

Term Rewriting Induction^{*}

Uday S. Reddy[†]

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
Net: reddy@a.cs.uiuc.edu

Abstract

An induction method called *term rewriting induction* is proposed for proving properties of term rewriting systems. It is shown that the Knuth-Bendix completion-based inductive proof procedures construct term rewriting induction proofs. It has been widely held heretofore that these procedures construct proofs by consistency, and cannot be justified as induction methods. Our formulation shows otherwise. Technically, our result goes beyond the earlier ones in that it is independent of the confluence or ground confluence of the rewrite systems involved. This addresses one of the major criticisms of the method raised in recent times.

1 Introduction

We present a new induction method for proving properties of term rewriting systems called *term rewriting induction*. Term rewriting systems arise in programming as generalizations of (first-order as well as typed higher-order) functional programs. They also arise in implementations of algebraic specifications. Using the term rewriting induction method, we are able to validate the widely studied inductive proof procedures based on Knuth-Bendix completion. The first such procedure was proposed by Musser [26] and various improvements have since been made [1, 5, 10, 12, 15, 19, 20, 22, 24, 25]. Our formulation shows that these procedures indeed construct inductive proofs in terms of term rewriting induction. This is in contrast to the traditional justification of these procedures in terms of “proof by consistency”. Moreover, our results generalize the conventional ones in that these procedures are shown to be valid even if the rewriting systems involved are *not* confluent or ground confluent.

The essential idea of term rewriting induction can be described in rather simple terms: If we have a uniformly terminating rewrite system (i.e., without any infinite rewrite sequences), then the corresponding rewrite relation is a well-founded order over terms and so can be used as the basis for induction. Noetherian induction specialized to the term rewriting order is what we call *term rewriting induction*. This form of induction was used by Huet [14] and others for proving confluence properties of rewrite systems, and we adapt it to proving inductive properties of the

^{*}This paper is to appear in the proceedings of Conference on Automated Deduction, Kaiserslautern, 1990.

[†]This work was supported in part by a NSF grant: NSF-CCR-87-00988.

structures described by rewrite systems. (There are several technicalities in doing induction on terms rather than values. We deal with these in section 2). Stated in this form, term rewriting induction is not all that new in computer science. It is closely related to Hoare induction [13] for proving partial correctness properties, as well as to fixed point induction [30].

Based on these ideas, we show that an inductive proof procedure based on Knuth-Bendix completion constructs term rewriting induction proofs. (This procedure is equivalent to those proposed by Fribourg [10] and Küchlin [24], and subsumes earlier procedures [5, 15, 19, 22, 26]). This formulation may be contrasted with Musser's formulation where it is shown that the procedure is complete in detecting inconsistencies. From the fact that a strongly complete proof system entails a formula iff it is consistent with the axioms, it follows that a formula that does not generate an inconsistency is an inductive theorem. See [20] for the theory of this argument.

We were prompted to this study by some recent criticism of the completion-based induction procedures [11]. Such procedures are based on *consistency* arguments, the critique maintained, and so hard to understand. They are also severely hampered by confluence restrictions. And so, conventional methods, such as structural induction, are superior. In contrast, our experience with completion-based methods shows the opposite conclusions. The completion-based procedures of our Focus program transformation system were indeed justified by *induction* arguments [29, 28] and so were many other results on completion [3, 14]. In fact, the induction method implicit in these arguments is crystallized as term rewriting induction in the present paper. This shift of focus from proof procedures to the induction methods has produced rewarding results. It showed us, in particular, the role confluence plays (or does not play) in the inductive proof procedures. Confluence is not necessary to *prove* valid properties using the completion methods, but it is necessary to *refute* invalid properties. Given that our interest is mainly in proving properties rather than refuting them, the confluence requirement is quite unnecessary.

Term rewriting induction is Noetherian induction specialized to a specific well-founded order (albeit a very general one). So, it cannot be complete (even in the relative sense). There would always be properties provable by some other induction method, but not provable by term rewriting induction. This does not necessarily decrease the value of the method. As we show, the advantage of term rewriting induction is the ease with which it can accommodate hierarchical inductive proofs. We can combine it with other methods whenever necessary. But, in our opinion, it is still an extremely convenient method both for mechanical and human use.

The remainder of the paper is organized as follows. Section 2 presents the basic notions of inductive proofs and the generalization of Noetherian induction to terms. Section 3 discusses term rewriting induction, and section 4 presents an automatic proof procedure based on the method. Section 5 relates term rewriting induction to the proof procedures based on Knuth-Bendix completion. In section 6, we consider the question of completeness. We formulate a notion of relative completeness and show that term rewriting induction and completion-based methods are not relatively complete in this sense.

2 Inductive Proofs

Definition 1 (inductive consequence) *Let H be a sorted (typed) language of ground terms*

$$\begin{aligned}
0 &: \text{nat}, s : \text{nat} \rightarrow \text{nat}, + : \text{nat} \times \text{nat} \rightarrow \text{nat} \\
N1 &: x + 0 = x \\
N2 &: x + s(y) = s(x + y)
\end{aligned}$$

Figure 1: A set of axioms for natural numbers

generated by a sorted alphabet of function symbols F . Let A be a set of axioms. A proposition P is an inductive consequence of A (in H), written $A \vdash_{\text{ind}} P$ iff every well-sorted ground instance of P (with instantiations from H) is a consequence of A .

The standard method of proving inductive consequences is the following:

Proposition 2 (Noetherian induction principle) *Let \succ be a well-founded order on H , A be a set of axioms, and $P[x]$ a proposition. $A \vdash_{\text{ind}} P[x]$ if $A \vdash (\forall y \prec x. P[y]) \Rightarrow P[x]$. (Note: x and y range over H). \square*

Now, let T be the set of ground as well as nonground terms, and \succ be a stable well-founded order on T .¹ We can formulate the following variation of the Noetherian induction principle:

Proposition 3 (Noetherian induction for terms) *Let \succ be a stable well-founded order on T , and $P[x^s]$ a proposition with a variable x of sort s . $A \vdash_{\text{ind}} P[x^s]$ if*

1. $\{t_i\}_i$ is a set of terms of sort s such that every ground term of sort s is an instance of some t_i , and
2. for each case $P[t_i]$, $A \vdash (\forall u \prec t_i. P[u]) \Rightarrow P[t_i]$.

(Note: t_i and u range over T , and may have variables).

As an example of how nonground terms can be directly used in inductive proofs, consider the signature and axioms of Figure 1 and the proposition $P[z] : 0 + z = z$. Assume a stable well-founded order whereby $s(t) \succ t$ for all $t \in T$. Since $s(y) \succ y$, we can assume $P[y]$ in proving $P[s(y)]$.

The first condition of Proposition 3 is stronger than necessary, and we weaken it using the notion of *cover sets*:

Definition 4 (cover sets) *Let A be a set of axioms and \succ a stable well-founded order. A set of terms $\{t_i\}_i$ of sort s is a \succ -cover set for s (with respect to A) if, for every ground term g of sort s , there is a t_i and substitution σ such that $g =_A t_i \sigma$ and $g \succeq t_i \sigma$.*

¹A stable order is an order preserved by substitution.

We can easily generalize this notion for multiple variables of possibly different sorts, and talk about a *cover set of substitutions* $\{\sigma_i\}_i$. Similarly, if e is a term or a proposition, a set of instances $\{e\sigma_i\}_i$ is a cover set of cases for e if $\{\sigma_i\}_i$ is a cover set of substitutions.

Bachmair’s notion of cover sets [1] and the complete test set notion of Kapur *et. al.* are obtained by restricting \succ to a rewrite relation \rightarrow_R^+ . The generalization incorporated in the above definition is significant. (See the discussion at the end of 5.1). Zhang *et. al.* [32] use a completely different notion of cover sets which incorporates additional information via so-called “cover functions”. It seems that the additional information gives their method more power. However, it is not clear how to automate the detection of cover functions in nontrivial cases. The relation between term rewriting induction and their cover set induction remains to be investigated.

Example 5 We use here *recursive path orders (rpo)* [7] to formulate well-founded orders.

Let \succ_1 be the rpo generated by the precedence order $+ > s > 0$. Then $N_1 = \{0, s(x)\}$ is a \succ_1 -cover set for nat .

Let \succ_2 be the rpo generated by the precedence order $s > + > 0$. Then $N_2 = \{0, s(0), x + y\}$ is a \succ_2 -cover set for nat .

Add an axiom $s(s(0)) = 0$ to the axioms of Figure 1. Let \succ_3 be the singleton $\{0 \succ_3 s(s(0))\}$. Then $N_3 = \{s(x)\}$ is a \succ_3 -cover set for nat . (Note that \succ_3 is not an rpo, or even a reduction order). \square

Proposition 6 *Let \succ be a stable well-founded order on T . $A \vdash_{ind} P[x^s]$ if*

1. $\{t_i\}_i$ is a \succ -cover set of cases for s , and
2. for each case $P[t_i]$, we have $A \vdash (\forall u \prec t_i. P[u]) \Rightarrow P[t_i]$.

The above generalization of induction to terms using the notion of stable well-founded orders, allows us to consider arbitrary subterms of propositions instead of just variables. Even more interestingly, we can consider several subterms of the proposition. This, of course, requires us to compare multisets of subterms and the natural candidate for such comparison is the *multiset order* [8]. If \succ is a well-founded order on T , the multiset order \succcurlyeq on multisets of T is defined by $X \cup \{y\} \succcurlyeq X \cup \{z_1, \dots, z_k\}$ whenever $y \succ z_i$ for all $i = 1, \dots, k$. If \succ is stable and well-founded, \succcurlyeq is stable and well-founded. Now, we have:

Corollary 7 *Let \succ be a stable well-founded order on T , and $P[e_1, \dots, e_k]$ a proposition with subterms e_1, \dots, e_k . $A \vdash_{ind} P[e_1, \dots, e_k]$ if*

1. $\{\sigma_i\}_i$ is a \succ -cover set of substitutions, and
2. for each case $P[e_1, \dots, e_k]\sigma_i$, we have

$$A \vdash (\forall \theta : \{e_1\theta, \dots, e_k\theta\} \prec \{e_1\sigma_i, \dots, e_k\sigma_i\}. P[e_1, \dots, e_k]\theta) \Rightarrow P[e_1, \dots, e_k]\sigma_i$$

We take Corollary 7 to be our operational principle of induction. A proof method based on it involves two tasks: (1) finding the cover set of cases $e\sigma_i$, and (2) choosing a well-founded order. We address the second issue first.

3 Term rewriting induction

In this section, we specialize Corollary 7 for rewrite relations and show that it yields automatable proof methods for proving inductive consequences of equational axioms.

If E is a set of equational axioms its one-step replacement relation (stable, symmetric and monotonic extension) is denoted \leftrightarrow_E . An equation $t = u$ is considered symmetric, and so is equivalent to $u = t$.

Suppose the equations of E can be oriented to form a *uniformly terminating* rewrite system, i.e., there is no infinite rewrite sequence in R . We denote the one-step rewrite relation (stable and monotonic extension) of R by \rightarrow_R , its inverse by \leftarrow_R , and its symmetric closure by \leftrightarrow_R . As usual, the transitive and reflexive-transitive closures of \rightarrow_R are denoted by \rightarrow_R^+ and \rightarrow_R^* respectively.

First, note that the rewrite relation \rightarrow_R^+ is a stable well-founded order on T . This fact now allows us to “factor out” termination aspects from inductive proofs. Once we have proved a rewrite system to be terminating, we can reuse this termination proof in inductive proofs without having to choose a suitable well-founded order again and again. This “factoring out” aspect is at the heart of the term rewriting induction method, and so the assumption of a terminating rewrite system is central to it. (This does not, however, rule out the use of unorientable axioms. See Remark 14).

Even though the rewrite relation \rightarrow_R^+ is suitable for induction, we find that it is not always suitable for specifying cover sets. So, we introduce the following orders:

Definition 8 (reduction order) *A well-founded order $>$ over a set of terms is called a reduction order if it is stable ($u > v \Rightarrow u\theta > v\theta$) and monotonic ($u > v \Rightarrow t[u] > t[v]$).*

Note that \rightarrow_R^+ is a reduction order.

Definition 9 (decreasing order) *The decreasing order generated by a reduction order $>_X$, denoted \succ_X , is $(>_X \cup \triangleright)^+$ where \triangleright is the strict super term order. The decreasing order generated by \rightarrow_R^+ is denoted \succ_R .*

It is not hard to see that a decreasing order \succ_X is equivalent to $>_X \cup \triangleright \cup (>_X \triangleright)$. So, a decreasing order is well-founded.

Definition 10 (compatible orders) *Two well-founded orders \succ_1 and \succ_2 are compatible if $(\succ_1 \cup \succ_2)^+$ is well-founded, or, equivalently, if they are both included in a well-founded order.*

The principle of term rewriting induction is nothing but Corollary 7 specialized by the use of \succ_R for induction.

Corollary 11 (Term rewriting induction) *Let R be a uniformly terminating rewrite system, and $e = e'$ an equational proposition. Let \succ be a stable well-founded order that is compatible with \succ_R . Then, $R \vdash_{ind} e = e'$ if*

1. $\{e\sigma_i\}_i$ is a \succ -cover set of cases for e , and
2. for each case $e\sigma_i = e'\sigma_i$, $R \vdash (\forall \theta : \{e\theta, e'\theta\} \ll_R \{e\sigma_i, e'\sigma_i\}. e\theta = e'\theta) \Rightarrow e\sigma_i = e'\sigma_i$.

Example 12 Consider the axioms of Figure 1, but now treat the equations as rewrite rules oriented left to right. Let $P[z] : 0 + z = z$ be the proposition to be proved. As noted before, $\{[z \rightarrow 0], [z \rightarrow s(y)]\}$ is a \rightarrow_R^+ -cover set of substitutions.

1. For $0 + 0 = 0$, we have $0 + 0 \rightarrow_{N1} 0$.
2. For $0 + s(y) = s(y)$, we have $0 + s(y) \rightarrow_{N2} s(0 + y) \leftrightarrow_P s(y)$.

Note that P is applied as inductive hypothesis for a smaller pair of terms: $\{0 + y, y\} \ll_R \{0 + s(y), s(y)\}$ (since $0 + s(Y) \rightarrow_{N2} s(0 + y) \triangleright 0 + y$, and $s(y) \triangleright y$). \square

The argument applied in this example can be generalized. All that is needed is to ensure that for each case of the proposition there exists a so-called “valley proof”, i.e. a proof of the form $\rightarrow^* \leftarrow^*$, and it is then guaranteed that the inductive hypothesis is only applied to smaller terms. This observation leads to Proposition 13.

We say that an equational proposition $e = e'$ is *orientable* if $R \cup \{e \rightarrow e'\}$ is uniformly terminating.

Proposition 13 (Term rewriting induction for orientable equations) *Let R be a rewrite system and $P : e \rightarrow e'$ an oriented equation such that $R \cup P$ is uniformly terminating. Let \succ be a stable well-founded order compatible with $\succ_{R \cup P}$. Then, $R \vdash_{ind} P$ if*

1. $\{e\sigma_i\}_i$ is a \succ -cover set of cases of e , and
2. for each case $e\sigma_i \rightarrow e'\sigma_i$, we have a and b such that $e\sigma_i \rightarrow_R a \rightarrow_{R \cup P}^* b \leftarrow_{R \cup P}^* e'\sigma_i$.

Proof: Choose $\succ_{R \cup P}$ as the well-founded order. Suppose $c \rightarrow_P c'$ is an application of the inductive hypothesis in the segment $a \rightarrow_{R \cup P}^* b$. Since $e\sigma_i \succ a \succeq c \succ c'$, we have $\{e\sigma_i, e'\sigma_i\} \succ \{c, c'\}$. So, every application of the hypothesis in this segment is for a smaller pair of terms.

Suppose $c' \leftarrow_P c$ is a rewrite step in the segment $b \leftarrow_{R \cup P}^* e'\sigma_i$. Since $e\sigma_i \succ e'\sigma_i \succeq c \succeq c'$, we have $\{e\sigma_i, e'\sigma_i\} \succ \{c, c'\}$.

Hence, by Corollary 11, P is an inductive consequence of R . \square

The pragmatic advantage of this method is that we do not need to explicitly *check* that the inductive hypothesis is applied to smaller terms. The very form of the proof $e\sigma_i \rightarrow_R a \rightarrow_{R \cup P}^* b \leftarrow_{R \cup P}^* e'\sigma_i$ guarantees that it is. The hypothesis P can be used in such a proof just like any other rewrite rule, as if it were a *universal closure* over all its variables. We exploit this in section 4 for constructing hierarchical induction proofs. However, note that the method also restricts the well-founded order to be used in inductive proofs to the decreasing order \succ_R generated by the rewrite system. It is possible that there exists an inductive proof for a proposition using some well-founded order, but not using \succ_R . In such cases, the method fails to find a proof. (Cf. section 6). There is also no guarantee that rewrite proofs of the above form exist for $e\sigma_i = e'\sigma_i$. Proving lemmas (section 4) can help in such a situation. Having a confluent R also helps.

Remark 14 For unorientable equational propositions (i.e., $R \cup P$ not terminating), this result is not directly applicable. But, note that the only purpose of termination restriction is to determine the instances permitted to be rewritten by the inductive hypothesis. We can thus replace $\rightarrow_{R \cup P}$ in the proof scheme of Proposition 13 by the so called rewriting modulo congruence relation: $\rightarrow_{R/P} = \leftrightarrow_P^* \rightarrow_R \leftrightarrow_P^*$. See [2, 18] for a discussion of this notion. The same solution can be used to handle unorientable axioms. If the axioms can be partitioned as $A \uplus R$ where A contains unorientable axioms such that R/A is uniformly terminating, $\succ_{R/A}$ can be used as the well-founded order. This addresses one of the criticisms of [11].

4 Hierarchical inductive proofs

In this section, we present an automatable proof procedure based on Proposition 13. However, a direct application of the proposition is not adequate for most problems in practice. We can often “reduce” an inductive proposition P to another proposition $P1$. $P1$ is then a “lemma” and needs its own inductive proof. Moreover, $P1$ may not be an *independent* inductive theorem, but may need to use appropriately smaller instances of P in its inductive proof. We call such proofs *hierarchical inductive proofs*.

Example 15 Consider the commutativity proposition $P[u, v] : u + v = v + u$. Using the recursive path order generated by $+ > s > 0$, its proof proceeds as follows:

1. $P[u, 0]$ reduces to $P1[u] : u = 0 + u$. (oriented as $0 + u \rightarrow u$).
 - (a) $P1[0]$ reduces to $0 = 0$.
 - (b) $P1[s(y)]$ reduces to $s(y) = s(0 + y)$. By $P1[y]$, it reduces to identity.
2. $P[u, s(z)]$ reduces to $P2[u, z] : s(u + z) = s(z) + u$. (oriented as $s(z) + u \rightarrow s(u + z)$).
 - (a) $P2[0, z]$ reduces to $s(0 + z) = s(z)$, and then to identity by $P1[z]$.
 - (b) $P2[s(y), z]$ reduces to $s(s(y) + z) = s(s(z) + y)$. Using $P2[z, y]$ and $P2[y, z]$ on the two sides respectively, it reduces to $s(s(z + y)) = s(s(y + z))$. Now we use $P[y, z]$ to reduce it to identity.

Note that $P1$ turns out to be an independent lemma. It has a proof independent of P . $P2$, on the other hand, is a subsidiary lemma. Its proof uses smaller instances of P , in addition to smaller instances of $P2$ itself. The use of inductive hypotheses is valid since a reduction has been made by the order \rightarrow_R^+ .

Constructing such hierarchical proofs using conventional induction is quite tricky because of the multiple inductive hypotheses that need to be maintained. Note that what is proved in step 2 is the proposition $(\forall u. \forall v < s(z). P[u, v]) \Rightarrow P2[u, z]$. An additional hypothesis is added in proving this by induction. \square

The proof procedure for constructing hierarchical inductive proofs is called *inductive completion*, owing to its similarity with Knuth-Bendix completion. It is parameterized by a reduction order $>$. The axioms are assumed to be expressed as a set of rewrite rules R included in $>$. The procedure then incrementally constructs and modifies two sets of equations:

1. E , containing the set of equations to be proved, and
2. H , containing the equations (oriented as rewrite rules) which have been reduced to other equations in E and so can be used as inductive hypotheses. \rightarrow_H is always included in $>$.

We express the procedure in terms of inference rules. Operationally, the rules are used *backwards* in a goal-reduction fashion. The inference system \mathbf{I} contains the following inference rules:

$$\begin{array}{l}
\textit{Expand} \quad \frac{(E \cup E', H \cup \{e \rightarrow e'\})}{(E \cup \{e = e'\}, H)} \quad \begin{array}{l} \text{if } e > e', \\ \{\sigma_i\}_i \text{ is a } >\text{-cover set of substitutions for } e, \text{ and} \\ E' = \{b = e'\sigma_i \mid e\sigma_i \rightarrow_R b\}_i \end{array} \\
\textit{Simplify} \quad \frac{(E \cup \{a' = b\}, H)}{(E \cup \{a = b\}, H)} \quad \text{if } a \rightarrow_{R \cup H} a' \\
\textit{Delete} \quad \frac{(E, H)}{(E \cup \{a = a\}, H)}
\end{array}$$

An \mathbf{I} -*derivation* is a sequence of states $(E_0, H_0) \dashv_{\mathbf{I}} (E_1, H_1) \dashv_{\mathbf{I}} \dots \dashv_{\mathbf{I}} (E_n, H_n)$ where $H_0 = \emptyset$, $E_n = \emptyset$. As an example, consider an initial state $(\{0 + u = u\}, \emptyset)$. By *Expand* using the cover set $\{[u \mapsto 0], [u \mapsto s(y)]\}$, we reduce it to

$$(\{0 = 0, s(0 + y) = s(y)\}, \{0 + u \rightarrow u\})$$

The second equation simplifies to identity using H , and both the equations are then deleted to obtain the final state $(\emptyset, \{0 + u \rightarrow u\})$.

This inductive proof procedure is similar in spirit to that of Fribourg [10] and K uchlin [24]. However, its correctness proof below is more general in that it does not assume either a confluence or a ground confluence property of R . This is significant because confluence is often hard to achieve, and ground confluence is known to be undecidable [21].

Lemma 16 *The proof relation $\leftrightarrow_{R \cup H \cup E}^*$ is invariant in an \mathbf{I} -derivation.*

Lemma 17 *The following assertion is invariant in an \mathbf{I} -derivation: For every pair of ground terms g and h such that $g \rightarrow_H h$, there is a proof $g \leftrightarrow_{R \cup E}^* h$ such that every \leftrightarrow_E step in the latter proof is for a multiset of terms smaller than $\{g\}$ by the order \gg .*

Proof: Since $H_0 = \emptyset$, the assertion is vacuously true for the initial state. Suppose it holds for (E_i, H_i) . The next state (E_{i+1}, H_{i+1}) is generated by using one of the inference rules backwards. Consider each rule:

Expand Let $g \rightarrow_{H_{i+1}} h$ be an instance of $e \rightarrow e'$. Since $\{\sigma_i\}_i$ is a $>$ -cover set of substitutions for e , there is a σ_i and substitution θ such that $g \leftrightarrow_R^* e\sigma_i\theta$, $g \geq e\sigma_i\theta$, $h \leftrightarrow_R^* e'\sigma_i\theta$ and $h \geq e'\sigma_i\theta$. So, we have a proof

$$g \leftrightarrow_R^* e\sigma_i\theta \rightarrow_R b\theta \leftrightarrow_{E_{i+1}} e'\sigma_i\theta \leftrightarrow_R^* h$$

Note that $g \geq e\sigma_i\theta > b\theta$ and $g > h \geq e'\sigma_i\theta$. So, clearly $\{g\} \gg \{b\theta, e'\sigma_i\theta\}$.

If $g \rightarrow_{H_{i+1}} h$ is not an instance of $e \rightarrow e'$ then $g \rightarrow_{H_i} h$. It has a proof using $R \cup E_i = R \cup E \cup \{e = e'\}$ with smaller \leftrightarrow_{E_i} steps. By the above argument, it also has a proof using $R \cup E_{i+1}$ with smaller $\leftrightarrow_{E_{i+1}}$ steps.

Simplify First, we show, by induction on \gg , that if $t \leftrightarrow_{E_i} u$ is a ground proof step, then there is a proof $t \leftrightarrow_{R \cup E_{i+1}}^* u$ such that every $\leftrightarrow_{E_{i+1}}$ step in it is smaller than or equal to $\{t, u\}$. If $t \leftrightarrow_{E_i} u$ is not an instance of $a = b$ then $t \leftrightarrow_{E_{i+1}} u$. If it is an instance of $a = b$, let θ be the substitution such that $t = a\theta$ and $u = b\theta$. By the inference rule, for each ground proof $a\theta \leftrightarrow_{E_i} b\theta$, there is a proof

$$a\theta \rightarrow_{R \cup H_i} a'\theta \leftrightarrow_{E_{i+1}} b\theta$$

If the first step is $a\theta \rightarrow_R a'\theta$ then the only $\leftrightarrow_{E_{i+1}}$ step is for $\{a'\theta, b\theta\} \ll \{a\theta, b\theta\}$. If the first step is $a\theta \rightarrow_{H_i} a'\theta$, then there is a proof $a\theta \rightarrow_{R \cup E_i} a'\theta$ with each \leftrightarrow_{E_i} step smaller than $\{a\theta\}$. By inductive hypothesis, every such \leftrightarrow_{E_i} step has a proof using $R \cup E_{i+1}$ with smaller $\leftrightarrow_{E_{i+1}}$ steps. The conclusion is thus proved.

Since $H_{i+1} = H_i$, $g \rightarrow_{H_{i+1}} h$ implies $g \rightarrow_{H_i} h$. The latter implies that there is a proof $g \leftrightarrow_{R \cup E_i}^* h$ using smaller \leftrightarrow_{E_i} steps. By the above argument, there is a proof $g \leftrightarrow_{R \cup E_{i+1}}^* h$ using smaller $\leftrightarrow_{E_{i+1}}$ steps.

Delete Trivial.

Hence, the assertion holds in (E_{i+1}, H_{i+1}) . \square

Proposition 18 (Correctness of inductive completion) *If there is an I-derivation starting from (E, \emptyset) and ending in (\emptyset, H) for some H , then all equations in E are inductive consequences of R .*

Proof: If $t \leftrightarrow_E u$ is a ground proof then, by lemma 16, $t \leftrightarrow_{R \cup H}^* u$, and, by lemma 17, $t \leftrightarrow_R^* u$. \square

4.1 Postulating lemmas

It is well-known that we need to postulate and prove subsidiary lemmas in proving inductive theorems. In the context of the inductive completion procedure, this is rather easy. We use the following (noneffective) inference rule:

$$\text{Postulate} \quad \frac{(E \cup E', H)}{(E, H)}$$

This allows us to add any set of equations E' to the equations to be proved. Moreover, the inductive completion procedure allows E and E' to be proved in a “mutually recursive” fashion, using each set as inductive hypotheses in the other’s proof. Example 21 illustrates such an intricately interrelated proof.

Let \mathbf{I}^+ be the inference system \mathbf{I} together with the rule *Postulate*.

Lemma 19 *The containment of proof relations $\leftrightarrow_{R \cup H_i \cup E_i}^* \subseteq \leftrightarrow_{R \cup H_{i+1} \cup E_{i+1}}$ holds in an \mathbf{I}^+ derivation.*

Proposition 20 (Correctness of \mathbf{I}^+) *If there is an \mathbf{I}^+ derivation starting from (E, \emptyset) and ending in (\emptyset, H) for some H , then all the equations in E are inductive consequences of R .*

<i>F1</i>	$f(0)$	\rightarrow	0
<i>F2</i>	$f(S0)$	\rightarrow	$S0$
<i>F3</i>	$f(SSn)$	\rightarrow	$f(Sn) + f(n)$
<i>G1</i>	$g(0)$	\rightarrow	$\langle S0, 0 \rangle$
<i>G2</i>	$g(Sn)$	\rightarrow	$np(g(n))$
<i>G3</i>	$np(\langle x, y \rangle)$	\rightarrow	$\langle x + y, x \rangle$
<i>G4</i>	$sum(\langle x, y \rangle)$	\rightarrow	$x + y$

Figure 2: Rewrite rules for Fibonacci numbers

Proof: If $t \leftrightarrow_E u$ is a ground proof, by lemma 19, $t \leftrightarrow_{R \cup H}^* u$. The invariant of lemma 17 holds in an \mathbf{I}^+ derivation as well. So, $t \leftrightarrow_R^* u$. \square

Example 21 Figure 2 shows a rewrite system defining the Fibonacci number function f , a related function g and two auxiliary functions sum and np . We would like to verify the correctness of g by proving the property

$$g(x) = \langle f(Sx), f(x) \rangle$$

Proving it, however, requires a lemma:

$$sum(g(x)) = f(Sx) + f(x)$$

Figure 3 shows a proof of the two equations constructed by the inductive completion procedure. The reduction order is the recursive path order generated by the precedence

$$f > g > np > sum > pair > + > s > 0$$

The set $\{0, Sn\}$ is uniformly chosen as the $>$ -cover set for nat (by the equations of Figure 1). The subterms shown in boxes are those used for the mandatory reduction step in each expansion.

Notice how the theorem and the lemma use each other as inductive hypotheses. The lemma (rule 2) is used in the proof of both the theorem as well as the lemma in the first two simplification steps. The theorem (via rule 5) is used in both in the last simplification step. \square

5 Relationship to Knuth-Bendix Completion

In this section, we relate the inductive completion procedure of the previous section to an inductive proof procedure based on Knuth-Bendix completion [10, 19, 24]. The correctness of the latter then follows immediately.

Rule	E_i	simplifications
<i>Expand</i>	1. $\langle \boxed{f(Sx)}, f(x) \rangle = g(x)$ 2. $\boxed{f(Sx)} + f(x) = \text{sum}(g(x))$	
<i>Simplify/</i> <i>Delete</i>	$\langle S0, f(0) \rangle = g(0)$ $\langle f(Sn) + f(n), f(n) \rangle = g(Sn)$ $S0 + f(0) = \text{sum}(g(0))$ $f(Sn) + f(n) + f(Sn) = \text{sum}(g(Sn))$	using R ; deleted using R using R ; deleted using R and 2
<i>Expand</i>	3. $\langle \text{sum}(\boxed{g(n)}), f(Sn) \rangle = \text{np}(g(n))$ 4. $\text{sum}(\boxed{g(n)}) + f(Sn) = \text{sum}(\text{np}(g(n)))$	
<i>Simplify/</i> <i>Delete</i>	$\langle \text{sum}(\langle S0, 0 \rangle), f(S0) \rangle = \text{np}(g(0))$ $\langle \text{sum}(\text{np}(g(n))), f(SSn) \rangle = \text{np}(g(Sn))$ $\text{sum}(\langle S0, 0 \rangle) + f(Sn) = \text{sum}(\text{np}(g(n)))$ $\text{sum}(\text{np}(g(n))) + f(SSn) = \text{sum}(\text{np}(g(Sn)))$	using R ; deleted using R and 2 using R ; deleted using R and 2
<i>Expand</i>	5. $\langle \text{sum}(\text{np}(g(n))), \text{sum}(g(n)) \rangle = \text{np}(\text{np}(\boxed{g(n)}))$ $\text{sum}(\text{np}(g(n))) + \text{sum}(g(n)) = \text{sum}(\text{np}(\text{np}(g(n))))$	
<i>Simplify/</i> <i>Delete</i>	$\text{np}(\text{np}(\langle S0, 0 \rangle)) = \langle \text{sum}(\text{np}(g(0))), \text{sum}(g(0)) \rangle$ $\text{np}(\text{np}(\text{np}(g(n)))) = \langle \text{sum}(\text{np}(g(Sn))), \text{sum}(g(Sn)) \rangle$ $\text{sum}(\text{np}(g(n))) + \text{sum}(g(n)) = \text{sum}(\text{np}(\text{np}(g(n))))$	using R ; deleted using R and 5; deleted using R and 5; deleted

Figure 3: Proof of the specification of g

5.1 Cover sets

The inference rule *Expand* places an important constraint on the choice of cover sets, viz., for each case $e\sigma_i = e'\sigma_i$, $e\sigma_i$ must be reducible by R . The completion method of inductive proofs attacks this problem by first trying to satisfy the requirement of reducible left hand sides. The set of cases is obtained by choosing σ_i such that $e\sigma_i$ is the most general instance of e rewritable by a rule in R at some position. The cases generated in this way are then examined for the cover set property.

Lemma 22 *Let e be a term, α a nonvariable position in e , and $R : l \rightarrow r$ a rewrite rule. The most general instance of e rewritable by R at α is $e\sigma$ where $\sigma = \text{mgu}(e/\alpha, l)$ is the most general unifier substitution.*

This property is often exhibited as a *narrowing* relation $e \rightsquigarrow_{R, \alpha, \sigma} e\sigma[\alpha \rightarrow r\sigma]$ (Cf. [9, 17, 27, 31]).

Definition 23 (cover set of narrowings) *Let e be a term, R a rewrite system. A set of narrowings $\{e \rightsquigarrow_{R, \alpha_i, \sigma_i} d_i\}$ is a $>$ -cover set of narrowings if $\{\sigma_i\}_i$ is a $>$ -cover set of substitutions for e .*

Just as we have used the rewrite relation \rightarrow_R^+ as the well-founded order for induction, we can also use it as the well-founded order to show cover sets. A necessary and sufficient condition for the existence of \rightarrow_R^+ -cover set of narrowings is inductive reducibility [5, 19].

Definition 24 (inductive reducibility) *A term e is inductively reducible by R (also called ground reducible and quasi-reducible) if every ground instance of e is reducible by R .*

Methods for testing inductive reducibility when R is left-linear are given in [19]. Methods for the general case, which are however efficient only for the left-linear case may be found in [22]. More efficient tests are possible when a set of constructors is available [15, 19].

Proposition 25 *Let R be a terminating rewrite system, then e has a \rightarrow_R^+ cover set of narrowings if and only if e is inductively reducible by R .*

It is easily seen that the set of all narrowings of e is a cover set if e is inductively reducible.

Optimizations Even though the set of *all narrowings* gives a cover set, it is often an over-kill. For instance, if e has an inductively reducible subterm e/α then the set of narrowings inside the subterm is adequate for a cover set. The optimizations presented in [10, 24] are based on such ideas. The methods for testing inductive reducibility [19, 22] can be adapted to finding an adequate cover set of narrowings. See [4] for recent results.

Limitations of inductive reducibility method Even though Proposition 25 shows that inductive reducibility is a necessary condition for the existence of \rightarrow_R^+ cover sets, it is not necessary for the existence of arbitrary cover sets of narrowings. For instance, as shown in Example 5, $N_2 = \{0, s(0), x + y\}$ is a \succ_2 -cover set for *nat*. But, it is not a \rightarrow_R^+ -cover set. (E.g., $s(s(0)) \succ_2 s(0) + s(0)$, but it is not reducible). Such a cover set is, however, useful for proving properties of the following definition:

$$\begin{aligned} x \times 0 &\rightarrow 0 \\ x \times s(0) &\rightarrow x \\ x \times (y + z) &\rightarrow (x \times y) + (x \times z) \end{aligned}$$

5.2 Critical pairs as lemmas

The inductive completion procedure **I**, with the use of cover sets of narrowings for generating cases, is very close to the Knuth-Bendix completion procedure [3, 6, 16, 23]. If $e \rightsquigarrow_{R, \alpha, \sigma} b$ is a narrowing, then the equational proof $e'\sigma \leftarrow e\sigma \rightarrow b$ is a *critical peak proof* and the equation $e'\sigma = b$ is a *critical pair*. Thus the inference rule *Expand* is essentially constructing critical pairs. However, the set of critical pairs generated by Knuth-Bendix completion is a superset of those needed for a cover set. It is sometimes found that the additional critical pairs have a useful role as lemmas. For instance, the crucial lemma involved in example 21 can be constructed from the critical peak proof:

$$sum(g(x)) \leftarrow_P sum(\langle f(Sx), f(x) \rangle) \rightarrow_R f(Sx) + f(x)$$

To relate the Knuth-Bendix completion procedure with the inductive completion procedure, we formulate a procedure \mathbf{I}^K as a specialization of \mathbf{I}^+ (for inductive completion with lemmas). The inference rule *Postulate* is specialized to:

$$\textit{Postulate} \quad \frac{(E \cup E', H)}{(E, H)} \quad E' \subseteq CP(R \cup H, R \cup H)$$

where CP is the set of critical pairs. The relation between \mathbf{I}^K and the use of Knuth-Bendix completion for inductive proofs [5, 15, 19, 22] is the following. Every inductive Knuth-Bendix derivation (derivation generating only inductively reducible left hand sides of rules) can be transformed to an \mathbf{I}^K derivation using \rightarrow_R^+ cover sets. Of the critical pairs generated by the Knuth-Bendix derivation, those belonging to the cover set are obtained by *Expand* and the others by *Postulate*. The fact that the Knuth-Bendix derivation is inductive guarantees that the critical pairs include \rightarrow_R^+ cover sets. Thus, the following result immediately follows from Proposition 20.

Corollary 26 *If there is an inductive Knuth-Bendix derivation starting from (E, R) and ending in (\emptyset, R') , then $R \vdash_{ind} E$.*

This generalizes the results of [1, 5, 10, 15, 19, 22, 24] in that it is independent of the ground confluence of the rewrite system R . Note that the purported confluence requirement is the main argument against the use of completion-based methods in [11, 32].

Not only does the inductive completion procedure \mathbf{I}^K simulate Knuth-Bendix completion, it is more powerful than the latter as it does not insist that all the critical pairs should have inductive proofs. This has been discussed elsewhere [1, 10, 24, 28].

6 Completeness

Inductive theories of equational systems are as hard as Peano arithmetic, and so, by Gödel's incompleteness theorem, are not semidecidable. There can be no procedure for inductive proofs which is complete in the real sense.

The oft-cited completeness result for the completion-based procedures is the following. A completion derivation is said to be *fair* if the set of persisting equations $E_\infty = \bigcup_i \bigcap_{j \geq i} E_j$ is empty.

Proposition 27 *If R is a ground confluent and terminating rewrite system, and there is a fair Knuth-Bendix derivation starting from (E, R) , then $R \vdash_{ind} E$ if and only if the derivation is inductive.*

Note that there is no guarantee that the completion derivation is finite. So, this may be called *completeness at ω* (or *co-semi-completeness* [22]). However, notice also the strong restriction that R must be ground confluent. Its only use is to refute false propositions, not to produce effective proofs. Given that we are mostly interested in *proving* propositions rather than *refuting* them, this requirement is of limited practical significance.

To talk about useful completeness properties, we need a notion of *relative completeness* which factors out the second-order aspects of inductive proofs and states that the first-order

aspects can be carried out by the proof procedure. The following definitions propose such a notion.

Definition 28 (inductive proof) *An inductive proof of $R \vdash_{ind} E$, using a stable well-founded order \succ , consists of a finite cover set of cases for each equation in E , and an equational proof for each case equation of the form $e\sigma_i \leftrightarrow_{R \cup E_{\prec}}^* e'\sigma_i$ where*

$$E_{\prec} = \{e\theta = e'\theta \mid (e = e') \in E, \{e\theta, e'\theta\} \ll \{e\sigma_i, e'\sigma_i\}\}$$

The proof thus constructed is “relative” to the lemmas provided in E .

Definition 29 (relative completeness) *An inductive proof procedure is relatively complete if, whenever there is an inductive proof of $R \vdash_{ind} E$ using \succ , the procedure succeeds in proving it.*

Example 30 Consider again the problem of Example 21, but without the function *sum*. The lemma used in that example is now inexpressible. However, an inductive proof exists for the proposition using the (decreasing order generated by the) given reduction order. Note:

$$g(0) \rightarrow_{G1} \langle S0, 0 \rangle \leftarrow_{\{F1, F2\}}^{\dagger} \langle f(S0), f(0) \rangle$$

$$g(Sn) \rightarrow_{G2} np(g(n)) \leftarrow_P np(\langle f(Sn), f(n) \rangle) \rightarrow_{G3} \langle f(Sn) + f(n), f(Sn) \rangle \leftarrow_{F3} \langle f(SSn), f(Sn) \rangle$$

But, there is no finite \mathbf{I}^K derivation for the proposition. (This may be checked by the reader). The problem is that the smaller instance condition:

$$\{g(Sn), \langle f(SSn), f(Sn) \rangle\} \not\gg \{g(n), \langle f(Sn), f(n) \rangle\}$$

holds for the decreasing order generated by the reduction order, but it does not hold for the decreasing order \succ_R generated by the rewrite relation. \square

Thus, we have that

Proposition 31 *Both inductive completion and Knuth-Bendix completion are relatively incomplete for proving inductive propositions.*

In the light of the earlier discussion, this is not surprising because the inductive completion procedure uses the rewrite relation as the well-founded order and so cannot find proofs based on other well-founded orders. This shortcoming suggests ways to improve the completion method so as to use other well-founded orders, but we do not explore those in this paper.

7 Summary

To summarize the results of the paper, we have formulated *Term rewriting induction* as Noetherian induction specialized by the use of a rewrite relation as the well-founded order. It is shown that the widely studied completion-based inductive proof procedures construct inductive proofs using this induction method. These procedures are shown to be correct independent of the *ground confluence* restriction of the axioms. A notion of relative completeness is proposed for inductive proof methods. Both the inductive completion procedure and the general Knuth-Bendix completion procedure are not relatively complete in this sense.

8 Acknowledgements

I am grateful to Francois Bronsard, Nachum Dershowitz, Deepak Kapur, Wolfgang Küchlin and Hantao Zhang for several discussions which led to the observations made here. The idea of relative completeness in section 6 was suggested by Dershowitz.

References

- [1] L. Bachmair. Proof by consistency. In *Symp. on Logic in Computer Science*, IEEE, 1988.
- [2] L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. In P. Lescanne, editor, *Rewriting Techniques and Applications*, pages 192–203, Springer-Verlag, 1987. (Lecture Notes in Comp. Science, Vol 256).
- [3] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *Symp. on Logic in Computer Science*, pages 346–357, IEEE, 1986.
- [4] R. Bündgen and W. Küchlin. Computing ground reducibility and inductively complete positions. In N. Dershowitz, editor, *Rewriting Techniques and Appl.*, pages 59–75, Springer-Verlag, Berlin, 1989.
- [5] N. Dershowitz. Applications of the Knuth–Bendix completion procedure. In *Proc. of the Seminaire d’Informatique Theorique, Paris*, pages 95–111, December 1982.
- [6] N. Dershowitz. Completion and its applications. In *Resolution of Equations in Algebraic Structures*, Academic Press, New York, 1988.
- [7] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.
- [8] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, August 1979.
- [9] M. Fay. First-order unification in an equational theory. In *Fourth Workshop on Automated Deduction*, pages 161–167, Austin, Texas, 1979.
- [10] L. Fribourg. A strong restriction of the inductive completion procedure. In *Intern. Conf. Aut., Lang. and Program.*, pages 105–115, Rennes, France, July 1986. (Springer Lecture Notes in Computer Science, Vol. 226).
- [11] S. J. Garland and J. V Gutttag. Inductive methods for reasoning about abstract data types. In *ACM Symp. on Princ. of Program. Languages*, pages 219–228, ACM, 1988.
- [12] J. A. Goguen. How to prove inductive hypotheses without induction. In *Conf. on Automated Deduction*, pages 356–372, Springer Verlag Lecture Notes in Computer Science, Vol 87, Jul 1980.

- [13] C. A. R. Hoare. Procedures and parameters: An axiomatic approach. In E. Engeler, editor, *Symp. Semantics of Algorithmic Languages*, pages 102–116, Springer-Verlag, 1971. (Lect. Notes in Math. Vo. 188).
- [14] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980. (Previous version in *Proc. Symp. Foundations of Computer Science*, Oct 1977).
- [15] G. Huet and J.-M. Hullot. Proofs by induction in equational theories with constructors. In *Symp. on Foundations of Computer Science*, pages 96–107, IEEE, Lake Placid, NY, October 1980.
- [16] G. Huet and D. C. Oppen. Equations and rewrite rules: A survey. In R. Book, editor, *Formal Language Theory: Perspectives and Open Problems*, pages 349–405, Academic Press, New York, 1980.
- [17] J.-M. Hullot. Canonical forms and unification. In *Conf. on Automated Deduction*, pages 318–334, 1980.
- [18] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. on Computing*, Nov 1986.
- [19] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in equational theories without constructors. In *Symp. on Logic in Computer Science*, pages 358–366, IEEE, Cambridge, MA., June 1986.
- [20] D. Kapur and D. R. Musser. Proof by consistency. In *Proc. of NSF Workshop on the Rewrite Rule Laboratory, Sep 4-6, 1983*, G.E. R&D Center Report GEN 84008, Schenectady, April 1984.
- [21] D. Kapur, P. Narendran, and F. Otto. *On Ground-Confluence of Term Rewriting Systems*. Technical Report 87-6, General Electric R & D Center, Schenectady, New York, 1987.
- [22] D. Kapur, P. Narendran, and H. Zhang. Proof by induction using test sets. In *Conf. on Automated Deduction*, Oxford, U.K., 1986.
- [23] D. Knuth and P. Bendix. Simple word problems in Universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297, Pergamon Press, 1970.
- [24] W. Küchlin. Inductive completion by ground proof transformation. In *Proc. 1987 MCC Colloquium on Resolution of Equations in Algebraic Structures*, MCC, Austin, Texas, 1988.
- [25] D. S. Lankford. *A Simple Explanation of Inductionless Induction*. Memo MTP-14, Dep. of Mathematics, Louisiana Tech. Univ., Aug 1981.
- [26] D. R. Musser. On proving inductive properties of abstract data types. In *ACM Symp. on Princ. of Program. Languages*, pages 154–162, ACM, Las Vegas, 1980.
- [27] U. S. Reddy. Narrowing as the operational semantics of functional languages. In *Symp. on Logic Program.*, pages 138–151, IEEE, Boston, 1985.

- [28] U. S. Reddy. Rewriting techniques for program synthesis. In N. Dershowitz, editor, *Rewriting Techniques and Appl.*, pages 388–403, Springer-Verlag, 1989. (LNCS Vol. 355).
- [29] U. S. Reddy. Transformational derivation of programs using the Focus system. In *Symp. Practical Software Development Environments*, pages 163–172, ACM, December 1988.
- [30] D. Scott. Data types as lattices. *SIAM Journal on Computing*, 5(3):522–587, Sept. 1976.
- [31] J. R. Slagle. Automated theorem-proving for theories with simplifiers, commutativity and associativity. *Journal of the ACM*, 21(4):622–642, 1974.
- [32] H. Zhang, D. Kapur, and M. S. Krishnamoorthy. A mechanizable induction principle for equational specifications. In E Lusk and R. Overbeek, editors, *Conf. on Automated Deduction*, pages 162–181, Springer-Verlag, Berlin, 1988. (LNCS Vol 310).