MECHANIZATION OF MATHEMATICAL INDUCTION

Robert S. Boyer, J S. Moore, Claus-Peter Wirth

(I would not mind to have Matt or Pete as additional authors. CP)

(I added "Mathematical" to the title because there is another article in the Handbook (Vol. 10) with the title "Mechanizing Induction: ...", dealing with inductive learning. CP)

(This is typeset using the official handbook style files, as they were in 2008, when I wrote my article on Jacques Herbrand. CP)

(This very first program sketch is, needless to say, to be understood as a suggestion, open to additions, deletions, and changes. CP)

1 PREFACE

(very short)

2 WHAT IS MATHEMATICAL INDUCTION?

In this section, we will introduce mathematical induction with its rich history since the 6th century B.C., and clarify the difference between *structural induction* and *Noetherian induction* with its traditional variant called *descente infinie*.

According to Aristotle, *induction* means to go from the special to the general, in particular to obtain *general laws* from special cases, which plays a major rôle in the generation of conjectures in mathematics and the natural sciences. Modern scientists design experiments to falsify such a law of nature, and they accept the law as a scientific fact only after many trials have all failed to falsify it. In the tradition of Euclid of Alexandria, mathematicians accept a conjectured mathematical law as a theorem only after a rigorous proof has been provided. According to Kant, induction is *synthetic* in the sense that it properly extends what we think to know — in opposition to *deduction*, which is *analytic* in the sense that all information we can obtain by deduction is implicitly contained in the initial judgments, though we can hardly be aware of all their deducible consequences before.

Surprisingly, in this well-established and time-honored terminology, *mathematical induction* is not induction, but a special form of deduction for which — in the 19th century[1] — the term "induction" was introduced and became standard in English and German mathematics. In spite of this misnomer, for the sake of brevity, the term "induction" will always refer to mathematical induction in what follows.

Although it received its current name only in 19th century, mathematical induction has been a standard method of every working mathematician at all times. Hippasus of Metapontum (Italy) (ca. 550 B.C.) is reported[2] to have proved the irrationality of the golden number by a form of mathematical induction, which later was named *descente infinie (ou indéfinie)* by Fermat. We find another form of induction, nowadays called *structural induction*, in a text of Plato (427–347 B.C.).[3] In the famous "Elements" of Euclid [ca. 300 B.C.], we find several applications of *descente infinie* and structural induction.[4] Structural induction was known to the Muslim mathematicians around the year 1000, and occurs in a Hebrew book of Levi ben Gerson (Orange and Avignon) (1288–1344).[5] Furthermore, structural induction was used by Francesco Maurolico (Messina) (1494–1575),[6] and by Blaise Pascal (1623–1662).[7] After an absence of more than one millennium, *descente infinie* was reinvented by Pierre Fermat (1607?–1665).[8]

---

[1] Cf. [Cajori, 1918].

[2] Cf. [Fritz, 1945].

[3] Cf. [Acerbi, 2000].

[4] An example for *descente infinie* is Proposition 31 of Vol. VII of the Elements, and an example for structural induction is Proposition 8 of Vol. IX, cf. [Wirth, 2010, § 2.4].

[5] Cf. [Katz, 1998].

[6] Cf. [Bussey, 1917].

[7] Cf. [Pascal, 1954, p. 103].

[8] See [Barner, 2001] for the correction on the Fermat's year of birth as compared to the wrong date in the title of [Mahoney, 1994]. The best-documented example of a proof by *descente infinie* of one of Fermat's many outstanding results in number theory is the proof of the following theorem: *The area of a Pythagorean triangle with positive integer side lengths is not the square of an integer*; cf. [Wirth, 2010].

In its modern standard meaning, the method of mathematical induction can easily be seen to be a form of deduction, simply because it can be formalized as the application of the *Theorem of Noetherian Induction* (after Emmy Noether (1882–1935)):

> A proposition $P(x)$ can be shown to hold (for all x) by *Noetherian induction* over a well-founded relation $<$ as follows: *Show (for every v) that $P(v)$ follows from the assumption that $P(u)$ holds for all $u < v$.*

A relation $<$ is *well-founded* if each proposition $Q(x)$ that is not constantly false holds for a $<$-minimal m, i.e. there is an m with $Q(m)$, for which there is no $w < m$ with $Q(w)$.

Writing "$\mathsf{Wellf}(<)$" for "$<$ is well-founded", we can formalize this definition together with the Theorem of Noetherian Induction (N) as follows:

$$(\mathsf{Wellf}(<)) \qquad \forall Q.\ \Big(\ \exists x.\ Q(x) \quad \Rightarrow \quad \exists m.\ \big(\ Q(m) \wedge \neg\exists w{<}m.\ Q(w)\ \big)\ \Big)$$

$$(\mathsf{N}) \qquad \forall P.\ \left(\ \forall x.\ P(x) \quad \Leftarrow \quad \exists {<}.\ \left(\begin{array}{ll} & \forall v.\big(P(v) \ \Leftarrow\ \forall u{<}v.\ P(u)\big) \\ \wedge & \mathsf{Wellf}(<) \end{array}\right)\right)$$

A main field of application of induction is the domain of the natural numbers 0, 1, 2, ... Let us formalize the natural numbers with the help of two constructors: the constant symbol

$$\mathsf{0} : \mathsf{nat}$$

for zero, and the the function symbol

$$\mathsf{s} : \mathsf{nat} \rightarrow \mathsf{nat}$$

for the direct successor of a natural number.

After the definition ($\mathsf{Wellf}(<)$) and the theorem (N), let us now consider some standard *axioms* for specifying the natural numbers, namely that a natural number is either zero or a direct successor of another natural number ($\mathsf{nat1}$), that zero is not a successor ($\mathsf{nat2}$), that the successor function is injective ($\mathsf{nat3}$), and the *Axiom of Structural Induction*; formally:

$$(\mathsf{nat1}) \qquad \forall x : \mathsf{nat}.\ \big(\ x = \mathsf{0} \quad \vee \quad \exists y : \mathsf{nat}.\ \big(\ x = \mathsf{s}(y)\ \big)\ \big)$$

$$(\mathsf{nat2}) \qquad \forall x : \mathsf{nat}.\ \big(\ \mathsf{s}(x) \neq \mathsf{0}\ \big)$$

$$(\mathsf{nat3}) \qquad \forall x, y : \mathsf{nat}.\ \big(\ \mathsf{s}(x) = \mathsf{s}(y) \quad \Rightarrow \quad x = y\ \big)$$

$$(\mathsf{S}) \qquad \forall P.\ \Big(\ \forall x : \mathsf{nat}.\ P(x) \quad \Leftarrow \quad P(\mathsf{0})\ \wedge\ \forall y : \mathsf{nat}.\ \big(\ P(\mathsf{s}(y)) \Leftarrow P(y)\ \big)\ \Big)$$

Note that analogous axioms can be used to specify any other data type given by constructors, such as lists of natural numbers or binary trees over such lists.

Richard Dedekind (1831–1916) proved the Axiom of Structural Induction (S) for his model of the natural numbers in [Dedekind, 1888], where he states that the proof method resulting from the application of this axiom is known under the name "vollständige Induction" ("complete induction").[9]

[9] The first occurrence of the name "vollständige Induction" with the meaning of mathematical induction seems to be on Page 46f. in [Fries, 1822].

The relation from a natural number to its direct successor can be formalized by the binary relation $\lambda x, y : \mathsf{nat}.\ (\mathsf{s}(x) = y)$. Then $\mathsf{Wellf}(\lambda x, y : \mathsf{nat}.\ (\mathsf{s}(x) = y))$ states the well-foundedness of this relation, which implies that its transitive closure — i.e. the irreflexive ordering of the natural numbers — is a well-founded ordering.[10]

Now the natural numbers can be specified up to isomorphism either[11]

- by (nat2), (nat3), and (S) — following Guiseppe Peano (1858–1932) —

or else

- by (nat1) and $\mathsf{Wellf}(\lambda x, y.\ (\mathsf{s}(x) = y))$ — following Mario Pieri (1860–1913).

In everyday mathematical practice of an advanced theoretical journal, the common inductive arguments are hardly ever carried out explicitly. Instead, the proof reads something like "by structural induction on n, q.e.d." or "by (Noetherian) induction on (x, y) over $<$, q.e.d.", expecting that the mathematically educated reader could easily expand the proof if in doubt. In contrast, difficult inductive arguments, sometimes covering several pages,[12] still require considerable ingenuity and have to be carried out. In case of a proof on natural numbers, the experienced mathematician engineers his proof roughly according to the following pattern:

> He starts with the conjecture and simplifies it by case analysis, typically based on the axiom (nat1). When he realizes that the current goal becomes similar to an instance of the conjecture, he applies the instantiated conjecture just like a lemma, but keeps in mind that he has actually applied an induction hypothesis. Finally, using the free variables of the conjecture, he constructs some ordering, whose well-foundedness follows from the axiom $\mathsf{Wellf}(\lambda x, y.\ (\mathsf{s}(x) = y))$, and in which all instances of the conjecture applied as induction hypotheses are smaller than the original conjecture.

The hard tasks of a proof by mathematical induction are thus:

(Induction-Hypotheses Task)
to find the numerous induction hypotheses,[13] and

(Induction-Ordering Task)
to construct an *induction ordering* for the proof, i.e. a well-founded ordering that satisfies the ordering constraints of all these induction hypotheses in parallel.[14]

---

[10] According to Lemma 2.1 of [Wirth, 2004, § 2.1.1], a relation is well-founded if and only if its transitive closure is a well-founded ordering.

[11] Cf. [Wirth, 2004, § 1.1.2].

[12] Such difficult inductive arguments, for example, are the proofs of Hilbert's *first ε-theorem* [Hilbert and Bernays, 1970], Gentzen's *Hauptsatz* [Gentzen, 1935], and confluence theorems such as the ones in [Gramlich and Wirth, 1996], [Wirth, 2009].

[13] As, e.g., in the proof of Gentzen's Hauptsatz on Cut-elimination.

[14] For instance, this was the hard part in the elimination of the ε-formulas in the proof of the 1st ε-theorem in [Hilbert and Bernays, 1970], and in the proof of the consistency of arithmetic by the ε-substitution method in [Ackermann, 1940].

The above induction method can be formalized as an application of the Theorem of Noetherian Induction. For non-trivial proofs, mathematicians indeed prefer the the axioms of Pieri's specification in combination with the Theorem of Noetherian Induction (N) to Peano's alternative with the Axiom of Structural Induction (S), because the instances for P and $<$ in (N) are often still easy to find when the instances for P in (S) are not.

The soundness of the above induction method is most easily seen when the argument is structured as a proof by contradiction, assuming a counterexample. For Fermat's historic reinvention of the method, it is thus just natural that he developed the method in terms of assumed counterexamples.[15] Here is Fermat's Method of *Descente Infinie* in modern language, very roughly speaking:

> A proposition $P(x)$ can be proved by *descente infinie* as follows: *Show that for each assumed counterexample of $P(x)$ there is a smaller counterexample of $P(x)$ w.r.t. a well-founded relation $<$, which does not depend on the counterexamples.*

If this method is executed successfully, we have proved $\forall x.\ P(x)$ because no counterexample can be $<$-minimal and so the well-foundedness of $<$ implies that there are no counterexamples at all.

Nowadays every logician immediately realizes that a formalization of the method of *descente infinie* is obtained from the Theorem of Noetherian Induction simply by replacing

$$P(v) \ \Leftarrow\ \forall u{<}v.\ P(u)$$

with its contrapositive

$$\neg P(v) \ \Rightarrow\ \exists u{<}v.\ \neg P(u).$$

For Fermat, however, it was still very hard to obtain a positive version of his method.[16] Moreover, a natural-language presentation via *descente infinie* is often simpler than via the *Theorem of Noetherian Induction*, because it is easier to speak of one counterexample v and to find one smaller counterexample u, than to administrate the dependences of universally quantified variables.

The following two proof-theoretical peculiarities of induction compared to first-order deduction are noteworthy:[17]

- As the theory of arithmetic is not enumerable according to [Gödel, 1931], a calculus for arithmetic cannot be complete.[18]
- According to Gentzen's Hauptsatz [Gentzen, 1935], a proof of a first-order theorem can be restricted to its "sub"-formulas. In contrast to lemma appli-

[15] Cf. [Fermat, 1891ff.], [Mahoney, 1994], [Bussotti, 2006], [Wirth, 2010].

[16] Fermat reported in his letter for Huygens that he had had problems to apply the Method of *Descente Infinie* to positive mathematical statements; see [Wirth, 2010, p. 11] and the references there, in particular [Fermat, 1891ff., Vol. II,p. 432].

[17] Note, however, that these peculiarities of induction do not make a difference to first-order deductive theorem proving *in practice*. See Notes 18 and 20.

[18] In practice, however, it does not matter whether our proof attempt fails because our theorem will not be enumerated ever or will not be enumerated before doomsday.

cation in a deductive proof tree, however, the application of induction hypotheses and lemmas inside an inductive reasoning cycle cannot generally be eliminated in the sense that the "sub"-formula property could be obtained.[19]

As a consequence, in first-order inductive theorem proving, "creativity" cannot be restricted to finding just the proper instances, but may require the invention of new lemmas and notions.[20]

In this section, we have presented a formalization and a first practical description of the induction method in its historical context. A proof method of a working mathematician, however, cannot be completely captured by the formulas he applies, and so we still have to develop effective heuristics for actually finding proofs by induction. This will be the subject of the following section.

J suggested to write on ordinal numbers in this section. CP does not know for which context, but guesses [Manolios and Vroon, 2003].

[19] Cf. [Kreisel, 1965].

[20] In practice, however, it does not matter whether we have to extend our proof search to additional lemmas and notions for principled reasons or for tractability, cf. [Baaz and Leitsch, 1995].

3 EXPLICIT INDUCTION

(very long)

- How it came to Explicit induction
- 1 section on how the base and step cases are obtained from Noetherian induction.
- 1 very long section on recursion analysis.
- 1 long section From the early 1970 to NQTHM.
- 1 section on INKA, (could be written by Dieter and Serge?)
- 1 section on ACL2.
- 1 section on the practical challenges
- 1 section on the limitations of explicit induction

4 ALTERNATIVE APPROACHES

4.1 Proof Planning

Suggestions on how to overcome an envisioned dead end in automated theorem proving were summarized at the end of the 1980s under the keyword *proof planning*. Beside its human-science aspects,[21] the main idea of proof planning[22] is to add a smaller and more human-oriented *higher-level search space* to the theorem-proving system on top of the *low level search space* of the logic calculus. We do not cover this subject here, and refer the reader to the article by Alan Bundy and Jörg Siekmann in this volume.

4.2 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 cannot cover this very well-documented subject here, but refer the reader to the monograph [Bundy *et al.*, 2005].[23] 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 as a lake with ripples (goal). 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, mirrored by the calmed surface of the lake.

4.3 Implicit Induction

Proof planning and rippling were applied to the automation of induction within the paradigm of explicit 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

[21] Cf. [Bundy, 1989].
[22] Cf. [Bundy, 1988] [Dennis *et al.*, 2005].
[23] Historically important are also the following publications on rippling: [Hutter, 1990], [Bundy *et al.*, 1991], [Basin and Walsh, 1996].

equational theories.[24] A sequence of papers on technical improvements[25] 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 end of the 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.[26]

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:[27] Systems for proof by consistency run some Knuth–Bendix[28] or superposition[29] completion procedure that 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 an inconsistency.

Proof by consistency has shown to be not competitive with explicit induction in practice, mainly due to too many superfluous inferences, typically infinite runs, and too restrictive admissibility conditions. Roughly speaking, the conceptual flaw in proof by consistency is that, instead of finding a sufficient set of reasonable inferences, the research follows the paradigm of ruling out as many irrelevant inferences as possible.

Implicit Induction Ordering: In the early implicit-induction systems, induction proceeds over a syntactical term ordering, which typically cannot be made explicit in the sense that there would be some predicate in the logical syntax that denotes this ordering in the intended models of the specification. The semantical orderings of explicit induction, however, cannot depend precisely on the 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 syntax is surprisingly high for powerful inference systems.[30]

---

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

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

[26] 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].

[27] 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].

[28] Cf. [Gramlich and Lindner, 1991].

[29] Cf. [Ganzinger and Stuber, 1992].

[30] Cf. [Wirth, 1997].

The early implicit-induction systems needed such sophisticated term orderings[31] 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,[32] after which the need for syntactical term orderings does not exist anymore in automated induction.

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 termination in implicit induction need not be confined to single inference steps.[33] The induction rule of explicit induction combines several 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 § 2, the name *descente infinie* was coined for it in [Wirth, 2004]. Researchers introduced to this aspect by [Protzen, 1994] (entitled "Lazy Generation of Induction Hypotheses") sometimes speak of "lazy induction" instead of *descente infinie*.

The interest in proof by consistency and implicit induction orderings today is either merely theoretical or merely historical, especially because these approaches cannot combined with the paradigm of explicit induction. For more information on these viewpoints on implicit induction see the handbook article [Comon, 2001] and its partial correction [Wirth, 2005].

In § 4.4 we will show, however, how *Descente infinie* ("lazy induction") goes well together with explicit induction and why we can hope that both the restrictions implied by induction axioms can be overcome and the usefulness of the excellent heuristic knowledge developed in explicit induction can be improved.[34]

4.4 QuodLibet

(along [Wirth, 2009] and [Schmidt-Samoa, 2006c])

[31] Cf. e.g. [Bachmair, 1988], [Steinbach, 1995].
[32] Cf. [Wirth and Becker, 1995].
[33] 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.
[34] Cf. [Wirth, 2012].

5 BEYOND INDUCTION

(short)

- Beyond Noetherian induction (Full axiom of choice instead of principle of dependent choices)
- 1 section on what the incredible success of NQTHM meant for the fields of ATP and AI
- Lessons we have learned for ATP useful outside induction.

6 CONCLUSION

(very short)

ACKNOWLEGDGEMENTS

We would like to thank Matt Kaufmann, Marianeh Rezaei.

We could split the bibliography into subsections, but this seems to be inappropriate here, because alternative approaches to explicit induction have few references, and because the sections on INKA, NQTHM, √ERIFUN, ACL2, λCLAM would heavily overlap)

BIBLIOGRAPHY

[Abrahams *et al.*, 1980] Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne, editors. *Conference Record of the 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, 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.

[Aubin, 1976] Raymond Aubin. *Mechanizing Structural Induction*. PhD thesis, Univ. Edinburgh, 1976.

[Aubin, 1979] Raymond Aubin. Mechanizing structural induction. *Theoretical Computer Sci.*, 9:329–345+347–362, 1979.

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

[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://www.ags.uni-sb.de/~cp/p/quodlibet`.

[Baader, 2003] Franz Baader, editor. *19th Int. Conf. on Automated Deduction, 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.

[Baaz *et al.*, 1997] Matthias Baaz, Uwe Egly, and Christian G. Fermüller. Lean induction principles for tableaus. 1997. In [Galmiche, 1997, pp. 62–75].

[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/PastProceedings`.

[Barner, 2001] Klaus Barner. Das Leben Fermats. *DMV-Mitteilungen*, 3/2001:12–26, 2001.

[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.

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

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

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

[Boyer and Moore, 1975] Robert S. Boyer and J S. Moore. Proving theorems about LISP functions. *J. of the ACM*, 22:129–144, 1975.

[Boyer and Moore, 1977] Robert S. Boyer and J S. Moore. A lemma driven automatic theorem prover for recursive function theory. 1977. In [Reddy, 1977, pp. 511–519].

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

[Boyer and Moore, 1988] Robert S. Boyer and J S. Moore. *A Computational Logic Handbook*. Academic Press (Elsevier), 1988.

[Boyer and Moore, 1989] Robert S. Boyer and J S. Moore. The addition of bounded quantification and partial functions to a computational logic and its theorem prover. 1989. In: Manfred Broy (ed.), Constructive Methods in Computing Science, NATO ASI Series, Vol. F 55, pp. 95–145, Springer.

[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, 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, 1969] R. M. Burstall. Proving properties of programs by structural induction. *The Computer Journal*, 12:48–51, 1969.

[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.

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

[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 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.

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

[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].

[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.

[Galmiche, 1997] Didier Galmiche, editor. *6th Int. Conf. on Tableaus and Related Methods, Pont-à-Mousson (France), 1997*, number 1227 in Lecture Notes in Artificial Intelligence. Springer, 1997.

[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, 1999] Harald Ganzinger, editor. *16th Int. Conf. on Automated Deduction, 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].

[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 B. 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].

[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. In *7th Int. Conf. on Rewriting Techniques and Applications (RTA), New Brunswick (NJ), 1996*, number 1103 in Lecture Notes in Computer Science, pages 245–259. Springer, 1996.

[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).

[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. 1st edn. (2nd 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. 2nd 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. 2nd 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.* College Publications, London, 2011. Commented, first English translation of the 2nd edn. [Hilbert and Bernays, 1968] by Claus-Peter Wirthet.al., incl. the German facsimile and the annotation and translation of all deleted texts of 1st German edn. [Hilbert and Bernays, 1934]. 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.

[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.

[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 60th 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].

[Hutter, 1994] Dieter Hutter. Synthesis of induction orderings for existence proofs. 1994. In [Bundy, 1994, pp. 29–41].

[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 Okada, 1991] Stéphane Kaplan and Mitsuhiro Okada, editors. *2nd 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.

[Katz, 1998] Victor J. Katz. *A History of Mathematics: An Introduction.* Addison-Wesley, 1998. 2nd edn..

[Kaufmann *et al.*, 2000] Matt Kaufmann, Panagiotis Manolios, and J S. Moore. *Computer-Aided Reasoning: An Approach.* Kluwer (Springer Science+Business Media), 2000.

[Kaufmann *et al.*, 2010] Matt Kaufmann, Panagiotis Manolios, and J S. Moore, editors. *Computer-Aided Reasoning:* ACL2 *Case Studies.* Kluwer (Springer Science+Business Media), 2010.

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

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

[Kraan *et al.*, 1995] Ina Kraan, David Basin, and Alan Bundy. *Middle-Out Reasoning for Synthesis and Induction.* 1995. DAI Research Paper No. 729, Dept. Artificial Intelligence, Univ. Edinburgh. Also in J. Automated Reasoning (1996) **16(1–2)**, pp. 113–145, Kluwer (Springer Science+Business Media).

[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, 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://www.ags.uni-sb.de/~cp/p/kuehlerdiss`.

[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.

[LICS, 1986] *Proc. 1st Annual IEEE Symposium on Logic In Computer Sci. (LICS), Cambridge (MA), 1986*. IEEE Press, 1986. `http://www2.informatik.hu-berlin.de/lics/archive/1986`.

[LICS, 1988] *Proc. 3rd Annual IEEE Symposium on Logic In Computer Sci. (LICS), Edinburgh, 1988*. IEEE Press, 1988. `http://www2.informatik.hu-berlin.de/lics/archive/1988`.

[Lusk and Overbeek, 1988] Ewing Lusk and Ross Overbeek, editors. *9th Int. Conf. on Automated Deduction, 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).

[Manolios and Vroon, 2003] Panagiotis Manolios and Daron Vroon. Algorithms for ordinal arithmetic. 2003. In [Baader, 2003, pp. 243–257].

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

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

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

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

[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.).

[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].

[Reddy, 1977] Ray Reddy, editor. *Proc. 5th Int. Joint Conf. on Artificial Intelligence (IJCAI), Cambridge, MA*. William Kaufman, 1977. `http://ijcai.org/PastProceedings`.

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

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

[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.

[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://www.ags.uni-sb.de/~cp/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://www.ags.uni-sb.de/~cp/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.

[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. In [Kodratoff, 1988, pp. 565–570].

[Stickel, 1990] Mark E. Stickel, editor. *10th Int. Conf. on Automated Deduction, 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.

[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.

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

[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, 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://www.ags.uni-sb.de/~cp/p/d.

[Wirth, 2005] 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://www.ags.uni-sb.de/~cp/p/zombie.

[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, 2010] 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. 2nd edn. (1st edn. 2006). http://arxiv.org/abs/0902.3623.

[Wirth, 2012] Claus-Peter Wirth. Human-oriented inductive theorem proving by descente infinie — a manifesto. *Logic J. of the IGPL*, 20, 2012. To appear. http://dx.doi.org/10.1093/jigpal/jzr048.

[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].