H. Comon and R. Nieuwenhuis

Induction = I−Axiomatization + First−Order Consistency

Research Report LSV−98−9, Oct. 1998

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Abstract

In the early 80's, there was a number of papers on what should be called *proofs by consistency.* They describe how to perform inductive proofs, without using an explicit induction scheme, in the context of equational specifications and ground-convergent rewrite systems. The method was explicitly stated as a first-order consistency proof in case of pure equational, constructor based, specifications.

In this paper, we show how, in general, inductive proofs can be reduced to first-order consistency and hence be performed by a first-order theorem prover. Moreover, we extend previous methods, allowing non-equational specifications (even non-Horn specifications), designing some specific strategies. Finally, we also show how to drop the ground convergence requirement (which is called *saturatedness* for general clauses).

Induction = I-Axiomatization + First-Order Consistency

Hubert Comon
L.S.V., ENS Cachan
61 avenue du président Wilson
94235 Cachan cedez, France
E-mail: comon@lsv.ens-cachan.fr

Robert Nieuwenhuis *
Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics
Campus Nord C6, Jordi Girona 1
E-08034 Barcelona, Spain.
E-mail:roberto@lsi.upc.es

October 8, 1998

1 Inductive proofs

First-order specifications are ubiquitous in virtually all areas of computer science. In many cases, the intended meaning of a specification $\mathcal{E}$ is not its standard first-order semantics, i.e., the class of all its models, but rather a more specific class $\mathcal{M}$. Perhaps the most well-known example is the *initial* or *minimal Herbrand model* semantics, where $\mathcal{M}$ consists of the unique minimal Herbrand model of a set of Horn clauses with or without equality $\mathcal{E}$ (an algebraic specification, a logic program, a deductive data base, ...). Other interesting semantics that are used in practice as well include *final* semantics, or, for possibly non-Horn $\mathcal{E}$, *perfect model* semantics or the class of all (minimal) Herbrand models.

In this paper we address the issue of inductive proofs in this broad sense: we wish to define general methods for (semi-)automatically (dis)proving the validity of conjectures C in classes of Herbrand models of $\mathcal{E}$, for universally quantified C and $\mathcal{E}$. In order to minimize user expertise, we do not want to require the user of such a (semi-)automatic inductive theorem prover to provide explicit induction schemes. Instead, induction will be based on simple, automatically generated, well-founded orderings $\succ$ on formulae. This is what distinguishes our approach from the *explicit induction* methods that will not be addressed in this paper.

Theory imposes severe limitations on our aims. Unlike it happens when proving standard first-order consequences (i.e., validity in the class of all models), from Gödels incompleteness theorem it follows that in general there are no complete proof systems that can be used to enumerate all inductively valid formulae, and hence inductive validity is not even semi-decidable. In Section 6 we will show that this is the case even for validity of equations in the initial model of very restrictive classes of specifications $\mathcal{E}$, like the $\mathcal{E}$ presented by purely equational, convergent, linear, right

*Both authors supported by CCL ESPRIT working group 22457. This work was presented by the first author as an invited talk at RTA'98.

shallow, constructor-based term rewrite systems, or the $\mathcal{E}$ presented by a convergent set of length-reducing word rewrite rules. This shows why almost no decidability results for inductive validity exist in the literature: even in very restricted and simple situations the problem remains undecidable. To our knowledge, the only decidability results are for *ω-complete* theories, i.e., where (for infinite $\mathcal{I}$) the inductive theory coincides with the equational theory, like the *shallow* equational theories of [CHJ94] or the *Catalog* Horn theories of [Nie96].

However, under reasonable assumptions it is possible to obtain what we will call *refutation complete* procedures for inductive validity: procedures that provide in finite time a disproof for any conjecture that is *false* (in $\mathcal{M}$). Then the situation is exactly reverse—but probably less useful in practice—to what happens in standard first-order logic, where all formulae *valid* (in the class of *all* models of $\mathcal{E}$) are provable in finite time.

It is interesting to observe that refutation completeness amounts to refutation completeness for *ground* conjectures: a conjecture C is valid in $\mathcal{M}$ if, and only if, all its ground instances are valid (since $\mathcal{M}$ is a class of Herbrand models) and one can simply enumerate (representatives of) all ground instances until a ground counterexample is found by the ground refutation procedure. This simple idea is in essence the basis of methods like the ones of [Red90, Zha88, BR95], and indeed refutation procedures or decision procedures for ground conjectures are available in many practical settings, like in the presence of a *convergent* term rewrite system or an adequate *saturated* clausal presentation for $\mathcal{E}$.

In fact, if $\mathcal{M}$ includes all minimal Herbrand models of $\mathcal{E}$ (and hence, in particular, if $\mathcal{M}$ is the initial model of $\mathcal{E}$) then refutation completeness for ground conjectures c, i.e., the co-semi-decidability of $\mathcal{M} \models c$, is equivalent to decidability of this problem: since $\mathcal{M}$ includes the minimal models, for every ground atom A we have $\mathcal{M} \models A$ if, and only if, $\mathcal{E} \models A$ (i.e., A is a first-order logical consequence of $\mathcal{E}$). Then, since $\mathcal{E} \models A$ is semi-decidable, $\mathcal{M} \models A$ is decidable and hence, for ground c, $\mathcal{M} \models c$ is decidable as well[1].

Our techniques will extensively deal with refutation completeness, among other reasons because a procedure that makes progress towards a disproof when given a false conjecture, usually also advances towards a proof when given a valid conjecture. In the class of methods described in this paper, progress is made by inference rules *reducing* (with respect to $\succ$) counter examples, i.e., false ground instances $c\sigma$ of a conjecture c. Once a minimal (irreducible) counter example appears, it is detected by other means. Here we build upon previous work on *proofs by consistency* for the purely equational case, formerly also called *inductionless induction* methods, where minimal counter examples were detected in several forms, like equations $true = false$ [Mus80], equations between constructors [HH82], or *ground irreducible* non-trivial equations [JK86, Bac88], among others. We generalize those methods for the

[1] Ganzinger and Stuber [GS92] state that if validity of ground conjectures is not decidable, "inductive theorem proving is hopeless anyway". Here we show that refutation completeness (rather than inductive theorem proving) becomes hopeless in that case only if $\mathcal{M}$ includes all minimal Herbrand models of $\mathcal{E}$, which is not true in their setting. Their perfect model is only one of the minimal models and hence a (rather hypothetical, however) co-semidecision procedure for ground validity would suffice for obtaining refutation completeness.

detection of inconsistencies by introducing the notion of *I-axiomatization.* These axiomatizations will be used in a uniform framework for proof by consistency that allows us to eliminate many restrictions on syntax and semantics (saturatedness requirements on $\mathcal{E}$, arbitrary universal formulae, more general redundancy notions and classes of models). The explicit use of axiomatizations was introduced in [Fri84] for free constructors equational specifications, but not further generalized in the following papers.

An I-axiomatization is a set $\mathcal{A}$ of first-order formulae such that $\mathcal{A} \cup \mathcal{E} \cup \{c\}$ is consistent if, and only if, $\mathcal{M} \models c$. Such I-axiomatizations hence allow us to reduce inductive proofs to first-order consistency proofs, which in turn makes it possible to apply general-purpose first-order theorem provers to inductive validity problems. Note that of course we do not require $\mathcal{A}$ to completely axiomatize $\mathcal{M}$, which is impossible in general.

The issue of how to compute I-axiomatizations automatically will be treated in Section 5. The other main question to be answered is of course how to design efficient procedures proving the inconsistency of $\mathcal{A} \cup \mathcal{E} \cup C$ for all false sets of conjectures C, and proving its consistency in as many cases as possible for valid C. The remainder of this paper is devoted to this in a number of different situations, depending on the intended semantics and the syntactic properties of $\mathcal{E}$ and C. An important step forwards for our aims is a completeness result for the following two-stage approach which is common to our different strategies: on the one hand, new consequences are computed from $\mathcal{E} \cup C$ in a restricted way; on the other, each new consequence is checked for inconsistency with $\mathcal{A}$. The latter can be done either by a standard first-order prover or by a dedicated —in some cases, decision—procedure (see Section 5).

In Section 2 we first concentrate on the case where $\mathcal{E}$ is Horn, $\mathcal{M}$ is the minimal Herbrand model, and C is any set of clauses. The results are hence directly applicable to the case where $\mathcal{M}$ is the class of all Herbrand models if C contains only positive clauses. Different special cases of constructor-based specifications are handled separately in Section 3. In Section 4 we generalize these ideas to non-Horn $\mathcal{E}$ and perfect model semantics.

This paper does not contain any difficult new results; its contributions are more at the conceptual level and many proofs are actually adaptations of standard proofs which are not reproduced here. We show however how inductive validity can be reduced to first-order consistency, without the usual assumptions of the proofs by consistency method. The main advantage is that any saturation based general purpose first-order theorem prover can be used for inductive validity. For instance, we experimented the use of *Saturate* [NN93, GNN95] for this purpose. On all (small size) examples we tried, this experiment was quite successful. We reproduce some of the *Saturate* proofs in the paper.

2 Inductive Proofs and I-Axiomatizations

In this section we assume a finite signature $\mathcal{F}$, and that $\mathcal{E}$ is a finite set of Horn clauses (the *axioms*), C is a set of clauses (the *conjectures*), $\mathcal{I}$ is the minimal Herbrand model of $\mathcal{E}$, and we address the question whether $\mathcal{I} \models C$. Finally, let $\succ$ be a total reduction ordering on $\mathcal{T}(\mathcal{F})$.

For simplicity, we can assume as well that equality is the only predicate symbol, since (positive or negative) atoms $P(t_1 \ldots t_n)$ can be expressed everywhere as (positive or negative) equations $P(t_1 \ldots t_n) = true$, where *true* is a new special symbol and P is considered as a (boolean) function symbol. Indeed, if $\mathcal{I}'$ is the resulting minimal Herbrand model, then clearly for every ground atom A it holds that $\mathcal{I} \models A$ iff $\mathcal{I}' \models A = true$ (note, however that $\mathcal{I}$ and $\mathcal{I}'$ are not isomorphic since two ground atoms that are false in $\mathcal{I}$ need not be in the same congruence class of $\mathcal{I}'$). In what follows, when P is a predicate symbol, we only allow equations (resp. disequations) of the form $P(s_1, \ldots, s_n) = true$ (resp. their negation) which will be sometimes abbreviated as $P(s_1, \ldots, s_n)$ (resp. $\neg P(s_1, \ldots, s_n)$). In other words, equations of the form $P(s_1, \ldots, s_n) = Q(t_1, \ldots, t_m)$ are ruled out by our syntax. This will be always consistent with ordering strategies, if we assume *true* to be the smallest term, which we will do in the rest of the paper.

Definition 1 *A set $\mathcal{A}$ of first-order formulas is an* I-axiomatization *of $\mathcal{I}$ if*

1. *$\mathcal{A}$ is recursive and contains only purely universal sentences (i.e. $\mathcal{A}$ is a set of clauses)*
2. *$\mathcal{I}$ is the only Herbrand model of $\mathcal{E} \cup \mathcal{A}$ up to isomorphism.*

In other words, an I-axiomatization must contain enough negative information to rule out all non-minimal Herbrand models of $\mathcal{E}$, or, equivalently, it must ensure the well-known concept of "no confusion". As said, in Section 5 we will extensively explain how to automatically derive $\mathcal{I}$-axiomatizations. Let us only remark at this point that, since an ordering on ground terms $\succ$ is available, a convenient and intuitive way for defining I-axiomatizations is given by the following:

Definition 2 *A ground term t is called* normal *if it is the (unique) minimal (w.r.t. $\succ$) representative of its congruence class in $\mathcal{I}$. Similarly, a ground clause c is normal if all terms occurring in c are normal.*

Lemma 1 *Let $\mathcal{A}$ be a set of first-order clauses such that $\mathcal{I} \models \mathcal{A}$ and $\mathcal{A} \models s \neq t$ for any two distinct normal terms s and t. Then $\mathcal{A}$ is an I-axiomatization.*

Proof: If J were a non minimal Herbrand model such that $J \models \mathcal{A} \cup \mathcal{E}$, then $J \models u = v$ for some ground u and v such that $\mathcal{I} \not\models u = v$, i.e., two different congruence classes in $\mathcal{I}$ are merged in J, which is impossible since $\mathcal{A} \models s \neq t$ for the normal representatives s and t of these classes. $\square$

In the following, the particular kind of I-axiomatization given by this lemma will be called *normal*, and, unless stated otherwise, we assume all I-axiomatizations to be normal.

For instance, consider for $\mathcal{E}$ the set of equations

$$\left\{ \begin{array}{rcl} 0 + x & = & x \\ s(x) + y & = & s(x + y) \end{array} \right.$$

Assuming that $F = \{0, s, +\}$, an $\mathcal{I}$-axiomatization could be:

$$\forall x, y. s(x) \neq 0 \wedge (s(x) = s(y) \Rightarrow x = y)$$

Note that there is no need to express differences between terms headed by $+$ because they are equivalent to terms without $+$. Indeed if terms with $+$ are bigger w.r.t. $\succ$ than terms built of only s and 0, then this I-axiomatization is normal.

Note that not all I-axiomatizations are normal: if a, b, c, d are constants with $a \succ b \succ c \succ d$ and $\mathcal{E} = \{a = b, c = d\}$ then $\mathcal{A} = \{a \neq c\}$ is an I-axiomatization that is not normal (any normal one must entail $b \neq d$).

The following key proposition allows us to reduce the problem of proving inductive theorem to the consistency of a finite set of clauses:

Proposition 1 *Let $\mathcal{A}$ be an I-axiomatization. Then $\mathcal{A} \cup \mathcal{E} \cup C$ is consistent, if, and only if, $\mathcal{I} \models C$.*

Proof: If $\mathcal{I} \models C$, then $\mathcal{I} \models \mathcal{A} \cup \mathcal{E} \cup C$ which is then consistent. Conversely, if $\mathcal{A} \cup \mathcal{E} \cup C$ is consistent, then it has a Herbrand model, as this is a set of purely universal formulas. Now, from the last property of I-axiomatizations, this model should be $\mathcal{I}$.

2.1 Inductive saturation

As said, our aim is to define an inference system that *reduces* (with respect to $\succ$) counter examples, i.e., the smallest false ground instance $c\sigma$ of some conjecture c in C. We will do this only if $c\sigma$ is not a normal clause, since for normal clauses we have the following:

Lemma 2 *Let $\mathcal{A}$ be a normal I-axiomatization, let c be a clause, and let $c\sigma$ be a normal clause such that $\mathcal{I} \not\models c\sigma$. Then $\mathcal{A} \cup \{c\}$ is inconsistent.*

Proof: $\mathcal{A} \cup \{c\sigma\}$ is inconsistent if (i) every model of $\mathcal{A}$ satisfies the negative literals of $c\sigma$ and (ii) no model of $\mathcal{A}$ satisfies any of the positive ones. If $\mathcal{I} \not\models c\sigma$ then $\mathcal{I} \models s = t$ for all negative equations $s = t$ in $c\sigma$. Since s and t are both normal, it must be the case that $s \equiv t$, which implies (i). For (ii), let $u = v$ be a positive equation $u = v$ in $c\sigma$. Since $\mathcal{I} \not\models c\sigma$, we have $u \not\equiv v$, and hence, since $\mathcal{A}$ is normal, $\mathcal{A} \models u \neq v$ and hence no model of $\mathcal{A}$ satisfies $u = v$. This implies the inconsistency of $\mathcal{A} \cup \{c\sigma\}$ and hence of $\mathcal{A} \cup \{c\}$.

Now it remains to design efficient procedures that are able to reduce any non-normal counter example. Since this is essentially a well-known problem in first-order saturation-based theorem proving, we can rely on a large amount of existing results from this field. We refer to [BG94] for more details and restate only the main results. Let us recall here only the ground versions of the following inference rules for Horn clauses in sequent notation, where $s \succ t, \Gamma$ denotes that $s \succ t$ and $s \succ u$ for all terms u occurring in Γ:

superposition right:

$$\frac{\Gamma' \rightarrow l = r \qquad \Gamma \rightarrow s = t}{\Gamma', \Gamma \rightarrow s[r]_p = t} \qquad \text{if} \qquad \begin{array}{l} s|_p \equiv l \text{ and} \\ l \succ r, \Gamma' \text{ and } s \succ t, \Gamma. \end{array}$$

superposition left:

$$\frac{\Gamma' \rightarrow l = r \qquad \Gamma, s = t \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p = t \rightarrow \Delta} \qquad \text{if} \qquad \begin{array}{l} s|_p \equiv l \text{ and} \\ l \succ r, \Gamma', \; s \succ t \text{ and } s \succeq \Gamma, \Delta. \end{array}$$

equality resolution:

$$\frac{\Gamma, s = s \rightarrow \Delta}{\Gamma \rightarrow \Delta} \qquad \text{if} \qquad s \succeq \Gamma, \Delta.$$

Non-ground versions of these rules are defined as usual. For example, by equality resolution on $\Gamma, s = t \rightarrow \Delta$, the conclusion $\Gamma\sigma \rightarrow \Delta\sigma$ is obtained if σ is the most general unifier of s and t and $s\sigma\theta$ can indeed be the maximal term of $(\Gamma, s = t \rightarrow \Delta)\sigma\theta$ for some ground θ (the latter condition is decidable if $\succ$ is some *recursive or lexicographic path ordering*; otherwise, approximations are used and some more inferences than needed may be computed). It is also possible to work with constrained formulae inheriting the generated unification and ordering restrictions as constraints [NR95].

It is well-known that if $\mathcal{E}$ is consistent and saturated under superposition and equality resolution then $\mathcal{I}$ is (isomorphic to) $\mathcal{T}(\mathcal{F})/_{=R}$, where R is a convergent ground term rewrite system (TRS) such that each rule $l \Rightarrow r$ in R is *generated* by a ground instance $\Gamma \rightarrow l = r$ of a clause in $\mathcal{E}$ with $l \succ r, \Gamma$ [BG94, NR95]. In what follows, let R denote this TRS. Clearly, a term (or a clause) is normal if, and only if, it is in normal form with respect to R. Similarly, we will call a ground substitution σ *normal* (or *irreducible* by R) if $x\sigma$ is normal (or irreducible by R) for all $x \in Dom(\sigma)$.

Let us assume from now on that $\mathcal{E}$ is saturated under superposition and equality resolution. In Section 3.2 we will show how this requirement can be weakened in many cases. We now define *conjecture superposition*, a form of superposition where the left premise is always a (definite) Horn clause of $\mathcal{E}$ and the right premise c is a conjecture in C:

conjecture superposition:

$$\frac{D \vee l = r \qquad c}{(D \vee c[r]_p)\sigma} \qquad \text{if} \qquad \begin{array}{l} \sigma = mgu(c|_p, l) \text{ and, for some ground } \theta, \\ l\sigma\theta \succ r\sigma\theta, D\sigma\theta, \text{ and, if } p \text{ is inside } s \text{ in a} \\ \text{negative literal } s = t \text{ of } c \text{ then } s\sigma\theta \succ t\sigma\theta \end{array}$$

Note that in this inference rule there are strong ordering restrictions on the left premise, the clause of $\mathcal{E}$, but only a weak ordering restriction on the conjecture clause c has been imposed so far. Furthermore, here we have given the standard non-ground version of this inference rule, but again it is possible to apply constraint inheritance here. For example, the basicness restriction can be imposed, i.e., no inferences are needed on terms introduced by unifiers of previous inferences generating ancestor conjectures (see [NR95, BGLS95] for the details).

We now give a number of definitions for the *redundancy* of conjectures and induction superposition inferences. They roughly coincide with well-known notions of redundancy in saturation-based first-order theorem proving, except that they include the use, without any ordering limitations, of formulae that are known to be valid in $\mathcal{I}$ (including lemmas proved by previous runs of our method or by any other means). The essence of induction is present in the fact that smaller (unproved) conjectures are applicable by what could be called the induction hypothesis:

In the following, let the ordering $\succ$ (ambiguously) denote as well as its multiset extension, used as an ordering on ground clauses (seen as the multiset of all their terms). Furthermore, if c is a ground clause and S is a set of clauses, we denote by $S^{\prec c}$ the set of all ground instances of clauses of S that are smaller than c (w.r.t. $\succ$). Finally, let $\mathcal{L}$ be a set of *lemmas*, i.e., arbitrary first-order clauses such that $\mathcal{I} \models \mathcal{L}$.

Definition 3 *A ground conjecture c is* redundant *in a set of conjectures C if $\mathcal{E} \cup \mathcal{A} \cup \mathcal{L} \cup C^{\prec c} \models c$. Similarly, a non-ground conjecture c is redundant if all its ground instances are.*

Definition 4 *A ground inference by conjecture superposition with right premise c and conclusion c' is* redundant *in a set of conjectures C if $\mathcal{E} \cup \mathcal{A} \cup \mathcal{L} \cup C^{\prec c} \models c'$. Similarly, a non-ground inference is redundant if all its ground instances are.*

Definition 5 *An* induction derivation *is a sequence of sets of conjectures $C_1.C_2, \ldots$ such that each C_{i+1} is obtained from C_i either by adding to C_i a logical consequence of $\mathcal{E}, \mathcal{A}, \mathcal{L}, C_i$ or by removing from C_i some conjecture that is redundant in C_i.*

A conjecture is persistent *in the derivation if for some j it belongs to all C_k with $k \geq j$.*

A derivation is fair *if every conjecture superposition inference with a persistent right premise is redundant in C_j for some j.*

The previous definitions are a simple adaptation for our purposes of the standard ones of saturated based first-order theorem proving. In particular, let us remark that in practical derivations usually the conjectures are stored in some (priority) queue insuring that every conjecture c is eventually either proved redundant or else considered for conjecture superposition; since under our ordering restrictions adding the conclusion of an inference makes the inference redundant, this implies the fairness of the derivation.

Theorem 1 *Let $C_0, C_1, \ldots$ be a fair induction derivation. Then $\mathcal{I} \models C_0$ if, and only if, there is no clause c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.*

Proof: If there is some c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent, then $\mathcal{I} \not\models C_0$, since all such c are logical consequences from $\mathcal{E}, \mathcal{A}, \mathcal{L}, C_0$ and $\mathcal{I} \models \mathcal{E}, \mathcal{A}, \mathcal{L}$.

For the reverse implication, assume $\mathcal{A} \cup \{c\}$ is consistent for all c in $\cup_i C_i$. We will derive a contradiction from the existence of a minimal (w.r.t. $\succ$) ground instance $c\sigma$ of a clause c in $\cup_i C_i$ such that $\mathcal{I} \not\models c\sigma$. If c is redundant in some C_j then from the definition of redundancy of conjectures, it follows that there is some false instance of a conjecture in C_j that is smaller than $c\sigma$, contradicting the minimality of $c\sigma$.

Otherwise c is persistent. By Lemma 2, $c\sigma$ is not a normal clause (otherwise $\mathcal{A} \cup \{c\}$ would be inconsistent). Furthermore, we can assume that σ is normal, since otherwise σ is reducible by R into some σ' such that $\mathcal{I} \not\models c\sigma'$ and $c\sigma \succ c\sigma'$, contradicting the minimality assumption on $c\sigma$.

Hence $c\sigma$ is reducible by some rule in R at some *skeleton* position, i.e., a position p in c. Let $D \vee l = r$ be the clause in $\mathcal{E}$ that generated the rule $l\theta \to r\theta$ of R that reduces $c\sigma$. Then $(c\sigma)|_p \equiv l\theta$ and hence $c|_p$ and l are unifiable by an mgu σ', and there exists some inference by conjecture superposition

$$\frac{D \vee l = r \qquad c}{(D \vee c[r]_p)\sigma'}$$

whose conclusion has an instance $c\sigma[r\theta]_p$ such that $\mathcal{I} \not\models c\sigma[r\theta]_p$ and moreover $c\sigma \succ c\sigma[r\theta]_p$. By fairness, this conclusion is redundant in some C_j. But then from the definition of redundancy of inferences, it follows that there is some false instance of a conjecture in C_j that is smaller than $c\sigma$, contradicting the minimality of $c\sigma$.

Now, it remains to show that superpositions on negative literals of a conjecture can be restricted to a maximal side of the equation. First note that, when $\mathcal{I} \not\models c\sigma$ and $c\sigma$ is not normal, then there is a literal L in c such that $\mathcal{I} \not\models L\sigma$ and $L\sigma$ is not normal (otherwise c can be written $c_1 \vee c_2$ where $\mathcal{I} \models c_1\sigma$ and each literal in $c_2\sigma$ is not normal. Then $\mathcal{I} \not\models c_2\sigma$, which contradicts lemma 2.) If L is a negative literal $g \neq d$, then $\mathcal{I} \models g\sigma = d\sigma$, hence the normal forms of $g\sigma$ and $d\sigma$ w.r.t. R are identical. Assuming $g\sigma \succ d\sigma$, g is maximal and $g\sigma$ is reducible by R. The above inference is then an overlap on g. □

2.2 More refined orderings

Up to now, for simplicity reasons, we have considered only an ordering $\succ$ on ground terms and clauses. However, in some redundancy proofs it is convenient to consider more refined orderings. In particular, *subsumption* cannot be handled by the redundancy notions defined up to now; for example, the equation $f(a) = b$ is not redundant in the presence of the equation $f(x) = b$.

This can be solved by techniques that are well-known from the field of saturation-based theorem proving [BG94, NR95], and which we do not want to treat in detail here. Let us only mention one possibility: compare ground instances $t\sigma$ and $s\theta$ of terms (or clauses) t and s by an ordering $\succ_p$ on pairs defined by: $(t,\sigma) \succ_p (s,\theta)$ if either $t\sigma \succ s\sigma$ or else $t\sigma \equiv s\sigma$ and s subsumes t but not vice versa. The definitions of redundancy can be adapted as follows according to this idea:

If $c\sigma$ is a ground instance of a clause c and S is a set of clauses, we denote by $S^{\prec c\sigma}$ the set of all ground instances $d\theta$ of clauses of S such that $(c,\sigma) \succ_p (d,\theta)$. A ground instance $c\sigma$ of a conjecture c is then redundant in a set of conjectures C if

$\mathcal{E} \cup \mathcal{A} \cup \mathcal{L} \cup C^{\prec c\sigma} \models c\sigma$, and a non-ground conjecture c is redundant if all its ground instances are.

2.3 Complete sets of positions

The only requirement for Theorem 1 to hold is that enough conjecture superposition inferences are computed in order to reduce the smallest false conjecture instance. In many cases, it is possible to determine, by analysis of a clause c, whether there is some subset P of the positions of c such that for all reducible (by R) $c\sigma$ where σ is normal, $c\sigma|_p$ is always reducible for some $p \in P$. In this case, we will call P a *complete* set of positions for c (and the given $\mathcal{E}$). In such a situation, it clearly suffices to compute inductive superposition inferences only at the positions in P.

This generalizes a number of notions defined in the literature for the case where $\mathcal{E}$ is purely equational, like the ones of [Fri86, Küc89]. The development of techniques for finding small complete sets of positions in our setting is related to the techniques of Section 5. In the case of a constructor discipline, described in the next section, the notion of complete set of positions will be especially useful.

2.4 Selection strategies

From, e.g., [BG94] it is known that, for first-order theorem proving, superposition remains complete with *selection:* in each clause c an arbitrary negative literal can be selected and the only inferences involving c are superpositions and equality resolution steps on this selected literal.

For our purposes of inductive validity proving, conjecture superposition remains complete with selection as well, provided we add equality resolution on selected literals in conjectures. In the following, in each conjecture clause c (the rightmost premise), a negative literal may have been selected:

conjecture superposition with selection:

$$\frac{D \vee l = r \qquad c}{(D \vee c[r]_p)\sigma}$$

if $\sigma = mgu(c|_p, l)$ and, for some ground θ, $l\sigma\theta \succ r\sigma\theta, D\sigma\theta$, and, if p is inside s in a negative literal $s = t$ of c then $s\sigma\theta \succ t\sigma\theta$, and if a literal l in c has been selected then p is in l.

negative equality resolution:

$$\frac{l \neq r \vee c}{c\sigma}$$

if $\sigma = mgu(l, r)$ and the literal $l \neq r$ has been selected in c.

Now, inductive derivations are defined according to these rules and using any given selection function:

Theorem 2 *Let $C_0, C_1, \ldots$ be a fair induction derivation with selection. Then $\mathcal{I} \models C_0$ if and only if there is no clause c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.*

Proof: We have almost the same proof as for theorem 1. Only the last paragraph has to be changed: Suppose $I \not\models c\sigma$, where σ is normal. Then $I \models s\sigma = t\sigma$ for all negative equations $s = t$ in c. If one such a negative equation $s = t$ is selected in c, either (i) $s\sigma$ and $t\sigma$ are the same term, and then there is an inference by equality resolution on $s = t$, or (ii) $s\sigma \succ t\sigma$ and hence $s\sigma$ is not normal, and then there is an inference by conjecture superposition on s. □

Example 1 *Let* gr *be the strict ordering on natural numbers defined by constructors* $0, s$. *The conjecture is the transitivity of the ordering. We use a negative literal selection strategy.*

```
 1    : axiom gr(s(x1),0)=true
 2    : axiom gr(x1,x2)=true -> gr(s(x1),s(x2))=true
 3    : conje gr(x1,x2)=true,gr(x2,x3)=true -> gr(x1,x3)=true

The total LPO precedence is [gr,s,0,true]

| ?- ind.

        Select negative literal in conje:
         3    : conje gr(x1,x2)=true,gr(x2,x3)=true -> gr(x1,x3)=true
        Type the number (left-to-right): 1.

Inference by conjecture superposition of 1 on 3 gives:
 4    : conje gr(0,x1)=true -> gr(s(x2),x1)=true

Inference by conjecture superposition of 2 on 3 gives:
 5    : conje gr(s(x1),x2)=true,gr(x3,x1)=true -> gr(s(x3),x2)=true

        Select negative literal in conje:
         4    : conje gr(0,x1)=true -> gr(s(x2),x1)=true
        Type the number (left-to-right): 1.

        Select negative literal in conje:
         5    : conje gr(s(x1),x2)=true,gr(x3,x1)=true -> gr(s(x3),x2)=true
        Type the number (left-to-right): 1.

Inference by conjecture superposition of 1 on 5 gives:
 6    : conje gr(x1,x2)=true -> gr(s(x1),0)=true

Inference by conjecture superposition of 2 on 5 gives:
 7    : conje true=true,gr(x1,x2)=true,gr(x2,x3)=true -> gr(s(x1),s(x3))=true

clause 6 is demodulated by rules [1] giving
 8    : conje gr(x1,x2)=true -> true=true (tautology)

Clause 7: gr(x1,x2)=true,gr(x2,x3)=true -> gr(s(x1),s(x3))=true is redundant by:
 2    : axiom gr(x1,x3)=true -> gr(s(x1),s(x3))=true
 3    : conje gr(x1,x2)=true,gr(x2,x3)=true,-->,gr(x1,x3)=true]
...

Induction derivation successfully terminated.
```

2.5 More ordering restrictions

The conjecture superposition rule may overlap clauses in $\mathcal{E}$ on any literal of the conjecture c and on any side of this literal, provided it is a positive one. This contrasts with the strict superposition rules where there are more ordering restrictions for the superposition. In general, we cannot do better as shown by the following simple example:

Example 2 *Assume we have only 3 constants a, b, c such that $a > b > c$ and $\mathcal{E}$ consists in the single equation $b = c$. Consider the normal axiomatization $\mathcal{A} = \{a \neq c\}$ and the (false) conjecture $a = b$. It is consistent with $\mathcal{A}$ and only superpositions on the small side are possible.*

This is a bit annoying since in the classical proof by consistency for the equational case, only superpositions on the maximal side of the equations are considered. This is because the axiomatization has some stronger properties.

Definition 6 *$\mathcal{A}$ is a* strongly normal axiomatization *if $\mathcal{I} \models \mathcal{A}$ and, moreover, for every ground terms s, t such that s is minimal in its congruence class, $s \succ t$ and $\mathcal{I} \not\models s = t$, then $\mathcal{A} \models s \neq t$.*

Actually most axiomatizations of section 5 are strongly normal.

Lemma 3 *Strongly normal axiomatizations are normal and hence are I-axiomatizations.*

Proof: This follows from the linearity of the ordering on ground terms. □

If we restrict now the conjecture superposition, allowing the overlaps on maximal sides of conjectures literals only, we get a new definition of induction derivation (let us call it *restricted induction derivation*) whose the classical inductive completion methods are an instance. And we still have the analog of theorem 1:

Theorem 3 *If $\mathcal{A}$ is a strongly normal axiomatization and $C_0, \ldots$ is a fair restricted induction derivation, then $\mathcal{I} \models C_0$ if and only if there is no clause in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.*

whose proof is only a slight modification of the proof of theorem 1.

3 Constructors

Normally, the structure of a specification with Herbrand model semantics can be seen as a set of constructor symbols $\mathcal{F}_0$ axiomatized by a set $\mathcal{E}_0$, to which (repeatedly) the complete definition of a new symbol has been added. For example, one can specify the natural numbers with constructors 0 and s, then define $+$ in terms of 0 and s, then define $*$ in terms of the $0, s, +$, then *power* in terms of $0, s, +, *$, etc.

More formally, along this section we assume the following setting. Let $\mathcal{F} = \cup_j \mathcal{F}_j$. Let $\mathcal{F}_0$ be a non-empty set of *constructor* symbols and let $\mathcal{E}_0$ be a saturated subset of $\mathcal{E}$ built over $\mathcal{T}(\mathcal{F}_0, \mathcal{X})$. Terms in $\mathcal{T}(\mathcal{F}_0, \mathcal{X})$ are called *constructor terms.* The constructors are called *free* if $\mathcal{E}_0$ is empty (or consists of only tautologies).

Symbols in $\mathcal{F} \setminus \mathcal{F}_0$ are called *defined* symbols, and will be denoted by $f_1, f_2, \ldots$. For $i > 0$, let $\mathcal{F}_i$ be $\mathcal{F}_{i-1} \cup \{f_i\}$. The ordering $\succ$ is defined according to this hierarchy, i.e., for ground s and t always $s[f_i(\ldots)] \succ t$ if $t \in \mathcal{T}(\mathcal{F}_j)$ with $i > j$. This is easily achieved in general-purpose orderings like LPO or RPO, for which it suffices to define the precedence on symbols as $f_{i+1} \succ f_i \succ g$ for all $i > 0$ and $g \in \mathcal{F}_0$.

In addition, we will assume that $\mathcal{E}$ is *sufficiently complete*, that is, for every ground term s there is some ground constructor term t such that $\mathcal{E} \models s = t$.

This specification method is very general and, at the same time, convenient for our proof techniques for three main reasons. First, if $\mathcal{A}$ is a normal I-axiomatization for $\mathcal{E}$, then it is one as well for the successive enrichments, since the normal terms are constructor terms; in the particular case of free constructors, $\mathcal{A}$ simply states that constructor terms are different. Second, by enriching in this way a saturated set $\mathcal{E}$ with the definition of a new symbol, usually the resulting set $\mathcal{E}'$ will be saturated as well. If this is not the case, we can still apply our techniques in many cases, as shown in Subsection 3.2. The third reason is explained in the next subsection.

3.1 Constructors and complete sets of positions

In this subsection, in addition to the hierarchical constructor-based setting, we assume $\mathcal{E}$ to be saturated. As said, for Theorem 1 to hold we need enough conjecture superposition inferences in order to reduce the smallest false conjecture instance. Since, by sufficient completeness, all ground terms headed by a defined symbol f must be reducible by R, we have the following result:

Lemma 4 *If p is an innermost occurrence of a defined symbol f in a conjecture c, then $P = \{p \cdot p' \mid p \cdot p'$ is a position of $c\}$ is a complete set of positions.*

Now suppose it is known that, for some defined symbol f, all terms of the form $f(t_1, \ldots, t_n)$ are reducible at the topmost position if the arguments t_i are constructor terms. Then the set $\{p\}$ on itself is already a complete set of positions if $c|_p$ is such a term $f(t_1, \ldots, t_n)$. In most specifications, this is indeed the case, since, in order to ensure sufficient completeness, the axioms defining f are usually precisely written like this. A simple particular case is:

Lemma 5 *In the case of free constructors, if p is an innermost occurrence of a defined symbol f in a conjecture c, then $\{p\}$ is a complete set of positions.*

Example 3 *If constructors are not free, $\{p\}$ needs not be complete. Let g, a, b be constructors axiomatized by the convergent TRS*

$$\{g(a) \to a, \;\; g(g(x)) \to g(x)\ \}$$

Let f be a completely defined by:

$$\{f(a) \to a,\ \ f(b) \to b,\ \ f(g(b)) \to b\}$$

Then some ground instances of $f(g(x))$ (by irreducible ground substitutions) are irreducible at topmost position. The root position alone is not inductively complete alone and, indeed, only overlapping at root position with the conjecture $f(g(x)) = b$ would yield to a tautology, whereas the conjecture is false ($f(g(a)) \to f(a) \to a$). □

In practice, for each defined symbol f it can be analyzed whether all $f(t_1, \ldots, t_n)$ with constructor arguments are reducible at the topmost position or not, and hence, whether $\{p\}$ is complete or the weaker result of Lemma 4 has to be applied.

In practical derivations, when and how is an innermost defined symbol of a conjecture c selected for determining a complete set of positions $P\Gamma$ A practical implementation includes a mechanism for insuring fairness, that eventually obliges every (apparently) persistent conjecture c to be considered for inference computation. Selection of P for a conjecture c can be done at the moment c is going to be considered for superposition or before. This can be done automatically, by some heuristic, or by user interaction.

3.2 Dropping saturatedness: reductive definitions

Sufficient completeness is an undecidable property even for finite convergent string rewrite systems [KNZ87]. For convergent TRS where $\succ$ fulfills the aforementioned requirements, this can be recovered: then the property of sufficient completeness is equivalent to the ground reducibility of $f(x_1, \ldots, x_n)$, where the x_i are pairwise distinct variables, for every defined symbol f [JK89].

But this result does not provide a means to effectively construct sufficiently complete specifications, and in cases when the result does not apply, like when $\mathcal{E}$ is not saturated, sufficient completeness has to be ensured in some other way.

The following is a standard method to do this, by axiomatizing each defined symbol f_i in such a way that all ground terms containing f_i are, in a certain general sense, reducible, but without any saturatedness requirement. Below we show that this more general notion of reducibility will suffice not only for obtaining sufficient completeness, but also for the applicability of our techniques for inductive theorem proving in non-saturated $\mathcal{E}$.

Definition 7 *Let $\mathcal{E}$ be a (possibly non saturated) hierarchical constructor-based specification where $\mathcal{E} = \cup_{i \geq 0} \mathcal{E}_i$ and $\mathcal{E}_i = \mathcal{E}_{i-1} \cup D_i$, where D_i for $i > 0$ is a set of Horn clauses.*

Furthermore, assume that for every ground term s of the form $f_i(t_1, \ldots, t_n)$ where $t_j \in \mathcal{T}(\mathcal{F}_0)$ for $j \in 1 \ldots n$, there is some clause in D_i with an instance $\Gamma \Rightarrow l = r$ such that $\mathcal{E}_{i-1} \models \Gamma, l = s$ and $s \succ \Gamma, r$.

Then $\mathcal{E}$ is called a reductive definition.

Lemma 6 *Every reductive definition $\mathcal{E}$ is sufficiently complete.*

Proof: By contradiction. Let u be the smallest term (w.r.t. $\succ$) in $\mathcal{T}(\mathcal{F}_i)$ such that $\mathcal{E}_i \models u = v$ for no v in $\mathcal{T}(\mathcal{F}_0)$. Let s be an innermost non constructor subterm of u. Then s is of the form $f_i(t_1, \dots, t_n)$, where the t_j are constructor terms. Hence there is some clause in D_i with an instance $\Gamma \Rightarrow l = r$ such that $\mathcal{E}_{i-1} \models \Gamma, l = s$ and $s \succ r$, and hence $u[s]_p \succ u[r]_p$. But then $\mathcal{E}_i \models s = r$, and hence $\mathcal{E}_i \models u[s]_p = u[r]_p$, contradicting the minimality of u. □

We believe that most practical cases of sufficiently complete non saturated specifications can be covered by this notion of reductive definition. For instance, consider the following example, borrowed from [KZ95]. It is interesting because for the authors it is supposed to illustrate the weakness of the proof by consistency approach.

Example 4

$$\mathcal{E} = \begin{cases} x + 0 &= x \\ s(x) + y &= s(x + y) \\ gcd(0, x) &= x \\ gcd(x, 0) &= x \\ gcd(x, x + y) &= gcd(x, y) \\ gcd(x + y, y) &= gcd(x, y) \end{cases}$$

This $\mathcal{E}$ is not saturated and cannot be turned into a finite saturated (or convergent) set of equations (even modulo the commutativity of $+$). Furthermore, note that the definition of gcd is clearly reductive: for every term s of the form $gcd(s^n(0), s^m(0))$, if $n = 0$ or $m = 0$ then s is equivalent to the smaller $s^n(0)$ or to $s^m(0)$; otherwise wlog. let $m = n + n'$; then $\mathcal{E} \models gcd(s^n(0), s^{n+n'}(0)) = gcd(s^n(0), s^n(0) + s^{n'}(0))$, which can be reduced by the fifth rule into $gcd(s^n(0), s^{n'}(0))$ wich is smaller w.r.t. $\succ$ than s. The definition of gcd is hence complete by the previous lemma. □

Indeed, since reductive definitions $\mathcal{E}$ are not saturated in general, our techniques as explained in Section 2 are not refutation complete anymore. For instance, in the previous example, we cannot disprove a false conjecture like $gcd(x, x) = 0$, since the only conjecture superposition inferences produce $0 = 0$, and $gcd(x, x) = 0$ can be consistent with a normal axiomatization $\mathcal{A}$.

The cause of this problem is that Theorem 1 requires every counter example to be reducible, which is not the case for $gcd(x, x) = 0$, whose minimal false instance $gcd(s(0), s(0)) = 0$ is not reducible by $\mathcal{E}$. However, a more careful analysis reveals that this problem does not appear if the false conjecture has some subterm of the form $gcd(x, y)$ where x and y are distinct variables. Then, for every instance $gcd(s, t)$ of it, another equivalent instance $gcd(s', t')$ will be reducible into some r such that $gcd(s, t) \succ r$, which suffices for Theorem 1. These ideas lead us to the following.

Definition 8 *A* definition pattern *is a term of the form $f_i(x_1, \dots, x_n)$ where f_i is a defined symbol and x_i and x_j are distinct variables for $1 \leq i, j \leq n$ where $i \neq j$.*

Lemma 7 *Let $\mathcal{E}$ be a reductive definition and let c be a conjecture such that $c|_p$ is a definition pattern.*

Then for every ground instance $c\sigma$ where σ is normal, there exists some inference by conjecture superposition at position p with a conclusion c', and a normal substitution σ' such that $\mathcal{I} \not\models c\sigma$ implies $\mathcal{I} \not\models c'\sigma'$, and furthermore, $c\sigma \succ c'\sigma'$.

Proof: Let $c\sigma$ be a clause $\Gamma' \Rightarrow \Delta', l$ and $l|_p$ be $f_i(x_1, \ldots, x_n) \equiv s$. By definition of a reductive definition, there is an instance $\Gamma \Rightarrow s\theta = r$ of a clause c_1 in D_i such that $\mathcal{E}_{i-1} \models \Gamma, s\theta = s\sigma$ and $s\sigma \succ \Gamma, r$. Since σ is normal, $s\theta \succeq s\sigma \succ r$, hence we can apply a conjecture superposition on c and c_1 yielding a clause c' which has the desired properties. □

Lemma 8 *Let $\mathcal{E}$ be a reductive definition and let c be a conjecture such that $c|_p$ is a definition pattern. Then $\{p\}$ is a complete set of positions.*

Proof: This is a consequence of lemma 5. □

As a consequence, in the remainder of this section we assume that every conjecture c has a selected position $p(c)$ (or simply p) such that the symbol at $c|_p$ is an innermost defined symbol.

Definition 9 *An induction derivation $C_1.C_2, \ldots$ is* reductively fair *if for every persistent conjecture c the term $c|_p$ is a definition pattern and every conjecture superposition on c at p is redundant in C_j for some j.*

Theorem 4 *Let $C_0.C_1, \ldots$ be a reductively fair induction derivation. Then $\mathcal{I} \models C_0$ if, and only if, there is no clause c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.*

Proof: We have again to replicate the proof of theorem 1 (or one of its extensions), without using the saturatedness property. This property was used in the implication $c\sigma$ not normal implies $c\sigma$ reducible by R, which we can no longer use. The purpose of this was to show that, for non-normal $c\sigma$ such that $\mathcal{I} \not\models c\sigma$, there is a conjecture superposition yielding a smaller false conjecture. However, this part can be replaced with the result of lemma 7 and the rest of the proof is the same. □

The previous theorem leaves us with the problem of how to achieve reductive fairness when a certain conjecture c appears such that c has no definition pattern and c cannot be proved redundant either. Hence this kind of derivation may *fail.*

In order to completely avoid failure, instead of requiring reductive fairness, we now slightly generalize the inference rule of conjecture superposition that handles persistent conjectures c without definition pattern in a different way. This is done by *abstracting out* some subterms of c, creating a logically equivalent conjecture c' that does have a definition pattern subterm:

Definition 10 *Let c be a clause and let $c|_q$ be a non-variable term t. Then the (logically equivalent) clause $x \neq t \vee c[x]_q$ is called a* variable abstraction *of c.*

Definition 11 *Let c be a conjecture and let p the selected innermost defined symbol in c. Let c' be the clause obtained from c by the smallest number of variable abstraction steps such that $c'|_p$ is a definition pattern. Furthermore, let d be a clause obtained by conjecture superposition on the position p in c'.*

Then we say that d can be obtained from c by an inference of conjecture superposition with abstraction.

Lemma 9 *Let $\mathcal{E}$ be a reductive definition. For every ground instance $c\sigma$ of a non constructor clause c, there exists some inference by conjecture superposition with abstraction whose conclusion d has an instance $d\sigma'$ such that $\mathcal{I} \not\models c\sigma$ implies $\mathcal{I} \not\models d\sigma'$, and moreover $c\sigma \succ d\sigma'$.*

Proof: Applying a variable abstraction, any clause can be replaced with a clause such that any innermost position of a term $f(t_1, \ldots, t_n)$ such that f is a defined symbol is a definition pattern. Then we use lemmas 7 and 8. □

Theorem 5 *Let $C_0.C_1, \ldots$ be a fair induction derivation with respect to conjecture superposition with abstraction. Then $\mathcal{I} \models C_0$ if, and only if, there is no clause c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.*

The proof is the same as for theorem 4, replacing lemma 7 with lemma 9.

Example 5 *Continuing with example 4, here we show the proof of commutativity of gcd by an experimental implementation in the* Saturate *system [NN93, GNN95]. Note that the commutativity of + is used as a lemma, and that indeed for all conjectures a definition pattern is selected.*

```
 1    : axiom   0+x1=x1
 2    : axiom   s(x1)+x2=s(x1+x2)
 3    : axiom   gcd(x1,0)=x1
 4    : axiom   gcd(0,x1)=x1
 5    : axiom   gcd(x1,x1+x2)=gcd(x1,x2)
 6    : axiom   gcd(x1+x2,x2)=gcd(x1,x2)
 7    : lemma   x1+x2=x2+x1
 8    : conje   gcd(x1,x2)=gcd(x2,x1)

The total LPO precedence is [gcd,+,s,0]

| ?- ind.

        Select innermost defined symbol in conje:
         8    : conje   gcd(x1,x2)=gcd(x2,x1)
        Type the number (left-to-right) of symbol: 1.

Inference by conjecture superposition of 3 on 8 gives:
 9    : conje   x1=gcd(0,x1)

Inference by conjecture superposition of 4 on 8 gives:
 10   : conje   x1=gcd(x1,0)
```

```
Inference by conjecture superposition of 5 on 8 gives:
 11   : conje   gcd(x1,x2)=gcd(x1+x2,x1)

Inference by conjecture superposition of 6 on 8 gives:
 12   : conje   gcd(x1,x2)=gcd(x2,x1+x2)

Clause 9: x1=gcd(0,x1) is redundant by instances:
         4    : axiom   gcd(0,x1)=x1
     clausal rewrite proof:
        by 4 we get x1=x1

Clause 10: x1=gcd(x1,0) is redundant by instances:
         3    : axiom   gcd(x1,0)=x1
     clausal rewrite proof:
        by 3 we get x1=x1

Clause 11: gcd(x1,x2)=gcd(x1+x2,x1) is redundant by instances:
 7    : lemma   x1+x2=x2+x1
 6    : axiom   gcd(x1+x2,x2)=gcd(x1,x2)
 8    : conje   gcd(x1,x2)=gcd(x2,x1)
     clausal rewrite proof:
        by 7 we get gcd(x1,x2)=gcd(x2+x1,x1)
        by 6 we get gcd(x2,x1)=gcd(x1,x2)
        by 8 we get gcd(x2,x1)=gcd(x2,x1)

Clause 12: gcd(x1,x2)=gcd(x2,x1+x2) is redundant by instances:
 7    : lemma   x1+x2=x2+x1
 5    : axiom   gcd(x1,x1+x2)=gcd(x1,x2)
 8    : conje   gcd(x1,x2)=gcd(x2,x1)
     clausal rewrite proof:
        by 7 we get gcd(x1,x2)=gcd(x2,x2+x1)
        by 5 we get gcd(x2,x1)=gcd(x1,x2)
        by 8 we get gcd(x2,x1)=gcd(x2,x1)

Induction derivation successfully terminated.
```

Example 6 *Again continuing with example 4, here we show the refutation of* $gcd(x, x) = 0$ *by* Saturate*. The input is the same as in the previous example, except that the conjecture, after abstraction for creating a definition pattern, is now:*

```
  8 :  conje x1=x2 -> gcd(x1,x2)=0

...

inconsistency detected:

 24   : conje  -> x1=0

Disproof:
input: 1    : axiom   0+x1=x1
input: 4    : axiom   gcd(0,x1)=x1
input: 6    : axiom   gcd(x1+x2,x2)=gcd(x1,x2)
input: 8    : conje   x1=x2 -> gcd(x1,x2)=0

conjecture superp. of 6 on 8
 12   : conje   x1+x2=x2 -> gcd(x1,x2)=0
```

```
conjecture superp. of 4 on 12
 21   : conje   0+x1=x1 -> x1=0

demodulation of 21 by 1
 24   : conje   x1=0
```

Example 7 *Without any lemmas, we now prove $gcd(x,x) = x$, which by abstraction becomes:* `7 :  conje x1=x2 -> gcd(x1,x2)=x1`

```
 1    : axiom 0+x1=x1
 2    : axiom s(x1)+x2=s(x1+x2)
 3    : axiom gcd(x1,0)=x1
 4    : axiom gcd(0,x1)=x1
 5    : axiom gcd(x1,x1+x2)=gcd(x1,x2)
 6    : axiom gcd(x1+x2,x2)=gcd(x1,x2)
 7    : conje x1=x2 -> gcd(x1,x2)=x1

...

Inference by conjecture superposition of 3 on 7 gives:
 8    : conje x1=0 -> x1=x1 (tautology)

Inference by conjecture superposition of 4 on 7 gives:
 9    : conje 0=x1 -> x1=0 (tautology)

Inference by conjecture superposition of 5 on 7 gives:
 10   : conje x1=x1+x2 -> gcd(x1,x2)=x1

Inference by conjecture superposition of 6 on 7 gives:
 11   : conje x1+x2=x2 -> gcd(x1,x2)=x1+x2

Inference by conjecture superposition of 1 on 10 gives:
 12   : conje 0=x1 -> gcd(0,x1)=0

Inference by conjecture superposition of 2 on 10 gives:
 13   : conje s(x1)=s(x1+x2) -> gcd(s(x1),x2)=s(x1)

clause 12 is demodulated by rules [4] giving
 14   : conje 0=x1 -> x1=0 (tautology)

Clause 13: s(x1)=s(x1+x2) -> gcd(s(x1),x2)=s(x1) is redundant by instances:
...

Inference by conjecture superposition of 1 on 11 gives:
 15   : conje x1=x1 -> gcd(0,x1)=0+x1

Inference by conjecture superposition of 2 on 11 gives:
 16   : conje s(x1+x2)=x2 -> gcd(s(x1),x2)=s(x1)+x2

clause 15 is demodulated by rules [1,4] giving
 17   : conje x1=x1 (tautology)

clause 16 is demodulated by rules [2] giving
```

```
 18   : conje s(x1+x2)=x2 -> gcd(s(x1),x2)=s(x1+x2)

Clause 18: s(x1+x2)=x2 -> gcd(s(x1),x2)=s(x1+x2) is redundant by instances:
...

Induction derivation successfully terminated.
```

The redundancy proofs of 13 and 18 are left out here, because they are bit long, although easy: for 13, instances $s(x1) + x2 = s(x1 + x2)$ *of 2 are used to enable the instance* $gcd(s(x1), s(x1) + x2) = gcd(s(x1), x2)$ *of 5 and then create* $s(x1) = s(x1+x2) \rightarrow gcd(s(x1), s(x1+x2)) = s(x1)$ *which is subsumed by 7. The redundancy proof of 18 is similar, using 6 instead of 5.*

4 Non-Horn Axioms

It is well-known that if $\mathcal{E}$ contains some non-Horn axiom, then in general no longer a unique minimal Herbrand model exists. For example, if $\mathcal{E} \equiv \{p \vee q\}$ then both the models $\{p\}$ and $\{q\}$ are minimal.

A total reduction ordering $\succ$ on ground literals provides a way to single out one of the minimal models, the so-called *perfect model* (of $\mathcal{E}$ and $\succ$). The prefect model is the minimal one with respect to the set extension $\succ_{set}^{-1}$ of $\succ^{-1}$. If $\mathcal{E} \equiv \{p \vee q\}$ then $\{p\} \succ_{set}^{-1} \{q\}$ and hence $\{q\}$ is the perfect model. (see [BG91] for details).

In logic programming, the ordering $\succ$ is usually induced from the way non-Horn clauses are written: one positive atom is written in the head of the clause, and the other ones are written negatively in the tail. For instance, $p \vee q$ can be written $p :- \neg q$ or $q :- \neg p$. Heads are made big in the ordering. If the resulting ordering is not contradictory, then the program has a perfect model: roughly, a logic program with negation $\mathcal{E}$ is called (locally) *stratified* if there is some ordering $\succ$ such that for all ground instances of clauses $A :- A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ it holds that $A \succeq A_i$ and $A \succ B_j$ for $i \in 1 \ldots n$ and $j \in 1 \ldots m$ [Prz88].

Local stratification is a too strong condition for the existence of a perfect model, and it has been relaxed into *weak stratification*, where, roughly, only ground instances contributing to the model need to fulfill the requirements [PP90]. These ideas are generalized and extended to arbitrary clausal specifications with equality in [BG91]. There it is shown that the perfect model of $\mathcal{E}$ is precisely $\mathcal{T}(\mathcal{F})/_{=R}$, where R is the TRS *generated* by the saturation of $\mathcal{E}$ with respect to $\succ$. Apart from the aforementioned inference rules of superposition and equality factoring, for non-Horn clauses rules for factoring and *merging paramodulation* are needed as well (see again [BG91] for details).

If we assume that $\mathcal{E}$ is saturated under this inference system, our techniques for inductive theorem proving for Horn $\mathcal{E}$ given in Section 2 smoothly extend to the non-Horn case and perfect model semantics.

We consider *normal* terms, clauses, substitutions, and I-axiomatizations with respect to the perfect model, and normality in this sense coincides again with irreducibility with respect to R, the ground TRS *generated* by the saturated set of axioms $\mathcal{E}$. The ordering restrictions for the inference rule for conjecture superposition are now slightly more complicated, since the left premise $D \vee l = r$ can now be non-Horn. In this case (again for some instance $\sigma\theta$, as in Section 4) one cannot require l to be the strictly maximal term anymore. Instead one can impose that $l \succ r$ and $l \succ u$ for all terms u occurring in negative equations in D, and $l \succeq l'$ for all other positive equations $l' = r'$, and if $l \equiv l'$ then $r \succ r'$.

Theorem 6 *Let $\mathcal{I}$ be the perfect model of a saturated set $\mathcal{E}$, let $\mathcal{A}$ be a normal I-axiomatization, and let C_0 be any set of universal conjectures. Let $C_0.C_1, \ldots$ be a fair induction derivation with respect to non-Horn conjecture superposition.*

Then $\mathcal{I} \models C_0$ if, and only if, there is no clause c in some C_j such that $\mathcal{A} \cup \{c\}$ is inconsistent.

The results of [GS92] are closely related to our previous theorem. In both approaches, validity is proved by finite saturation, in our case under conjecture superposition, in their case, after an encoding by `gnd` predicates, by a larger set of inference rules with selection. They refer to perfect model semantics, as we do here, and also for the first time they state the close relationship between redundancy (defined very similarly to the way it is done here) and inductive theorem proving.

But the methods are different in essence, however. We derive minimal counter examples with respect to normal I-axiomatizations, whereas in [GS92] counter examples are required to be ground: it is assumed that validity of ground clauses is decidable, and essentially the saturation with their encoding by `gnd` predicates amounts to an enumeration of the ground instances of the conjectures. Hence their method is more similar to the other ground instance enumeration methods [Red90, Zha88, BR95] we mentioned in Section 1, than to proof by consistency. It would be interesting to compare the behaviour of their restricted inference rules and refined abstract redundancy notions with the other enumeration methods, as well as all these methods with our techniques.

5 Computing I-axiomatizations

The work which has been done on proofs by consistency in equational theories can be reformulated in our framework: for instance, each of the procedures given in [Mus80, HH82, Bac88, JK89] is an inductive saturation procedure corresponding to some (implicit) I-axiomatization. We give here briefly the axiomatization corresponding to these known procedures and we sketch then some other possible automatic computations of (strongly normal) I-axiomatizations.

5.1 Some examples from the literature

5.1.1 Musser's approach

D. Musser in [Mus80] assumes that $\mathcal{E}$ is a convergent rewrite system such that there is a particular function symbol eq satisfying, for every terms t, t' not containing

$true, false, eq$: $eq(s,t) =_{\mathcal{E}} true \Leftrightarrow s =_{\mathcal{E}} t$ and $eq(s,t) =_{\mathcal{E}} false \Leftrightarrow s \neq_{\mathcal{E}} t$. Moreover, $true$ and $false$ are assumed to be irreducible.

This corresponds to a very simple I-axiomatization:

Lemma 10 *With Musser's assumptions $\{true \neq false\}$ is an I-axiomatization.*

Proof: $\mathcal{I} \models \mathcal{A}$ as $true \neq_{\mathcal{E}} false$ as soon as the initial algebra is not trivial. If $\mathcal{M} \models \mathcal{E} \cup \mathcal{A}$, then, for any two ground terms s, t, if $\mathcal{I} \not\models s = t$, then the normal forms of s and t are different. It follows that $\mathcal{E} \models eq(s,t) = false$, hence $\mathcal{M} \models eq(s,t) = false$. Since $\mathcal{M} \models true \neq false$, $\mathcal{M} \not\models eq(s,t) = true$, hence $\mathcal{M} \not\models s = t$.

For any two ground terms, we have $\mathcal{M} \models s = t$ iff $\mathcal{I} \models s = t$. If $\mathcal{M}$ is an Herbrand model, it is isomorphic to $\mathcal{I}$. □

This axiomatization is not normal. However, a strongly normal I-axiomatization can be obtained in an easy way. Consider

$$\mathcal{A} = \mathcal{D} \cup \{s \neq t \Leftrightarrow eq(s,t) = false\}$$

where $\mathcal{D}$ is the set of equations which defines eq.

Lemma 11 *With Musser's assumptions $\mathcal{A}$ is a (strongly normal) I-axiomatization.*

Proof: $\mathcal{I} \models \mathcal{A}$ follows from the assumptions. Ground terms that are minimal in their equivalence class are irreducible by convergence of $\mathcal{E}$. Now, if $s \succ t$ and s is an irreducible ground term, then $s \neq_{\mathcal{E}} t$, hence $eq(s,t) =_{\mathcal{D}} false$ by hypothesis. Then $\mathcal{A} \models eq(s,t) = false$, hence $\mathcal{A} \models s \neq t$. □

Note that the definition of eq needs not to be put inside $\mathcal{E}$ any more and that we do not need the property $eq(s,t) =_{\mathcal{E}} true \Leftrightarrow s =_{\mathcal{E}} t$. Actually, Musser's conditions imply that, roughly, there is an *equational* I-axiomatization.

5.1.2 Huet and Hullot's approach

G. Huet and J.-M. Hullot introduced [HH82] the constructor disciplines. This corresponds to the following axiomatization, as noticed by L. Fribourg [Fri84]:

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\ c(x_1, \ldots, x_n) = c(y_1, \ldots, y_n) \Rightarrow x_1 = y_1 \wedge \ldots \wedge x_n = y_n$$

For every constructor symbol c

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_m.\ c(x_1, \ldots, x_n) \neq c'(y_1, \ldots, y_m)$$

For every pair of distinct constructors c, c'

Huet and Hullot assume moreover that $\mathcal{E}$ is given by a convergent rewrite system and that pure constructors ground terms are smaller than terms which contain at least a defined symbol. (Besides, every definition of a symbol is supposed to be sufficiently complete with respect to the constructors).

Lemma 12 *Under these assumptions, the above axioms form a (strongly) normal I-axiomatization.*

Proof: if s, t are ground terms such that $s \succ t$ and s is minimal in its equivalence class. By convergence, s is irreducible, hence s is a constructor term. Moreover, t being smaller is also a constructor term. Since s, t are distinct, it follows that $\mathcal{A} \models s \neq t$. □

This extends in a straightforward way to reductive specifications:

Lemma 13 *If $\mathcal{E}$ is a reductive definition, then Huet and Hullot's axiomatization is strongly normal.*

Proof: This follows from lemma 6. □

5.1.3 Jouannaud and Kounalis approach

J.-P. Jouannaud and E. Kounalis introduced in [JK86] the notion of *ground reducibility*. The equation $s = t$ is inconsistent when $s \succ t$ and s is not ground reducible. (At this point, we use a lifting of the ordering to the non-ground level; this assumes another property of the ordering: its compatibility with substitutions). This corresponds to the following axiomatization:

$$\{\forall \vec{x}.\ s = t \Rightarrow Red(s) \mid s \succ t\}$$

where $\vec{x}$ is the set of variables occurring in s, t and Red, the *reducibility predicate* is defined by the following set of clauses:

$Red(l)$	For every left hand side l of a rule in $\mathcal{E}$
$Red(f(x_1, \ldots, x_n)) \Leftarrow Red(x_i)$	For every function symbol f and index i

$\mathcal{A}$ contains the definition of $\neg Red$, as it can be computed from Red using the method described in section 5.2. $\mathcal{A}$ consists actually of the definition of an automaton with disequality constraints recognizing the set of irreducible ground terms (see e.g. [CJ97]).

Example 8 *Consider the following very simple example of integers with addition: function symbols are $0, s, p, +$ and $\mathcal{E} = \{0 + x = x;\ \ s(x) + y = s(x + y),\ \ p(x) + y = p(x+y),\ \ s(p(x)) = x,\ \ p(s(x)) = x\}$. The computation of $\mathcal{A}$ consists of the following universally quantified axioms:*

$$\left\{\begin{array}{ll} s = t \Rightarrow Red(s) & \textit{For every terms } s, t \textit{ such that } s \succ t \\ \forall \vec{x}. \neg Red(x) \Leftrightarrow P(x) \vee N(x) & \\ P(0) & \\ N(0) & \\ P(s(x)) \Leftarrow P(x) & \\ N(p(x)) \Leftarrow N(x) & \end{array}\right.$$

Red is a recursive predicate on ground terms. This result is however technically difficult to prove and there are several proofs in the literature (from [Pla85] to [CJ97]). This implies the decidability of the consistency of an equation with $\mathcal{A}$.

Lemma 14 *When $\mathcal{E}$ is a finite ground convergent term rewriting system, the above set $\mathcal{A}$ is a strongly normal I-axiomatization.*

Proof: (sketch) $\mathcal{A}$ is infinite, but recursive, as the ordering on terms is assumed to be recursive.

$\mathcal{I} \models \mathcal{A}$ since, by ground convergence of $\mathcal{E}$, $s\sigma = t\sigma$ and $s \succ t$ implies that the normal forms of $s\sigma$ and $t\sigma$ are identical.

For any minimal ground term s in its equivalence class and every ground term t such that $s \succ t$, $\mathcal{A} \models \neg \mathit{Red}(s)$ (by ground convergence), hence, by definition, $\mathcal{A} \models s \neq t$. □

5.1.4 Bachmair's approach

L. Bachmair implicitly uses in [Bac91] a slightly more general definition (with the same assumptions on $\mathcal{E}$):

$$\mathcal{B} = \begin{cases} s = t \Rightarrow \mathit{Red}(s) \vee \mathit{Red}(t) \vee s \equiv t & \\ s = t \Rightarrow \mathit{Red}(s) & \text{For every terms } s, t \text{ such that } s \succ t \end{cases}$$

Note that the first axiom is not necessary a consequence of the second set of formulas when s, t are not comparable at the non ground level. Similarly, the second set of formulas is not a consequence of the first axiom since s may be both irreducible and strictly larger than a reducible term t.

Lemma 15 *$\mathcal{B}$ is a (strongly) normal I-axiomatization.*

Another slight extension would consist in merging the two cases, considering the more elegant axiomatization:

$$(s = t \wedge s > t) \Rightarrow \mathit{Red}(s)$$

of which all axioms in Bachmair's axiomatization would be a consequence. This is again a (strongly) normal I-axiomatization under the same hypotheses. The decidability of the consistency of such a new $\mathcal{A}$ with an equation is however an open question.

5.2 Domain closure and axiomatization computation

Now we want to design a general procedure for the computation of (strongly) (normal) I-axiomatizations. For this purpose, we start with a known procedure for Horn-clauses without equality which we borrow from the logic programming area.

The idea is to use a domain closure axiom to compute a set of Horn clauses which defines the negation of the predicates. Such a computation is valid in the least fixed point of the set of Horn clauses and is used in the "explicit negation as failure" [Stu91].

The following is known as *Clark's completion* [Cla78]: write every set of clauses whose head has a top predicate P as a single implication

$$P(x_1, \ldots, x_n) \Leftarrow \phi$$

where $x_1, \ldots, x_n$ are variables and ϕ is a disjunction of conjunctions of the form

$$\exists \vec{y}. x_1 \equiv t_1 \wedge \ldots \wedge x_n \equiv t_n \wedge B$$

B being the body of the clause whose head is $P(t_1, \ldots, t_n)$. In the least fixed point of the set of clauses, the converse implication holds, hence, in this model, we have

$$\neg P(x_1, \ldots, x_n) \Leftarrow \neg\phi$$

Now, we may use a quantifier elimination procedure for the theory of finite trees and get a definition of $\neg P$. (See e.g. [Com91] for more details on the quantifier elimination procedures). The only weakness is that, if some clauses contain variables in the body which do not appear in the head, then universal quantifiers cannot always be eliminated. However, if this is not the case, then the result is an I-axiomatization.

Lemma 16 *If we assume that all variables occurring in the body of the clauses also occur in the head of the clause, then the above procedure computes a strongly normal I-axiomatization for Horn clauses without equality.*

Proof: (sketch) $\mathcal{I} \models \mathcal{A}$ and $\mathcal{A}$ satisfies $\mathcal{E} \not\models P(t_1, \ldots, t_n) = true$ iff $\mathcal{A} \models P(t_1, \ldots, t_n) \neq true$. Since $true$ is the smallest term, this guarantees the strong normality property. □

Example 9 *Let us consider the most simple example of Horn clauses.*

$$\begin{array}{rcl} E(0) & & \\ E(s(s(x))) & \Leftarrow & E(x) \end{array}$$

Then we write

$$E(x) \Leftarrow (x \equiv 0) \vee (\exists y. x \equiv s(s(y)) \wedge E(y))$$

by negating both members we get

$$\neg E(x) \Leftarrow (x \not\equiv 0) \wedge (\forall y. x \not\equiv s(s(y)) \vee \neg E(y))$$

which gives after quantifier elimination:

$$\neg E(x) \Leftarrow (x \equiv s(0) \vee (\exists z. x \equiv s(s(z)) \wedge \neg E(z)))$$

and hence the I-axiomatization:

$$\mathcal{A} = \left\{ \begin{array}{rcl} \neg E(s(0)) & & \\ \neg E(s(s(x))) & \Leftarrow & \neg E(x) \end{array} \right.$$

Another simple example is the definition of the strict ordering on natural numbers:

Example 10

$$\begin{array}{rcl} s(x) > 0 & & \\ s(x) > s(y) & \Leftarrow & x > y \end{array}$$

whose axiomatization can be computed, yielding:

$$\mathcal{A} = \left\{ \begin{array}{rcl} 0 \not> x & & \\ s(x) \not> s(y) & \Leftarrow & x \not> y \end{array} \right.$$

Note that in the case of Horn clauses without equality, proof by consistency is actually a negation as failure rule.

5.3 Computing I-axiomatizations for arbitrary Horn clauses

For arbitrary Horn clauses with equality, the reducibility predicate can also be defined as before, we need however the equality:

$$\begin{array}{rcl} Red(s) & \Leftarrow & s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge s > t \\ & & \text{for every clause } s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n \vee s = t \\ Red(f(x_1, \ldots, x_n)) & \Leftarrow & Red(x_i) \end{array}$$

Using the method described in the previous section, we can compute an axiomatization (assuming that there is no variable occurring in the conditions which does not occur in the corresponding body). We get a finite set of clauses $\mathcal{S}_r$ of the form :

$$\begin{array}{l} \neg Red(s) \Leftarrow s'_1 \neq t'_1 \wedge \ldots s'_m \neq t'_m \wedge \neg Red(u_1) \wedge \ldots \wedge \neg Red(u_k) \\ \qquad\qquad\qquad \wedge v_1 > w_1 \wedge \ldots \wedge v_p > w_p \wedge x_1 \not\equiv t_1 \wedge \ldots \wedge x_q \not\equiv t_q \end{array}$$

using the linearity of the ordering on ground terms. Now, we get an automaton with semantic constraints. In the pure equational case, we get an automaton with disequality and inequality constraints. If, moreover, every equation can be oriented we are back to an automaton with disequality constraints. Finally, if $t \equiv t_1 \equiv \ldots \equiv t_n \equiv true$ we are back in the scope of the previous section.

Unfortunately, nothing guarantees in general that the semantic constraints can be eliminated. This is a problem: even if the theory in which the disequality constraints are interpreted is restricted to AC (associativity and commutativity), *Red* may be no longer recursive [KNRZ91]. Even if there are no disequality constraints $s'_i \neq t'_i$, *Red* is no longer recursive, as shown as a side result in [CNNR98]).

Lemma 17 *$\mathcal{S}_r \cup \mathcal{B} \cup \mathcal{E}$ is a strongly normal I-axiomatization.*

This lemma is not very useful in general as $\mathcal{E}$ is part of $\mathcal{A}$. However, if the disequalities $s'_i \neq t'_i$ are not present in $\mathcal{S}_r$, we do not need $\mathcal{E}$ in the above lemma. This is what happens in the next example.

Example 11 *This is a natural definition of finite sets (of, say, natural numbers, but it does not matter here). Function symbols include ins (binary), $\emptyset$ (constant) and $\in$ (binary). $\in$ and ins are defined by:*

$$\left\{\begin{array}{rcl} ins(x,l) = l & \Leftarrow & x \in l \\ x \in ins(y,l) & \Leftarrow & x \in l \\ x \in ins(x,l) & & \end{array}\right.$$

(the = true have been removed for the predicates definitions). Note that there is no finite set of free constructors and that there is no purely equational specification.

The first computation step gives a definition of the membership predicate and of the reducibility predicate for lists:

$$\left\{\begin{array}{rcl} x \in l & \Leftarrow & \exists y, l'.x \in l' \wedge l \equiv ins(y,l') \vee \exists l'.l \equiv ins(x,l') \\ Red(l) & \Leftarrow & \exists l'.l = ins(x,l') \wedge x \in l' \end{array}\right.$$

The second step negates both sides:

$$\left\{\begin{array}{rcl} x \notin l & \Leftarrow & (\forall y, l'.x \notin l' \vee l \not\equiv ins(y,l')) \wedge \forall l'.l \not\equiv ins(x,l') \\ \neg Red(l) & \Leftarrow & \forall l'.l \not\equiv ins(x,l') \vee x \notin l' \end{array}\right.$$

The last step is quantifier elimination, the patterns being inserted back in the body of the clause:

$$\mathcal{S}_r = \left\{\begin{array}{rcl} x \notin \emptyset & & \\ x \notin ins(y,l') & \Leftarrow & y \not\equiv x \wedge x \notin l' \\ \neg Red(\emptyset) & & \\ \neg Red(ins(x,l)) & \Leftarrow & \neg Red(l) \wedge x \notin l \end{array}\right.$$

And $\mathcal{A} = \mathcal{B} \cup \mathcal{S}_r$ is a normal I-axiomatization. ($\mathcal{E}$ is not needed in $\mathcal{A}$ as there is no (semantic) disequality constraint in $\mathcal{S}_r$).

6 Limitations

To our knowledge, the only (syntactically defined) classes of axiomatizations $\mathcal{E}$ with a decidable inductive validity problem are the *w-complete* ones, i.e., (for infinite I) the inductive theory coincides with the equational theory. This is the case for classes like the *shallow* equations of [CHJ94] or the *Catalog* Horn axiomatizations of [Nie96].

It seems to be *folk* knowledge that it is difficult to go beyond. In this section we very briefly give some insight for the reason why: even for (syntactically) very restricted and simple $\mathcal{E}$ the problem remains undecidable. We slightly adapt two results from [KNO90] (used there for showing the undecidability of *ground confluence*) in order to obtain the following:

Theorem 7 *The following problem is undecidable:*
INSTANCE: A finite, convergent, left- and right-linear (no variable occurs more

than once per side of a rule), right monadic (right hand sides have depth at most 1), constructor-based term rewrite system R and an equation $s = t$ over $\mathcal{T}(\mathcal{F}, \mathcal{X})$
PROBLEM: Is $s = t$ valid in $\mathcal{T}(\mathcal{F})/_{=R}$?

Proof: In [KNO90] for each instance $(\{u_1, \ldots, u_n\}, \{v_1, \ldots, v_n\})$ of the modified Post correspondence problem (MPCP), a rewrite system R (with the aforementioned properties) is shown to exist, such that, very roughly, all ground terms of the form $f(\ u_{i_1}(\ldots(u_{i_k}(a)\ldots),\ v_{i_1}(\ldots(v_{i_k}(a)\ldots)\ )$ for $i_j \in 1 \ldots n$ rewrite into $\top$, and all others into $-$ (the symbols of the MPCP become unary function symbols and $\top, -, a$ are constants).

Applying this construction to inductive validity: it is not difficult to see that $f(x, x) = -$ is valid in $\mathcal{T}(\mathcal{F})/_{=R}$ if, and only if, the instance of MPCP has no solution. □.

Theorem 8 *There exists a finite, length reducing, convergent (string) rewrite system R over $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $\mathcal{F}$ consists of only unary symbols (and hence there is only one variable at each side of each rule) such that, given some constant symbol a, the following inductive validity problem is undecidable:*
INSTANCE: an equation $s = t$ over $\mathcal{T}(\mathcal{F}, \mathcal{X})$
PROBLEM: Is $s = t$ valid in $\mathcal{T}(\mathcal{F} \cup \{a\})/_{=R}$?

Proof: The word TRS S given in [KNO90] has the required properties and has an undecidable *right equivalence problem*: given two words x and y, does $xw \leftrightarrow_S^* yw$ hold for all wΓ This amounts to an undecidable inductive validity problem. □

7 Conclusion

The method of proofs by consistency which was a success in the early 80s has long been forgotten. The main reasons were its requirements: pure equational specifications, ground convergent presentations,... Powerful saturation methods have been developed since and we believe that it was time to generalize the technique, taking advantage of the new developments. This was possible because of the generalization of ideas already present in [Fri84, KM87] for pure constructor systems: give explicitly a first-order axiomatization which reduces inductive proofs to proofs by consistency.

Besides this generalization (the introduction of I-axiomatizations) we have shown how to generalize the proofs by consistency methods to Horn clauses and arbitrary clauses, taking advantage of ordered strategies, we have also shown how to drop the ground convergence requirement for constructor specifications. Now we are seeking for more significant examples that could be automatized using the *Saturate* system.

Acknowledgements

We thank Laurent Fribourg for some interesting discussions.

References

[Bac88] Leo Bachmair. Proof by consistency in equational theories. In *Proceedings, Third Annual Symposium on Logic in Computer Science*, pages 228–233, Edinburgh, Scotland, 5–8 July 1988. IEEE Computer Society.

[Bac91] Leo Bachmair. *Canonical Equational Proofs.* Birkhäuser, Boston, 1991.

[BG91] Leo Bachmair and Harald Ganzinger. Perfect model semantics for logic programs with equality. In Koichi Furukawa, editor, *Logic Programming, Proceedings of the Eighth International Conference*, pages 645–659, Paris, France, June 24–28, 1991. The MIT Press.

[BG94] Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BGLS95] L. Bachmair, H. Ganzinger, Chr. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[BR95] Adel Bouhoula and Michael Rusinowitch. Implicit induction in conditional theories. *Journal of Automated Reasoning*, 14(2):189–235, 1995.

[CHJ94] Hubert Comon, Marianne Haberstrau, and Jean-Pierre Jouannaud. Syntacticness, Cycle-Syntacticness and Shallow Theories. *Information and Computation*, 111(1):154–191, 1994.

[CJ97] Hubert Comon and Florent Jacquemard. Ground reducibility is exptime-complete. In *Proc. IEEE Symp. on Logic in Computer Science*, Varsaw, June 1997. IEEE Comp. Soc. Press.

[Cla78] K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases.* Plenum, New York, 1978.

[CNNR98] Hubert Comon, Paliath Narendran, Robert Nieuwenhuis, and Michael Rusinowitch. Decision problems in ordered rewriting. In *Proc. IEEE Symp. Logic in Computer Science*, Indianapolis, 1998.

[Com91] Hubert Comon. Disunification: a survey. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson.* MIT Press, 1991.

[Fri84] L. Fribourg. A narrowing procedure for theories with constructors. In R. Shostak, editor, *Proc. 7th Int. Conf. on Automated Deduction*, volume 170 of *Lecture Notes in Computer Science*, pages 259–281, Napa, CA., 1984. Springer-Verlag.

[Fri86] Laurent Fribourg. A strong restriction of the inductive completion procedure. In Laurent Kott, editor, *Automata, Languages and Programming, 13th International Colloquium*, volume 226 of *Lecture Notes in Computer Science*, pages 105–115, Rennes, France, 15–19 July 1986. Springer-Verlag.

[GNN95] Harald Ganzinger, Robert Nieuwenhuis, and Pilar Nivela. The Saturate System, 1995. Software and documentation available: `http://www.mpi-sb.mpg.de/SATURATE/Saturate.html`.

[GS92] Harald Ganzinger and Jürgen Stuber. Inductive theorem proving by consistency for first-order clauses (extended abstract). In M[ichaël] Rusinowitch and J[ean-]L[uc] Rémy, editors, *The Third International Workshop on Conditional Term Rewriting Systems, Extended Abstracts*, pages 130–135, Pont-à-Mousson, France, July 8–10, 1992. Centre de Recherche en Informatique de Nancy and INRIA Lorraine.

[HH82] Gérard Huet and Jean-Marie Hullot. Proofs by induction in equational theories with constructors. *J. of Computer and System Sciences*, 25:239–266, 1982.

[JK86] Jean-Pierre Jouannaud and Emmanuel Kounalis. Automatic proofs by induction in equational theories without constructors. In *Proceedings, Symposium on Logic in Computer Science*, pages 358–366, Cambridge, Massachusetts, 16–18 June 1986. IEEE Computer Society.

[JK89] Jean-Pierre Jouannaud and Emmanuel Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82(1):1–33, July 1989.

[KM87] Deepak Kapur and D. Musser. Proof by consistency. *Artificial Intelligence*, 31(2), February 1987.

[KNO90] Deepak Kapur, Paliath Narendran, and Friedrich Otto. On ground confluence of term rewriting systems. *Inform. Comput.*, 86(1):14–31, 1990.

[KNZ87] Deepak Kapur, Paliath Narendran, and Hantao Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, 1987.

[KNRZ91] Deepak Kapur, Paliath Narendran, Daniel Rosenkrantz, and Hantao Zhang. Sufficient completeness, ground reducibility and their complexity. *Acta Informatica*, 28:311–350, 1991.

[Küc89] W. W. Küchlin. Inductive completion by ground proof transformation. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2 of *Rewriting Techniques*, chapter 7. Academic Press, New York, 1989.

[KZ95] Deepak Kapur and Hantao Zhang. An overview of Rewrite Rule Laboratory (RRL). *Journal of Mathematics and Computation*, 1995.

[Mus80] D. R. Musser. On proving inductive properties of abstract data types. In *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada*, pages 154–162. ACM, 1980.

[Nie96] Robert Nieuwenhuis. Basic paramodulation and decidable theories. In *Eleventh Annual IEEE Symposium on Logic in Computer Science*, pages 473–482, New Brunswick, New Jersey, USA, July 27–30, 1996. IEEE Computer Society Press.

[NN93] Pilar Nivela and Robert Nieuwenhuis. Practical results on the saturation of full first-order clauses: Experiments with the saturate system. (system description). In C. Kirchner, editor, *5th International Conference on Rewriting Techniques and Applications*, LNCS 690, Montreal, Canada, June 16–18, 1993. Springer-Verlag.

[NR95] Robert Nieuwenhuis and Albert Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *J. of Symbolic Computation*, 19(4):321–351, April 1995.

[Pla85] David Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65:182–215, 1985.

[PP90] Halina Przymusinska and Teodor Przymusinski. Weakly Stratified Logic Programs. *Fundamenta Informaticae*, XIII:51–65, 1990.

[Prz88] T. Przymusiński. On the declarative semantics of deductive databases and logic programs. In *Foundations of deductive databases and logic programming*, pages 193–216, Los Altos, CA., 1988. Morgan Kaufmann.

[Red90] Uday S. Reddy. Term rewriting induction. In Mark E. Stickel, editor, *10th International Conference on Automated Deduction*, LNAI 449, pages 162–177, Kaiserslautern, FRG, July 24–27, 1990. Springer-Verlag.

[Stu91] P. Stuckey. Constructive negation for constraint logic programming. In *Proc. 6th IEEE Symp. Logic in Computer Science, Amsterdam*, 1991.

[Zha88] H Zhang. *Reduction, Superposition, and Induction: Automated Reasoning in an Equational Logic.* PhD thesis, Renselaer Polytechnic Institute, 1988.