CHAPTER 1

The Early History of Automated Deduction

DEDICATED TO THE MEMORY OF HAO WANG

Martin Davis

SECOND READERS: Peter Andrews, Wolfgang Bibel, Alan Robinson, and Jörg Siekmann.

Contents

HANDBOOK OF AUTOMATED REASONING
Edited by Alan Robinson and Andrei Voronkov

With the ready availability of serious computer power, deductive reasoning, especially as embodied in mathematics, presented an ideal target for those interested in experimenting with computer programs that purported to implement the "higher" human faculties. This was because mathematical reasoning combines objectivity with creativity in a way difficult to find in other domains. For this endeavor, two paths presented themselves. One was to try to understand what people do when they create proofs and to write programs emulating that process. The other was to make use of the systematic work of the logicians in reducing logical reasoning to standard canonical forms on which algorithms could be based. Each path confronted daunting obstacles. The difficulty with the first approach was that available information about how creative mathematicians go about their business was and remains vague and anecdotal. On the other hand, the well-known unsolvability results of Church and Turing showed that the kind of algorithm on which a programmer might want to base a theorem-proving program simply did not exist. Moreover, it was all too obvious that an attempt to generate a proof of something non-trivial by beginning with the axioms of some logical system and systematically applying the rules of inference in all possible directions was sure to lead to a gigantic combinatorial explosion.

Each of these approaches has led to important and interesting work. Unfortunately, for many years the proponents of the two approaches saw themselves as opponents and engaged in polemics in which they largely spoke past each other. One problem was that whereas they appeared to be working on the same problems, they tended to differ not only in their approaches, but also in their fundamental goals. Those whose method was the emulation of the human mathematician tended to see their research as part of an effort to help understand human thought. Those who proposed to use the methods of mathematical logic tended to see the goal as the development of useful systems of automated deduction. Ultimately, the most successful developments incorporated insights deriving from both approaches.

For a brief account of the history of the developments in logic that provided the background for research in this field see [Davis 1983*c*]. An interesting account of the two approaches and their mutual interaction can be found in [MacKenzie 1995]. The volume [Siekmann and Wrightson 1983] is a useful anthology of the principal articles on automated deduction to appear in the years through 1966.

1. Presburger's Procedure

In 1929, M. Presburger had shown that the first-order theory of addition in the arithmetic of integers is decidable, that is he had provided an algorithm which would be able to determine for a given sentence of that language, whether or not it is true. In 1954, Martin Davis programmed this algorithm for the vacuum tube computer at the Institute for Advanced Study in Princeton. As was stated by Davis [1983*c*]

> Since it is now known that Presburger's procedure has worse than exponential complexity, it is not surprising that this program did not perform very well. Its

great triumph was to prove that the sum of two even numbers is even.

2. Newell, Shaw & Simon, and H. Gelernter

The propositional calculus is the most elementary part of mathematical logic, dealing as it does with the connectives $\neg \wedge \vee \supset$. Its treatment constitutes Section A of Part I (37 pages) of Whitehead and Russell's *Principia Mathematica*, their monumental three volume effort purporting to demonstrate that all of mathematics can be viewed as a part of logic. Their treatment proceeds from five particular formulas, that may be called *axioms* or *primitive propositions* to which are applied the explicitly stated rule of *modus ponens* or *detachment*[1] and implicit rules permitting substitutions for propositional variables and replacing defined symbols using their definitions. Newell, Shaw and Simon set themselves the problem of producing a computer program that emulates the process by which a person might seek proofs in the propositional calculus of *Principia.* Although the formalism is simple enough so that such a program would be feasible, the process requires enough ingenuity that the problem was hardly trivial.

In Newell, Shaw and Simon's [1957] report on experiments with their "Logic Theory Machine," (developed around the same time as Davis's Presburger program) the authors are very explicit about their goals:

Our explorations ... represent a step in a program of research ... aimed at developing a theory ... and applying [it] to such fields as computer programming and human learning and problem-solving

Although it would be difficult to claim that this work has helped very much with such an ambitious agenda, it did provide a paradigm employed by many theorem-provers developed later, and this was surely its lasting influence. Among the techniques made explicit were forward and backward chaining, the generation of useful subproblems, and seeking substitutions that produce desired matches.

The authors emphasize that their program is "heuristic" rather than "algorithmic," and this purported distinction has given rise to much dissension and confusion. In this context, "heuristic" seems to mean little more than the lack of a guarantee that the process will always work (given sufficient space and time). The algorithm they contrast with their own procedure is the "British Museum algorithm" by which all possible proofs are generated until one leading to the desired result is reached. Indeed, the authors seem to have been unaware that Post's proof of the completeness of the Principia propositional calculus using truth tables had, in effect supplied a simple algorithm by means of which a demonstration by truth tables could be converted into a proof in *Principia* [Post 1921].

Wang and Gao [1987] presented a Gentzen-style proof system for the propositional calculus designed for efficiency. Unlike the program of Newell et al, Wang's system is complete: for any input, processing eventually halts, yielding either a

[1]Actually Whitehead and Russell's tendency to confuse object and meta-language led them to state this confusingly as "Anything implied by a true proposition is true." But this lapse is not important for the present discussion.

proof or a disproof. The simple examples that are explicitly listed in *Principia*, including those that stumped the Logic Theory Machine, were easily disposed of. Although Wang seems not to have quite understood that producing an efficient generator of proofs in the propositional calculus was not what the Logic Theory Machine designers were after, they did leave themselves open to Wang's criticism by giving the impression that the absurd British Museum algorithm was the only possible "non-heuristic" proof-generating system for the propositional calculus.

Like the propositional calculus, the elementary geometry of the plane can be specified by a formal system for which an algorithmic decision procedure is available. This is seen by introducing a coordinate system and relying on the reduction of geometry to algebra and Tarski's decision procedure for the algebra of the real numbers. However, unlike the case of truth table methods for the propositional calculus, this method is utterly unfeasible. Although theoretical confirmation of this did not come until much later, it was already apparent from Davis's experience with the much simpler Presburger procedure. Herbert Gelernter's Geometry Machine is very much in the spirit of Newell at al. A clue to Gelernter's orientation is provided by his statement [Gelernter 1959]:

> ... geometry provides illustrative material in treatises and experiments in human problem-solving. It was felt that we could exchange valuable insights with behavioral scientists ...

Technically, in addition to the repertoire of The Logic Machine (backward chaining, the use of subproblems), the geometry machine introduced two interesting innovations: the systematic use of symmetries to abbreviate proofs and the use of a coordinate system to simulate the carefully drawn diagram a student of geometry might employ. This last was used to tip off the prover to the fact that certain pairs of line segments and of angles "appeared" to be equal to one another, and thereby to guide the search for a proof.

3. First-Order Logic

Unlike the cases of propositional logic and elementary geometry, there is no general decision procedure for first-order logic. On the other hand, given appropriate axioms as premises, all mathematical reasoning can be expressed in first- order logic, and that is why so much attention has been paid to proof procedures for this domain. Investigations by Skolem and Herbrand in the 1920s and early 1930s provided the basic tools needed for theorem-proving programs for first-order logic [Davis 1983*c*].

In 1957 a five week Summer Institute for Symbolic Logic held at Cornell University was attended by almost every logician working in the United States. Many of the more theoretically inclined researchers from the nearby IBM facilities were also present; FORTRAN, a brand-new innovation in programming practice was unveiled. After discussions with Gelernter, the logician Abraham Robinson was led to give a short talk [Robinson 1957] in which he pointed to Skolem functions and "Herbrand's theorem" as useful tools for general purpose theorem-provers. He also made the provocative remark that the auxilliary points, lines, or circles "constructed" as part of the solution to a geometry problem can be thought of as being elements of what is now called the Herbrand universe for the problem.

The first theorem-provers for first-order logic to be implemented based on Herbrand's theorem employed a completely unguided search of the Herbrand universe. Instead of using Skolem functions to deal with instantiations, variables were replaced by parameters; so the program had to provide a capability for keeping track of dependencies among these parameters. Tests for truth-functional satisfiability used either simple truth table calculations or expansion into disjunctive normal form. Not surprisingly, these programs were capable of proving only the simplest theorems. Among the first of these programs, that by Gilmore [1960] served as a particularly useful stimulus for further investigations.

In his later commentary, Prawitz [1983*a*] explained that the development of new proof procedures and completeness proofs for first- order logic together with the availability of computational resources tempted him to become involved in implementing such a procedure. He adopted a modified form of the method of semantic tableaux, and formulated his own high level algorithmic language in which the procedure could be written. The detailed implementation was accomplished by Prawitz, Prawitz and Voghera [1960]. Despite being based on an up-to-date underlying logical system, this program suffered from the same limitations as Gilmore's.

Martin Davis and Hilary Putnam noted that Gilmore's program failed on some rather simple examples because of its reliance on expansions into disjunctive normal form for satisfiability testing. This led them to the optimistic (and in retrospect rather naive) conclusion that the lack of effective methods for testing large formulas of the propositional calculus for satisfiability was the main obstacle to be surmounted. Although their interest in algorithms for what came to be known as the *satisfiability problem* was only because they wanted to use such methods as part of a proof procedure for first-order logic, they secured support from the National Security Agency, to spend the summer of 1958 working on this problem. In their unpublished report to the NSA [Davis and Putnam 1958], they emphasized the use of conjunctive normal form for satisfiability testing. The specific reduction methods whose use together have been linked to the names Davis-Putnam are all present in this report. These are:

1. The *one literal rule* also known as the *unit rule.*
2. The *affirmative-negative rule* also known as the *pure literal rule.*
3. The *rule for eliminating atomic formulas*
4. The *splitting rule,* called in the report, the *rule of case analysis*

The Davis-Putnam paper usually cited [Davis and Putnam 1960] was written a year later. The proposed procedure would accept as input a formula that had been preprocessed by first using Skolem functions to eliminate existential quantifiers and then expanding the formula into conjunctive normal form. Many theorem-provers (including some that have been very successful) have used this approach. Satisfiability testing was to be carried out using rules 1,2,3 above, and it was noted that an example that stumped Gilmore's program could easily be done by hand computation. When George Logemann and Donald Loveland attempted to implement the program they found that the *rule for eliminating atomic formulas* (later called *ground resolution*) which replaced a formula

$$(p \vee A) \wedge (\neg p \vee B) \wedge C$$

by

$$(A \vee B) \wedge C$$

used too much RAM. So it was proposed to use instead the *splitting rule* which generates the pair of formulas

$$A \wedge C \qquad B \wedge C$$

The idea was that a stack for formulas to be tested could be kept in external storage (in fact a tape drive) so that formulas in RAM never became too large.[2] Although testing for satisfiability was performed very efficiently, it soon became clear that no very interesting results could be obtained without first devising a method for avoiding the generation of spurious elements of the Herbrand universe [Davis, Logemann and Loveland 1962].

During the same years, Hao Wang was attempting to apply some of the more sophisticated work that had been done in proof theory and on solvable cases of Hilbert's Entscheidungsproblem to automatic deduction programs. He announced a computer program that proved all of the theorems (about 400) of Whitehead and Russell's *Principia Mathematica* of first-order logic with equality [Wang and Zhi 1998, Wang and Zhi 1998]. However, this apparently momentous achievement in automating deduction was (as Wang himself pointed out) possible only because all of these theorems can be brought into prenex form with the simple prefix $\forall \ldots \forall \exists \ldots \exists$. Wang concluded that:

> The most interesting lesson from these results is perhaps that even in a fairly rich domain, the theorems actually proved are mostly ones which call on a very small portion of the available resources of the domain. ([Wang 1963*c*] p. 32)

Prawitz's [1960] influential paper taught the growing automated deduction community that unnecessary terms in the Herbrand expansion could be avoided by using algorithms that did not generate elements of the Herbrand universe until needed. Most later progress was based on this key insight. Prawitz's procedure worked by obtaining expansions into disjunctive normal form before replacing variables by

[2]Unfortunately both procedures using rules 1,2, and 3 and procedures using rules 1,2, and 4 have been called the "Davis-Putnam procedure" in the literature; the first is generally considered for worst case analysis while it is the second that is ordinarily implemented.

Wolfgang Bibel has kindly pointed out to me that the "rule for eliminating atomic formulas" otherwise known as "ground resolution" was first proposed in A. Blake's dissertation in 1937 and (in its dual form) was also mentioned by W.V. Quine in 1955 under the name "consensus rule". For further information, see [Bibel 1993]. Otherwise, as far as I know, the other rules mentioned occurred for the first time in [Davis and Putnam 1958].

It should also be mentioned that rules 2 and 4 were found independently by Dunham, Fridsal and Sward [1959]. They emphasized that a program based on these rules performs very effectively without using "heuristic" devices.

elements of the Herbrand universe. The algorithm thus generated disjunctive normal forms of increasing length seeking one with the property that some substitution from the Herbrand universe would yield a truth-functionally unsatisfiable formula.[3] Since this condition amounts to each disjunctive clause including a pair of literals of the form $\ell, \neg\ell$, it can be formulated as the need to satisfy a system of equations in the parameters of the expansion.[4]

Prawitz's procedure was a great improvement over what had been done previously because no spurious elements of the Herbrand universe were generated. Unfortunately, the huge expansions into disjunctive normal form that would be generated by all but the simplest problems made it clear that, at least as presented, this was still an unsatisfactory procedure. However, it contained the seminal idea of searching for substitutions that would transform pairs of literals into negations of one another. Moreover if existential quantifiers are eliminated in favor of Skolem functions at the outset, instead of systems of equations, one has the simple problem of *unifying* pairs of terms.

In his survey paper, Davis [1963] proposed

> ...a new kind of procedure which seeks to combine the virtues of the Prawitz procedure and those of the Davis-Putnam procedure.

The idea, also noted by Dunham and North [1962], was that by the "pure literal rule" from the Davis-Putnam procedure (Rule 2, above), substitutions can help to render a conjunctive set of disjunctive clauses unsatisfiable only if they succeed in transforming a literal from one of these clauses into the negation of a literal in another clause. A theorem-proving program based on these ideas was written by D. McIlroy at Bell Laboratories and was improved and corrected by Peter Hinman. The program included an implementation of the ordinary unification algorithm [Chinlund, Davis, Hinman and McIlroy 1964].

Merely the existence of this volume makes it abundantly clear that automated reasoning is a thriving field with a huge literature. The bimonthly publication *The Journal of Automated Reasoning* is devoted entirely to this field. If one event can be pinpointed as marking its emergence as a mature subject, it would be the publication [Robinson 1965*b*] in which J.A. Robinson announced the resolution principle. [Robinson 1965*b*] was Robinson's second paper in the area, and it is helpful in tracing his thought to begin with the first [Robinson 1963]. He began with the basic framework of Davis-Putnam: existential quantifiers eliminated in favor of Skolem functions and conjunctive normal form. He noted Prawitz's technique for avoiding spurious elements of the Herbrand universe and Davis's survey paper. Evidently Davis's sketch of his proposed procedure was insufficiently clear, and Robinson wrote:[5]

---

[3]This account is not quite accurate because in Prawitz's paper matters are expressed in terms of finding a proof rather than a refutation. So what he actually did is precisely the dual of what is stated above.

[4]As pointed out to the author by Gérard Huet, this same use of equations occurs already in Herbrand's [1930, p. 145] thesis.

[5]In the interest of clarity, the reference numbers in this quote were replaced by the numbers in the present bibliography corresponding to the same papers.

> Davis [1963] has therefore proposed a way of exploiting Prawitz' powerful idea while avoiding Prawitz' disasterous use of normal forms–in much the same way that the techniques of Davis and Putnam [1960] avoid the use of normal forms which caused Gilmore's [1960] program to be unable even to solve [an easy problem]. From the few remarks at the end of [Davis 1963] it does not yet seem clear just how Davis will proceed, and one waits with great interest his further researches along these lines.

The rest of the paper has a number of interesting computer proofs generated by using the Davis-Putnam "one literal clause" rule, and, when that fails, requiring the user to pre-specify the elements of the Herbrand universe needed to obtain a proof. Finding these elements was conjectured to be "the really 'creative' part of the art of proof-construction."

Robinson's method of resolution introduced in his highly influential [1965*b*] revolutionized the subject. Robinson found a single rule of inference, easily performable by computer, that was complete for first- order logic. Using resolution required no separate procedure for dealing with propositional calculus. Starting with the usual pre-processed conjunctive set of disjunctive clauses, Robinson's technique was to seek all possible "unifications" that would make it possible to express the set of clauses as

$$(\ell \vee A) \wedge (\neg\ell \vee B) \wedge C$$

where ℓ is a literal that doesn't occur in C. This yields the "resolvent"

$$(A \vee B) \wedge C$$

which after $(A \vee B)$ is "multiplied out" yields a new set of clauses that is unsatisfiable just in case the original set was. This was similar to Davis's proposal [Davis 1963, Chinlund et al. 1964], in seeking unifications that generate complementary literals. It differs not only in not requiring separate truth functional testing, but also in not requiring, as part of the input, specification of the number of instances of each clause to participate in the final proof. [Robinson 1965*b*] is striking for its combinatorial simplicity, as well as for the sheer mathematical elegance of the presentation. Unfortunately, as soon became apparent, the bare resolution method could easily produce many thousands of clauses without reaching a proof. Finding a proof using resolution becomes the problem of providing criteria for the order in which resolutions are to be sought. Early attempts to cut down the search space were Robinson's own elegant hyperresolution [Robinson 1965*a*], and the strategies of unit preference [Wos, Carson and Robinson 1964] and set of support [Wos, Robinson and Carson 1965].

The three decades since the first implementations of resolution have seen an outpouring of research devoted to automated reasoning systems. While some of the most successful are based on resolution, others have proceeded in different directions. For further information, the reader is referred to the other articles in this volume.

Bibliography

BIBEL W. [1993], *Deduction. Automated Logic.*, Academic Press. with the assistance of Steffen Hölldobler.

CHINLUND T. J., DAVIS M., HINMAN P. AND MCILROY M. [1964], Theorem proving by matching. Bell Laboratories.

DAVIS M. [1957], A computer program for presburger's algorithm, *in* 'Summaries of Talks Presented at the Summer Institute for Symbolic Logic', Institute for Defense Analysis. 2nd edition, published in 1960. Reprinted as [Davis 1983*a*].

DAVIS M. [1963], Eliminating the irrelevant from mechanical proofs, *in* 'Proc. Symp. Applied Math.', Vol. XV, pp. 15–30. Reprinted as [Davis 1983*b*].

DAVIS M. [1983*a*], A computer program for Presburger's algorithm, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 41–48. Originally published as [Davis 1957].

DAVIS M. [1983*b*], Eliminating the irrelevant from mechanical proofs, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 315–330. Originally published as [Davis 1963].

DAVIS M. [1983*c*], The prehistory and early history of automated deduction, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 1–28.

DAVIS, M., ED. [1994], *Solvability, Provability, Definability: The Collected Works of Emil L. Post*, Birkhäuser.

DAVIS M., LOGEMANN G. AND LOVELAND D. [1962], 'A machine program for theorem proving', *Communications of the ACM* 5(1962), 394–397. Reprinted as [Davis, Logemann and Loveland 1983].

DAVIS M., LOGEMANN G. AND LOVELAND D. [1983], A machine program for theorem proving, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 267–270. Originally published as [Davis et al. 1962].

DAVIS M. AND PUTNAM H. [1958], Computational methods in the propositional calculus, unpublished report, Rensselaer Polytechnic Institute.

DAVIS M. AND PUTNAM H. [1960], 'A computing procedure for quantification theory', *Journal of the ACM* 7(3), 201–215. Reprinted as [Davis and Putnam 1983].

DAVIS M. AND PUTNAM H. [1983], A computing procedure for quantification theory, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 125–150. Originally published as [Davis and Putnam 1960].

DUNHAM B., FRIDSAL R. AND SWARD G. [1959], A non-heuristic program for proving elementary logical theorems, *in* 'Proc. IFIP Congr.', pp. 282–285.

DUNHAM B. AND NORTH J. [1962], Theorem testing by computer, *in* 'Symp. Math. Theory Machines', Brooklyn Poly. Inst., pp. 172–177. Reprinted as [Dunham and North 1983].

DUNHAM B. AND NORTH J. [1983], Theorem testing by computer, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 271–275. Originally published as [Dunham and North 1962].

GELERNTER H. [1959], Realization of a geometry-theorem proving machine, *in* 'Proc. Intern. Conf. on Inform. Processing', UNESCO House, pp. 273–282. Reprinted as [Gelernter 1983].

GELERNTER H. [1983], Realization of a geometry-theorem proving machine, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 99–122. Originally published as [Gelernter 1959].

GILMORE P. [1960], 'A proof method for quantification theory: its justification and realization', *IBM J. of Research and Development* 4, 28–35. Reprinted as [Gilmore 1983].

GILMORE P. [1983], A proof method for quantification theory: its justification and realization, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 151–158. Originally published as [Gilmore 1960].

HERBRAND J. [1930], Recherches sur la théorie de la démonstration, PhD thesis, University of Paris, Austin. Translated as [Herbrand 1971].

HERBRAND J. [1971], Investigations in proof theory, *in* W. Goldfarb, ed., 'Jacques Herbrand—Logical Writings', Harvard University Press. Originally published as [Herbrand 1930].

MACKENZIE D. [1995], 'The automation of proof: an historical and sociological exploration', *IEEE Annals of the History of Computing* **17**(3), 7–29.
URL: *http://dream.dai.ed.ac.uk/papers/donald/donald.html*

NEWELL A., SHAW J. AND SIMON H. [1957], Empirical explorations with the logic theory machine, *in* 'Proc. West. Joint Comp. Conf.', pp. 218–239. Reprinted as [Newell, Shaw and Simon 1983].

NEWELL A., SHAW J. AND SIMON H. [1983], Empirical explorations with the logic theory machine, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 49–73. Originally appeared as [Newell et al. 1957].

POST E. [1921], 'Introduction to a general theory of elementary propositions', *American Journal of Mathematics* **43**, 163–185. reprinted as [Post 1967] and in [Davis 1994].

POST E. [1967], Introduction to a general theory of elementary propositions, *in* J. van Heijenoort, ed., 'From Frege to Gödel: a Source Book in Mathematical Logic, 1879–1931', Harvard University Press, Cambridge, MA, pp. 264–283.

PRAWITZ D. [1960], 'An improved proof procedure', *Theoria* **26**(2), 102–139. Reprinted as [Prawitz 1983].

PRAWITZ D. [1983*a*], Comments on [Prawitz 1960] and [Prawitz et al. 1960], *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 159–161,200–201.

PRAWITZ D. [1983*b*], An improved proof procedure, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 162–199. Originally published as [Prawitz 1960].

PRAWITZ D., PRAWITZ H. AND VOGHERA N. [1960], 'A mechanical proof procedure and its realization in an electronic computer', *Journal of the ACM* **7**(2), 102–128. Reprinted as [Prawitz, Prawitz and Voghera 1983].

PRAWITZ D., PRAWITZ H. AND VOGHERA N. [1983], A mechanical proof procedure and its realization in an electronic computer, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 202–228. Originally published as [Prawitz et al. 1960].

ROBINSON A. [1957], Proving theorems, as done by man, machine and logician, *in* 'Summaries of Talks Presented at the Summer Institute for Symbolic Logic', Institute for Defense Analysis. 2nd edition, published in 1960. Reprinted as [Robinson 1983*a*].

ROBINSON A. [1983*a*], Proving theorems, as done by man, machine and logician, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 74–76. Originally published as [Robinson 1957].

ROBINSON J. [1963], 'Theorem-proving on the computer', *Journal of the ACM* **10**, 163–174. Reprinted as [Robinson 1983*d*].

ROBINSON J. [1965*a*], 'Automatic deduction with hyper-resolution', *International Journal of Computer Mathematics* **1**, 227–234. Reprinted as [Robinson 1983*b*].

ROBINSON J. [1965*b*], 'A machine-oriented logic based on the resolution principle', *Journal of the ACM* **12**(1), 23–41. Reprinted as [Robinson 1983*c*].

ROBINSON J. [1983*b*], Automatic deduction with hyperresolution, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 416–423. Originally published as [Robinson 1965*a*].

ROBINSON J. [1983*c*], A machine oriented logic based on the resolution principle, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1965, Springer, pp. 397–415. Originally published as [Robinson 1965*b*].

ROBINSON J. [1983*d*], Theorem-proving on the computer, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 372–383. Originally published as [Robinson 1963].

SIEKMANN, J. AND WRIGHTSON, G., EDS [1983], *Automation of Reasoning. Classical Papers on Computational Logic, 1957-1966*, Vol. I,II, Springer Verlag.

WANG H. [1960*a*], 'Proving theorems by pattern recognition I', *Communications of the ACM* **3**(1960), 220–234. Reprinted as [Wang 1983*a*].

WANG H. [1960*b*], 'Towards mechanical mathematics', *IBM J. of Research and Development* **4**, 2–22. Reprinted as [Wang 1983*b*].

WANG H. [1961], 'Proving theorems by pattern recognition II', *Bell Syst. Tech. J.* **40**, 1–41.

WANG H. [1963*a*], Mechanical mathematics and inferential analysis, *in* Braffort and Hirschberg, eds, 'Computer Programming and Formal Systems', North-Holland, pp. 1–20.

WANG H. [1963*b*], Mechanical mathematics and inferential analysis, *in* Braffort and Hirschberg, eds, 'Computer Programming and Formal Systems', North-Holland, pp. 1–20.

WANG H. [1963*c*], The mechanization of mathematical arguments, *in* 'Proc. Symp. in Applied Math.', Vol. XV, pp. 31–40.

WANG H. [1983*a*], Proving theorems by pattern recognition I, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1960, Springer Verlag, pp. 229–243. Originally published as [Wang and Hu 1987].

WANG H. [1983*b*], Towards mechanical mathematics, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 244–264. Originally published as [Wang and Gao 1987].

WOS L., CARSON D. AND ROBINSON G. [1964], The unit preference strategy in theorem proving, *in* 'IFIPS 1964 Fall Joint Comp. Conf.', Vol. 26, pp. 616–621. Reprinted in [Wos, Carson and Robinson 1983].

WOS L., CARSON D. AND ROBINSON G. [1983], The unit preference strategy in theorem proving, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 387–393. Originally appeared as [Wos et al. 1964].

WOS L., ROBINSON G. AND CARSON D. [1965], 'Efficiency and completeness of the set of support strategy in theorem proving', *Journal of the ACM* **12**, 536–541. Reprinted in [Wos, Robinson and Carson 1983].

WOS L., ROBINSON G. AND CARSON D. [1983], Efficiency and completeness of the set of support strategy in theorem proving, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer, pp. 484–489. Originally appeared as [Wos et al. 1965].

Index