6 ALTERNATIVE APPROACHES BESIDES EXPLICIT INDUCTION

6.1 Proof Planning

Suggestions on how to overcome an envisioned dead end in automated theorem proving were summarized in the end of the 1980s under the keyword *proof planning.* Besides its human-science aspects,[173] the main idea[174] of proof planning is to extend a theorem-proving system — on top of the *low-level search space* of the logic calculus of a proof checker — with a *higher-level search space*, which is typically smaller or better organized w.r.t. searching, more abstract, and more human-oriented.

The most extensive and sophisticated subject of proof planning is not especially related to induction, but addresses automated theorem proving in general. We cannot cover it here and have to refer the reader to the article by Alan Bundy and Jörg Siekmann in this volume.

6.2 Rippling

Rippling is a technique for augmenting rewrite rules with information that helps to find a way to rewrite one expression (*goal*) into another (*target*), more precisely to reduce the difference between the goal and the target by rewriting the goal. We had to mention rippling already in § 5.6 several times, but this huge and well-documented area of research cannot be covered here, and we have to refer the reader to the monograph [Bundy *et al.*, 2005].[175] Let us explain here, however, why rippling can be most helpful in the automation of simple inductive proofs.

Roughly speaking, the remarkable success in proving *simple* theorems by induction automatically, can be explained as follows: If we look upon the task of proving a theorem as reducing it to a tautology, then we have more heuristic guidance when we know that we probably have to do it by mathematical induction: Tautologies can have arbitrary subformulas, but the induction hypothesis we are going to apply can restrict the search space tremendously.

In a cartoon of Alan Bundy's, the original theorem is pictured as a zigzagged mountainscape and the reduced theorem after the unfolding of recursive operators according to recursion analysis (goal) is pictured as the reflection of the mountainscape on the surface of a lake with ripples. To apply the induction hypothesis (target), instead of the uninformed search for an arbitrary tautology, we have to *get rid of the ripples* to be able to apply an instance of the theorem as induction hypothesis to the mountainscape mirrored by the calmed surface of the lake.

Until today, rippling was applied to the automation of induction only within the paradigm of explicit induction, whereas it is clearly not limited to this paradigm, and we expect it to be more useful in areas of automated theorem proving with bigger search spaces and, in particular, in *descente infinie.*

[173] Cf. [Bundy, 1989].

[174] Cf. [Bundy, 1988], [Dennis *et al.*, 2005].

6.3 Implicit Induction

The alternative approaches to mechanize mathematical induction *not* subsumed by explicit induction, however, are united under the name "implicit induction".

Triggered by the success of Boyer and Moore [1979], work on these alternative approaches started already in the year 1980 in purely equational theories.[176] A sequence of papers on technical improvements[177] was topped by [Bachmair, 1988], which gave rise to a hope to develop the method into practical usefulness, although it was still restricted to purely equational theories. Inspired by this paper, in the late 1980s and the first half of the 1990s several researchers tried to understand more clearly what implicit induction means from a theoretical point of view and whether it could be useful in practice.[178]

While it is generally accepted that [Bachmair, 1988] is about implicit induction and [Boyer and Moore, 1979] is about explicit induction, there are the following three different viewpoints on what the essential aspect of implicit induction actually is.

Proof by Consistency:[179] Systems for proof by consistency run some Knuth–Bendix[180] or superposition[181] completion procedure which produces a huge number of irrelevant inferences under which the ones relevant for establishing the induction steps can hardly be made explicit. A proof attempt is successful when the prover has drawn all necessary inferences and stops without having detected any inconsistency.

Proof by consistency has shown to perform far worse than any other known form of mechanizing mathematical induction; mainly because it requires the generation of far too many superfluous inferences, and because its runs are typically infinite, and its admissibility conditions are too restrictive for most applications. Roughly speaking, the conceptual flaw in proof by consistency is that, instead of finding a sufficient set of reasonable inferences, the research follows the idea of ruling out as many irrelevant inferences as possible.

Implicit Induction Ordering: In the early implicit-induction systems,[182] induction proceeds over a syntactical term ordering, which typically cannot be made explicit in the sense that there would be some predicate term in the

---

[175] Historically important are also the following publications on rippling: [Hutter, 1990], [Bundy *et al.*, 1991], [Basin and Walsh, 1996].

[176] Cf. [Goguen, 1980], [Huet and Hullot, 1980], [Lankford, 1980], [Musser, 1980].

[177] Cf. [Göbel, 1985], [Jouannaud and Kounalis, 1986], [Fribourg, 1986], [Küchlin, 1989].

[178] Cf. e.g. [Zhang *et al.*, 1988], [Kapur and Zhang, 1989], [Bevers and Lewi, 1990], [Reddy, 1990], [Gramlich and Lindner, 1991], [Ganzinger and Stuber, 1992], [Bouhoula and Rusinowitch, 1995], [Padawitz, 1996].

[179] The name "proof by consistency" was coined in the title of [Kapur and Musser, 1987], which is the later published forerunner of its outstanding improved version [Kapur and Musser, 1986].

[180] See UNICOM [Gramlich and Lindner, 1991] for such a system, following [Bachmair, 1988] with several improvements. See [Knuth and Bendix, 1970] for the Knuth–Bendix completion procedure.

[181] See [Ganzinger and Stuber, 1992] for such a system.

[182] See [Gramlich and Lindner, 1991] and [Ganzinger and Stuber, 1992] for such systems.

logical syntax that denotes this ordering in the intended models of the specification. The semantical orderings of explicit induction, however, cannot depend on the precise syntactical term structure of a weight w, but only on the value of w under an evaluation in the intended models.

The price one has to pay for the possibility to have induction orderings that can also depend on the precise syntactical structure of terms is surprisingly high for powerful inference systems.[183]

The early implicit-induction systems needed such sophisticated term orderings,[184] because they started from the induction conclusion and every inference step reduced the formulas w.r.t. the induction ordering again and again, but an application of an induction hypothesis was admissible to greater formulas only. This deterioration of the ordering information with every inference step was overcome by the introduction of explicit weight terms in [Wirth and Becker, 1995], which obviate the former need for syntactical term orderings as induction orderings.

Descente Infinie ("Lazy Induction"): Contrary to explicit induction, where induction is introduced into an otherwise merely deductive inference system only by the explicit application of induction axioms in the induction rule, the cyclic arguments and their well-foundedness in implicit induction need not be confined to single inference steps.[185] The induction rule of explicit induction generates all induction hypotheses in a single inference step. To the contrary, in implicit induction, the inference system "knows" what an induction hypothesis is, i.e. it includes inference rules that provide or apply induction hypotheses, given that certain ordering conditions resulting from these applications can be met by an induction ordering. Because this aspect of implicit induction can facilitate the human-oriented induction method described in § 3.6, the name *descente infinie* was coined for it (cf. § 3.7). Researchers introduced to this aspect by [Protzen, 1994] (entitled "Lazy Generation of Induction Hypotheses") sometimes speak of "lazy induction" instead of *descente infinie*.

The entire handbook article [Comon, 2001] (with corrections in [Wirth, 2005a]) is dedicated to the two aspects of *proof by consistency* and *implicit induction orderings*. Today, however, the interest in these two aspects tends to be historical or theoretical, especially because these aspects can hardly be combined with explicit induction.

To the contrary, *descente infinie* synergetically combines with explicit induction, as witnessed by the QuodLibet system, which we will discuss in § 6.4.

[183] Cf. [Wirth, 1997].

[184] Cf. e.g. [Bachmair, 1988], [Steinbach, 1988; 1995], [Geser, 1996].

[185] For this reason, the funny name "inductionless induction" was originally coined for implicit induction in the titles of [Lankford, 1980; 1981] as a short form for "induction without induction rule". See also the title of [Goguen, 1980] for a similar phrase.

6.4 QuodLibet

In the last years of the Collaborative Research Center SFB 314 "Artificial Intelligence" (cf. § 5.6), after extensive experiments with several inductive theorem proving systems, such as Nqthm (cf. § 5.4), Inka (cf. § 5.6), Rrl (cf. § 5.6), and the implicit induction system Unicom [Gramlich and Lindner, 1991], Claus-Peter Wirth (*1963) and Ulrich Kühler (*1964) came to the conclusion that — in spite of the excellent interaction concept of Unicom[186] — *descente infinie* was actually the only aspect of implicit induction that deserved further investigation. Moreover, the coding of recursive functions in unconditional equations in Unicom turned out to be inadequate for inductive theorem proving in practice, where positive/negative-conditional equations were in demand for specification, as well as clausal logic for theorem proving [Kühler, 1991, pp. 134, 138].

Therefore, a new system had to be created, which was given the name QuodLibet (Latin for "as you like it"), because it should enable its users to avoid overspecification by admitting partial function specifications, and to execute proofs whose crucial proof steps mirror exactly the intended ones.

A concept for partial function specification instead of the totality requirement of explicit induction was easily obtained by elaborating the first part of [Wirth, 1991] into the framework for positive/negative-conditional rewrite systems of [Wirth and Gramlich, 1994a]. After inventing constructor variables in [Wirth *et al.*, 1993], the monotonicity of validity w.r.t. consistent extension of the partial specifications was easily achieved [Wirth and Gramlich, 1994b], so that the induction proofs did not have to be re-done after such an extension of a partially defined function.

Although the confluence criterion defining admissibility of function definitions in QuodLibet and guaranteeing (object-level) consistency (cf. § 4.1) of QuodLibet's functional specifications was completely presented in an appropriate form not before [Wirth, 2009], the essential admissibility requirements were already clear in 1996.[187]

The weak admissibility conditions of QuodLibet — mutually recursive functions, possibly partially defined because of missing cases or non-termination — are of practical relevance in applications. For instance, Bernd Löchner (*1967) (a user, not a developer of QuodLibet) concludes in [Löchner, 2006, p. 76]:

> "The translation of the different specifications into the input language of the inductive theorem prover QuodLibet [Avenhaus *et al.*, 2003] was straightforward. We later realized that this is difficult or impossible with several other inductive provers as these have problems with mutual recursive functions and partiality" . . .

[186]For the assessment of Unicom's interaction concept see [Kühler, 1991, p. 134ff.].

[187]See [Kühler and Wirth, 1996] for the first publication of the object-level consistency of the specifications that are admissible and supported with strong induction heuristics in QuodLibet. In [Kühler and Wirth, 1996], a huge proof from the original 1995 edition of [Wirth, 2005b] guaranteed the consistency; moreover, the most relevant of the seven inductive validities of [Wirth and Gramlich, 1994b] is chosen (no longer the initial or free models typical for implicit induction).

Based on the *descente infinie* inference system for clausal first-order logic of [Wirth and Kühler, 1995],[188] the system development of QuodLibet in Common Lisp (cf. § 5.5), mostly by Kühler and Tobias Schmidt-Samoa (*1973), lasted from 1995 to 2006. The system was described and demonstrated at the 19th Int. Conf. on Automated Deduction (CADE), Miami Beach (FL), 2003 [Avenhaus *et al.*, 2003]. The extension of the *descente infinie* inference systems of QuodLibet to the full [modal] higher-order logic of [Wirth, 2004; 2013] has not been implemented yet.

To the best of our knowledge, QuodLibet is the first theorem prover whose proof state is an and-or-tree (of clauses); actually, a forest of such trees, so that in a mutual induction proof each conjecture providing induction hypotheses has its own tree [Kühler, 2000]. An extension of the recursion analysis of [Boyer and Moore, 1979] for constructor-style specifications (cf. § 4.4) was developed by writing and testing tactics in QuodLibet's Pascal-like[189] meta-language Qml [Kühler, 2000]. To achieve an acceptable run-time performance (but not competitive with ACL2, of course), Qml tactics are compiled before execution.

In principle, termination proofs are not required, simply because termination is not an admissibility restriction in QuodLibet. Instead, definition-time recursion analysis uses induction lemmas (cf. § 5.3.7) to prove lemmas on function domains by induction.[190] At proof time, recursion analysis is used by the standard tactic only to determine the induction variables from the induction templates: As seen in Example 3 (as compared to Examples 12 and 23), subsumption and merging of schemes are not required in *descente infinie*.[191]

An enormous speed-up of QuodLibet and an extension of its automatically provable theorems was achieved by Schmidt-Samoa during his PhD work with the system in 2004–2006. He developed a marking concept for the tagging of rewrite lemmas (cf. § 5.3.1), where the elements of a clause can be marked as Forbidden, Mandatory, Obligatory, and Generous, to control the recursive relief of

---

[188] Later improvements of this inference system are found in [Wirth, 1997], [Kühler, 2000], and [Schmidt-Samoa, 2006b].

[189] See [Wirth, 1971] for the programming language Pascal. The critical decision for an imperative instead of a functional tactics language turned out to be most appropriate during the ten years of using Qml.

[190] While domain lemmas for totally defined functions use to be found without interaction and total functions do not provide relevant overhead in QuodLibet, the user often has to help in case of partial function definitions by providing *domain lemmas* such as Def $\mathsf{delfirst}(x, l)$, $\mathsf{mbp}(x, l) \neq \mathsf{true}$, for delfirst defined via (delfirst1–2) of § 3.5.

[191] Although it is not a must and not part of the standard tactic, induction hypotheses may be generated eagerly in QuodLibet to enhance generalization as in Example 5, in which case subsumption and merging of induction schemes as described in § 5.3.8 are required. Moreover, the concept of flawed induction schemes of QuodLibet (taken over from Thm as well, cf. § 5.3.8) depends on the mergeability of schemes. Furthermore, QuodLibet actually applies some merging techniques to plan case analyses optimized for induction [Kühler, 2000, § 8.3.3]. The question why QuodLibet adopts the great ideas of recursion analysis from Thm, but does not follow them precisely, has two answers: First, it was necessary to extend the heuristics of Thm to deal with constructor-style definitions. The second answer was already given in § 5.3.9: Testing is the only judge on heuristics.

conditions in contextual rewriting [Schmidt-Samoa, 2006b; 2006c]. Moreover, a very simple, but most effective reuse mechanism analyzes during a proof attempt whether it actually establishes a proof of some sub-clause, and uses this knowledge to crop conjunctive branches that do not contribute to the actual goal [Schmidt-Samoa, 2006b]. Finally, an even closer integration of linear logic (cf. Note 165) with excellent results [Schmidt-Samoa, 2006a; 2006b] questioned one of the basic principles of QuodLibet, namely the idea that the prover does not try to be clever, but stops early if there is no progress visible, and presents the human user the proof state in a nice graphical tree representation: The expanded highly-optimized formulation of arithmetic by means of special functions for the decidable fragment results in clauses that do not easily admit human inspection anymore. We did not find means to overcome this, because we did not find a way to fold theses clauses to achieve a human-oriented higher level of abstraction.

All in all, QuodLibet has proved that *descente infinie* ("lazy induction") goes well together with explicit induction and that we have reason to hope that eager induction-hypotheses generation can be overcome for theorems with difficult induction proofs, sacrificing neither efficiency nor the usefulness of the excellent heuristic knowledge developed in explicit induction. Why *descente infinie* and human-orientedness should remain on the agenda for induction in mathematics assistance systems is explained in the manifesto [Wirth, 2012c].

7 LESSONS LEARNED

What lessons can we draw from the history of the automation of induction?

Do not be too inclined to follow the current fads. Choose a hard problem, give thought to the "right" foundations, and then pursue its solution with patience and perseverance.

Another piece of oft-repeated advice to the young researcher: start simply. From the standpoint of formalizing microprocessors, investing in a theorem prover supporting only `NIL` and `CONS` is clearly inadequate. From the standpoint of understanding induction and simplification, however, it presents virtually all the problems, and its successors then gradually refined and elaborated the techniques. The four key provers discussed here — the Pure LISP Theorem Prover, Thm, Nqthm, and ACL2 — are clearly "of a kind". The lessons learned from one tool directly informed the design of the next.

If you are interested in building an inductive theorem prover, do not make the mistake of focusing merely on an induction principle and the heuristics for controlling it. A successful inductive theorem prover must be able to simplify and generalize. Ideally, it must be able to invent new concepts to express inductively provably theorems.

If theorems and proofs are simple and obvious for humans, a good automatic theorem prover ought not to struggle with them. If it takes a lot of time and machinery to prove obvious theorems, then truly interesting theorems are out of reach.

Do not be too eager to add features that break old ones. Instead, truly explore the extent to which new problems can be formalized within the existing framework so as to exploit the power of the existing system. Had Boyer and Moore adopted higher-order logic initially or attempted to solve the problem solely by exhaustive searching in a general purpose logic calculus, the discovery of many powerful techniques would have been delayed.

We strongly recommend collecting all your successful proofs into a regression suite and re-running your improved provers on this suite regularly. It is remarkably easy to "improve" a theorem prover such that it discovers a new proof at the cost of failing to re-discover old ones. The ACL2 regression suite contains over 90,000 `DEFTHM` commands, i.e. conjectures to be proved. It is an invaluable resource to Kaufmann and Moore when they explore new heuristics.

Finally, Boyer and Moore did not give names to their provers before ACL2, and so they became most commonly known under the name *the Boyer–Moore theorem prover*. So here is some advice to young researchers who want to become well-known: Build a good system, but do not give it a name, so that people have to attach your name to it!

8 CONCLUSION

> "One of the reasons our theorem prover is successful is that we trick the user into telling us the proof. And the best example of that, that I know, is: If you want to prove that there exists a prime factorization — that is to say a list of primes whose product is any given number — then the way you state it is: You define a function that takes a natural number and delivers a list of primes, and then you prove that it does that. And, of course, the definition of that function is everybody else's proof. The absence of quantifiers and the focus on constructive, you know, recursive definitions forces people to do the work. And so then, when the theorem prover proves it, they say 'Oh what wonderful theorem prover!', without even realizing they sweated bullets to express the theorem in that impoverished logic."

said Moore, and Boyer agreed laughingly.[192]

ACKNOWLEGDGEMENTS

We would like to thank Fabio Acerbi, Klaus Barner, Anne O. Boyer, Robert S. Boyer, Catherine Goldstein, Bernhard Gramlich, Matt Kaufmann, Ulrich Kühler, Jo Moore, Peter Padawitz, Marianeh Rezaei, and Tobias Schmidt-Samoa.

[192] [Wirth, 2012d].

BIBLIOGRAPHY

[Abrahams *et al.*, 1980] Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne, editors. *Conference Record of the 7th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Las Vegas (NV), 1980.* ACM Press, 1980. `http://dl.acm.org/citation.cfm?id=567446`.

[Acerbi, 2000] Fabio Acerbi. Plato: Parmenides 149a7–c3. A proof by complete induction? *Archive for History of Exact Sciences*, 55:57–76, 2000.

[Ackermann, 1928] Wilhelm Ackermann. Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99:118–133, 1928. Received Jan. 20, 1927.

[Ackermann, 1940] Wilhelm Ackermann. Zur Widerspruchsfreiheit der Zahlentheorie. *Mathematische Annalen*, 117:163–194, 1940. Received Aug. 15, 1939.

[Aït-Kaci and Nivat, 1989] Hassan Aït-Kaci and Maurice Nivat, editors. *Proc. of the Colloquium on Resolution of Equations in Algebraic Structures (CREAS), Lakeway, TX, 1987.* Academic Press (Elsevier), 1989.

[Anon, 2005] Anon. Advanced Architecture MicroProcessor 7 Government (AAMP7G) microprocessor. Rockwell Collins, Inc. WWW only: `http://www.rockwellcollins.com/sitecore/content/Data/Products/Information_Assurance/Cryptography/AAMP7G_Microprocessor.aspx`, 2005.

[Armando *et al.*, 2008] Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors. *4th Int. Joint Conf. on Automated Reasoning (IJCAR), Sydney, Australia, 2008*, number 5195 in Lecture Notes in Artificial Intelligence. Springer, 2008.

[Autexier *et al.*, 1999] Serge Autexier, Dieter Hutter, Heiko Mantel, and Axel Schairer. System description: INKA 5.0 – a logical voyager. 1999. In [Ganzinger, 1999, pp. 207–211].

[Autexier, 2005] Serge Autexier. On the dynamic increase of multiplicities in matrix proof methods for classical higher-order logic. 2005. In [Beckert, 2005, pp. 48–62].

[Avenhaus *et al.*, 2003] Jürgen Avenhaus, Ulrich Kühler, Tobias Schmidt-Samoa, and Claus-Peter Wirth. How to prove inductive theorems? QUODLIBET! 2003. In [Baader, 2003, pp. 328–333], `http://wirth.bplaced.net/p/quodlibet`.

[Baader, 2003] Franz Baader, editor. *19th Int. Conf. on Automated Deduction (CADE), Miami Beach (FL), 2003*, number 2741 in Lecture Notes in Artificial Intelligence. Springer, 2003.

[Baaz and Leitsch, 1995] Matthias Baaz and Alexander Leitsch. Methods of functional extension. *Collegium Logicum — Annals of the Kurt Gödel Society*, 1:87–122, 1995.

[Bachmair *et al.*, 1992] Leo Bachmair, Harald Ganzinger, and Wolfgang J. Paul, editors. *Informatik – Festschrift zum 60. Geburtstag von Günter Hotz.* B. G. Teubner Verlagsgesellschaft, 1992.

[Bachmair, 1988] Leo Bachmair. Proof by consistency in equational theories. 1988. In [LICS, 1988, pp. 228–233].

[Bajscy, 1993] Ruzena Bajscy, editor. *Proc. 13th Int. Joint Conf. on Artificial Intelligence (IJCAI), Chambery, France.* Morgan Kaufman (Elsevier), 1993. `http://ijcai.org/Past%20Proceedings`.

[Barendregt, 1981] Hen(dri)k P. Barendregt. *The Lambda Calculus — Its Syntax and Semantics.* Number 103 in Studies in Logic and the Foundations of Mathematics. North-Holland (Elsevier), 1981. 1st edn. (final rev. edn. is [Barendregt, 2012]).

[Barendregt, 2012] Hen(dri)k P. Barendregt. *The Lambda Calculus — Its Syntax and Semantics.* Number 40 in Studies in Logic. College Publications, London, 2012. 6th rev. edn. (1st edn. is [Barendregt, 1981]).

[Barner, 2001a] Klaus Barner. Pierre Fermat (1601?–1665) — His life beside mathematics. *European Mathematical Society Newsletter*, 43 (Dec. 2001):12–16, 2001. Long version in German is [Barner, 2001b]. `www.ems-ph.org/journals/newsletter/pdf/2001-12-42.pdf`.

[Barner, 2001b] Klaus Barner. Das Leben Fermats. *DMV-Mitteilungen*, 3/2001:12–26, 2001. Extensions in [Barner, 2007]. Short versions in English are [Barner, 2001c; 2001a].

[Barner, 2001c] Klaus Barner. How old did Fermat become? *NTM Internationale Zeitschrift für Geschichte und Ethik der Naturwissenschaften, Technik und Medizin, Neue Serie, ISSN 00366978*, 9:209–228, 2001. Long version in German is [Barner, 2001b]. New results on the subject in [Barner, 2007].

[Barner, 2007] Klaus Barner. Neues zu Fermats Geburtsdatum. *DMV-Mitteilungen*, 15:12–14, 2007. *(Further support for the results of [Barner, 2001c], narrowing down Fermat's birth date from 1607/8 to Nov. 1607).*

[Basin and Walsh, 1996] David Basin and Toby Walsh. A calculus for and termination of rippling. *J. Automated Reasoning*, 16:147–180, 1996.

[Becker, 1965] Oscar Becker, editor. *Zur Geschichte der griechischen Mathematik.* Wissenschaftliche Buchgesellschaft, Darmstadt, 1965.

[Beckert, 2005] Bernhard Beckert, editor. *14^{th} Int. Conf. on Tableaux and Related Methods, Koblenz (Germany), 2005*, number 3702 in Lecture Notes in Artificial Intelligence. Springer, 2005.

[Benzmüller *et al.*, 2008] Christoph Benzmüller, Frank Theiss, Larry Paulson, and Arnaud Fietzke. LEO-II — a cooperative automatic theorem prover for higher-order logic. 2008. In [Armando *et al.*, 2008, pp. 162–170].

[Berka and Kreiser, 1973] Karel Berka and Lothar Kreiser, editors. *Logik-Texte – Kommentierte Auswahl zur Geschichte der modernen Logik.* Akademie Verlag GmbH, Berlin, 1973. 2^{nd} rev. edn. (1^{st} edn. 1971; 4^{th} rev. rev. edn. 1986).

[Bevers and Lewi, 1990] Eddy Bevers and Johan Lewi. Proof by consistency in conditional equational theories. Tech. Report CW 102, Dept. Comp. Sci., K. U. Leuven, 1990. Rev. July 1990. Short version in [Kaplan and Okada, 1991, pp. 194–205].

[Bevier *et al.*, 1989] William R. Bevier, Warren A. Hunt, J Strother Moore, and William D. Young. An approach to systems verification. *J. Automated Reasoning*, 5:411–428, 1989.

[Bevier, 1989] William R. Bevier. Kit and the short stack. *J. Automated Reasoning*, 5:519–530, 1989.

[Bibel and Kowalski, 1980] Wolfgang Bibel and Robert A. Kowalski, editors. *5^{th} Int. Conf. on Automated Deduction (CADE), Les Arcs, France, 1980*, number 87 in Lecture Notes in Computer Science. Springer, 1980.

[Biundo *et al.*, 1986] Susanne Biundo, Birgit Hummel, Dieter Hutter, and Christoph Walther. The Karlsruhe inductive theorem proving system. 1986. In [Siekmann, 1986, pp. 673–675].

[Bledsoe and Loveland, 1984] W. W. Bledsoe and Donald W. Loveland, editors. *Automated Theorem Proving: After 25 Years.* Number 29 in Contemporary Mathematics. American Math. Soc., Providence (RI), 1984. Proc. of the Special Session on Automatic Theorem Proving, 89^{th} Annual Meeting of the American Math. Soc., Denver (CO), Jan. 1983.

[Bledsoe *et al.*, 1971] W. W. Bledsoe, Robert S. Boyer, and William H. Henneman. Computer proofs of limit theorems. 1971. In [Cooper, 1971, pp. 586–600]. Long version is [Bledsoe *et al.*, 1972].

[Bledsoe *et al.*, 1972] W. W. Bledsoe, Robert S. Boyer, and William H. Henneman. Computer proofs of limit theorems. *Artificial Intelligence*, 3:27–60, 1972. Short version is [Bledsoe *et al.*, 1971].

[Bledsoe, 1971] W. W. Bledsoe. Splitting and reduction heuristics in automatic theorem proving. *Artificial Intelligence*, 2:55–77, 1971.

[Bouajjani and Maler, 2009] Ahmed Bouajjani and Oded Maler, editors. *Proc. 21^{st} Int. Conf. on Computer Aided Verification (CAV), Grenoble (France), 2009*, volume 5643 of *Lecture Notes in Computer Science.* Springer, 2009.

[Bouhoula and Rusinowitch, 1995] Adel Bouhoula and Michaël Rusinowitch. Implicit induction in conditional theories. *J. Automated Reasoning*, 14:189–235, 1995.

[Bourbaki, 1939] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Fascicule De Résultats.* Number 846 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1939. 1^{st} edn., VIII + 50 pp.. Review is [Church, 1946]. 2^{nd} rev. extd. edn. is [Bourbaki, 1951].

[Bourbaki, 1951] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Fascicule De Résultats.* Number 846-1141 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1951. 2^{nd} rev. extd. edn. of [Bourbaki, 1939]. 3^{rd} rev. extd. edn. is [Bourbaki, 1958b].

[Bourbaki, 1954] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre I & II.* Number 1212 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1954. 1^{st} edn.. 2^{nd} rev. edn. is [Bourbaki, 1960].

[Bourbaki, 1956] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre III.* Number 1243 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1956. 1st edn., II + 119 + 4 (mode d'emploi) + 23 (errata no. 6) pp.. 2nd rev. extd. edn. is [Bourbaki, 1967].

[Bourbaki, 1958a] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre IV.* Number 1258 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1958. 1st edn.. 2nd rev. extd. edn. is [Bourbaki, 1966a].

[Bourbaki, 1958b] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Fascicule De Résultats.* Number 1141 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1958. 3rd rev. extd. edn. of [Bourbaki, 1951]. 4th rev. extd. edn. is [Bourbaki, 1964].

[Bourbaki, 1960] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre I & II.* Number 1212 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1960. 2nd rev. extd. edn. of [Bourbaki, 1954]. 3rd rev. edn. is [Bourbaki, 1966b].

[Bourbaki, 1964] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Fascicule De Résultats.* Number 1141 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1964. 4th rev. extd. edn. of [Bourbaki, 1958b]. 5th rev. extd. edn. is [Bourbaki, 1968b].

[Bourbaki, 1966a] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre IV.* Number 1258 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1966. 2nd rev. extd. edn. of [Bourbaki, 1958a]. English translation in [Bourbaki, 1968a].

[Bourbaki, 1966b] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitres I & II.* Number 1212 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1966. 3rd rev. edn. of [Bourbaki, 1960], VI + 143 + 7 (errata no. 13) pp.. English translation in [Bourbaki, 1968a].

[Bourbaki, 1967] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Chapitre III.* Number 1243 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1967. 2nd rev. extd. edn. of [Bourbaki, 1956], 151 + 7 (errata no. 13) pp.. 3rd rev. edn. results from executing these errata. English translation in [Bourbaki, 1968a].

[Bourbaki, 1968a] Nicolas Bourbaki. *Elements of Mathematics — Theory of Sets.* Actualités Scientifiques et Industrielles. Hermann, Paris, jointly published with AdiWes International Series in Mathematics, Addison–Wesley, Reading (MA), 1968. English translation of [Bourbaki, 1966b; 1967; 1966a; 1968b].

[Bourbaki, 1968b] Nicolas Bourbaki. *Éléments des Mathématique — Livre 1: Théorie des Ensembles. Fascicule De Résultats.* Number 1141 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1968. 5th rev. extd. edn. of [Bourbaki, 1964]. English translation in [Bourbaki, 1968a].

[Boyer and Moore, 1971] Robert S. Boyer and J Strother Moore. The sharing of structure in resolution programs. Memo 47, Univ. Edinburgh, Dept. of Computational Logic, 1971. II + 24 pp.. Revised version is [Boyer and Moore, 1972].

[Boyer and Moore, 1972] Robert S. Boyer and J Strother Moore. The sharing of structure in theorem-proving programs. 1972. In [Meltzer and Michie, 1972, pp. 101–116].

[Boyer and Moore, 1973] Robert S. Boyer and J Strother Moore. Proving theorems about LISP functions. 1973. In [Nilsson, 1973, pp. 486–493]. `http://ijcai.org/Past%20Proceedings/IJCAI-73/PDF/053.pdf`. Rev. version, extd. with a section "Failures", is [Boyer and Moore, 1975].

[Boyer and Moore, 1975] Robert S. Boyer and J Strother Moore. Proving theorems about LISP functions. *J. of the ACM*, 22:129–144, 1975. Rev. extd. edn. of [Boyer and Moore, 1973]. Received Sept. 1973, Rev. April 1974.

[Boyer and Moore, 1977a] Robert S. Boyer and J Strother Moore. A fast string searching algorithm. *Comm. ACM*, 20:762–772, 1977. `http://doi.acm.org/10.1145/359842.359859`.

[Boyer and Moore, 1977b] Robert S. Boyer and J Strother Moore. A lemma driven automatic theorem prover for recursive function theory. 1977. In [Reddy, 1977, Vol. I, pp. 511–519]. `http://ijcai.org/Past%20Proceedings/IJCAI-77-VOL1/PDF/089.pdf`.

[Boyer and Moore, 1979] Robert S. Boyer and J Strother Moore. *A Computational Logic.* Academic Press (Elsevier), 1979.

[Boyer and Moore, 1981a] Robert S. Boyer and J Strother Moore, editors. *The Correctness Problem in Computer Science.* Academic Press (Elsevier), 1981.

[Boyer and Moore, 1981b] Robert S. Boyer and J Strother Moore. Metafunctions: Proving them correct and using them efficiently as new proof procedures. 1981. In [Boyer and Moore, 1981a, pp. 103–184].

[Boyer and Moore, 1984a] Robert S. Boyer and J Strother Moore. A mechanical proof of the Turing completeness of pure LISP. 1984. In [Bledsoe and Loveland, 1984, pp. 133–167].

[Boyer and Moore, 1984b] Robert S. Boyer and J Strother Moore. A mechanical proof of the unsolvability of the halting problem. *J. of the ACM*, 31:441–458, 1984.

[Boyer and Moore, 1984c] Robert S. Boyer and J Strother Moore. Proof checking the RSA public key encryption algorithm. *American Mathematical Monthly*, 91:181–189, 1984.

[Boyer and Moore, 1985] Robert S. Boyer and J Strother Moore. Program verification. *J. Automated Reasoning*, 1:17–23, 1985.

[Boyer and Moore, 1987] Robert S. Boyer and J Strother Moore. The addition of bounded quantification and partial functions to a computational logic and its theorem prover. Technical Report ICSCA-CMP-52, Inst. for Computing Science and Computing Applications, The University of Texas at Austin, 1987. Printed Jan. 1987. Also published as [Boyer and Moore, 1988a; 1989].

[Boyer and Moore, 1988a] Robert S. Boyer and J Strother Moore. The addition of bounded quantification and partial functions to a computational logic and its theorem prover. *J. Automated Reasoning*, 4:117–172, 1988. Received Feb. 11, 1987. Also pubished as [Boyer and Moore, 1987; 1989].

[Boyer and Moore, 1988b] Robert S. Boyer and J Strother Moore. *A Computational Logic Handbook*. Number 23 in Perspectives in Computing. Academic Press (Elsevier), 1988. 2^{nd} rev. extd. edn. is [Boyer and Moore, 1998].

[Boyer and Moore, 1988c] Robert S. Boyer and J Strother Moore. Integrating decision procedures into heuristic theorem provers: A case study of linear arithmetic. 1988. In [Hayes *et al.*, 1988, pp. 83–124].

[Boyer and Moore, 1989] Robert S. Boyer and J Strother Moore. The addition of bounded quantification and partial functions to a computational logic and its theorem prover. 1989. In [Broy, 1989, pp. 95–145] (received Jan. 1988). Also published as [Boyer and Moore, 1987; 1988a].

[Boyer and Moore, 1990] Robert S. Boyer and J Strother Moore. A theorem prover for a computational logic. 1990. In [Stickel, 1990, pp. 1–15].

[Boyer and Moore, 1998] Robert S. Boyer and J Strother Moore. *A Computational Logic Handbook*. International Series in Formal Methods. Academic Press (Elsevier), 1998. 2^{nd} rev. extd. edn. of [Boyer and Moore, 1988b], rev. to work with NQTHM–1992, a new version of NQTHM.

[Boyer and Yu, 1992] Robert S. Boyer and Yuan Yu. Automated correctness proofs of machine code programs for a commercial microprocessor. 1992. In [Kapur, 1992, 416–430].

[Boyer and Yu, 1996] Robert S. Boyer and Yuan Yu. Automated proofs of object code for a widely used microprocessor. *J. of the ACM*, 43:166–192, 1996.

[Boyer *et al.*, 1973] Robert S. Boyer, D. Julian M. Davies, and J Strother Moore. The 77-editor. Memo 62, Univ. Edinburgh, Dept. of Computational Logic, 1973.

[Boyer *et al.*, 1976] Robert S. Boyer, J Strother Moore, and Robert E. Shostak. Primitive recursive program transformations. 1976. In [Graham *et al.*, 1976, pp. 171–174]. `http://doi.acm.org/10.1145/800168.811550`.

[Boyer, 1971] Robert S. Boyer. *Locking: a restriction of resolution*. PhD thesis, The University of Texas at Austin, 1971.

[Boyer, 2012] Robert S. Boyer. E-mail to Claus-Peter Wirth, Nov. 19, 2012.

[Brock and Hunt, 1999] Bishop Brock and Warren A. Hunt. Formal analysis of the Motorola CAP DSP. 1999. In [Hinchey and Bowen, 1999, pp. 81–116].

[Brotherston and Simpson, 2007] James Brotherston and Alex Simpson. Complete sequent calculi for induction and infinite descent. 2007. In [LICS, 2007, pp. 51–62?]. Thoroughly rev. version in [Brotherston and Simpson, 2011].

[Brotherston and Simpson, 2011] James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *J. Logic and Computation*, 21:1177–1216, 2011. Thoroughly rev. version of [Brotherston, 2005] and [Brotherston and Simpson, 2007]. Received April 3, 2009. Published online Sept. 30, 2010, `http://dx.doi.org/10.1093/logcom/exq052`.

[Brotherston, 2005] James Brotherston. Cyclic proofs for first-order logic with inductive definitions. 2005. In [Beckert, 2005, pp. 78–92]. Thoroughly rev. version in [Brotherston and Simpson, 2011].

[Broy, 1989] Manfred Broy, editor. *Constructive Methods in Computing Science*, number F 55 in NATO ASI Series. Springer, 1989.

[Buch and Hillenbrand, 1996] Armin Buch and Thomas Hillenbrand. WALDMEISTER*: Development of a High Performance Completion-Based Theorem Prover.* SEKI-Report SR–96–01 (ISSN 1860–5931). SEKI Publications, FB Informatik, Univ. Kaiserslautern, 1996. agent.informatik.uni-kl.de/seki/1996/Buch.SR-96-01.ps.gz.

[Bundy *et al.*, 1989] Alan Bundy, Frank van Harmelen, Jane Hesketh, Alan Smaill, and Andrew Stevens. A rational reconstruction and extension of recursion analysis. 1989. In [Sridharan, 1989, pp. 359–365].

[Bundy *et al.*, 1990] Alan Bundy, Frank van Harmelen, Christian Horn, and Alan Smaill. The OYSTER/CLAM system. 1990. In [Stickel, 1990, pp. 647–648].

[Bundy *et al.*, 1991] Alan Bundy, Andrew Stevens, Frank van Harmelen, Andrew Ireland, and Alan Smaill. *Rippling: A Heuristic for Guiding Inductive Proofs.* 1991. DAI Research Paper No. 567, Dept. Artificial Intelligence, Univ. Edinburgh. Also in Artificial Intelligence (1993) **62**, pp. 185–253.

[Bundy *et al.*, 2005] Alan Bundy, Dieter Hutter, David Basin, and Andrew Ireland. *Rippling: Meta-Level Guidance for Mathematical Reasoning.* Cambridge Univ. Press, 2005.

[Bundy, 1988] Alan Bundy. *The use of Explicit Plans to Guide Inductive Proofs.* 1988. DAI Research Paper No. 349, Dept. Artificial Intelligence, Univ. Edinburgh. Short version in [Lusk and Overbeek, 1988, pp. 111–120].

[Bundy, 1989] Alan Bundy. *A Science of Reasoning.* 1989. DAI Research Paper No. 445, Dept. Artificial Intelligence, Univ. Edinburgh. Also in [Lassez and Plotkin, 1991, pp. 178–198].

[Bundy, 1994] Alan Bundy, editor. *12th Int. Conf. on Automated Deduction (CADE), Nancy, 1994*, number 814 in Lecture Notes in Artificial Intelligence. Springer, 1994.

[Bundy, 1999] Alan Bundy. *The Automation of Proof by Mathematical Induction.* Informatics Research Report No. 2, Division of Informatics, Univ. Edinburgh, 1999. Also in [Robinson and Voronkow, 2001, Vol. 1, pp. 845–911].

[Burstall *et al.*, 1971] Rod M. Burstall, John S. Collins, and Robin J. Popplestone. *Programming in* POP–2. Univ. Edinburgh Press, 1971.

[Burstall, 1969] Rod M. Burstall. Proving properties of programs by structural induction. *The Computer Journal*, 12:48–51, 1969. Received April 1968, rev. Aug. 1968.

[Bussey, 1917] W. H. Bussey. The origin of mathematical induction. *American Mathematical Monthly*, XXIV:199–207, 1917.

[Bussotti, 2006] Paolo Bussotti. *From Fermat to Gauß: indefinite descent and methods of reduction in number theory.* Number 55 in Algorismus. Dr. Erwin Rauner Verlag, Augsburg, 2006.

[Cajori, 1918] Florian Cajori. Origin of the name "mathematical induction". *American Mathematical Monthly*, 25:197–201, 1918.

[Church, 1946] Alonzo Church. Review of [Bourbaki, 1939]. *J. Symbolic Logic*, 11:91, 1946.

[Cohn, 1965] Paul Moritz Cohn. *Universal Algebra.* Harper & Row, New York, 1965. 1st edn.. 2nd rev. edn. is [Cohn, 1981].

[Cohn, 1981] Paul Moritz Cohn. *Universal Algebra.* Number 6 in Mathematics and Its Applications. D. Reidel Publ., Dordrecht, now part of Springer Science+Business Media, 1981. 2nd rev. edn. (1st edn. is [Cohn, 1965]).

[Comon, 1997] Hubert Comon, editor. *8th Int. Conf. on Rewriting Techniques and Applications (RTA), Sitges (Spain), 1997*, number 1232 in Lecture Notes in Computer Science. Springer, 1997.

[Comon, 2001] Hubert Comon. Inductionless induction. 2001. In [Robinson and Voronkow, 2001, Vol. I, pp. 913–970].

[Constable *et al.*, 1985] Robert L. Constable, Stuart F. Allen, H. M. Bromly, W. R. Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the* NUPRL *Proof Development System.* Prentice–Hall, Inc., 1985. http://www.nuprl.org/book.

[Cooper, 1971] D. C. Cooper, editor. *Proc. 2nd Int. Joint Conf. on Artificial Intelligence (IJCAI), Sept. 1971, Imperial College, London.* Morgan Kaufman, Los Altos (CA), 1971. http://ijcai.org/Past%20Proceedings/IJCAI-1971/CONTENT/content.htm.

[DAC, 2001] *Proc. 38th Design Automation Conference (DAC), Las Vegas (NV), 2001.* ACM, 2001.

[Davis, 2009] Jared Davis. *A Self-Verifying Theorem Prover.* PhD thesis, The University of Texas at Austin, 2009.

[Dedekind, 1888] Richard Dedekind. *Was sind und was sollen die Zahlen.* Vieweg, Braunschweig, 1888. Also in [Dedekind, 1930–32, Vol. 3, pp. 335–391]. Also in [Dedekind, 1969].

[Dedekind, 1930–32] Richard Dedekind. *Gesammelte mathematische Werke.* Vieweg, Braunschweig, 1930–32. Ed. by Robert Fricke, Emmy Noether, and Öystein Ore.

[Dedekind, 1969] Richard Dedekind. *Was sind und was sollen die Zahlen? Stetigkeit und irrationale Zahlen.* Friedrich Vieweg und Sohn, Braunschweig, 1969.

[Dennis *et al.*, 2005] Louise A. Dennis, Mateja Jamnik, and Martin Pollet. On the comparison of proof planning systems λCLAM, ΩMEGA and ISAPLANNER. *Electronic Notes in Theoretical Computer Sci.*, 151:93–110, 2005.

[Dershowitz and Jouannaud, 1990] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. 1990. In [Leeuwen, 1990, Vol. B, pp. 243–320].

[Dershowitz and Lindenstrauss, 1995] Nachum Dershowitz and Naomi Lindenstrauss, editors. *4th Int. Workshop on Conditional Term Rewriting Systems (CTRS), Jerusalem, 1994*, number 968 in Lecture Notes in Computer Science, 1995.

[Dershowitz, 1989] Nachum Dershowitz, editor. *3rd Int. Conf. on Rewriting Techniques and Applications (RTA), Chapel Hill (NC), 1989*, number 355 in Lecture Notes in Computer Science. Springer, 1989.

[Euclid, ca. 300 B.C.] Euclid, of Alexandria. *Elements.* ca. 300 B.C.. Web version without the figures: `http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.01.0085`. English translation: Thomas L. Heath (ed.). *The Thirteen Books of Euclid's Elements.* Cambridge Univ. Press, 1908; web version without the figures: `http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.01.0086`. English web version (incl. figures): D. E. Joyce (ed.). *Euclid's Elements.* `http://aleph0.clarku.edu/~djoyce/java/elements/elements.html`, Dept. Math. & Comp. Sci., Clark Univ., Worcester (MA).

[Fermat, 1891ff.] Pierre Fermat. *Œuvres de Fermat.* Gauthier-Villars, Paris, 1891ff.. Ed. by Paul Tannery, Charles Henry.

[Fitting, 1990] Melvin Fitting. *First-order logic and automated theorem proving.* Springer, 1990. 1st edn. (2nd rev. edn. is [Fitting, 1996]).

[Fitting, 1996] Melvin Fitting. *First-order logic and automated theorem proving.* Springer, 1996. 2nd rev. edn. (1st edn. is [Fitting, 1990]).

[FOCS, 1980] *Proc. 21st Annual Symposium on Foundations of Computer Sci., Syracuse, 1980.* IEEE Press, 1980. `http://ieee-focs.org/`.

[Fowler, 1994] David Fowler. Could the Greeks have used mathematical induction? Did they use it? *Physis*, XXXI(1):253–265, 1994.

[Freudenthal, 1953] Hans Freudenthal. Zur Geschichte der vollständigen Induktion. *Archives Internationales d'Histoire des Sciences*, 6:17–37, 1953.

[Fribourg, 1986] Laurent Fribourg. A strong restriction of the inductive completion procedure. 1986. In [Kott, 1986, pp. 105–116]. Also in J. Symbolic Computation **8**, pp. 253–276, Academic Press (Elsevier), 1989.

[Fries, 1822] Jakob Friedrich Fries. *Die mathematische Naturphilosophie nach philosophischer Methode bearbeitet – Ein Versuch.* Christian Friedrich Winter, Heidelberg, 1822.

[Fritz, 1945] Kurt von Fritz. The discovery of incommensurability by Hippasus of Metapontum. *Annals of Mathematics*, 46:242–264, 1945. German translation: *Die Entdeckung der Inkommensurabilität durch Hippasos von Metapont* in [Becker, 1965, pp. 271–308].

[Fuchi and Kott, 1988] Kazuhiro Fuchi and Laurent Kott, editors. *Programming of Future Generation Computers II: Proc. of the 2nd Franco-Japanese Symposium.* North-Holland (Elsevier), 1988.

[Gabbay and Woods, 2004ff.] Dov Gabbay and John Woods, editors. *Handbook of the History of Logic.* North-Holland (Elsevier), 2004ff..

[Gabbay *et al.*, 1994] Dov Gabbay, Christopher John Hogger, and J. Alan Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 2: Deduction Methodologies.* Oxford Univ. Press, 1994.

[Ganzinger and Stuber, 1992] Harald Ganzinger and Jürgen Stuber. Inductive Theorem Proving by Consistency for First-Order Clauses. 1992. In [Bachmair *et al.*, 1992, pp. 441–462]. Also in [Rusinowitch and Remy, 1993, pp. 226–241].

[Ganzinger, 1996] Harald Ganzinger, editor. *7th Int. Conf. on Rewriting Techniques and Applications (RTA), New Brunswick (NJ), 1996*, number 1103 in Lecture Notes in Computer Science. Springer, 1996.

[Ganzinger, 1999] Harald Ganzinger, editor. *16th Int. Conf. on Automated Deduction (CADE), Trento, Italy, 1999*, number 1632 in Lecture Notes in Artificial Intelligence. Springer, 1999.

[Gentzen, 1935] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1935. Also in [Berka and Kreiser, 1973, pp. 192–253]. English translation in [Gentzen, 1969].

[Gentzen, 1969] Gerhard Gentzen. *The Collected Papers of Gerhard Gentzen.* North-Holland (Elsevier), 1969. Ed. by Manfred E. Szabo.

[Geser, 1995] Alfons Geser. A principle of non-wellfounded induction. 1995. In [Margaria, 1995, pp. 117–124].

[Geser, 1996] Alfons Geser. An improved general path order. *J. Applicable Algebra in Engineering, Communication and Computing (AAECC)*, 7:469–511, 1996.

[Gillman, 1987] Leonard Gillman. *Writing Mathematics Well.* The Mathematical Association of America, 1987.

[Göbel, 1985] Richard Göbel. Completion of globally finite term rewriting systems for inductive proofs. 1985. In [Stoyan, 1985, pp. 101–110].

[Gödel, 1931] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931. With English translation also in [Gödel, 1986ff., Vol. I, pp. 145–195]. English translation also in [Heijenoort, 1971, pp. 596–616] and in [Gödel, 1962].

[Gödel, 1962] Kurt Gödel. *On formally undecidable propositions of Principia Mathematica and related systems.* Basic Books, New York, 1962. English translation of [Gödel, 1931] by Bernard Meltzer. With an introduction by R. B. BRAITHWAITE. 2nd edn. by Dover Publications, 1992.

[Gödel, 1986ff.] Kurt Gödel. *Collected Works.* Oxford Univ. Press, 1986ff. Ed. by Sol(omon) Feferman, John W. Dawson Jr., Warren Goldfarb, Jean van Heijenoort, Stephen C. Kleene, Charles Parsons, Wilfried Sieg, *et al.*.

[Goguen, 1980] Joseph Goguen. How to prove algebraic inductive hypotheses without induction. 1980. In [Bibel and Kowalski, 1980, pp. 356–373].

[Goldstein, 2008] Catherine Goldstein. Pierre Fermat. 2008. In [Gowers *et al.*, 2008, § VI.12, pp. 740–741].

[Gore *et al.*, 2001] Rajeev Gore, Alexander Leitsch, and Tobias Nipkow, editors. *1st Int. Joint Conf. on Automated Reasoning (IJCAR), Siena, Italy, 2001*, number 2083 in Lecture Notes in Artificial Intelligence. Springer, 2001.

[Gowers *et al.*, 2008] Timothy Gowers, June Barrow-Green, and Imre Leader, editors. *The Princeton Companion to Mathematics.* Princeton Univ. Press, 2008.

[Graham *et al.*, 1976] Susan L. Graham, Robert M. Graham, Michael A. Harrison, William I. Grosky, and Jeffrey D. Ullman, editors. *Conference Record of the 3rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Atlanta (GA), Jan. 1976.* ACM Press, 1976. `http://dl.acm.org/citation.cfm?id=800168`.

[Gramlich and Lindner, 1991] Bernhard Gramlich and Wolfgang Lindner. *A Guide to* UNICOM, *an Inductive Theorem Prover Based on Rewriting and Completion Techniques.* SEKI-Report SR–91–17 (ISSN 1860–5931). SEKI Publications, FB Informatik, Univ. Kaiserslautern, 1991. `http://agent.informatik.uni-kl.de/seki/1991/Lindner.SR-91-17.ps.gz`.

[Gramlich and Wirth, 1996] Bernhard Gramlich and Claus-Peter Wirth. Confluence of terminating conditional term rewriting systems revisited. 1996. In [Ganzinger, 1996, pp. 245–259].

[Hayes *et al.*, 1988] J. E. Hayes, Donald Michie, and Judith Richards, editors. *Proceedings of the 11th Annual Machine Intelligence Workshop (Machine Intelligence 11), Univ. Strathclyde, Glasgow, 1985.* Clarendon Press, Oxford (Oxford Univ. Press), 1988.

[Heijenoort, 1971] Jean van Heijenoort. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931.* Harvard Univ. Press, 1971. 2nd rev. edn. (1st edn. 1967).

[Herbelin, 2009] Hugo Herbelin, editor. *The 1st* COQ *Workshop.* Inst. für Informatik, Tech. Univ. München, 2009. TUM-I0919, `http://www.lix.polytechnique.fr/coq/files/coq-workshop-TUM-I0919.pdf`.

[Hilbert and Bernays, 1934] David Hilbert and Paul Bernays. *Die Grundlagen der Mathematik — Erster Band.* Number XL in Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen. Springer, 1934. 1st edn. (2nd edn. is [Hilbert and Bernays, 1968]).

[Hilbert and Bernays, 1939] David Hilbert and Paul Bernays. *Die Grundlagen der Mathematik — Zweiter Band*. Number L in Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen. Springer, 1939. 1^{st} edn. (2^{nd} edn. is [Hilbert and Bernays, 1970]).

[Hilbert and Bernays, 1968] David Hilbert and Paul Bernays. *Die Grundlagen der Mathematik I*. Number 40 in Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen. Springer, 1968. 2^{nd} rev. edn. of [Hilbert and Bernays, 1934].

[Hilbert and Bernays, 1970] David Hilbert and Paul Bernays. *Die Grundlagen der Mathematik II*. Number 50 in Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen. Springer, 1970. 2^{nd} rev. edn. of [Hilbert and Bernays, 1939].

[Hilbert and Bernays, 2011] David Hilbert and Paul Bernays. *Grundlagen der Mathematik I — Foundations of Mathematics I, Part A: Prefaces and §§ 1–2*. College Publications, London, 2011. First English translation. Bilingual facsimile edn. of the 2^{nd} German edn. [Hilbert and Bernays, 1968], incl. the annotation and translation of all differences of the 1^{st} German edn. [Hilbert and Bernays, 1934]. Translated and commented by Claus-Peter Wirth. Ed. by Claus-Peter Wirth, Jörg Siekmann, Michael Gabbay, Dov Gabbay. Advisory Board: Wilfried Sieg (chair), Irving H. Anellis, Steve Awodey, Matthias Baaz, Wilfried Buchholz, Bernd Buldt, Reinhard Kahle, Paolo Mancosu, Charles Parsons, Volker Peckhaus, William W. Tait, Christian Tapp, Richard Zach. ISBN 978–1–84890–033–2.

[Hillenbrand and Löchner, 2002] Thomas Hillenbrand and Bernd Löchner. The next WALDMEISTER loop. 2002. In [Voronkov, 2002, pp. 486–500]. `http://www.waldmeister.org`.

[Hinchey and Bowen, 1999] Michael G. Hinchey and Jonathan P. Bowen, editors. *Industrial-Strength Formal Methods in Practice*. Formal Approaches to Computing and Information Technology (FACIT). Springer, 1999.

[Howard and Rubin, 1998] Paul Howard and Jean E. Rubin. *Consequences of the Axiom of Choice*. American Math. Society, 1998.

[Hudlak *et al.*, 1999] Paul Hudlak, John Peterson, and Joseph H. Fasel. A gentle introduction to HASKELL. Web only: `http://www.haskell.org/tutorial`, 1999.

[Huet and Hullot, 1980] Gérard Huet and Jean-Marie Hullot. Proofs by induction in equational theories with constructors. 1980. In [FOCS, 1980, pp. 96–107]. Also in J. Computer and System Sci. **25**, pp. 239–266, Academic Press (Elsevier), 1982.

[Huet, 1980] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the ACM*, 27:797–821, 1980.

[Hunt and Swords, 2009] Warren A. Hunt and Sol Swords. Centaur technology media unit verification. 2009. In [Bouajjani and Maler, 2009, pp. 353–367].

[Hunt, 1985] Warren A. Hunt. *FM8501: A Verified Microprocessor.* PhD thesis, The University of Texas at Austin, 1985. Also published as [Hunt, 1994].

[Hunt, 1989] Warren A. Hunt. Microprocessor design verification. *J. Automated Reasoning*, 5:429–460, 1989.

[Hunt, 1994] Warren A. Hunt. *FM8501: A Verified Microprocessor*. Number 795 in Lecture Notes in Artificial Intelligence. Springer, 1994. Already published as [Hunt, 1985].

[Hutter and Bundy, 1999] Dieter Hutter and Alan Bundy. The design of the CADE-16 Inductive Theorem Prover Contest. 1999. In [Ganzinger, 1999, pp. 374–377].

[Hutter and Sengler, 1996] Dieter Hutter and Claus Sengler. INKA: the next generation. 1996. In [McRobbie and Slaney, 1996, pp. 288–292].

[Hutter and Stephan, 2005] Dieter Hutter and Werner Stephan, editors. *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60^{th} Birthday.* Number 2605 in Lecture Notes in Artificial Intelligence. Springer, 2005.

[Hutter, 1990] Dieter Hutter. Guiding inductive proofs. 1990. In [Stickel, 1990, pp. 147–161].

[Ireland and Bundy, 1994] Andrew Ireland and Alan Bundy. *Productive Use of Failure in Inductive Proof.* 1994. DAI Research Paper No. 716, Dept. Artificial Intelligence, Univ. Edinburgh. Also in: J. Automated Reasoning (1996) **16(1-2)**, pp. 79–111, Kluwer (Springer Science+Business Media).

[Jouannaud and Kounalis, 1986] Jean-Pierre Jouannaud and Emmanuël Kounalis. Automatic proofs by induction in equational theories without constructors. 1986. In [LICS, 1986, pp. 358–366]. Also in Information and Computation **82**, pp. 1–33, Academic Press (Elsevier), 1989.

[Kaplan and Jouannaud, 1988] Stéphane Kaplan and Jean-Pierre Jouannaud, editors. *1^{st} Int. Workshop on Conditional Term Rewriting Systems (CTRS), Orsay (France), 1987*, number 308 in Lecture Notes in Computer Science, 1988.

[Kaplan and Okada, 1991] Stéphane Kaplan and Mitsuhiro Okada, editors. *2^{nd} Int. Workshop on Conditional Term Rewriting Systems (CTRS), Montreal, 1990*, number 516 in Lecture Notes in Computer Science, 1991.

[Kapur and Musser, 1986] Deepak Kapur and David R. Musser. Inductive reasoning with incomplete specifications. 1986. In [LICS, 1986, pp. 367–377].

[Kapur and Musser, 1987] Deepak Kapur and David R. Musser. Proof by consistency. *Artificial Intelligence*, 31:125–157, 1987.

[Kapur and Zhang, 1989] Deepak Kapur and Hantao Zhang. An overview of Rewrite Rule Laboratory (RRL). 1989. In [Dershowitz, 1989, pp. 559–563]. Journal version is [Kapur and Zhang, 1995].

[Kapur and Zhang, 1995] Deepak Kapur and Hantao Zhang. An overview of Rewrite Rule Laboratory (RRL). *Computers and Mathematics with Applications*, 29(2):91–114, 1995.

[Kapur, 1992] Deepak Kapur, editor. *11^{th} Int. Conf. on Automated Deduction (CADE), Saratoga Springs (NY), 1992*, number 607 in Lecture Notes in Artificial Intelligence. Springer, 1992.

[Katz, 1998] Victor J. Katz. *A History of Mathematics: An Introduction.* Addison–Wesley, Reading (MA), 1998. 2^{nd} edn..

[Kaufmann *et al.*, 2000a] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore, editors. *Computer-Aided Reasoning:* ACL2 *Case Studies.* Number 4 in Advances in Formal Methods. Kluwer (Springer Science+Business Media), 2000. With a foreword from the series editor Mike Hinchey.

[Kaufmann *et al.*, 2000b] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach.* Number 3 in Advances in Formal Methods. Kluwer (Springer Science+Business Media), 2000. With a foreword from the series editor Mike Hinchey.

[Knuth and Bendix, 1970] Donald E Knuth and Peter B. Bendix. Simple word problems in universal algebra. 1970. In [Leech, 1970, pp. 263–297].

[Kodratoff, 1988] Yves Kodratoff, editor. *Proc. 8^{th} European Conf. on Artificial Intelligence (ECAI).* Pitman Publ., London, 1988.

[Kott, 1986] Laurent Kott, editor. *13^{th} Int. Colloquium on Automata, Languages and Programming (ICALP), Rennes, France*, number 226 in Lecture Notes in Computer Science. Springer, 1986.

[Kowalski, 1988] Robert A. Kowalski. The early years of logic programming. *Comm. ACM*, 31:38–43, 1988.

[Kreisel, 1965] Georg Kreisel. Mathematical logic. 1965. In [Saaty, 1965, Vol. III, pp. 95–195].

[Küchlin, 1989] Wolfgang Küchlin. Inductive completion by ground proof transformation. 1989. In [Aït-Kaci and Nivat, 1989, Vol. 2, pp. 211–244].

[Kühler and Wirth, 1996] Ulrich Kühler and Claus-Peter Wirth. *Conditional Equational Specifications of Data Types with Partial Operations for Inductive Theorem Proving.* SEKI-Report SR–1996–11 (ISSN 1437–4447). SEKI Publications, FB Informatik, Univ. Kaiserslautern, 1996. 24 pp., http://wirth.bplaced.net/p/rta97. Short version is [Kühler and Wirth, 1997].

[Kühler and Wirth, 1997] Ulrich Kühler and Claus-Peter Wirth. Conditional equational specifications of data types with partial operations for inductive theorem proving. 1997. In [Comon, 1997, pp. 38–52]. Extended version is [Kühler and Wirth, 1996].

[Kühler, 1991] Ulrich Kühler. Ein funktionaler und struktureller Vergleich verschiedener Induktionsbeweiser. (English translation of title: "A functional and structural comparsion of several inductive theorem-proving systems" (INKA, LP (Larch Prover), NQTHM, RRL, UNICOM)). vi+143 pp., Diplomarbeit (Master's thesis), FB Informatik, Univ. Kaiserslautern, 1991.

[Kühler, 2000] Ulrich Kühler. *A Tactic-Based Inductive Theorem Prover for Data Types with Partial Operations.* Infix, Akademische Verlagsgesellschaft Aka GmbH, Sankt Augustin, Berlin, 2000. PhD thesis, Univ. Kaiserslautern, ISBN 1586031287, http://wirth.bplaced.net/p/kuehlerdiss.

[Lambert, 1764] Johann Heinrich Lambert. *Neues Organon oder Gedanken über die Erforschung und Bezeichnung des Wahren und dessen Unterscheidung von Irrthum und Schein.* Johann Wendler, Leipzig, 1764. Vol. I (Dianoiologie oder die Lehre von den Gesetzen des Denkens, Alethiologie oder Lehre von der Wahrheit) (http://books.google.de/books/about/Neues_Organon_oder_Gedanken_Uber_die_Erf.html?id=ViS3XCuJEw8C) & Vol. II (Semiotik oder Lehre von der Bezeichnung der Gedanken und Dinge, Phänomenologie oder Lehre von dem Schein) (http://books.google.de/books/about/Neues_Organon_oder_Gedanken_

%C3%BCber_die_Er.html?id=X8UAAAAAcAAj). Facsimile reprint by Georg Olms Verlag, Hildesheim, 1965, with a German introduction by Hans Werner Arndt.

[Lankford, 1980] Dallas S. Lankford. Some remarks on inductionless induction. Memo MTP-11, Math. Dept., Louisiana Tech. Univ., Ruston, LA, 1980.

[Lankford, 1981] Dallas S. Lankford. A simple explanation of inductionless induction. Memo MTP-14, Math. Dept., Louisiana Tech. Univ., Ruston, LA, 1981.

[Lassez and Plotkin, 1991] Jean-Louis Lassez and Gordon D. Plotkin, editors. *Computational Logic — Essays in Honor of J. Alan Robinson*. MIT Press, 1991.

[Leech, 1970] John Leech, editor. *Computational Word Problems in Abstract Algebra — Proc. of a Conf. held at Oxford, under the auspices of the Science Research Council, Atlas Computer Laboratory, 29th Aug. to 2nd Sept. 1967*. Pergamon Press, Oxford, 1970. With a foreword by J. Howlett.

[Leeuwen, 1990] Jan van Leeuwen, editor. *Handbook of Theoretical Computer Sci.*. MIT Press, 1990.

[LICS, 1986] *Proc. 1st Annual IEEE Symposium on Logic In Computer Sci. (LICS), Cambridge (MA), 1986*. IEEE Press, 1986. http://lii.rwth-aachen.de/lics/archive/1986.

[LICS, 1988] *Proc. 3rd Annual IEEE Symposium on Logic In Computer Sci. (LICS), Edinburgh, 1988*. IEEE Press, 1988. http://lii.rwth-aachen.de/lics/archive/1988.

[LICS, 2007] *Proc. 22nd Annual IEEE Symposium on Logic In Computer Sci. (LICS), Wrocław (i.e. Breslau, Silesia), 2007*. IEEE Press, 2007. http://lii.rwth-aachen.de/lics/archive/2007.

[Löchner, 2006] Bernd Löchner. Things to know when implementing LPO. *Int. J. Artificial Intelligence Tools*, 15:53–79, 2006.

[Lusk and Overbeek, 1988] Ewing Lusk and Ross Overbeek, editors. *9th Int. Conf. on Automated Deduction (CADE), Argonne National Laboratory (IL), 1988*, number 310 in Lecture Notes in Artificial Intelligence. Springer, 1988.

[Mahoney, 1994] Michael Sean Mahoney. *The Mathematical Career of Pierre* de *Fermat 1601–1665*. Princeton Univ. Press, 1994. 2nd rev. edn. (1st edn. 1973).

[Marchisotto and Smith, 2007] Elena Anne Marchisotto and James T. Smith. *The Legacy of Mario Pieri in Geometry and Arithmetic*. Birkhäuser (Springer), 2007.

[Margaria, 1995] Tiziana Margaria, editor. *Kolloquium Programmiersprachen und Grundlagen der Programmierung*, 1995. Tech. Report MIP–9519, Univ. Passau.

[McCarthy *et al.*, 1965] John McCarthy, Paul W. Abrahams, D. J. Edwards, T. P. Hart, and M. I. Levin. *LISP 1.5 Programmer's Manual*. MIT Press, 1965.

[McRobbie and Slaney, 1996] Michael A. McRobbie and John K. Slaney, editors. *13th Int. Conf. on Automated Deduction (CADE), New Brunswick (NJ), 1996*, number 1104 in Lecture Notes in Artificial Intelligence. Springer, 1996.

[Meltzer and Michie, 1972] Bernard Meltzer and Donald Michie, editors. *Proceedings of the 7th Annual Machine Intelligence Workshop (Machine Intelligence 7), Edinburgh, 1971*. Univ. Edinburgh Press, 1972.

[Meltzer, 1975] Bernard Meltzer. Department of A.I. – Univ. of Edinburgh. *ACM SIGART Bulletin*, 50:5, 1975.

[Moore *et al.*, 1998] J Strother Moore, Thomas Lynch, and Matt Kaufmann. A mechanically checked proof of the correctness of the kernel of the AMD5K86 floating point division algorithm. *IEEE Transactions on Computers*, 47:913–926, 1998.

[Moore, 1973] J Strother Moore. *Computational Logic: Structure Sharing and Proof of Program Properties*. PhD thesis, Dept. Artificial Intelligence, Univ. Edinburgh, 1973.

[Moore, 1975a] J Strother Moore. Introducing iteration into the Pure LISP Theorem Prover. Technical Report CSL 74–3, Xerox, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto (CA), 1975. ii+37 pp., Received Dec. 1974, rev. March 1975. Short version is [Moore, 1975b].

[Moore, 1975b] J Strother Moore. Introducing iteration into the Pure LISP Theorem Prover. *IEEE Transactions on Software Engineering*, 1:328–338, 1975. http://doi.ieeecomputersociety.org/10.1109/TSE.1975.6312857. Long version is [Moore, 1975a].

[Moore, 1979] J Strother Moore. A mechanical proof of the termination of Takeuti's function. *Information Processing Letters*, 9:176–181, 1979. Received July 13, 1979. Rev. Sept. 5, 1979. http://dx.doi.org/10.1016/0020-0190(79)90063-2.

[Moore, 1981] J Strother Moore. Text editing primitives — the TXDT package. Technical Report CSL 81–2, Xerox, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto (CA), 1981.

[Moore, 1989a] J Strother Moore. A mechanically verified language implementation. *J. Automated Reasoning*, 5:461–492, 1989.

[Moore, 1989b] J Strother Moore. System verification. *J. Automated Reasoning*, 5:409–410, 1989.

[Moskewicz *et al.*, 2001] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. CHAFF: Engineering an efficient SAT solver. 2001. In [DAC, 2001, pp. 530–535].

[Musser, 1980] David R. Musser. On proving inductive properties of abstract data types. 1980. In [Abrahams *et al.*, 1980, pp. 154–162].

[Nilsson, 1973] Nils J. Nilsson, editor. *Proc. 3rd Int. Joint Conf. on Artificial Intelligence (IJCAI), Stanford (CA)*. Stanford Research Institute, Publications Dept., Stanford (CA), 1973. `http://ijcai.org/Past%20Proceedings/IJCAI-73/CONTENT/content.htm`.

[Padawitz, 1996] Peter Padawitz. Inductive theorem proving for design specifications. *J. Symbolic Computation*, 21:41–99, 1996.

[Pascal, 1954] Blaise Pascal. *Œuvres Complètes*. Gallimard, Paris, 1954. Jacques Chevalier (ed.).

[Péter, 1951] Rósza Péter. *Rekursive Funktionen*. Akad. Kiadó, Budapest, 1951.

[Pieri, 1908] Mario Pieri. Sopra gli assiomi aritmetici. *Il Bollettino delle seduta della Accademia Gioenia di Scienze Naturali in Catania*, Series 2, 1–2:26–30, 1908. Written Dec. 1907. Received Jan. 8, 1908. English translation *On the Axioms of Arithmetic* in [Marchisotto and Smith, 2007, § 4.2, pp. 308–313].

[Presburger, 1930] Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Sprawozdanie z I Kongresu metematyków krajów słowianskich, Warszawa 1929 (Comptes-rendus du 1re Congrès des Mathématiciens des Pays Slaves, Varsovie 1929)*, pages 92–101+395, 1930. Remarks and English translation in [Stansifer, 1984].

[Protzen, 1994] Martin Protzen. Lazy generation of induction hypotheses. 1994. In [Bundy, 1994, pp. 42–56].

[Protzen, 1995] Martin Protzen. *Lazy Generation of Induction Hypotheses and Patching Faulty Conjectures*. Infix, Akademische Verlagsgesellschaft Aka GmbH, Sankt Augustin, Berlin, 1995. PhD thesis.

[Protzen, 1996] Martin Protzen. Patching faulty conjectures. 1996. In [McRobbie and Slaney, 1996, pp. 77–91].

[Rabinovitch, 1970] Nachum L. Rabinovitch. Rabbi Levi ben Gerson and the origins of mathematical induction. *Archive for History of Exact Sciences*, 6:237–248, 1970. Received Jan. 12, 1970.

[Reddy, 1977] Ray Reddy, editor. *Proc. 5th Int. Joint Conf. on Artificial Intelligence (IJCAI), Cambridge (MA)*. Dept. of Computer Sci., Carnegie Mellon Univ., Cambridge (MA), 1977. `http://ijcai.org/Past%20Proceedings`.

[Reddy, 1990] Uday S. Reddy. Term rewriting induction. 1990. [Stickel, 1990, pp. 162–177].

[Riazanov and Voronkov, 2001] Alexander Riazanov and Andrei Voronkov. Vampire 1.1 (system description). 2001. In [Gore *et al.*, 2001, pp. 376–380].

[Robinson and Voronkow, 2001] Alan Robinson and Andrei Voronkow, editors. *Handbook of Automated Reasoning*. Elsevier, 2001.

[Rubin and Rubin, 1985] Herman Rubin and Jean E. Rubin. *Equivalents of the Axiom of Choice*. North-Holland (Elsevier), 1985. 2nd rev. edn. (1st edn. 1963).

[Rusinowitch and Remy, 1993] Michaël Rusinowitch and Jean-Luc Remy, editors. *3rd Int. Workshop on Conditional Term Rewriting Systems (CTRS), Pont-à-Mousson (France), 1992*, number 656 in Lecture Notes in Computer Science, 1993.

[Russinoff, 1998] David M. Russinoff. A mechanically checked proof of IEEE compliance of a register-transfer-level specification of the AMD-K7 floating-point multiplication, division, and square root instructions. *London Mathematical Society Journal of Computation and Mathematics*, 1:148–200, 1998.

[Saaty, 1965] T. L. Saaty, editor. *Lectures on Modern Mathematics*. John Wiley & Sons, New York, 1965.

[Schmidt-Samoa, 2006a] Tobias Schmidt-Samoa. An even closer integration of linear arithmetic into inductive theorem proving. *Electronic Notes in Theoretical Computer Sci.*, 151:3–20, 2006. `http://wirth.bplaced.net/p/evencloser`, `http://dx.doi.org/10.1016/j.entcs.2005.11.020`.

[Schmidt-Samoa, 2006b] Tobias Schmidt-Samoa. *Flexible Heuristic Control for Combining Automation and User-Interaction in Inductive Theorem Proving.* PhD thesis, Univ. Kaiserslautern, 2006. `http://wirth.bplaced.net/p/samoadiss`.

[Schmidt-Samoa, 2006c] Tobias Schmidt-Samoa. Flexible heuristics for simplification with conditional lemmas by marking formulas as forbidden, mandatory, obligatory, and generous. *Journal of Applied Non-Classical Logics*, 16:209–239, 2006. `http://dx.doi.org/10.3166/jancl.16.208-239`.

[Schoenfield, 1967] Joseph R. Schoenfield. *Mathematical Logic.* Addison–Wesley, Reading (MA), 1967.

[Shankar, 1986] Natarajan Shankar. *Proof-checking Metamathematics.* PhD thesis, The University of Texas at Austin, 1986. Thoroughly revised version is [Shankar, 1994].

[Shankar, 1988] Natarajan Shankar. A mechanical proof of the Church–Rosser theorem. *J. of the ACM*, 35:475–522, 1988. Received May 1985, rev. Aug. 1987. See also Chapter 6 in [Shankar, 1994].

[Shankar, 1994] Natarajan Shankar. *Metamathematics, Machines, and Gödel's Proof.* Cambridge Univ. Press, 1994. Originally published as [Shankar, 1986]. Paperback reprint 1997.

[Siekmann, 1986] Jörg Siekmann, editor. *8th Int. Conf. on Automated Deduction (CADE), Oxford, 1986*, number 230 in Lecture Notes in Artificial Intelligence. Springer, 1986.

[Sridharan, 1989] N. S. Sridharan, editor. *Proc. 11th Int. Joint Conf. on Artificial Intelligence (IJCAI), Detroit, MI.* Morgan Kaufman (Elsevier), 1989. `http://ijcai.org/Past%20Proceedings`.

[Stansifer, 1984] Ryan Stansifer. Presburger's Article on Integer Arithmetic: Remarks and Translation. Technical Report TR 84–639, Dept. of Computer Sci., Cornell Univ., Ithaca, NY, 1984. `http://hdl.handle.net/1813/6478`.

[Steele, 1990] Guy L. Steele, Jr.. COMMON LISP — *The Language.* Digital Press (Elsevier), 1990. 2nd edn. (1st edn. 1984).

[Steinbach, 1988] Joachim Steinbach. *Term Orderings With Status.* SEKI-Report SR–88–12 (ISSN 1437–4447). SEKI Publications, FB Informatik, Univ. Kaiserslautern, 1988. 57 pp., `http://wirth.bplaced.net/SEKI/welcome.html#SR-88-12`.

[Steinbach, 1995] Joachim Steinbach. Simplification orderings — history of results. *Fundamenta Informaticae*, 24:47–87, 1995.

[Stevens, 1988] Andrew Stevens. *A Rational Reconstruction of Boyer and Moore's Technique for Constructing Induction Formulas.* 1988. DAI Research Paper No. 360, Dept. Artificial Intelligence, Univ. Edinburgh. Also in [Kodratoff, 1988, pp. 565–570].

[Stickel, 1990] Mark E. Stickel, editor. *10th Int. Conf. on Automated Deduction (CADE), Kaiserslautern (Germany), 1990*, number 449 in Lecture Notes in Artificial Intelligence. Springer, 1990.

[Stoyan, 1985] Herbert Stoyan, editor. *9th German Workshop on Artificial Intelligence (GWAI), Dassel (Germany), 1985*, number 118 in Informatik-Fachberichte. Springer, 1985.

[Toyama, 1988] Yoshihito Toyama. Commutativity of term rewriting systems. 1988. In [Fuchi and Kott, 1988, pp. 393–407]. Also in [Toyama, 1990].

[Toyama, 1990] Yoshihito Toyama. *Term Rewriting Systems and the Church–Rosser Property.* PhD thesis, Tohoku Univ. / Nippon Telegraph and Telephone Corporation, 1990.

[Unguru, 1991] Sabetai Unguru. Greek mathematics and mathematical induction. *Physis*, XXVIII(2):273–289, 1991.

[Verma, 2005?] Shamit Verma. Interview with Charles Simonyi. WWW only: `http://www.shamit.org/charles_simonyi.htm`, 2005?

[Voicu and Li, 2009] Răzvan Voicu and Mengran Li. *Descente Infinie* proofs in COQ. 2009. In [Herbelin, 2009, pp. 73–84].

[Voronkov, 1992] Andrei Voronkov, editor. *Proc. 3rd Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, number 624 in Lecture Notes in Artificial Intelligence. Springer, 1992.

[Voronkov, 2002] Andrei Voronkov, editor. *18th Int. Conf. on Automated Deduction (CADE), København, 2002*, number 2392 in Lecture Notes in Artificial Intelligence. Springer, 2002.

[Walther, 1988] Christoph Walther. Argument-bounded algorithms as a basis for automated termination proofs. 1988. In [Lusk and Overbeek, 1988, pp. 601–622].

[Walther, 1992] Christoph Walther. Computing induction axioms. 1992. In [Voronkov, 1992, pp. 381–392].

[Walther, 1993] Christoph Walther. Combining induction axioms by machine. 1993. In [Bajscy, 1993, pp. 95–101].

[Walther, 1994a] Christoph Walther. Mathematical induction. 1994. In [Gabbay *et al.*, 1994, pp. 127–228].

[Walther, 1994b] Christoph Walther. On proving termination of algorithms by machine. *Artificial Intelligence*, 71:101–157, 1994.

[Wirth and Becker, 1995] Claus-Peter Wirth and Klaus Becker. Abstract notions and inference systems for proofs by mathematical induction. 1995. In [Dershowitz and Lindenstrauss, 1995, pp. 353–373].

[Wirth and Gramlich, 1994a] Claus-Peter Wirth and Bernhard Gramlich. A constructor-based approach to positive/negative-conditional equational specifications. *J. Symbolic Computation*, 17:51–90, 1994. `http://dx.doi.org/10.1006/jsco.1994.1004`, `http://wirth.bplaced.net/p/jsc94`.

[Wirth and Gramlich, 1994b] Claus-Peter Wirth and Bernhard Gramlich. On notions of inductive validity for first-order equational clauses. 1994. In [Bundy, 1994, pp. 162–176], `www.ags.uni-sb.de/~cp/p/cade94`.

[Wirth and Kühler, 1995] Claus-Peter Wirth and Ulrich Kühler. *Inductive Theorem Proving in Theories Specified by Positive/Negative-Conditional Equations.* SEKI-Report SR–95–15 (ISSN 1437–4447). SEKI Publications, Univ. Kaiserslautern, 1995. iv+126 pp..

[Wirth *et al.*, 1993] Claus-Peter Wirth, Bernhard Gramlich, Ulrich Kühler, and Horst Prote. *Constructor-Based Inductive Validity in Positive/Negative-Conditional Equational Specifications.* SEKI-Report SR–93–05 (SFB) (ISSN 1437–4447). SEKI Publications, FB Informatik, Univ. Kaiserslautern, 1993. iv+58 pp., `http://wirth.bplaced.net/SEKI/welcome.html#SR-93-05`. Rev. extd. edn. of 1st part is [Wirth and Gramlich, 1994a], rev. edn. of 2nd part is [Wirth and Gramlich, 1994b].

[Wirth *et al.*, 2009] Claus-Peter Wirth, Jörg Siekmann, Christoph Benzmüller, and Serge Autexier. Jacques Herbrand: Life, logic, and automated deduction. 2009. In [Gabbay and Woods, 2004ff., Vol. 5: Logic from Russell to Church, pp. 195–254].

[Wirth, 1971] Niklaus Wirth. The programming language Pascal. *Acta Informatica*, 1:35–63, 1971.

[Wirth, 1991] Claus-Peter Wirth. Inductive theorem proving in theories specified by positive/negative-conditional equations. Diplomarbeit (Master's thesis), FB Informatik, Univ. Kaiserslautern, 1991.

[Wirth, 1997] Claus-Peter Wirth. *Positive/Negative-Conditional Equations: A Constructor-Based Framework for Specification and Inductive Theorem Proving*, volume 31 of *Schriftenreihe Forschungsergebnisse zur Informatik.* Verlag Dr. Kovač, Hamburg, 1997. PhD thesis, Univ. Kaiserslautern, ISBN 386064551X, `www.ags.uni-sb.de/~cp/p/diss`.

[Wirth, 2004] Claus-Peter Wirth. Descente Infinie + Deduction. *Logic J. of the IGPL*, 12:1–96, 2004. `http://wirth.bplaced.net/p/d`.

[Wirth, 2005a] Claus-Peter Wirth. History and future of implicit and inductionless induction: Beware the old jade and the zombie! 2005. In [Hutter and Stephan, 2005, pp. 192–203], `http://wirth.bplaced.net/p/zombie`.

[Wirth, 2005b] Claus-Peter Wirth. *Syntactic Confluence Criteria for Positive/Negative-Conditional Term Rewriting Systems.* SEKI-Report SR–95–09 (ISSN 1437–4447). SEKI Publications, Univ. Kaiserslautern, 2005. Rev. edn. Oct. 2005, ii+188 pp., `http://arxiv.org/abs/0902.3614`.

[Wirth, 2006] Claus-Peter Wirth. lim+, δ^+, *and Non-Permutability of* β*-Steps.* SEKI-Report SR–2005–01 (ISSN 1437–4447). SEKI Publications, Saarland Univ., 2006. Rev. edn. July 2006, ii+36 pp., `http://arxiv.org/abs/0902.3635`. Thoroughly improved version is [Wirth, 2012b].

[Wirth, 2009] Claus-Peter Wirth. Shallow confluence of conditional term rewriting systems. *J. Symbolic Computation*, 44:69–98, 2009. `http://dx.doi.org/10.1016/j.jsc.2008.05.005`.

[Wirth, 2010a] Claus-Peter Wirth. *Progress in Computer-Assisted Inductive Theorem Proving by Human-Orientedness and Descente Infinie?* SEKI-Working-Paper SWP–2006–01 (ISSN 1860–5931). SEKI Publications, Saarland Univ., 2010. Rev. edn. Dec 2010, ii+36 pp., `http://arxiv.org/abs/0902.3294`.

[Wirth, 2010b] Claus-Peter Wirth. *A Self-Contained and Easily Accessible Discussion of the Method of Descente Infinie and Fermat's Only Explicitly Known Proof by Descente Infinie.* SEKI-Working-Paper SWP–2006–02 (ISSN 1860–5931). SEKI Publications, DFKI Bremen GmbH, Safe and Secure Cognitive Systems, Cartesium, Enrique Schmidt Str. 5, D–28359 Bremen, Germany, 2010. Rev. ed. Dec. 2010, ii+36 pp., `http://arxiv.org/abs/0902.3623`.

[Wirth, 2012a] Claus-Peter Wirth. Herbrand's Fundamental Theorem in the eyes of Jean van Heijenoort. *Logica Universalis*, 6:485–520, 2012. Received Jan. 12, 2012. Published online June 22, 2012, `http://dx.doi.org/10.1007/s11787-012-0056-7`.

[Wirth, 2012b] Claus-Peter Wirth. $\lim +$, δ^+, and Non-Permutability of β-Steps. *J. Symbolic Computation*, 47:1109–1135, 2012. Received Jan. 18, 2011. Published online July 15, 2011, `http://dx.doi.org/10.1016/j.jsc.2011.12.035`. More funny version is [Wirth, 2006].

[Wirth, 2012c] Claus-Peter Wirth. Human-oriented inductive theorem proving by descente infinie — a manifesto. *Logic J. of the IGPL*, 20:1046–1063, 2012. Received July 11, 2011. Published online March 12, 2012, `http://dx.doi.org/10.1093/jigpal/jzr048`.

[Wirth, 2012d] Claus-Peter Wirth. Unpublished Interview of Robert S. Boyer and J Strother Moore at Boyer's place in Austin (TX) on Thursday, Oct. 7. 2012.

[Wirth, 2013] Claus-Peter Wirth. *A Simplified and Improved Free-Variable Framework for Hilbert's epsilon as an Operator of Indefinite Committed Choice.* SEKI Report SR–2011–01 (ISSN 1437–4447). SEKI Publications, DFKI Bremen GmbH, Safe and Secure Cognitive Systems, Cartesium, Enrique Schmidt Str. 5, D–28359 Bremen, Germany, 2013. Rev. edn. Jan. 2013, ii+65 pp., `http://arxiv.org/abs/1104.2444`.

[Wolff, 1728] Christian Wolff. *Philosophia rationalis sive Logica, methodo scientifica pertractata et ad usum scientiarium atque vitae aptata.* Rengerische Buchhandlung, Frankfurt am Main & Leipzig, 1728. 1st edn..

[Wolff, 1740] Christian Wolff. *Philosophia rationalis sive Logica, methodo scientifica pertractata et ad usum scientiarium atque vitae aptata.* Rengerische Buchhandlung, Frankfurt am Main & Leipzig, 1740. 3rd extd. edn. of [Wolff, 1728]. Facsimile reprint by Georg Olms Verlag, Hildesheim, 1983, with a French introduction by Jean École.

[Young, 1989] William D. Young. A mechanically verified code generator. *J. Automated Reasoning*, 5:493–518, 1989.

[Zhang *et al.*, 1988] Hantao Zhang, Deepak Kapur, and Mukkai S. Krishnamoorthy. A mechanizable induction principle for equational specifications. 1988. In [Lusk and Overbeek, 1988, pp. 162–181].

[Zygmunt, 1991] Jan Zygmunt. Mojżesz Presburger: Life and work. *History and Philosophy of Logic*, 12:211–223, 1991.