History and Future of Implicit and Inductionless Induction: Beware the old jade and the zombie!

Claus-Peter Wirth

Dept. of Computer Sci., Saarland Univ., D–66123 Saarbrücken, Germany
wirth@logic.at

Abstract. In this survey on implicit induction I recollect some memories on the history of implicit induction as it is relevant for future research on computer-assisted theorem proving, especially memories that significantly differ from the presentation in a recent handbook article on "inductionless induction". Moreover, the important references excluded there are provided here. To clear the fog a little, there is a short introduction to inductive theorem proving and a discussion of connotations of implicit induction like "*descente infinie*", "inductionless induction", "proof by consistency", implicit induction orderings (term orderings), and refutational completeness.

1 What is Inductive Theorem Proving (*ITP*)?

Inductive reasoning can be seen as extending deductive reasoning in that infinite deductive proofs may be represented in a finite cyclic form, as suggested in the following example, where $\Gamma(x_0, y)$ is a proposition over the natural numbers, where 's' denotes the successor function ($x \mapsto x{+}1$), and where the formulas below each line (sub-goals) imply the formula (goal) above: An infinite deductive proof of $\Gamma(x_0, y)$

$$
\begin{array}{lllll}
\Gamma(x_0,y) & & & & \\
\hline
\Gamma(0,y) & \Gamma(\mathsf{s}(x_1),y) & & & \\
\cline{2-3}
\vdots & \Gamma(\mathsf{s}(0),y) & \Gamma(\mathsf{s}(\mathsf{s}(x_2)),y) & & \\
\cline{1-1} \cline{3-4}
 & \vdots & \Gamma(\mathsf{s}(\mathsf{s}(0)),y) & \ddots & \\
\cline{2-3}
 & & \vdots & & \ddots \\
\cline{3-3}
 & & & & \quad\ddots
\end{array}
$$

(using $x_i = 0 \vee \exists x_{i+1}.\ x_i = \mathsf{s}(x_{i+1})$) should be captured in something like

$$
\cfrac{\Gamma(x_0,y)}{\cfrac{\Gamma(0,y)}{\cfrac{\vdots}{}} \qquad \cfrac{\Gamma(\mathsf{s}(x_1),y)}{\cfrac{\Delta \Rightarrow \Gamma(\mathsf{s}(x_1),y)}{\cfrac{\vdots}{\cfrac{\Delta' \vee \Gamma(x_1,s)}{\text{(back to top)}}}} \qquad \cfrac{\Pi \Rightarrow \Gamma(\mathsf{s}(x_1),y)}{\cfrac{\vdots}{\cfrac{\Pi' \vee \Gamma(x_1,t)}{\text{(back to top)}} \quad \cfrac{\Pi''}{\cfrac{\vdots}{}}}}}}
$$

using $\Delta \vee \Pi$. This kind of *cyclic* argument—namely inferring $\Gamma(x_1,s)$ and $\Gamma(x_1,t)$ from $\Gamma(x_0,y)$ —is sound if for each (ground) instantiation of the theorem (here: $\Gamma(\mathsf{s}^m(\mathsf{0}),\mathsf{s}^n(\mathsf{0}))$) the deductive proof terminates. This can be guaranteed by requiring that each cycle in the proof (graph) *terminate*, i.e. its preconditions (usually called *induction hypotheses* — here: $\Gamma(x_1,s)$ and $\Gamma(x_1,t)$) be smaller than the *"induction" conclusion* (here: $\Gamma(x_0,y)$ or $\Gamma(\mathsf{s}(x_1),y)$) w.r.t. some wellfounded ordering, called *induction ordering* (here e.g. the usual ordering on the natural numbers applied to the first argument of Γ). Thus, while the property of being an inductive theorem depends only on the specification (i.e. a language and a set of axioms) and the choice of a specific notion of inductive validity, cf. [74], an inductive proof of an inductive theorem also depends on an additional parameter, namely some induction ordering which must be chosen appropriately during the proof.

2 Explicit versus Implicit Induction

Although there is no generally accepted characterization of the two paradigms of *explicit* and *implicit* induction in the research community, in [67], which is a comprehensive survey on explicit induction, the following is said:

> Research on automated induction these days is based on two competing paradigms: *Implicit induction* (also termed *inductive completion*, *inductionless induction*, or, less confusingly, *proof by consistency*) evolved from the *Knuth–Bendix Completion Procedure* .... The other research paradigm ... is called *explicit induction* and resembles the more familiar idea of induction theorem proving using induction axioms.

In accordance with this view, one reason to call the latter paradigm "*explicit*" is that in the underlying inference systems every cyclic argument must be made explicit in a single inference step applying a so-called *induction rule*. Besides generating *induction base* formulas, this step joins induction hypotheses and conclusions in *induction step* formulas and explicitly guarantees the termination of their cycles by a sub-proof or -mechanism for the wellfoundedness of the induction ordering resulting from the step formulas. In the explicit induction proof corresponding to the example induction proof of § 1 the induction base formula is $\Gamma(0,y)$ and the induction step formulas are $\Delta \Rightarrow (\ \Gamma(x_1,s) \Rightarrow \Gamma(\mathsf{s}(x_1),y)\ )$ and $\Pi \Rightarrow (\ \Gamma(x_1,t) \Rightarrow \Gamma(\mathsf{s}(x_1),y)\ )$. The explicit induction proof then has the following form:

$$
\begin{array}{ccc}
\multicolumn{3}{c}{\Gamma(x_0, y)} \\ \hline
\Gamma(0, y) & \Delta \Rightarrow (\ \Gamma(x_1, s) \Rightarrow \Gamma(\mathsf{s}(x_1), y)\ ) & \Pi \Rightarrow (\ \Gamma(x_1, t) \Rightarrow \Gamma(\mathsf{s}(x_1), y)\ ) \\ \hline
\vdots & \vdots & \vdots \\ \hline
\end{array}
$$

Note that the first inference in this proof is an application of an *induction axiom* in the sense of [67].

As the example induction proof of § 1 illustrates, the cyclic arguments and their termination in *implicit* induction proofs need not be confined to single inference steps, as is in explicit induction. Therefore, the induction axioms corresponding to the cyclic arguments in a finite implicit induction proof can only be determined by analyzing the whole proof, whereas in the case of explicit induction each applied induction axiom is given by a single application of the induction rule.

The phrase "more familiar idea" attributed to explicit induction in the above quotation requires some remarks. Implicit induction in the style of *descente infinie* was already known to the ancient Greeks and—as will be explained below—is the standard method of mathematical induction since Pierre Fermat (1607?–1665) rediscovered the method and named it *descente infinie* (*ou indéfinie*). Nevertheless more familiar is the idea of explicit induction to computer scientists who—inspired by J. Alan Robinson's resolution method of the year 1963—wanted to solve all problems of logical inference via reduction to machine-oriented inference systems. Instead of implementing mathematical ITP, they decided to reduce it via the induction rule (cf. above) in the following fashion: Apply the induction rule backwards; then do purely deductive reasoning; if this fails, repeat the process! The so called "waterfall" of [15] refines this process into a fascinating heuristic and the ITP system NQTHM [15, 16] shows that in this case the reduction approach was quite successful—so successful actually that it is hard to understand why implicit induction was able to gain ground again, as we will see in the next section.

3 1988—Start of Practical Interest in Implicit Induction

Cf. e.g. [62, 19] for historical surveys on implicit induction, which, however, have to be taken with a grain of salt. After several papers on implicit induction in purely equational theories already in the year 1980, [32, 36, 47, 51], there was a sequence of papers on technical improvements, [37, 23, 30, 44], which was topped by Leo Bachmair's paper [6] in the year 1988. Up to 1987 the papers on implicit induction had a theoretical nature, but some of them were very inspiring, [38] being my favorite.

Bachmair's paper [6] is the first one that gave good reason for a hope to develop the method into practical usefulness. And—in 1990—there was real hope, cf. [34, p. 1]:

> This approach has some very attractive and promising characteristics compared to classical approaches for explicit inductive theorem proving based on induction schemas, which will be pointed out later on.

As a consequence, in the end of the 1980s and the first half of the 1990s several researchers tried to clearly understand what "*implicit induction*" means from a theoretical point of view and whether it would be useful for practical ITP.

4 1996—End of Most Interest in Implicit Induction

During the Induction Workshop on the 13$^{\text{th}}$ Int. Conf. on Automated Deduction, New Brunswick (NJ), 1996, there was an agreement on better not to use the term "implicit induction" in the future, for the following two reasons, which will be discussed in the next two subsections.

1. Different researchers understand this term differently.
2. The notion has lost its potential relevance for the practice of ITP.

4.1 What is *Implicit* Induction?

While it is generally accepted that [6] is implicit and [15] is explicit, we report the following different views on what it is that makes induction implicit:

Descente Infinie. In explicit induction there is something like an "induction rule" (cf. § 2) whose addition turns a deductive inference system into an inductive inference system without further changes on the deductive part. Concerning the concept of induction hypothesis, the explicit induction "hides" several (applications of) induction hypotheses in a single inference step. To the contrary, in *descente infinie*, 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.

Note that *descente infinie* is important for human-oriented ITP because this is the style in which working mathematicians do induction since Pierre Fermat (1607?–1665) rediscovered and named the method which was already known to the ancient Greeks. The working mathematician applies it in the following fashion.

> He starts with the conjecture and simplifies it by case analysis. 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, he searches for some well-founded ordering in which all the instances of the conjecture he has applied as induction hypotheses are smaller than the original conjecture.

This view on the notion of "implicit induction" (i.e. *descente infinie*) was the one of the majority on the Induction Workshop in 1996.

The name *descente infinie* for this aspect of implicit induction was coined later in [71] after suggestions and complaints on the occurrence of "implicit induction" in the title of [70]. Researchers introduced to *descente infinie* by [57] (entitled "Lazy Generation of Induction Hypotheses") sometimes speak of "*lazy induction*" instead of *descente infinie*.

Inductionless Induction / Proof by Consistency.
"Inductionless Induction" means that no induction can be observed explicitly. E.g. some Knuth–Bendix [35] or superposition-calculus [25] completion procedure produces a huge number of irrelevant inferences under which the ones relevant for induction can hardly be made explicit in an automatic way. "Inductionless induction" has shown to be practically useless, mainly due to too many superfluous inferences, typically infinite runs, and too restrictive admissibility conditions. The approach in [25] is an interesting theoretical possibility but definitely useless for practical purposes. Roughly speaking, the conceptual flaw in "inductionless induction" seems to be that, instead of finding a sufficient set of reasonable inferences, the research follows the paradigm of ruling out as many irrelevant inferences as possible. A proof attempt is successful when the prover has drawn all necessary inferences and stops without having detected an inconsistency (empty clause).

Christoph Walther (cf. [67]) prefers to use the name "*proof by consistency*" instead of the *contradictio in eo ipso* "inductionless induction", which, however, is used again in the year 2001 in the title of [19]. Indeed, "proof by consistency" seems to be a better name for this aspect of implicit induction than "inductionless induction" because the former highlights on the fact that—roughly speaking—the derivation of the empty clause (or inconsistency) means disproof, not proof. E.g., in the case of the toy example proofs in [20] the number of irrelevant inferences is cut down to zero (unless they are hidden in the omitted parts of the proofs) and it is quite obvious where the induction takes place, so that "proof by consistency" is still appropriate, but "inductionless induction" is not.

While the opinion that the "inductionless induction" aspect (i.e. no induction explicitly observable) is the crucial one for "implicit induction" is now shared by several researchers who had worked on implicit induction in the past, at the Induction Workshop in 1996, only a minority held this opinion. Note that this view on the notion of implicit induction is held by Martin Protzen and places his important work [57] (which is implicit induction in the sense of *descente infinie*) on the side of explicit induction, where the practically useful ITP systems use to dwell, such as the powerful up-to-date systems INKA [3] and ACL2 [41].

The name "proof by consistency" was coined in [39], which is the forerunner of the earlier published improved version [38]. The name "inductionless induction" was coined already before [48] (to which it is erroneously attributed in

[19, 62]), namely in [47] in the year 1980, when also the title of [32] included a similar phrase.

Implicit Induction Ordering. This means that there is no explicit induction ordering in the signature and the model. Instead of such semantical orderings, the induction is performed on syntactical term orderings that are not part of the logic language. The semantical orderings (cf. Definition 13.7 of [69]) cannot depend on the syntactical term structure of a weight w but only on the value of w under the evaluation function. In [69] we have rigorously investigated the price one has to pay for the possibility to have induction orderings also depending on the syntax of weights. For powerful concrete inference systems this price turned out to be surprisingly high. Besides this, after improving the ordering information in *descente infinie* by our introduction of explicit weights (cf. [72]) the former necessity of sophisticated induction orderings that exploit the term structure (cf. e.g. [6, 64]) does not seem to exist anymore.

This view on the notion of "implicit induction" (i.e. no explicit induction ordering in the language) is held by a minority, e.g. by Peter Padawitz. Note that his view places his ITP system EXPANDER, [54, 55], on the explicit side, where the practically useful ITP systems use to dwell. Note that EXPANDER is also explicit in the sense that it does not perform "inductionless induction". EXPANDER is, however, implicit in the sense that it realizes *descente infinie*, and—to my knowledge—it is the only integration of *descente infinie* into a framework of refutational resolution (or—more precisely—inverse method, [49]) where—roughly speaking—the empty clause (or inconsistency) means proof, not disproof.

Moreover, this view on the notion of "implicit induction" (i.e. implicit induction ordering) is also the view found in [19], which, however, concentrates on "inductionless induction".

It does not seem to be completely superfluous to note that "*refutational completeness*" in itself cannot be crucial for implicitness of induction, simply because it is not a very important property:

- For practical purposes, it is not important that the invalidity of a theorem would be detected sometime (= refutational completeness), but that it is detected efficiently, cf. [56].
- In theoretical terms, refutational completeness used to be trivial for the very restricted logics in the scope of implicit induction.

Nevertheless, since Leo Bachmair's important paper [6] various authors in the field of implicit induction emphasized refutational completeness as if it were a major asset. So let us have a final look at it:

In general, for all significant notions of inductive validity, the set of inductively valid theorems is not enumerable, cf. [31, 52, 50]. Therefore, refutational completeness is an optimal theoretical quality of inference systems for ITP. In practice, however, refutational completeness by itself does not help in refuting invalid conjectures or in finding finite proofs for inductively valid formulas. Only theoreticians completely detached from reality can consider nonterminating proof attempts in refutationally complete inference systems to be successful proofs.

4.2 End of the Schism

In 1996, the theoretical aspects of "inductionless induction" were clearly understood by the experts. Moreover, the severe limitations of "inductionless induction" were admitted by all researchers in the field, at least by those who attended the Induction Workshop. There was the general opinion that "inductionless induction" was *dead*.

The practitioners in automated theorem proving were also not too interested in the remaining aspects of implicit induction, i.e. "*descente infinie*" and "implicit induction orderings". My main interest, however, was and still is to convince the community of the practical importance of "*descente infinie*", cf. [71].

All participants agreed that the general challenge would be the development of practically more useful ITP systems where the separation of implicit vs. explicit would not play a splitting rôle because (contrary to "inductionless induction") "*descente infinie*" goes together well with the standard ideas of explicit induction. The general opinion was—roughly speaking—the following.

> To succeed in proving an inductive theorem in finite time, implicit inductive theorem provers have to solve the same problems as explicit inductive theorem provers, namely to find a finite cyclic representation for an infinite deductive proof as well as an induction ordering guaranteeing the termination of its cycles.

Thus, the hatchet between the tribe of explicit ITP and the (rather small) tribe of implicit ITP was buried at the Induction Workshop in 1996, at least between the attendees.

5 The Aftermath

Since the year 1996 there was only a small number of publications on implicit induction. The following seems to be a complete list of the research papers: "Inductionless induction" is only found in [20]. Besides this, *descente infinie* is treated in [10, 61, 69, 70, 45, 71, 65, 2]. Implicit induction orderings are found only in [10, 69, 20, 65, 2].

Recently, in the Handbook of Automated Reasoning [60], Hubert Comon published an article [19] on "inductionless induction" that gave a very biased account on the history of "implicit induction". It seems to me that Hubert Comon neglects the more practice-oriented research on implicit induction because this would invalidate his resurrection of "inductionless induction" in [20] as of merely theoretical interest. I think this to be very problematic because this was already misleading in the past (cf. [62], where [20] is mistaken to be "the cutting edge", [63]) and probably will further mislead newcomers who take the handbook [60] for a complete high-quality reference for entering a research area. Therefore, I would like to give the advice:

> *Do not to fumble around with the zombie of "inductionless induction"!*
> The experts in implicit induction have spent a lot of time with it, mostly to bury it.

Less problematic than the biased handbook article [19] is the technical report [20], although it does not represent a significant practical progress as compared to [25]. It is more of theoretical interest. Moreover, sentences like "The method of proofs by consistency has lost part of its popularity since it was developed in the early 80s." are misleading because the method was never popular, not even among researchers in ITP, not in the 1980s, not even in the the early 1990s.

6 Conclusion

While there was reason to hope to develop "inductionless induction" into practical applicability in 1988, it is a theoretician's game for more than half a decade now. Also "implicit induction orderings" do not seem to be useful in practice.

The remaining aspect of implicit induction, however, "*descente infinie*", is going to play an important rôle in the practice of ITP when mathematics assistance systems are applied to proof problems that are beyond the scope of the recursion analysis and the eager hypothesis generation of explicit ITP.

7 A Word to the Wise

In communications on previous versions of this paper I was surprised that the confusion on the meaning of implicitness of induction is just as total today as it was in 1996; only the confidence in the justifiedness of the respective personal views seems to have increased with the time gone by.

A sinner myself, cf. [70], regarding future research, I therefore propose to finally bury the phrases "implicit induction" and "inductionless induction".

The notions of "*descente infinie*", "proof by consistency", and "implicit induction ordering" are the relays that can take us further than the old jade "implicit induction" and the zombie of "inductionless induction".

Étoit il lors temps de moi taire?
— FRANÇOIS VILLON; POÉSIES DIVERSES; XVI

Caveat

This article depends on my personal possibly erroneous (but careful and honest) judgments and memories gathered over a dozen years of work on this very complex subject.

References

The following list contains publications on implicit induction with a special emphasis on those that should have been but are not cited in [19]. Additionally, it contains some outstanding or important papers on other kinds and aspects of induction and theorem proving.

[1] H. Ait-Kaci, M. Nivat (eds.) (1989). *Resolution of Equations in Algebraic Structures.* Academic Press (Elsevier).

[2] Alessandro Armando, Michaël Rusinowitch, Sorin Stratulat (2002). *Incorporating Decision Procedures in Implicit Induction.* J. Symbolic Computation **34**, pp. 241–258, Academic Press (Elsevier).

[3] Serge Autexier, Dieter Hutter, Heiko Mantel, Axel Schairer (1999). INKA *5.0 — A Logical Voyager.* 16th CADE 1999, LNAI 1632, pp. 207–211, Springer.

[4] Jürgen Avenhaus, Klaus Madlener (1995). *Theorem Proving in Hierarchical Clausal Specifications.* SEKI-Report SR–95–14 (SFB), FB Informatik, Univ. Kaiserslautern. `http://www-madlener.informatik.uni-kl.de/seki/1995/Avenhaus.SR-95-14.ps.gz` (March 07, 2002).

[5] Matthias Baaz, Uwe Egly, Christian G. Fermüller (1997). *Lean Induction Principles for Tableaus.* 6th TABLEAUX 1997, LNAI 1227, pp. 62–75, Springer.

[6] Leo Bachmair (1988). *Proof By Consistency in Equational Theories.* 3rd IEEE Symposium on Logic In Computer Sci., pp. 228–233, IEEE Press.

[7] Leo Bachmair (1991). *Proof By Consistency.* Birkhauser, Boston.

[8] Klaus Becker (1993). *Proving Ground Confluence and Inductive Validity in Constructor-Based Equational Specifications.* TAPSOFT 1993, LNCS 668, pp. 46–60, Springer.

[9] Klaus Becker (1994). *Rewrite Operationalization of Clausal Specifications with Predefined Structures.* PhD thesis, FB Informatik, Univ. Kaiserslautern.

[10] Klaus Becker (1996). *How to Prove Ground Confluence.* SEKI-Report SR–96–02, FB Informatik, Univ. Kaiserslautern.

[11] Rudolf Berghammer (1993). *On the Characterization of the Integers: The Hidden Function Problem Revisited.* Acta Cybernetica **11**, pp. 85–96, Szeged.

[12] Eddy Bevers, Johan Lewi (1990). *Proof by Consistency in Conditional Equational Theories.* Report CW 102, rev. July 1990. Dept. Comp. Sci., K. U. Leuven. Short version in: 2nd CTRS 1990, LNCS 516, pp. 194–205, Springer.

[13] Adel Bouhoula, Michaël Rusinowitch (1995). *Implicit Induction in Conditional Theories.* J. Automated Reasoning **14**, pp. 189–235, Kluwer (Springer).

[14] Adel Bouhoula, Emmanuël Kounalis, Michaël Rusinowitch (1992). *Automated Mathematical Induction.* Technical Report 1636, INRIA.

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

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

[17] Robert S. Boyer, J S. Moore (1989). *The Addition of Bounded Quantification and Partial Functions to A Computational Logic and Its Theorem Prover.* In: Manfred Broy (ed.), Constructive Methods in Computing Science, NATO ASI Series, Vol. F 55, pp. 95–145, Springer.

[18] Alan Bundy (1988). *The Use of Explicit Proof Plans to Guide Inductive Proofs.* 9th CADE 1988, LNAI 310, pp. 111–120, Springer.

[19] Hubert Comon (2001). *Inductionless induction.* In: [60], Vol. I, pp. 913–970.

[20] Hubert Comon, Robert Nieuwenhuis (1998). *Induction = I-Axiomatization + First-order Consistency.* Technical Report, ENS Cachan.

[21] Ulrich Fraus (1993). *A Calculus for Conditional Inductive Theorem Proving.* 3rd CTRS 1992, LNCS 656, pp. 357–362, Springer.

[22] Ulrich Fraus (1994). *Mechanizing Inductive Theorem Proving in Conditional Theories.* PhD thesis, Univ. Passau.

[23] Laurent Fribourg (1986). *A Strong Restriction of the Inductive Completion Procedure.* 13th ICALP 1986, LNCS 226, pp. 105–116, Springer. Also in: J. Symbolic Computation **8**, pp. 253–276, Academic Press (Elsevier), 1989.

[24] Dov Gabbay, C. J. Hogger, J. Alan Robinson (eds.) (1993ff.). *Handbook of Logic in Artificial Intelligence and Logic Programming.* Clarendon Press.

[25] Harald Ganzinger, Jürgen Stuber (1992). *Inductive Theorem Proving by Consistency for First-Order Clauses.* In: Informatik-Festschrift zum 60. Geburtstag von Günter Hotz. pp. 441–462. Teubner Verlag, Stuttgart. Also in: 3rd CTRS 1992, LNCS 656, pp. 226–241, Springer, 1993.

[26] Gerhard Gentzen (1938). *Die gegenwärtige Lage in der mathematischen Grundlagenforschung – Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie.* Forschungen zur Logik und zur Grundlegung der exakten Wissenschaften, Folge 4, Leipzig.

[27] Gerhard Gentzen (1943). *Beweisbarkeit und Unbeweisbarkeit von Anfangsfällen der transfiniten Induktion in der reinen Zahlentheorie.* Mathematische Annalen **119**, pp. 140–161.

[28] Alfons Geser (1995). *A Principle of Non-Wellfounded Induction.* In: Tiziana Margaria (ed.). Kolloquium Programmiersprachen und Grundlagen der Programmierung, MIP–9519, pp. 117–124, Univ. Passau.

[29] Martin Giese (1998). *Integriertes automatisches und interaktives Beweisen: die Kalkülebene.* Master's thesis, Univ. Karlsruhe. `http://i11www.ira.uka.de/~giese/da.ps.gz` (May 09, 2000).

[30] Richard Göbel (1985). *Completion of Globally Finite Term Rewriting Systems for Inductive Proofs.* 9th German Workshop on AI, IFB 118, Springer.

[31] Kurt Gödel (1931). *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I.* Monatshefte für Mathematik und Physik **38**, pp. 173–198.

[32] Joseph Goguen (1980). *How to Prove Algebraic Inductive Hypotheses Without Induction.* 5th CADE 1980, LNCS 87, pp. 356–373, Springer.

[33] Bernhard Gramlich (1989). *Inductive Theorem Proving Using Refined Unfailing Completion Techniques.* SEKI-Report SR–89–14 (SFB), FB Informatik, Univ. Kaiserslautern. Short version in: *9th European Conf. on Artificial Intelligence (ECAI 1990)*, pp. 314–319, Pitman Publ., London.

[34] Bernhard Gramlich (1990). *Completion Based Inductive Theorem Proving: A Case Study in Verifying Sorting Algorithms.* SEKI-Report SR–90–04, FB Informatik, Univ. Kaiserslautern.

[35] Bernhard Gramlich, Wolfgang Lindner (1991). *A Guide to* UNICOM*, an Inductive Theorem Prover Based on Rewriting and Completion Techniques.* SEKI-Report SR–91–17 (SFB), FB Informatik, Univ. Kaiserslautern. `http://agent.informatik.uni-kl.de/seki/1991/Lindner.SR-91-17.ps.gz` (May 09, 2000).

[36] Gérard Huet, Jean-Marie Hullot (1980). *Proofs by Induction in Equational Theories with Constructors.* 21st FOCS 1980, pp. 96–107. Also in: J. Computer and System Sci. **25**, pp. 239–266, Academic Press (Elsevier), 1982.

[37] Jean-Pierre Jouannaud, Emmanuël Kounalis (1986). *Automatic Proofs by Induction in Equational Theories Without Constructors.* 1st IEEE Symposium on Logic In Computer Sci., pp. 358–366, IEEE Press. Also in: Information and Computation **82**, pp. 1–33, Academic Press (Elsevier), 1989.

[38] Deepak Kapur, David R. Musser (1986). *Inductive Reasoning with Incomplete Specifications.* 1st IEEE Symposium on Logic In Computer Sci., pp. 367–377, IEEE Press.

[39] Deepak Kapur, David R. Musser (1987). *Proof by Consistency.* Artificial Intelligence **31**, pp. 125–157.

[40] Deepak Kapur, Hantao Zhang (1989). *An Overview of Rewrite Rule Laboratory (*RRL*).* 3rd RTA 1989, LNCS 355, pp. 559–563, Springer.

[41] Matt Kaufmann, Panagiotis Manolios, J S. Moore (2000). *Computer-Aided Reasoning: An Approach.* Kluwer (Springer).

[42] Emmanuël Kounalis, Michaël Rusinowitch (1990). *Mechanizing Inductive Reasoning.* 8th AAAI 1990, pp. 240–245, MIT Press.

[43] Georg Kreisel (1965). *Mathematical Logic.* In: T. L. Saaty (ed.). Lectures on Modern Mathematics, Vol. III, pp. 95–195, John Wiley & Sons, New York.

[44] Wolfgang Küchlin (1987). *Inductive Completion by Ground Proof Transformation.* Colloquium on Resolution of Equations in Algebraic Structures (CREAS). Also in: [1], Vol. 2, pp. 211–244.

[45] Ulrich Kühler (2000). *A Tactic-Based Inductive Theorem Prover for Data Types with Partial Operations.* PhD thesis, Infix, Akademische Verlagsgesellschaft Aka GmbH, Sankt Augustin, Berlin.

[46] Ulrich Kühler, Claus-Peter Wirth (1996). *Conditional Equational Specifications of Data Types with Partial Operations for Inductive Theorem Proving.* SEKI-Report SR–96–11, FB Informatik, Univ. Kaiserslautern. Short version in: 8th RTA 1997, LNCS 1232, pp. 38–52, Springer. `http://www.ags.uni-sb.de/~cp/p/rta97 (Aug. 05, 2001).`

[47] Dallas S. Lankford (1980). *Some Remarks on Inductionless Induction.* Memo MTP-11, Math. Dept., Louisiana Tech. Univ., Ruston (LA).

[48] Dallas S. Lankford (1981). *A Simple Explanation of Inductionless Induction.* Memo MTP-14, Math. Dept., Louisiana Tech. Univ., Ruston (LA).

[49] Vladimir A. Lifschitz (1989). *What Is the Inverse Method?.* J. Automated Reasoning **5**, pp. 1–23, Kluwer (Springer).

[50] David B. MacQueen, Donald T. Sannella (1985). *Completeness of Proof Systems for Equational Specifications.* IEEE Transactions on Software Engineering **11**, pp. 454–461, IEEE Press.

[51] David R. Musser (1980). *On Proving Inductive Properties of Abstract Data Types.* 7th POPL 1980, pp. 154–162, ACM Press.

[52] Cyrus F. Nourani (1994). *Types, Induction, and Incompleteness.* Bull. EATCS **53**, pp. 226–247.

[53] Peter Padawitz (1990). *Horn Logic and Rewriting for Functional and Logic Program Design.* MIP–9002, Univ. Passau.

[54] Peter Padawitz (1996). *Inductive Theorem Proving for Design Specifications.* J. Symbolic Computation **21**, pp. 41–99, Academic Press (Elsevier).

[55] Peter Padawitz (1998). EXPANDER. *A System for Testing and Verifying Functional Logic Programs.* `http://fldit-www.cs.uni-dortmund.de/~peter/ExpaTex.ps.gz` (March 24, 2006).

[56] Martin Protzen (1992). *Disproving Conjectures.* 11th CADE 1992, LNAI 607, pp. 340–354, Springer.

[57] Martin Protzen (1994). *Lazy Generation of Induction Hypotheses.* 12th CADE 1994, LNAI 814, pp. 42–56, Springer. Long version in: [58].

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

[59] Uday S. Reddy (1990). *Term Rewriting Induction.* 10th CADE 1990, LNAI 449, pp. 162–177, Springer.

[60] J. Alan Robinson, Andrei Voronkov (eds.) (2001). *Handbook of Automated Reasoning.* Elsevier.

[61] Christof Sprenger (1996). *Über die Beweissteuerung des induktiven Theorembeweisers* QUODLIBET *mit Taktiken.* Master's thesis, FB Informatik, Univ. Kaiserslautern.

[62] Graham Steel (1999). *Inductionless Induction (aka Implicit Induction or Proof by Consistency): A Literature Survey.* `http://www.dai.ed.ac.uk/homes/grahams/papers/lit-survey.ps.gz` (April 28, 2002).

[63] Graham Steel, Alan Bundy, Ewen Denney (2002). *Using Implicit Induction to Guide a Parallel Search for Inconsistency.* `http://www.dai.ed.ac.uk/homes/grahams/papers/abstract.pdf` (April 28, 2002).

[64] Joachim Steinbach (1995). *Simplification Orderings — History of Results.* Fundamenta Informaticae **24**, pp. 47–87.

[65] Sorin Stratulat (2001). *A General Framework to Build Contextual Cover Set Induction Provers.* J. Symbolic Computation **32**, pp. 403–445, Academic Press (Elsevier).

[66] Christoph Walther (1992). *Computing Induction Axioms.* 3rd LPAR 1992, LNAI 624, pp. 381–392, Springer.

[67] Christoph Walther (1994). *Mathematical Induction.* In: [24], Vol. 2, pp. 127–228.

[68] Claus-Peter Wirth (1991). *Inductive Theorem Proving in Theories Specified by Positive/Negative-Conditional Equations.* Master's thesis, FB Informatik, Univ. Kaiserslautern. Abstract in: 1st Workshop on Construction of Computational Logics, Val d'Ajol (France), 1992, Rapport Interne CRIN 93–R–023, p. 38, Villers-les-Nancy, 1993.

[69] Claus-Peter Wirth (1997). *Positive/Negative-Conditional Equations: A Constructor-Based Framework for Specification and Inductive Theorem Proving.* PhD thesis, Verlag Dr. Kovač, Hamburg.

[70] Claus-Peter Wirth (1999). *Full First-Order Free Variable Sequents and Tableaus in Implicit Induction.* 8th TABLEAUX 1999, LNAI 1617, pp. 293–307, Springer. `http://www.ags.uni-sb.de/~cp/p/d/tab99 (Aug. 05, 2001).`

[71] Claus-Peter Wirth (2000). *Descente Infinie + Deduction.* Report 737/2000, FB Informatik, Univ. Dortmund. Rev. version also in: Logic J. of the IGPL (2004) **12**, pp. 1–96, Oxford Univ. Press. `http://www.ags.uni-sb.de/~cp/p/d (Sept. 12, 2003).`

[72] Claus-Peter Wirth, Klaus Becker (1995). *Abstract Notions and Inference Systems for Proofs by Mathematical Induction.* 4th CTRS 1994, LNCS 968, pp. 353–373, Springer. `http://www.ags.uni-sb.de/~cp/p/ctrs94 (Aug. 05, 2001).`

[73] Claus-Peter Wirth, Bernhard Gramlich (1994). *A Constructor-Based Approach for Positive/Negative-Conditional Equational Specifications.* J. Symbolic Computation **17**, pp. 51–90, Academic Press (Elsevier). `http://www.ags.uni-sb.de/~cp/p/jsc94 (Aug. 05, 2001).`

[74] Claus-Peter Wirth, Bernhard Gramlich (1994). *On Notions of Inductive Validity for First-Order Equational Clauses.* 12th CADE 1994, LNAI 814, pp. 162–176, Springer. `http://www.ags.uni-sb.de/~cp/p/cade94 (Aug. 05, 2001).`

[75] Claus-Peter Wirth, Ulrich Kühler (1995). *Inductive Theorem Proving in Theories Specified by Positive/Negative-Conditional Equations.* SEKI-Report SR–95–15 (SFB), FB Informatik, Univ. Kaiserslautern. `http://www.ags.uni-sb.de/~cp/p/sr9515 (Aug. 05, 2001).`

[76] Hantao Zhang, Deepak Kapur, Mukkai S. Krishnamoorthy (1988). *A Mechanizable Induction Principle for Equational Specifications.* 9th CADE 1988, LNAI 310, pp. 162–181, Springer.