touched the heart of automated reasoning. Larry *Wos* died on August 20, 2020, in Chicago, USA.



Acknowledgements This work was done in part while Maria Paola Bonacina was participating in a program at the Simons Institute for the Theory of Computing. Thanks go to Christoph Weidenbach for inviting all living CADE Presidents to sign an obituary for Larry Wos that was written by Maria Paola Bonacina, and appeared in AAR Newsletter No. 123 of September 2020.³ The introduction of this paper is a revised version of that obituary.

References

1. Aigner, M., Ziegler, G.: *Proofs from The Book*, 6th edn. Springer, Berlin (2018)
2. Armando, A., Bonacina, M., Ranise, S., Schulz, S.: New results on rewrite-based satisfiability procedures. *ACM Trans. Comput. Log.* **10**(14), 1–15 (2009)
3. Bachmair, L.: *Canonical Equational Proofs*. Birkhäuser (1991)
4. Bachmair, L., Dershowitz, N.: Equational inference, canonical proofs, and proof orderings. *J. Assoc. Comput. Mach.* **41**(2), 236–276 (1994)
5. Bachmair, L., Dershowitz, N., Plaisted, D.: Completion Without Failure. In: Ait-Kaci, H., Nivat, M. (eds.) *Resolution of Equations in Algebraic Structures*, pp. 1–30. Academic Press (1989)
6. Bachmair, L., Ganzinger, H.: Rewrite-based equational theorem proving with selection and simplification. *J. Log. Comput.* **4**(3), 217–247 (1994)
7. Bachmair, L., Ganzinger, H.: Equational reasoning in saturation-based theorem proving. In: Bibel, W., Schmitt, P. (eds.) *Automated Deduction, A Basis for Applications I: Foundations—Calculi and Methods*, no. 10 in *Applied Logic Series*, pp. 352–397. Kluwer Academic Publishers (1998)
8. Beeson, M., Veroff, R., Wos, L.: Double-negation elimination in some propositional logics. *Stud. Logica.* **80**(2–3), 195–234 (2005)
9. Beeson, M., Wos, L.: OTTER proofs in Tarskian geometry. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) *Proceedings of the 7th International Joint Conference on Automated Reasoning*, no. 8562 in *Lecture Notes in Computer Science*, pp. 495–510 (2014)
10. Beeson, M., Wos, L.: Finding proofs in Tarskian geometry. *J. Autom. Reason.* **58**(1), 181–207 (2017)
11. Bentkamp, A., Blanchette, J., Cruanes, S., Waldmann, U.: Superposition for Lambda-free higher-order logic. In: Galniche, D., Schulz, S., Sebastiani, R. (eds.) *Proceedings of the 9th International Joint Conference on Automated Reasoning*, no. 10900 in *Lecture Notes in Computer Science*, pp. 2–46 (2018)
12. Bentkamp, A., Blanchette, J., Tourret, S., Vukmirović, P., Waldmann, U.: Superposition with Lambdas. In: Fontaine, P. (ed.) *Proceedings of the 27th International Conference on Automated Deduction*, no. 11716 in *Lecture Notes in Computer Science*, pp. 55–73. Springer (2019)
13. Bhayat, A., Reger, G.: Set of Support for Higher-order Reasoning. In: Konev, B., Rümmer, P., Urban, J. (eds.) *Proceedings of the 6th Workshop on Practical Aspects of Automated Reasoning*, no. 2162 in *CEUR Workshop Proceedings*, pp. 2–16 (2018)

³ <http://aarinc.org/Newsletters/132-2020-09.html>.

14. Bhayat, A., Reger, G.: Restricted Combinatory Unification. In: Fontaine, P. (ed.) Proceedings of the 27th International Conference on Automated Deduction, no. 11716 in Lecture Notes in Computer Science, pp. 74–93. Springer-Verlag (2019)
15. Bhayat, A., Reger, G.: A combinator-based superposition calculus for higher-order logic. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Proceedings of the 10th International Joint Conference on Automated Reasoning, no. 12166 in Lecture Notes in Artificial Intelligence, pp. 278–296 (2020)
16. Bibel, W.: Early history and perspectives of automated deduction. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) Proceedings of the 30th German Annual Conference on Artificial Intelligence, no. 4667 in Lecture Notes in Computer Science, pp. 2–18. Springer (2007)
17. Bonacina, M.P.: On the reconstruction of proofs in distributed theorem proving: a modified clause-diffusion method. *J. Symb. Comput.* **21**(4–6), 507–522 (1996)
18. Bonacina, M.P.: A taxonomy of theorem-proving strategies. In: Wooldridge, M., Veloso, M. (eds.) *Artificial Intelligence Today*, no. 1600 in Lecture Notes in Artificial Intelligence, pp. 43–84. Springer-Verlag (1999)
19. Bonacina, M.P.: Parallel theorem proving. In: Hamadi, Y., Sais, L. (eds.) *Handbook of Parallel Constraint Reasoning*, pp. 179–235. Springer (2018)
20. Bonacina, M.P., Furbach, U., Sofronie-Stokkermans, V.: On first-order model-based reasoning. In: Martí-Oliet, N., Olveczky, P., Talcott, C. (eds.) *Logic, Rewriting, and Concurrency: Essays Dedicated to José Meseguer*, no. 9200 in Lecture Notes in Computer Science, pp. 181–204. Springer-Verlag (2015)
21. Bonacina, M.P., Hsiang, J.: The clause-diffusion methodology for distributed deduction. *Fund. Inform.* **24**(1–2), 177–207 (1995)
22. Bonacina, M.P., Hsiang, J.: Towards a foundation of completion procedures as semidecision procedure. *Theoret. Comput. Sci.* **146**, 199–242 (1995)
23. Bonacina, M.P., Lynch, C., de Moura, L.: On deciding satisfiability by theorem proving with speculative inferences. *J. Autom. Reason.* **47**(2), 161–189 (2011)
24. Bonacina, M.P., Plaisted, D.: Semantically-guided goal-sensitive reasoning: model representation. *J. Autom. Reason.* **56**(2), 113–141 (2016)
25. Bonacina, M.P., Plaisted, D.: Semantically-guided goal-sensitive reasoning: inference system and completeness. *J. Autom. Reason.* **59**(2), 165–218 (2017)
26. Brand, D.: Proving theorems with the modification method. *SIAM J. Comput.* **4**(4), 412–430 (1975)
27. Cruanes, S.: Extending Superposition with Integer Arithmetic, Structural Induction, and Beyond. Ph.D. thesis, Ecole Polytechnique, Paris, France (2015)
28. Davis, M., Putnam, H.: A computing procedure for quantification theory. *J. ACM* **7**(3), 201–215 (1960)
29. Duarte, A., Korovin, K.: Implementing superposition in iProver. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Proceedings of the 10th International Joint Conference on Automated Reasoning, no. 12167 in Lecture Notes in Artificial Intelligence, pp. 388–397 (2020)
30. Ernst, Z., Fitelson, B., Harris, K., Wos, L.: Shortest axiomatizations of implicational S_4 and S_5 . *Notre Dame J. Formal Log.* **43**(3), 169–179 (2002)
31. Fietzke, A., Weidenbach, C.: Superposition as a decision procedure for timed automata. *J. Math. Comput. Sci.* **6**(4), 409–425 (2012)
32. Fitelson, B., Wos, L.: Axiomatic proofs through automated reasoning. *Bull. Sect. Log.* **29**(3), 125–136 (2000)
33. Ganzinger, H., de Nivelle, H.: A Superposition decision procedure for the guarded fragment with equality. In: Proceedings of the 14th Symposium on Logic in Computer Science, pp. 1043–6871. IEEE Computer Society (1999)
34. Gleiss, B., Suda, M.: Layered clause selection for theory reasoning. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Proceedings of the 10th International Joint Conference on Automated Reasoning, no. 12166 in Lecture Notes in Computer Science, pp. 402–409 (2020)
35. Hillenbrand, T.: Citius altius fortius: lessons learned from the theorem prover Waldmeister. In: Dahn, I., Vigneron, L. (eds.) Proceedings of the 4th International Workshop on First-Order Theorem Proving, no. 86.1 in Electronic Notes in Theoretical Computer Science, pp. 1–13 (2003)
36. Hsiang, J., Rusinowitch, M.: A new method for establishing refutational completeness in theorem proving. In: Siekmann, J.H. (ed.) Proceedings of the 8th International Conference on Automated Deduction, no. 230 in Lecture Notes in Computer Science, pp. 141–152. Springer-Verlag (1986)
37. Hsiang, J., Rusinowitch, M.: On word problems in equational theories. In: Ottman, T. (ed.) Proceedings of the 14th International Colloquium on Automata, Languages, and Programming, no. 267 in Lecture Notes in Computer Science, pp. 54–71. Springer-Verlag (1987)
38. Hsiang, J., Rusinowitch, M.: Proving refutational completeness of theorem proving strategies: the transfinite semantic tree method. *J. Assoc. Comput. Machinery.* **38**(3), 559–587 (1991)

39. Huet, G.: A complete proof of correctness of the Knuth–Bendix completion algorithm. *J. Comput. Syst. Sci.* **23**(1), 11–21 (1981)
40. Kinyon, M.: Proof simplification and automated theorem proving. *Philos. Trans. R Soc. A* **377**(2140) (2019)
41. Knuth, D., Bendix, P.: Simple word problems in universal algebras. In: John, L. (ed.) *Computational Problems in Abstract Algebras*, pp. 263–297. Pergamon Press, Oxford (1970)
42. Kovacs, L., Voronkov, A.: First-order theorem proving and vampire. In: Sharygina, N., Veith, H. (eds.) *Proceedings of the 25th International Conference on Computer Aided Verification*, no. 8044 in *Lecture Notes in Artificial Intelligence*, pp. 1–35. Springer (2013)
43. Lusk, E.: Controlling redundancy in large search spaces: argonne-style theorem proving through the years. In: Voronkov, A. (ed.) *Proceedings of the 3rd International Conference on Logic Programming and Automated Reasoning*, no. 624 in *Lecture Notes in Artificial Intelligence*, pp. 96–106. Springer (1992)
44. Lusk, E., McCune, W., Overbeek, R.: ITP at Argonne national laboratory. In: Siekmann, J.H. (ed.) *Proceedings of the 8th International Conference on Automated Deduction*, no. 230 in *Lecture Notes in Computer Science*, pp. 697–698. Springer (1986)
45. McCharen, J., Overbeek, R., Wos, L.: Problems and experiments for and with automated theorem-proving programs. *IEEE Trans. Comput.* **C-25**(8), 773–782 (1976)
46. McCune, W.: Prover9. <http://www.cs.unm.edu/~mccune/prover9/>
47. McCune, W.: 33 Basic test problems: a practical evaluation of some paramodulation strategies. In: Veroff, R. (ed.) *Automated Reasoning and its Applications: Essays in Honor of Larry Wos*, pp. 71–114. MIT Press (1997)
48. McCune, W.: Solution of the Robbins problem. *J. Autom. Reason.* **19**(3), 263–276 (1997)
49. McCune, W.: Otter 3.3 Reference Manual. Tech. Rep. ANL/MS-C-25(8), Argonne National Laboratory, Argonne, USA (2003)
50. McCune, W., Veroff, R., Fitelson, B., Harris, K., Feist, A., Wos, L.: Short single axioms for Boolean algebra. *J. Autom. Reason.* **29**(1), 1–16 (2002)
51. McCune, W., Wos, L.: A case study in automated theorem proving: finding sages in combinatory logic. *J. Autom. Reason.* **3**(1), 91–107 (1987)
52. McCune, W., Wos, L.: Otter: the CADE-13 competition incarnations. *J. Autom. Reason.* **18**(2), 211–220 (1997)
53. Nieuwenhuis, R., Rubio, A.: Paramodulation-based theorem proving. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, pp. 371–443. Elsevier Science (2001)
54. Overbeek, R.: An implementation of hyper-resolution. *Comput. Math. Appl.* **1**, 201–214 (1975)
55. Parrello, B., Kabat, W., Wos, L.: Job-shop scheduling using automated reasoning: a case study of the car-sequencing problem. *J. Autom. Reason.* **2**(1), 1–42 (1986)
56. Peterson, G.: A technique for establishing completeness results in theorem proving with equality. *SIAM Journal of Computation* **12**(1), 82–100 (1983)
57. Plaisted, D.: Equational reasoning and term rewriting systems. In: Gabbay, D., Hogger, C., Robinson, J. (eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 1, pp. 273–364. Oxford University Press (1993)
58. Plaisted, D., Zhu, Y.: *The Efficiency of Theorem Proving Strategies*. Friedr. Vieweg & Sohn (1997)
59. Regehr, G., Suda, M., Voronkov, A.: Playing with AVATAR. In: Felty, A., Middeldorp, A. (eds.) *Proceedings of the 25th International Conference on Automated Deduction*, no. 9195 in *Lecture Notes in Computer Science*, pp. 399–415. Springer-Verlag (2015)
60. Robinson, G., Wos, L.: Paramodulation and theorem proving in first-order theories with equality. *Mach. Intell.* **4**, 135–150 (1969)
61. Robinson, J.: A machine-oriented logic based on the resolution principle. *J. ACM* **12**(1), 23–41 (1965)
62. Robinson, J.: Automatic deduction with hyper-resolution. *Int. J. Comput. Math.* **1**, 227–234 (1965)
63. Rusinowitch, M.: *Démonstration Automatique par des Techniques de Réécriture*. Ph.D. thesis, Université de Nancy I, Nancy, France (1987). Published in the series *Collection Science Informatique*, InterEdition, Paris, France (1989)
64. Rusinowitch, M.: Theorem-proving with resolution and superposition. *J. Symb. Comput.* **11**(1–2), 21–50 (1991)
65. Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) *Proceedings of the 27th International Conference on Automated Deduction*, no. 11716 in *Lecture Notes in Computer Science*, pp. 495–507. Springer (2019)
66. Schulz, S., Möhrmann, M.: Performance of clause selection heuristics for saturation-based theorem proving. In: Olivetti, N., Tiwari, A. (eds.) *Proceedings of the 8th International Joint Conference on Automated Reasoning*, no. 9706 in *Lecture Notes in Artificial Intelligence*, pp. 330–345 (2016)

67. Slagle, J.: Automatic theorem proving with renamable and semantic resolution. *J. ACM* **14**(4), 687–697 (1967)
68. Smith, B.: Reference manual for the environmental theorem prover: an incarnation of AURA. Tech. Rep. ANL/88-2, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, USA (1988)
69. Smith, B., Wos, L.: An unnatural attack on the structure problem for the free Jordan ring on 3 letters: an application of quad arithmetic. In: Beck, R., Kolman, B. (eds.) *Computers in nonassociative rings and algebras. Special Session of the 82nd Annual Meeting of the American Mathematical Society*, pp. 45–138. Academic Press (1977)
70. Smullyan, R.: *To Mock a Mocking Bird and Other Logic Puzzles*. Knopf (1978)
71. Stickel, M.: PTPP and linked inference. In: R. Boyer (ed.) *Automated Reasoning: Essays in Honor of Woody Bledsoe, Automated Reasoning Series*, pp. 283–296. Kluwer Academic Publishers (1991)
72. Thiele, R.: Hilbert's twenty-fourth problem. *Amer. Math. Monthly* **110**(1), 1–24 (2003)
73. Thiele, R., Wos, L.: Hilbert's Twenty-fourth problem. *J. Autom. Reason.* **29**(1), 67–89 (2002)
74. Venker, G.: Beweisalgorithmen für die Prädikatenlogik. *Computing* **2**(3), 263–283 (1967)
75. Veroff, R.: *Automated Reasoning and its Applications: Essays in Honor of Larry Wos*. MIT Press (1997)
76. Veroff, R.: A Wos Challenge Met. *J. Autom. Reason.* (2022). <https://doi.org/10.1007/s10817-021-09614-y>
77. Vukmirović, P., Blanchette, J., Cruanes, S., Schulz, S.: Extending a brainiac prover to lambda-free higher-order logic. In: T. Vojnar, L. Zhang (eds.) *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, no. 11427 in *Lecture Notes in Computer Science*, pp. 192–210. Springer (2019)
78. Vukmirović, P., Nummelin, V.: Boolean reasoning in a higher-order superposition prover. In: Fontaine, P., Rümmer, P., Tourret, S. (eds.) *Proceedings of the 7th Workshop on Practical Aspects of Automated Reasoning*, no. 2752 in *CEUR Workshop Proceedings*, pp. 148–166 (2020)
79. Weidenbach, C., Dimova, D., Fietzke, A., Kumar, R., Suda, M., Wischniewski, P.: SPASS Version 3.5. In: Schmidt, R. (ed.) *Proceedings of the 22nd International Conference on Automated Deduction*, no. 5663 in *Lecture Notes in Artificial Intelligence*, pp. 140–145. Springer (2009)
80. Winker, S., Wos, L.: Automated generation of models and counterexamples and its application to open questions in ternary Boolean algebra. In: *Proceedings of the 8th International Symposium on Multiple-Valued Logic*, pp. 251–256. IEEE Press (1978)
81. Winker, S., Wos, L., Lusk, E.: Semigroups, antiautomorphisms, and involutions: a computer solution to an open problem. *Math. Comput.* **37**(156), 533–545 (1981)
82. Wos, L.: *Automated Reasoning—33 Basic Research Problems*. Prentice-Hall (1988)
83. Wos, L.: The impossibility of the automation of logical reasoning. In: Kapur, D. (ed.) *Proceedings of the 11th International Conference on Automated Deduction*, no. 607 in *Lecture Notes in Artificial Intelligence*, pp. 1–3. Springer-Verlag (1992)
84. Wos, L.: The Kernel strategy and its use for the study of combinatory logic. *J. Autom. Reason.* **10**(3), 287–344 (1993)
85. Wos, L.: Searching for circles of pure proofs. *J. Autom. Reason.* **15**(3), 279–315 (1995)
86. Wos, L.: OTTER and the Moufang identity problem. *J. Autom. Reason.* **17**(2), 215–257 (1996)
87. Wos, L.: *The Automation of Reasoning: An Experimenter's Notebook with OTTER Tutorial*. Academic Press (1996)
88. Wos, L.: Conquering the Meredith single Axiom. *J. Autom. Reason.* **27**(2), 175–199 (2001)
89. Wos, L., Carson, D., Robinson, G.: The unit preference strategy in theorem proving. In: *Proceedings of the AFIPS 1964 Fall Joint Computer Conference*, pp. 615–621. Spartan Books (1964)
90. Wos, L., Fitelson, B.: Automating the search for answers to open questions. In: Aagaard, M., Harrison, J. (eds.) *Proceedings of the 13th International Conference on Theorem Proving in Higher Order Logics*, no. 1869 in *Lecture Notes in Computer Science*, pp. 519–525 (2000)
91. Wos, L., Overbeek, R., Lusk, E.: Subsumption, a sometimes undervalued procedure. In: Lassez, J.L., Plotkin, G. (eds.) *Computational Logic—Essays in Honor of Alan Robinson*, pp. 3–40. MIT Press (1991)
92. Wos, L., Pereira, F., Hong, R., Boyer, R., Moore, J., Bledsoe, W., Henschen, L., Buchanan, B., Wrightson, G., Green, C.: An overview of automated reasoning and related fields. *J. Autom. Reason.* **1**(1), 5–48 (1985)
93. Wos, L., Pieper, G.: *A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning*. World Scientific (1999)
94. Wos, L., Pieper, G.: *Automated Reasoning and the Discovery of Missing and Elegant Proofs*. Rinton Press (2003)
95. Wos, L., Robinson, G., Carson, D.: Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM* **12**(4), 536–541 (1965)

96. Wos, L., Robinson, G., Carson, D., Shalla, L.: The concept of demodulation in theorem proving. *J. ACM* **14**(4), 698–709 (1967)
97. Wos, L., Thiele, R.: Hilbert's new problem. *Bull. Sect. Log.* **30**(3), 165–175 (2001)
98. Wos, L., Ulrich, D., Fitelson, B.: Vanquishing the XCB question: the methodological discovery of the last shortest single Axiom for the equivalential calculus. *J. Autom. Reason.* **29**(2), 107–124 (2002)
99. Wos, L., Ulrich, D., Fitelson, B.: XCB, the last of the shortest single axioms for the classical equivalential calculus. *Bull. Sect. Log.* **32**(3), 131–136 (2003)
100. Wos, L., Winker, S., Veroff, R., Smith, B., Henschen, L.: Questions concerning possible shortest single axioms in equivalential calculus: an application of automated theorem proving to infinite domains. *Notre Dame J. Formnal Log.* **24**(2), 205–223 (1983)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.