AVERROES Vérification de propriétés quantitatives et fonctionnelles
Analysis and VERification for the Reliability Of Embedded Systems

Lot 5.3

Technologie de vérification

Mécanisation de la déduction

Narrowing Based Inductive Proof Search

Description :	We present in this paper a narrowing-based proof search method for inductive theorems. It has the specificity to be grounded on deduction modulo and to yield a direct translation from a successful proof search derivation to a proof in the sequent calculus. The method is shown to be correct and refutationally complete in a proof theoretical way. **Keywords:** Noetherian induction, rule based induction, deduction modulo, proof assistant, term rewriting, proof terms.
Auteur(s) :	Claude Kirchner, Hélène Kirchner, Fabrice Nahon
Référence :	AVERROES / Lot 5.3 / Fourniture 5 / V1.1
Date :	mars 2006
Statut :	validé
Version :	1.1

Réseau National des Technologies Logicielles
Projet subventionné par le Ministère de la Recherche et des Nouvelles Technologies

CRIL Technology, France Télécom R&D, INRIA-Futurs, LaBRI (Univ. de Bordeaux – CNRS), LIX (Ecole Polytechnique, CNRS) LORIA, LRI (Univ. de Paris Sud – CNRS), LSV (ENS de Cachan – CNRS)

Introduction

Proof by induction is a main reasoning principle and is of prime interest in informatics. Typically in hardware and software verification problems, reasoning on complex data structures with infinite data or states make a prominent use of induction. Two main approaches have been developed for automated induction proof: explicit induction, used in proof assistants, and implicit induction by rewriting, used in automated theorem provers. This work was motivated by the need to have a better understanding of the relation between them. Thanks to the *deduction modulo* framework, explicit induction is applied to generate smaller instances of the property to be proved. These instances can then be used by the modulo part to implicitly simplify the goals, thanks to a sequent calculus modulo.

In this context, we provide a proof search mechanism for such inductive proofs. We show how the induction step can be performed by narrowing at innermost positions when the theory is axiomatized by a sufficiently complete and convergent rewrite system. This allows us to make precise the relationship between rewrite-based automated inductive theorem provers like Spike or RRL and case analysis in proof assistants like Coq or PVS.

We provide a proof theoretic foundation to the proof search procedure which is described by deduction rules that are proved valid in the sequent calculus modulo. This provides the ability to build a proof term for a proof assistant and therefore to be able to formally validate the proof search result. So, starting from the (inductive) proposition to be proved, the proof search mechanism builds a proof in the sequent calculus modulo, from which a proof term can be computed if needed.

This paper is built over the works and results on deduction modulo [DHK03], first-order presentation of higher-order logic [DHK01], formalization of induction in deduction modulo [Dep02, DK04] and on preliminary results on narrowing for induction presented in [DKKN03]. We provide first a summary of these approaches in Section 1 to motivate the main idea of narrowing based induction proof search. Section 2 introduces two basic ingredients of the method: ordering on equalities and narrowing with sufficiently complete rewrite systems. Then Section 3 presents the proof search system for inductive proofs, which is proved correct and refutationally complete.

For the main notations and classical results on term rewriting, we refer to the books on that topics like [BN98] or [KK99].

1 Deduction modulo and the Noetherian induction principle

Proofs by structural induction are of main use in proof assistants where the structural induction principle is generally automatically generated from the definition of the inductive data types. However, by using sophisticated termination orderings, proofs by Noetherian induction performed by rewriting are much

more expressive than structural induction. We recall in this section how deduction modulo can provide the description, at the proof theoretical level, of proof by Noetherian induction.

1.1 Deduction modulo

Let $\mathcal{T}(\Sigma, \mathcal{X})$ be the set of terms build over the signature Σ and the denumerable set of variables $\mathcal{X}$. We assume for simplicity Σ to be one-sorted, so that any term is of sort τ. Terms are denoted by letters s, t, u, v, l, r, variables by x, y, z, X, Y, Z, vectors of variables by $\overrightarrow{x}$, and substitutions on terms by Greek letters α, β, γ. $Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$ denotes the set of substitutions on $\mathcal{T}(\Sigma, \mathcal{X})$.

Provided a Noetherian relation R and a user defined theory Th_u, we are looking for a proof of a proposition P using a Noetherian induction principle denoted $NoethInd$, in the sense of finding a derivation of the sequent:

$$NoethInd(R), Th_u \vdash P$$

The Noetherian induction principle being by essence a second order proposition, this is indeed a sequent in higher-order logic.

Since we want to make a primarily use of first-order rewrite concepts and techniques and to consider first-order theories, we need a first-order presentation of higher-order logic. We use the so-called $HOL_{\lambda\sigma}$ introduced in [DHK01] which is based on deduction modulo [DHK03] and reveals to be particularly well-suited for our concerns. It is clearly out of the scope of this paper to explain in detail the full approach, and we only sketch here the main ideas. The reader can refer to [Dep02] and to [DK04] for a detailed exposition.

In deduction modulo, terms but also propositions can be identified modulo a congruence. We use a congruence that can typically be defined by conditional equalities and that takes into account the application context to evaluate the conditions. Furthermore, since the congruence application should be controlled closely, an appropriate notion of protective symbol is used, see [Dep02]: actually the congruence is not allowed to act below a protective symbol. In deduction modulo, the notions of term and proposition come from many-sorted first-order logic. We consider theories described by a set of axioms Γ and a congruence, denoted $\sim$, defined on terms and propositions. This congruence takes three arguments: the two objects to be compared and a set of axioms Γ called a local context. When we want to emphasize this, we denote the congruence $\sim^{\Gamma}$. The deduction rules of the sequent calculus take this equivalence into account. For instance, the right rule for the conjunction is not stated as usual

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

but is formulated

$$\frac{\Gamma \vdash_{\sim} A, \Delta \quad \Gamma \vdash_{\sim} B, \Delta}{\Gamma \vdash_{\sim} D, \Delta} \text{ if } D \sim^{\Gamma} A \wedge B.$$

We recall in Figure 1, the definition of the *sequent calculus modulo*. In these rules, Γ and Δ are finite multisets of propositions, P and Q denote propositions. Substituting the variable x by the term u in Q is denoted $Q\{u/x\}$. When the congruence $\sim$ is simply identity, this sequent calculus collapses to the usual one [GLT89]. In that case, sequents are written as usual with the $\vdash$ symbol.

$$\frac{}{\Gamma, P \vdash_\sim Q}\text{axiom} \quad \text{if } P \sim^\Gamma Q$$

$$\frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim \Delta}\text{cut} \quad \text{if } P \sim^\Gamma Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{contr-l} \quad \text{if } (\mathbf{A})$$

$$\frac{\Gamma \vdash_\sim Q_1, Q_2, \Delta}{\Gamma \vdash_\sim P, \Delta}\text{contr-r} \quad \text{if } (\mathbf{A})$$

$$\frac{\Gamma \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{weak-l}$$

$$\frac{\Gamma \vdash_\sim \Delta}{\Gamma \vdash_\sim P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\wedge\text{-l} \quad \text{if } R \sim^\Gamma (P \wedge Q)$$

$$\frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\wedge\text{-r} \quad \text{if } R \sim^\Gamma (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\vee\text{-l} \quad \text{if } (\mathbf{B})$$

$$\frac{\Gamma \vdash_\sim P, Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\vee\text{-r} \quad \text{if } (\mathbf{B})$$

$$\frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\Rightarrow\text{-l} \quad \text{if } (\mathbf{C})$$

$$\frac{\Gamma, P \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\Rightarrow\text{-r} \quad \text{if } (\mathbf{C})$$

$$\frac{\Gamma \vdash_\sim P, \Delta}{\Gamma, R \vdash_\sim \Delta}\neg\text{-l} \quad \text{if } R \sim^\Gamma \neg P$$

$$\frac{\Gamma, P \vdash_\sim \Delta}{\Gamma \vdash_\sim R, \Delta}\neg\text{-r} \quad \text{if } R \sim^\Gamma \neg P$$

$$\frac{}{\Gamma, P \vdash_\sim \Delta}\bot\text{-l} \quad \text{if } P \sim^\Gamma \bot$$

$$\frac{\Gamma, Q\{t/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, t)\ \forall\text{-l} \quad \text{if } (\mathbf{D})$$

$$\frac{\Gamma \vdash_\sim Q\{y/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, y)\ \forall\text{-r} \quad \text{if } (\mathbf{E})$$

$$\frac{\Gamma, Q\{y/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, y)\ \exists\text{-l} \quad \text{if } (\mathbf{F})$$

$$\frac{\Gamma \vdash_\sim Q\{t/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, t)\ \exists\text{-r} \quad \text{if } (\mathbf{G})$$

$\mathbf{A} = P \sim^\Gamma Q_1 \sim^\Gamma Q_2$, $\mathbf{B} = R \sim^\Gamma (P \vee Q)$ $\mathbf{C} = R \sim^\Gamma (P \Rightarrow Q)$, $\mathbf{D} = P \sim^\Gamma \forall x\ Q$, $\mathbf{E} = P \sim^\Gamma \forall x\ Q, y$ fresh variable, $\mathbf{F} = P \sim^\Gamma \exists x\ Q, y$ fresh variable, $\mathbf{G} = P \sim^\Gamma \exists x\ Q$

Figure 1: The sequent calculus modulo

Proof checking decidability for the sequent calculus modulo reduces to the decidability of the relation $\sim^\Gamma$, since we can check for each rule that the conditions of application are satisfied and we provide the needed information in the quantifier rules. When $\sim^\Gamma$ is not decidable, we still can use instances for which one can check the conditions of application, typically using a constraint based approach [Hue72, KKR90]

We can now introduce the fundamental notion of compatibility: a theory (a set of propositions) $\mathcal{T}$ is said to be compatible with a congruence $\sim$ when:

$$\mathcal{T}, \Gamma \vdash \Delta \text{ if and only if } \Gamma \vdash_\sim \Delta.$$

As shown in [Dep02, DK04], this property is modular: if $\mathcal{T}_1$ is compatible with a congruence C_1 and $\mathcal{T}_2$ is compatible with C_2 then $\mathcal{T}_1 \cup \mathcal{T}_2$ is compatible with $C_1 \cup C_2$.

Using the above equivalence, we can internalize propositions into the congruence, and we call this operation "push". We can also recover them at the level of the logic, and we call this operation "pop". Moreover, thanks to modularity, this can be done dynamically during the proof. This duality between computation and deduction is very conveniently reflected by the compatibility property. In [DHK03], internalization has been done statically and used to identify computation within the deduction process. Our aim here is to do internalization dynamically and to use it to design rules for induction by rewriting and an adequate strategy for Noetherian induction.

In what follows, we consider congruences generated by conditional class rewrite systems denoted $\mathcal{RE}$ and composed of (conditional) term rewrite rules, (conditional) term equational axioms, (conditional) proposition rewrite rules, (conditional) proposition equational axioms. Moreover, we assume that the left-hand side of a proposition rewrite rule and both sides of a proposition equational axiom have to be atomic propositions. Conditions may be arbitrary propositions. The variables in the right-hand side and condition of a rule must occur in the left-hand side. In the case of equational axioms, variables in both sides have to be the same and (free) variables in the condition have to be a subset of those.
We assume here that $\approx$ is a binary relation symbol which satisfies the axioms of equality (the classical denotation $=$ will only represent syntactical equality). In this case, to any conditional class rewrite system $\mathcal{RE}$ is associated the theory denoted $T_{\mathcal{RE}}$ as follows: for each conditional rewrite rule $(l \to r \text{ if } c)$ or conditional equality $(l \approx r \text{ if } c)$ in $\mathcal{RE}$, $T_{\mathcal{RE}}$ contains the proposition:

- $\forall \overline{x}(c \Rightarrow (l \Leftrightarrow r))$ when l and r are propositions,
- $\forall \overline{x}(c \Rightarrow (l \approx r))$ when l and r are terms,

where all free variables of l, denoted $\overline{x}$, are universally quantified.

It is proved in [Dep02] that $T_{\mathcal{RE}}$ is compatible with the congruence generated by $\mathcal{RE}$ (see also [Dow99] and [DHK03]). This allows us to freely use the "pushing and popping" operations. This also ensures that deduction modulo a congruence represented by a conditional class rewrite system is not a proper extension of first-order logic, but only a different presentation of it.

1.2 Deduction modulo for inductive proofs

This short introduction to deduction modulo now allows us to give a proof theoretic understanding of induction by rewriting. In the context of deduction modulo, the induction hypotheses arising from equational goals can be (dynamically) internalized into the congruence. When doing this, the computational part of the deduction modulo appears to perform induction by rewriting as done for instance by systems like Spike [BKR92] or RRL [KZ95].

The powerful principle of these approaches is to allow application of rewrite rules of the theory at any position of the current goal, as well as application of

induction hypotheses and current conjecture, provided that the applied formula is smaller in the Noetherian induction ordering than the current goal.

When the ordering contains the relation induced by a terminating rewrite system, a smaller formula is obtained as soon as a rewrite step is performed. Moreover, in Spike for instance, the choice of the induction variables and instantiation schemas is done using pre-calculated induction positions and schemas called test-sets. In the approach described below, we show how to use narrowing to automatically and completely perform these choices.

Given a property P and a relation R defined on a sort τ, the Noetherian induction principle $NoethInd(P, R, \tau)$ is defined as follows:

$$\forall x\,((x \in \tau \wedge \forall y\,((y \in \tau \wedge R(x, y)) \Rightarrow P(y))) \Rightarrow P(x)) \Rightarrow \forall x\,(x \in \tau \Rightarrow P(x))$$

and we write $Noeth(R, \tau)$ to state that R is a Noetherian relation over τ.

Proving that P inductively holds in a user theory Th_u, denoted $Th_u \models_{Ind} P$, amounts to derive the sequent:

$$\forall R \forall \tau\,(Noeth(R, \tau) \Rightarrow \forall P\, NoethInd(P, R, \tau)), Th_u \vdash P.$$

Of course to finish the proof, one should also provide a proof of $Noeth(R, \tau)$. To get a better intuition, let us consider an equational goal Q of the form $\forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x))$, the whole problem is formalized in $HOL_{\lambda\sigma}$. The remainder of this section gives the main steps which are detailed in [Dep02]. We start from the sequent:

$$\begin{array}{l}\forall R \forall \tau\,(Noeth(R, \tau) \Rightarrow \forall P\, NoethInd(P, R, \tau)), Th_u\\ \vdash\\ \forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x))\end{array}$$

In the following, we will denote NI the proposition:

$$\forall R \forall \tau\,(Noeth(R, \tau) \Rightarrow \forall P\, NoethInd(P, R, \tau))$$

Choosing a specific relation R (written $\prec$) and a type still denoted τ, we get:

$$Noeth(\prec, \tau) \Rightarrow \forall P\, NoethInd(P, \prec, \tau)), Th_u \vdash \forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x)).$$

From this, by the rule $\Rightarrow$-l of the sequent calculus, we get on one hand the sequent $Th_u \vdash Noeth(\prec, \tau)$ corresponding to the proof that $\prec$ is indeed Noetherian, on the other hand the sequent

$$\forall P\, NoethInd(P, \prec, \tau)), Th_u \vdash \forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x))$$

corresponding to the use of the induction principle to prove our goal.

We instantiate P as the equality to prove and we get:

$$\begin{array}{l}\forall x\,((x \in \tau \wedge \forall \underline{x}\,((\underline{x} \in \tau \wedge \underline{x} \prec x) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(x) \approx t_2(x))\\ \Rightarrow \forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x)), Th_u \vdash \forall x\,(x \in \tau \Rightarrow t_1(x) \approx t_2(x))\end{array}$$

where we have renamed y to $\underline{x}$ to emphasize that $\underline{x}$ is a smaller instance of x. A few easy steps of the sequent calculus later, we get:

$$Th_u \vdash \forall x\,((x \in \tau \wedge \forall \underline{x}\,((\underline{x} \in \tau \wedge \underline{x} \prec x) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(x) \approx t_2(x))$$

We then instantiate x by a fresh variable that we call X to emphasize this status, and we get:

$$Th_u \vdash (X \in \tau \wedge \forall \underline{x}\,((\underline{x} \in \tau \wedge \underline{x} \prec X) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(X) \approx t_2(X).$$

The $\Rightarrow$-r and $\wedge$-l rules of the sequent calculus lead to the discovery of the induction hypothesis:

$$Th_u, X \in \tau, \forall \underline{x}\,((\underline{x} \in \tau \wedge \underline{x} \prec X) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \vdash t_1(X) \approx t_2(X).$$

Using what we have seen on compatible theories, this hypothesis can now be internalized as a conditional equality denoted in general $\mathcal{RE}_{ind}(Q, \prec, \tau)(X)$:

$$t_1(\underline{x}) \approx t_2(\underline{x}) \text{ if } \underline{x} \in \tau \wedge \underline{x} \prec X \tag{1}$$

Note that because of its status of free fresh variable, X behaves like a constant, while $\underline{x}$ is universally quantified.

What is crucial in using the induction hypothesis (1) as an equality or a rewrite rule, is to check its condition. For any many-sorted theory, the $\underline{x} \in \tau$ part of the condition just expresses that the variable is sorted. More interestingly, the $\underline{x} \prec X$ condition is *always* satisfied provided the following hypotheses (called $\mathcal{H}$) are imposed:
(i) the theory Th_u can be oriented into a Noetherian rewrite system $\mathcal{R}$,
(ii) we choose for $\prec$ the reduction ordering induced by $\mathcal{R}$,
(iii) (1) is only applied on a subterm of the goal $t_1 \approx t_2$ or on a $\mathcal{R}$-reduced form of this goal.

Under these hypotheses, we are left to derive the sequent

$$Th_u, X \in \tau \vdash_{\mathcal{R}, t_1(\underline{x}) \approx t_2(\underline{x})} t_1(X) \approx t_2(X)$$

in the sequent calculus modulo. To be able to satisfy the (iii) part of the $\mathcal{H}$ hypotheses, we need in general to use the information that $X \in \tau$ in order to instantiate X by the free constructors of τ. This idea is exploited in the following to provide the proof search strategy. One of the main technical point handled in the paper is to justify that in most cases, the condition $\underline{x} \prec X$ is always satisfied when an induction hypothesis like (1) is internalized and used as a simplification rewrite rule.

2 Ordering and narrowing

Before describing the proof search system, we describe in this section the two main tools of the method, namely orderings on terms and equalities, and the narrowing properties in sufficiently complete rewrite systems. Most importantly, we provide the main result (Lemma 1) relating induction as deduction modulo as presented in the previous section and the Noetherian ordering induced by a terminating rewrite relation.

2.1 Orders and quasi-orders on terms and equalities

The set of positions in a term t is denoted $\mathcal{D}om(t)$, the subterm of t at position ω is denoted $t_{|\omega}$ and the symbol at position ω in t by $t(\omega)$. The notation $t[u]_\omega$ means that the term t contains the subterm u at position ω. These notations extend to goals $t_1 \approx t_2$ seen as a term with top symbol $\approx$ of arity 2. $\mathcal{V}ar(t)$ denotes the set of (free) variables of the term t and $|\mathcal{V}ar(t)|$ its cardinality. We define $\overrightarrow{\mathcal{V}ar(t)}$ as the vector of variables assumed linearly ordered by their name. These notations are extended to equalities, rewrite rules and goals.

From now on, we assume given a quasi simplification order $\leqslant$ on $\mathcal{T}(\Sigma, \mathcal{X})$ (see for example [DP01]). We denote $<$ its proper part, $\geqslant$ its associated equivalence (*i.e.* $\geqslant \;=\; (\leqslant \cap \geqslant)$) and $[t]$ the class of a term t for this equivalence. We assume that $<$ and $\geqslant$ are closed under substitutions and contexts. For instance, it is shown in [Fer95] that if $\leqslant$ is a recursive path ordering (rpo) with status then $<$ and $\geqslant$ are closed under substitutions and contexts.

In order to compare n-tuple of terms, for any natural n, we will use the standard extension on the Cartesian product $\leqslant_n$ of $\leqslant$:

$$\forall \overrightarrow{u}, \overrightarrow{v} \in \mathcal{T}(\Sigma, \mathcal{X})^n \quad \overrightarrow{u} \leqslant_n \overrightarrow{v} \Leftrightarrow (\forall i \; 1 \leq i \leq n \Rightarrow u_i \leqslant v_i)$$

If we denote $<_n$ the proper part of this quasi-order, then $<_n$ is Noetherian on the set $\mathcal{T}(\Sigma, \mathcal{X})^n$ provided $<$ is Noetherian.

Definition 1 Let Q and Q' be two equational goals, $Q' \leq_e Q$ whenever there exists a finite sequence of equalities $(Q_i = s_i \approx t_i)_{0 \leq i \leq n}$ such that:

1. $Q = Q_0$ and $Q' = Q_n$,
2. for any i, $s_{i+1} \leqslant s_i$ and $t_i = t_{i+1}$ or $t_{i+1} \leqslant t_i$ and $s_i = s_{i+1}$.

Now, since $\leqslant$ is stable under substitution, we get:

Lemma 1 $\leq_e$ is stable under substitution.

Moreover, to compare goals in a finer way, we also will make use of another ordering on goals similar to the one in [Dep02].

Definition 2 Let C be the following complexity measure on equalities:

$$C(s \approx t) \;=\; \begin{array}{ll} (\{\{[s]\}\}, \{\{[t]\}\}) & \text{if} \quad [t] < [s] \\ (\{\{[t]\}\}, \{\{[s]\}\}) & \text{if} \quad [s] < [t] \\ (\{\{[s], [t]\}\}, \emptyset) & \text{otherwise} \end{array}$$

We define a quasi ordering on equalities $\leqslant_e$ by

$$s \approx t \leqslant_e s' \approx t' \text{ if } C(s \approx t) \ll_{lex} C(s' \approx t') \text{ or } (s \geqslant s' \text{ and } t \geqslant t')$$

where $\ll_{lex}$ is the lexicographic extension of the multiset extension of $<$. We denote $<_e$ the proper part of $\leqslant_e$.

Let us remark that the order $<_e$ is well-suited for equalities, since it is invariant under symmetry of equality: for all $t, t', u, u' \in \mathcal{T}(\Sigma, \mathcal{X})$, we have: $t \approx t' <_e u \approx u'$ if and only if $t' \approx t <_e u \approx u'$ if and only if $t \approx t' <_e u' \approx u$. But it is not stable under substitution: for example with the substitution $\sigma = \{x \mapsto x_1,\ y \mapsto x_1,\ z \mapsto z_1\}$, we have:

1. $z \approx x + z <_e y \approx x + z$ since
   $C(z \approx x + z) = (\{\{[x + z]\}\}, \{\{[z]\}\})$ and
   $C(y \approx x + z) = (\{\{[x + z], [y]\}\}, \emptyset)$

2. but $z\sigma \approx x\sigma + z\sigma \not<_e y\sigma \approx x\sigma + z\sigma$ since
   $C(z\sigma \approx x\sigma + z\sigma) = (\{\{[x_1 + z_1]\}\}, \{\{[z_1]\}\})$ and
   $C(y\sigma \approx x\sigma + z\sigma) = (\{\{[x_1 + z_1]\}\}, \{\{[x_1]\}\})$

Notice the difference between $\leqslant_e$ and $\leq_e$, the latter being included in the former as it can be checked by a simple case analysis. Indeed, stability by substitution is in particular needed when considering optimized version of the proof search method developed in [KKN06].

2.2 Induction hypothesis and ordering on goals

Taking into account vectors of variables, we are now in position to instantiate the Noetherian induction hypothesis $\mathcal{RE}_{ind}(Q, \prec, \tau)(X)$ defined in Section 1.2.

For any equality Q, for any integer n such that $n = |\mathcal{V}ar(Q)|$, for any $\overrightarrow{x} \in \mathcal{X}^n$ such that $\overrightarrow{x}$ is the vector of variables of Q, we have:

$$\mathcal{RE}_{ind}(Q, <_n, \mathcal{T}(\Sigma)^n) \triangleq (\overrightarrow{x} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{x} <_n \overrightarrow{x}) \Rightarrow Q\{\overrightarrow{x}/\overrightarrow{x}\}$$

In order to simplify the notations, and when no confusion can occur, we denote it simply $\mathcal{RE}_{ind}(Q, <)$.

In the same way, we introduce the following notations, where σ is any substitution:

- $\mathcal{RE}_{ind}(Q, <)\sigma \triangleq (\overrightarrow{x} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{x} <_n \overrightarrow{x}\sigma) \Rightarrow Q\{\overrightarrow{x}/\overrightarrow{x}\}$
- $\mathcal{RE}_{ind}(Q, \leqslant) \triangleq (\overrightarrow{x} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{x} \leqslant_n \overrightarrow{x}) \Rightarrow Q\{\overrightarrow{x}/\overrightarrow{x}\}$
- $\mathcal{RE}_{ind}(Q, \leqslant)\sigma \triangleq (\overrightarrow{x} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{x} \leqslant_n \overrightarrow{x}\sigma) \Rightarrow Q\{\overrightarrow{x}/\overrightarrow{x}\}$

A crucial point in inductive proofs will be to compare different instances of a same equational goal: this is the purpose of the next proposition.

Lemma 2 For any equational goal Q with $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(Q)}$ and $n = |\overrightarrow{x}|$, for all substitutions $\sigma,\ \mu \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$, for all $t, t' \in \mathcal{T}(\Sigma, \mathcal{X})$:

1. If $t \leqslant t'$ then $Q[t]_\omega \leqslant_e Q[t']_\omega$
2. If $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$ then $Q\sigma \leqslant_e Q\mu$.
3. If $Q\sigma <_e Q\mu$ and $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$ then $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$.

Proof. 1. Let i and ω', such that $\omega = i.\omega'$. Since $t \leqslant t'$, and since $\leqslant$ is a reduction ordering, we have:

$$Q_{|i}[t]_{\omega'} \leqslant Q_{|i}[t']_{\omega'} \tag{2}$$

Now, one can easily check the following proposition:

$$\forall s\ \forall s'\ s \approx t \leqslant_e s' \approx t \tag{3}$$

And (2) and (3) above lead to $Q[t]_\omega \leqslant_e Q[t']_\omega$

2. is obtained from 1 by an easy induction based on the number of occurrences of the variables x_i in Q
3. Assume $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$ and $\overrightarrow{x}\sigma \not<_n \overrightarrow{x}\mu$. Then $\overrightarrow{x}\sigma \geqslant_n \overrightarrow{x}\mu$, hence $\overrightarrow{x}\mu \leqslant_n \overrightarrow{x}\sigma$, thus $Q\mu \leqslant_e Q\sigma$ by 2, and this contradicts the assumption $Q\sigma <_e Q\mu$.

□

In other words, for any equational goal Q, for any vector of variables $\overrightarrow{x}$ of Q in $\mathcal{X}^n$, and for all $\sigma, \mu \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$, in order to prove the proposition $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$, and whenever $Q\sigma <_e Q\mu$, it suffices to check all inequalities $\sigma(x_i) \leqslant \mu(x_i)$ for all component x_i of $\overrightarrow{x}$. Indeed, we are going to see in next Lemma that the inequality $Q\sigma <_e Q\mu$ can be automatically checked in many cases.

The next lemma relates the strict ordering $<_e$ on goals with a rewrite relation $\rightarrow$. It is a crucial step to justify the correct use of Noetherian rewriting as the main ingredient to perform Noetherian induction.

Indeed, under technical conditions that can be syntactically checked, this result ensures that $Q\sigma <_e Q\mu$. It is therefore possible in most of the cases to use an equational goal Q to reduce an instance of itself, $Q\mu$, as soon as a rewrite step has been previously performed on $Q\mu$.

Theorem 1 (Main compatibility theorem) *Let Q_1, Q_2, Q_3 and Q_4 be equational goals, $l \rightarrow r$ a rewrite rule (remember that then $l > r$), κ_0 be either a rewrite rule $l_{\kappa_0} \rightarrow r_{\kappa_0}$ or an equality $l_{\kappa_0} \approx r_{\kappa_0}$ Consider the inequality $I :\ (l_{\kappa_0} \approx r_{\kappa_0})\sigma <_e Q_1$ and assume:*

1. $Q_1 \rightarrow_{l\rightarrow r,\ j.\omega_j,\ \theta} Q_2$
2. $Q_2 \geq Q_3$
3. $Q_3 \rightarrow_{\kappa_0,\ i.\omega_i,\ \sigma} Q_4$
4. $Q_3 \geqslant_e Q_4$

Then:

1. *I is satisfied whenever $\omega_i \neq \varepsilon$ or $i = j$*
2. *If $\omega_i = \varepsilon$ and $i \neq j$:*

(a) If $l_{\kappa_0} > r_{\kappa_0}$*, then:*

$$I \Leftrightarrow ((Q_{1|i} \geqslant l_{\kappa_0}\sigma) \wedge (Q_{1|j} < Q_{1|i}) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma))$$

(b) If $l_{\kappa_0} \geqslant r_{\kappa_0}$*, then:*

$$I \Leftrightarrow ((Q_{1|i} \geqslant l_{\kappa_0}\sigma) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma))$$

(c) Otherwise:

$$I \Leftrightarrow (\quad ((Q_{1|i} \geqslant l_{\kappa_0}\sigma) \wedge ((Q_{1|j} < Q_{1|i}) \vee (l_{\kappa_0}\sigma \geqslant r_{\kappa_0}\sigma)) \wedge (r_{\kappa_0}\sigma \leqslant l_{\kappa_0}\sigma)) \Rightarrow (Q_{1|j} > r_{\kappa}\sigma) \ )$$

Proof. The proof of this crucial result is given in [KKN06]. It is based on a technical case analysis. □

A variant of this lemma is given in [Dep02] for an ordering between goals based on a complexity C using a set ordering instead of multiset ordering as here.

2.3 Narrowing

To make precise the use of narrowing in the induction process, let us first introduce a few concepts and notations. Narrowing will be performed only with rewrite rules, i.e. formulas $l \to r$ with $l > r$, but not with equalities. Let $\mathcal{R}$ be a rewrite system on $\mathcal{T}(\Sigma, \mathcal{X})$. The signature Σ is partitioned into a set of constructors $\mathcal{C}$ and a set of defined symbols $\mathcal{D}$. Constructors are function symbols which do not occur as a head symbol of a rule left-hand side. A constructor term is a term built only with constructor symbols. $\mathcal{T}(\mathcal{C}, \mathcal{X})$ denotes the set of constructor terms. A ground substitution is a substitution mapping each variable to a ground term, i.e. a term without variables. Let $Subst^{\mathcal{T}(\Sigma)}$ be the set of all ground substitutions on $\mathcal{T}(\Sigma)$. A rewrite system is said to be *ground convergent* if it is confluent and terminating over the set of ground terms. for any ground convergent rewrite system $\mathcal{R}$, for any term t, t is said to be ground $\mathcal{R}$-reducible if $t\alpha$ is $\mathcal{R}$-reducible for any ground substitution α. Furthermore, a symbol $f \in \mathcal{D}$ of arity n is *completely defined* if $f(t_1, \ldots, t_n)$ is reducible for all $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, and a ground convergent rewrite system $\mathcal{R}$ is said to be *sufficiently complete* if all symbols in $\mathcal{D}$ are completely defined.

For ground convergent and sufficiently complete rewrite systems, it is possible to specify particular positions in terms where reductions must apply, and where case analysis by rewriting can usefully be done.

Definition 3 For any $t \in \mathcal{T}(\Sigma, \mathcal{X})$, a position ω in t is called *defined-innermost*, and we denote $\omega \in DI(t)$, if $t(\omega) \in \mathcal{D}$ and $t(\omega') \in \mathcal{C} \cup \mathcal{X}$ whenever $\omega < \omega'$.

For example, considering Peano's naturals, 0 and s are constructors, $+$ is a defined symbol and in $s((0+0)+s(0+s(x)))$ the occurrence 1.2.1 is defined-innermost but 1 is not.

Lemma 3 For any ground convergent rewrite system $\mathcal{R}$, for any term t, and for any position $\omega \in \mathcal{D}om(t)$, if $t(\omega)$ is completely defined and ω is defined-innermost in $\mathcal{D}om(t)$, then, for any irreducible ground substitution α, $t\alpha$ is reducible at position ω.

Proof. Classical and by case analysis □

Definition 4 A goal Q is narrowed into Q' at a position ω with the rule $l \to r$ and the substitution σ, if σ is the most general unifier (mgu for short) of l and $Q_{|\omega}$, and $Q' = Q[r]_\omega\sigma$. This narrowing step is denoted $Q \leadsto_{l\to r,\omega,\sigma} Q'$.

Indeed, every defined-innermost occurrence is narrowable:

Corollary 1 For any ground convergent rewrite system $\mathcal{R}$, for any equational goal Q, for any defined-innermost position $\omega \in \mathcal{D}om(Q)$, for any ground substitution α and for any finite set V of variables such that $\mathcal{V}ar(Q) \cup \mathcal{D}om(\alpha\downarrow) \subseteq V$, there exists a rule $l \to r \in \mathcal{R}$, a unifier σ of $Q_{|\omega}$ and l, and a ground substitution μ such that $\sigma\mu_{|V} = (\alpha\downarrow)_{|V}$.

Proof. It is a consequence of the previous lemma and the classical narrowing lifting lemma [Hul80, KK99]. □

Thanks to these settings, we present in the next section, an induction based proof search system, relying on a main induction rule that uses narrowing to choose both the induction variables and the instantiation schema.

3 A proof search system for induction

The proof search system IndNarrow for inductive proofs introduced in this section is based on narrowing and rewriting. The main rule, called **Induce**, performs the induction step. This is the key point that provides a bridge between the implicit and explicit approaches of induction. Correctness and refutational completeness of this system are proved.

3.1 The proof search system IndNarrow

The rules in Figure 2 apply on sequents modulo of the form $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$, where Γ_1 is the deduction part of the definitions, $\mathcal{RE}_1$ is their computational part; Γ_2 is the deduction part for other statements, $\mathcal{RE}_2$ is their computational part; Q is an equational goal.

The distinction between $\Gamma_1/\mathcal{RE}_1$ and $\Gamma_2/\mathcal{RE}_2$ is needed because in the **Induce** rule, only $\mathcal{RE}_1$ is used for narrowing. For simplicity, we assume that $\mathcal{RE}_1$ contains only unconditional rules or equalities and we assume from now on, that $\mathcal{RE}_1$ is sufficiently complete.

Γ_2 is initialized with the proposition NI defined in subsection 1.2:

$$NI : \quad \forall R\ \forall \tau\ Noeth(R,\tau) \Rightarrow \forall P\ NoethInd(P,R,\tau)$$

Induce $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[t]_\omega \rightarrowtail$
$\bullet_{\substack{\kappa \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma \cup \mathcal{RE}_{ind}(Q,<)\sigma} (Q[r]_\omega)\sigma$
if $\kappa = l \rightarrow r$ and $\omega \in DI(Q)$

Orient $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \rightarrow r\}|\mathcal{RE}_2} Q$
if $\kappa = l \approx r$ or $\kappa = r \approx l$ and $l > r$

Push$_1$ $\Gamma_1, l \approx r|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\}|\mathcal{RE}_2} Q$

Push$_2$ $\Gamma_1|\Gamma_2, l \approx r \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{l \approx r\}} Q$

Rewrite$_1$ $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} Q[l\sigma]_\omega \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} Q[r\sigma]_\omega$
if $\kappa = l \rightarrow r$ or $\kappa = l \approx r$ or $\kappa = r \approx l$

Rewrite$_2$ $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\kappa\}} Q[l\sigma]_\omega \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\kappa\}} Q[r\sigma]_\omega$
if $\kappa = l \approx r$ or $\kappa = r \approx l$ or
$\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ or $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu$
and $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$ where $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(l \approx r)}$

Trivial $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} t \approx t \rightarrowtail \diamond$

Refutation $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail$ Refutation
when no other rules can be applied

Figure 2: The proof search system IndNarrow

and may contain other lemmas. $\mathcal{RE}_2$ receives the induction hypotheses provided by some application of the rule **Induce**. So $\mathcal{RE}_2$ may contain conditional equalities. Sequents are gathered in a multiset structure modeled with the $\bullet$ operator that is an AC operator on sequents with $\diamond$ as a neutral element.

The main rule is **Induce** as it performs the induction step. It uses narrowing to choose both the induction variable(s) and the instantiation schema. Narrowing is applied only at defined innermost positions (see Definition 3) $DI(Q)$ of the current goal Q. The other rules are doing the following: **Trivial** eliminates a trivial equation, **Push** pushes an equational hypothesis from the deduction part to the computational part, **Orient** orients an equation in the computational part into a rewrite rule, according to the term ordering, **Rewrite** (1 or 2) rewrites using a rule, an equation, or a smaller instance of a previous goal. **Push** and **Rewrite** are duplicated because of the $\Gamma_1/\mathcal{RE}_1$ and $\Gamma_2/\mathcal{RE}_2$ distinction.

3.2 A simple example

To get a better understanding of the way this set of rules is working, let us look at the proof of addition commutativity in Peano arithmetic. So, the goal is to prove:

$x + 0 \approx x, x + s(y) \approx s(x + y) | NI \vdash_{\emptyset|\emptyset} X + Y \approx Y + X$

Applying **Push$_1$** twice, we get:

$\emptyset | NI \vdash_{x+0\approx x, x+s(y)\approx s(x+y)|\emptyset} X + Y \approx Y + X$

Then, applying **Orient** twice gives us:

$\emptyset | NI \vdash_{x+0\to x, x+s(y)\to s(x+y)|\emptyset} X + Y \approx Y + X$

We can now apply **Induce** since $\mathcal{RE}_1 = \{x + 0 \to x, x + s(y) \to s(x + y)\}$ is confluent, terminating and sufficiently complete. This could be done at occurrence 1 or 2 of the goal. We arbitrary chose occurrence 1 and this leads us to prove the two sequents:

$\emptyset | NI \vdash_{\mathcal{RE}_1|\mathcal{RE}_{ind}(X+Y\approx Y+X, <, T^2_\Sigma)\{X\mapsto X_1; Y\mapsto 0\}} X_1 \approx 0 + X_1$

$\emptyset | NI \vdash_{\mathcal{RE}_1|\mathcal{RE}_{ind}(X+Y\approx Y+X, <, T^2_\Sigma)\{X\mapsto X_1; Y\mapsto s(Y_1)\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$

We have now to prove in particular that 0 is left-neutral. The only applicable rule on that goal is **Induce** again and we get the two new subgoals:

$\emptyset | NI \vdash_{\mathcal{RE}_1|} {}^{\mathcal{RE}_{ind}(X + Y \approx Y + X, <, T^2_\Sigma)\{X \mapsto 0; Y \mapsto 0\}}_{\mathcal{RE}_{ind}(X_1 \approx 0 + X_1, <, T_\Sigma)\{X_1 \mapsto 0\}} \quad 0 \approx 0$

$\emptyset | NI \vdash_{\mathcal{RE}_1|} {}^{\mathcal{RE}_{ind}(X + Y \approx Y + X, <, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}}_{\mathcal{RE}_{ind}(X_1 \approx 0 + X_1, <, T_\Sigma)\{X_1 \mapsto s(X_2)\}} \quad s(X_2) \approx s(0 + X_2)$

Trivial gets rid of the first one. **Rewrite$_2$** can be applied on the second one since, because of narrowing, the goal has been reduced and therefore the induction hypothesis can now be used. We get:

$\emptyset | NI \vdash_{\mathcal{RE}_1|} {}^{\mathcal{RE}_{ind}(X + Y \approx Y + X, <, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}}_{\mathcal{RE}_{ind}(X_1 \approx 0 + X_1, <, T_\Sigma)\{X_1 \mapsto s(X_2)\}} \quad s(X_2) \approx s(X_2 + 0)$

Applying now **Rewrite$_1$** proves that 0 is left-neutral for addition. We are left with the goal $s(X_1 + Y_1) \approx s(Y_1) + X_1$ and we will make precise later on how the proof search finishes.

3.3 Soundness of IndNarrow

Soundness amounts to show that for each rule of the proof search system IndNarrow of the form:

$$\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \bullet_{i\in I}\ \Gamma_1^i | \Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$$

then $\Gamma_1 | \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ is derivable provided all the $\Gamma_1^i | \Gamma_2^i, \overrightarrow{x^i} \in \mathcal{T}(\Sigma)^{n^i} \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$ are. In what follows, we assume that all variables in Γ are universally quantified.

Let us first state a few basic rules which are needed in the soundness proof.

Lemma 4 The following rules are derivable in the sequent calculus modulo:

1.
$$\frac{\Gamma \vdash_{\mathcal{RE}} P_1 \Rightarrow P_2, \Delta}{\Gamma, P_1 \vdash_{\mathcal{RE}} P_2, \Delta}\ imp$$

2.
$$\frac{}{\Gamma, x = y \vdash x \approx y}\ ref$$

3.
$$\frac{\Gamma \vdash_{\mathcal{RE}} \forall x\ \alpha(x) \approx \beta(x) \qquad \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta}\ r_e$$

4.
$$\frac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P, \Delta}{\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta}\ r_\alpha \quad \text{if} \begin{cases} \alpha \in Subst^\Sigma \\ \overrightarrow{x} \text{ is the vector of free variables of } \mathcal{RE} \cup P \end{cases}$$

5.
$$\frac{\bigwedge\limits_{\alpha \in Subst^\Sigma} \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P}\ r_{\overrightarrow{x}} \quad \text{if } \overrightarrow{x} \text{ is the vector of free variables of } \mathcal{RE} \cup P$$

6. For any proposition P and for any integer n, if $|\mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE})| = n$, if the proposition P is inductive in some context $\Gamma \cup \mathcal{RE}$ with respect to the order $<_n$, and if this order is Noetherian in this context, then the proposition P is valid in the context $\Gamma \cup \mathcal{RE}$, whenever it contains the proposition $NI = \forall R \forall \tau\, (Noeth(R, \tau) \Rightarrow \forall P\, NoethInd(P, R, \tau))$ (see subsection 3.1)

7.
$$\frac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P,<)} P \qquad \Gamma \vdash_{\mathcal{RE}} Noeth(<_n, \mathcal{T}(\Sigma)^n)}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P}\ r_I$$

if $\overrightarrow{x}$ is the vector of free variables of $\mathcal{RE} \cup P$

We are ready now to prove soundness of IndNarrow in the sequent calculus modulo by considering in turn each inference rule of IndNarrow.

Theorem 2 *For all contexts Γ_1, Γ_2, rewrite systems $\mathcal{RE}_1, \mathcal{RE}_2$, equational goal Q, occurrence $\omega \in DI(Q)$ and integer n, let us assume that:*

1. **Induce** *is applied on*
   $\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q[t]_\omega$
   to get
   - $\underset{\sigma = mgu(t,\, l)}{l \to r \in \mathcal{RE}_1}\ \Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2 \sigma \cup \mathcal{RE}_{ind}(Q,<)\sigma} (Q[r]_\omega)\sigma$;

2. $\mathcal{RE}_1$ *is ground convergent and sufficiently complete;*

3. $<$ *is Noetherian, so that* $\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Noeth(<_n, \mathcal{T}(\Sigma)^n)$;

4. *for any rewrite rule* $l \to r \in \mathcal{RE}_1$, *when* $\sigma = mgu(t, l)$ *and* $\overrightarrow{x}_\sigma \in \mathcal{X}^{n_\sigma}$ *is the vector of free variables of* $\mathcal{RE}\sigma \cup Q\sigma$, *the sequent*
$\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma$
is derivable in the sequent calculus modulo.

Then, the sequent
$\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q[t]_\omega$
is derivable in the sequent calculus modulo.

Proof. First, let us introduce the following notations:

$$\begin{array}{lll} \Gamma & \text{will denote} & \Gamma_1 \cup \Gamma_2 \\ \mathcal{RE} & \text{will denote} & \mathcal{RE}_1 \cup \mathcal{RE}_2 \\ \mathcal{RE}' & \text{will denote} & \mathcal{RE} \cup \{\mathcal{RE}_{ind}(Q,<)\} \\ \mathcal{RE}\sigma & \text{will denote} & \mathcal{RE}_1 \cup \mathcal{RE}_2\sigma \\ \mathcal{RE}'\sigma & \text{will denote} & \mathcal{RE}\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\}\sigma \end{array} \tag{4}$$

Let α be a ground substitution such that $Dom(\alpha) \subseteq V$ and $\alpha\downarrow$ be its $\mathcal{RE}_1$-normal form. According to the narrowing lemma, we have:

$$\sigma\mu_{|V} = (\alpha\downarrow)_{|V} \tag{5}$$

for some substitution μ. Let us consider the following derivations.

Π_1
$$\frac{}{\forall x\ x(\alpha\downarrow) = x\sigma\mu \vdash_{\mathcal{RE}} \forall x\ x\alpha = x\sigma\mu}\ Ax$$

Π_2
$$\frac{\vdash \forall x\ x(\alpha\downarrow) = x\sigma\mu \quad (by\ 5)}{\vdash_{\mathcal{RE}} \forall x\ x(\alpha\downarrow) = x\sigma\mu, \forall x\ x\alpha = x\sigma\mu}\ w + push$$

Π_3
$$\frac{\dfrac{\Pi_1 \quad \Pi_2}{\vdash_{\mathcal{RE}} \forall x\ x\alpha = x\sigma\mu}\ cut}{\Gamma \vdash_{\mathcal{RE}} \forall x\ x\alpha = x\sigma\mu, \forall x\ x\alpha \approx x\sigma\mu}\ w$$

Π_4:
$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, x\alpha = x\sigma\mu \vdash x\alpha \approx x\sigma\mu}\ ref}{\Gamma, \forall x\ x\alpha = x\sigma\mu \vdash x\alpha \approx x\sigma\mu}\ \forall - l}{\Gamma, \forall x\ x\alpha = x\sigma\mu \vdash \forall x\ x\alpha \approx x\sigma\mu}\ \forall - r}{\Gamma, \forall x\ x\alpha = x\sigma\mu \vdash_{\mathcal{RE}} \forall x\ x\alpha \approx x\sigma\mu}\ w + push$$

Π_5:
$$\frac{\Pi_3 \quad \Pi_4}{\Gamma \vdash_{\mathcal{RE}} \forall x\ x\alpha \approx x\sigma\mu}\ cut$$

Π_6:
$$\frac{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'\sigma} Q\sigma[r\sigma]_{|\omega}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'\sigma} Q\sigma, Q\sigma[r\sigma]_{|\omega}}\ w$$

Π_7

$$\dfrac{\dfrac{}{Q\sigma[r\sigma]_{|\omega} \vdash_{\mathcal{RE}_1} Q\sigma[l\sigma]_{|\omega}}\ Ax}{\dfrac{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n, Q\sigma[r\sigma]_{|\omega} \vdash_{\mathcal{RE}'\sigma} Q\sigma[l\sigma]_{|\omega}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n, Q\sigma[r\sigma]_{|\omega} \vdash_{\mathcal{RE}'\sigma} Q}\ w}$$

(since $l\sigma = t\sigma$ and $Q = Q[t]_{|\omega}$)
$\Pi_{1,\sigma}$:

$$\dfrac{\Pi_6 \qquad \Pi_7}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'\sigma} Q\sigma}\ cut$$

Denoting $\mathcal{PE}_{ind}(Q)$ the canonical proposition associated to $\mathcal{RE}_{ind}(Q, <)$, this leads to:
$\Pi_{2,\sigma}$

$$\dfrac{\dfrac{\Pi_{1,\sigma}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q)\sigma \vdash_{\mathcal{RE}\sigma} Q\sigma}\ pop}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}\sigma} \mathcal{PE}_{ind}(Q)\sigma \Rightarrow Q\sigma}\ \Rightarrow -r$$

Since the proposition $\mathcal{PE}_{ind}(Q)\sigma \Rightarrow_{\mathcal{RE}} Q\sigma$ is equivalent to $(\mathcal{PE}_{ind}(Q) \Rightarrow_{\mathcal{RE}} Q)\sigma$, we have:
$\Pi_{\sigma,\mu}$:

$$\dfrac{\Pi_{2,\sigma}}{\Gamma \vdash_{\mathcal{RE}\sigma\mu} (\mathcal{PE}_{ind}(Q) \Rightarrow Q)\sigma\mu}\ r_\mu$$

Π_α:

$$\dfrac{\Pi_{\sigma,\mu} \quad \Pi_5}{\Gamma \vdash_{\mathcal{RE}\alpha} (\mathcal{PE}_{ind}(Q) \Rightarrow Q)\alpha}\ r_e$$

And since α is any ground substitution, we have:
$\Pi_{\overrightarrow{x}}$:

$$\dfrac{\dfrac{\dfrac{\bigwedge\limits_{\alpha \in Subst^\Sigma} \Pi_\alpha}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} \mathcal{PE}_{ind}(Q) \Rightarrow Q}\ r_{\overrightarrow{x}}}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q) \vdash_{\mathcal{RE}} Q}\ imp}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(Q,<)} Q}\ push$$

Π:

$$\dfrac{\Pi_{\overrightarrow{x}} \qquad \Gamma \vdash_{\mathcal{RE}} Noeth(<_n, \mathcal{T}(\Sigma)^n)}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q}\ r_I$$

and we are done. □

Soundness of **Push** is simply a consequence of soundness of the sequent calculus modulo.

Let us now look at the **Rewrite** inferences.

Theorem 3 *For all contexts Γ_1, Γ_2, for all rewrite systems $\mathcal{RE}_1, \mathcal{RE}_2$, for any equational goal Q, let us assume that:*

1. **Rewrite** *is applied on*
   $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$
   to get:
   $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q'$
2. *The sequent* $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} (Q[r\sigma]_\omega)$ *admits a proof.*

Then, the sequent $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[l\sigma]_\omega$ *is derivable in the sequent calculus modulo.*

Proof. Let us use the same notations as in the previous theorem. By assumption 1, and by definition of the rule **Rewrite**, there exist $\kappa \in \mathcal{RE}$, $(l, r) \in \mathcal{T}(\Sigma, \mathcal{X})^2$, $\omega \in \mathcal{D}om(Q)$, and $\sigma \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$, such that:

- ($\kappa = l \to r$ or $\kappa = l \approx r$ or $\kappa = r \approx l$ or $\exists \mu\ (\mu \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})})$ and $(\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ or $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu))$
- $Q = Q[l\sigma]_{|\omega}$
- $Q' = Q[r\sigma]_{|\omega}$

Now, let us consider the following derivations:
Π_1:

$$\dfrac{\dfrac{}{Q[r\sigma]_\omega \vdash_{\{\kappa\}} Q[l\sigma]_\omega}\ Ax}{\Gamma, Q[r\sigma]_\omega \vdash_{\mathcal{RE}} Q[l\sigma]_\omega}\ w + push$$

Π_2:

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} Q[r\sigma]_\omega \quad \text{(assumed)}}{\Gamma \vdash_{\mathcal{RE}} Q[r\sigma]_\omega, Q[l\sigma]_\omega}\ w$$

Π:

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma \vdash_{\mathcal{RE}} Q[l\sigma]_\omega}\ cut$$

which concludes the proof. □

We have already proved soundness of the rewrite system IndNarrow \ {**Orient**}. Now, it is easy to see that, for all contexts Γ, Γ', for all rewrite systems $\mathcal{RE}$, $\mathcal{RE}'$, and for all equational goals Q, Q', one can build a derivation $\Gamma \vdash_{\mathcal{RE}} Q \overset{*}{\rightarrowtail}_{\mathsf{IndNarrow}\setminus\{\mathbf{Orient}\}} \Gamma' \vdash_{\mathcal{RE}'} Q'$ whenever there exists a derivation $\Gamma \vdash_{\mathcal{RE}} Q \overset{*}{\rightarrowtail}_{\mathsf{IndNarrow}} \Gamma' \vdash_{\mathcal{RE}'} Q'$. Therefore, soundness of IndNarrow is a consequence of soundness of IndNarrow \ {**Orient**}.

3.4 Example (continued)

Remember that we need to prove:

$\emptyset | NI \vdash_{\mathcal{RE}_1 | \mathcal{RE}_{ind}(X+Y \approx Y+X, <, T^2_\Sigma)\{X \mapsto X_1; Y \mapsto s(Y_1)\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$

We can apply **Induce** at position 1.1, leading to:

$\emptyset | NI \vdash_{\mathcal{RE}_1 | \mathcal{RE}'_2} s(0 + Y_3) \approx s(Y_3)$

$\emptyset | NI \vdash_{\mathcal{RE}_1 |} {}_{\substack{\mathcal{RE}_{ind}(X+Y \approx Y+X, <, T^2_\Sigma)\sigma_1 \\ \mathcal{RE}_{ind}(s(X_1+Y_1) \approx s(Y_1)+X_1, <, T^2_\Sigma)\sigma_2}} \quad s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$

where $\mathcal{RE}'_2$ is easy to explicit and $\sigma_1 = \{X_1 \mapsto 0, Y_1 \mapsto Y_3\}$, $\sigma_2 = \{X \mapsto 0; Y \mapsto s(Y_3)\}$ In the same way as before, the goal $s(0+Y_3) \approx s(Y_3)$ is solved. Reducing with the **Rewrite** rules and using Theorem 1 to check the conditions leads directly to the proof of the last goal, therefore finishing the proof.

Notice that, following the soundness proof above, the proof search developed in the example can be straightforwardly expanded into a sequent calculus proof.

3.5 Refutational correctness

Refutational correctness amounts to show that for each rule of the proof search system IndNarrow of the form:

$$\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q \rightarrowtail \bullet_{i \in I} \ \Gamma_1^i | \Gamma_2^i \vdash_{\mathcal{RE}_1^i | \mathcal{RE}_2^i} Q^i$$

then all the $\Gamma_1^i | \Gamma_2^i, \overrightarrow{x^i} \in \mathcal{T}(\Sigma)^{n^i} \vdash_{\mathcal{RE}_1^i | \mathcal{RE}_2^i} Q^i$ are derivable provided $\Gamma_1 | \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q$ is.

We detail here the most delicate point which is again the case of the rule **Induce**, addressed in the following theorem.

Theorem 4 *For all contexts Γ_1, Γ_2, for all rewrite systems $\mathcal{RE}_1$, $\mathcal{RE}_2$, for any equational goal Q, for any $\omega \in DI(Q)$, and for any integer n,*

If

$$\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q[t]_\omega \quad \overset{Induce}{\rightarrowtail} \quad \bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}} \ \Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2 \sigma \cup \mathcal{RE}_{ind}(Q, <)\sigma} Q[r]_w \sigma$$

and if the sequent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q[t]_\omega$ (where $\overrightarrow{x} \in \mathcal{X}^n$ denotes the vector of free variables of $\mathcal{RE}_2 \cup Q$) admits a proof in sequent calculus modulo,
then, for any $\sigma = mgu(t,l)$, for any integer n_σ, for any vector of free variables $\overrightarrow{x}_\sigma$ of $\mathcal{RE}\sigma \cup Q\sigma$ in $\mathcal{X}^{n_\sigma}$, one can build a proof of

$$\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma \cup \{\mathcal{RE}_{ind}(Q, <)\sigma\}} Q[r]_\omega \sigma$$

Proof. Recall the notations 4. Let $\sigma = mgu(t,l)$. For any ground substitution μ, we have:

$\Pi_{\sigma,\mu}$

$$\frac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q}{\Gamma \vdash_{\mathcal{RE}\sigma\mu} Q\sigma\mu} \ r_{\sigma\mu}$$

Now, let us consider the following derivations:

$\Pi_{1,\sigma}$:

$$\frac{\bigwedge\limits_{\mu \in Subst^{\Sigma}} \Pi_{\sigma,\mu}}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} Q\sigma} \; r_X$$

$\Pi_{2,\sigma}$:

$$\frac{\Pi_{1,\sigma}}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} Q\sigma, Q\sigma[r\sigma]_{|\omega}} \; w$$

Denoting $Th_{\mathcal{RE}_2\sigma}$ the canonical theory associated to $\mathcal{RE}_2\sigma$, and since $\mathcal{RE}\sigma = \mathcal{RE}_1 \cup \mathcal{RE}_2\sigma$, we obtain:

$\Pi_{3,\sigma}$:

$$\dfrac{\dfrac{\dfrac{}{Q\sigma \vdash_{\mathcal{RE}_1} Q\sigma[r\sigma]_{|\omega}} \; Ax}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma}, Q\sigma, Th_{\mathcal{RE}_2\sigma} \vdash_{\mathcal{RE}_1} Q\sigma[r\sigma]_{|\omega}} \; w}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma}, Q\sigma \vdash_{\mathcal{RE}\sigma} Q\sigma[r\sigma]_{|\omega}} \; push$$

Denoting $\mathcal{PE}_{ind}(Q)$ the canonical proposition associated to $\mathcal{RE}_{ind}(Q, <)$, this leads to:

$\Pi_{4,\sigma}$:

$$\dfrac{\dfrac{\dfrac{\Pi_{2,\sigma} \quad \Pi_{3,\sigma}}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} Q\sigma[r\sigma]_{|\omega}} \; cut}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma}, \mathcal{PE}_{ind}(Q)\sigma \vdash_{\mathcal{RE}\sigma} (Q\sigma[r\sigma]_{|\omega})} \; w}{\Gamma, \overrightarrow{x}_{\sigma} \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma \cup \mathcal{RE}_{ind}(Q)\sigma} (Q\sigma[r\sigma]_{|\omega})} \; push$$

and we are done. □

As a corollary of Theorem 4, we get:

Theorem 5 *The proof search system* IndNarrow *is refutationally correct.*

Proof. **Induce** being handled in Theorem 4, the other inference rules **Rewrite** and **Orient** are proved refutationally correct, in similar ways. Correctness of the other rules is a consequence of correctness of deduction modulo. □

3.6 Refutational completeness

Refutational completeness is achieved thanks to the **Refutation** rule which applies when no other rule of IndNarrow can be applied.

Lemma 5 For all contexts $\Gamma_1,\ \Gamma_2$, for all rewrite systems $\mathcal{RE}_1,\ \mathcal{RE}_2$, if:

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \text{Refutation}$$

then, the sequent $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ has no proof.

Proof. If Q contains a defined symbol, there exists a defined-innermost position in $Dom(Q)$, therefore one can apply the rule $Induce$, and there is a contradiction. Since the rule $Trivial$ cannot be applied either, we have $Q = t \approx t'$, with $t,\ t'$ constructor terms that are not syntactically equal. Therefore, the sequent $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ has no proof, since the constructors are assumed to be free and $\approx$ satisfies the axioms of equality. □

Lemma 6 For all contexts $\Gamma_1,\ \Gamma_2$, for all rewrite systems $\mathcal{RE}_1,\ \mathcal{RE}_2$, if there exists an IndNarrow-derivation

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \overset{*}{\rightarrowtail}_{\mathsf{IndNarrow}} \text{Refutation}$$

then, the sequent $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ has no proof.

Proof. Assume: $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \overset{*}{\rightarrowtail}_{IndNarrow}$ Refutation
There exist contexts $\Gamma'_1,\ \Gamma'_2$, rewrite systems $\mathcal{RE}'_1,\ \mathcal{RE}'_2$ and an equational goal Q' such that:

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \overset{*}{\rightarrowtail}_{\mathsf{IndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q' \rightarrowtail_{\mathsf{IndNarrow}} \text{Refutation}$$

And, by lemma 5, the sequent $\Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'$ has no proof. Therefore, by the refutation correctness of IndNarrow (Theorem 5), $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ has no proof either. □

As a corollary, we get:

Theorem 6 *The proof search system IndNarrow is refutationally complete.*

4 Conclusion

We have shown how narrowing can provide the inference mechanism to perform induce proof search. Instead of pre-computing induction schemata and induction variables, this has the advantages to target exactly which variables should be instantiated and how. Moreover, because the method derives directly from the deduction modulo framework, we take benefit from a direct translation from a successful proof search derivation to a sequent calculus modulo proof. Last but not least, the fact that we are precisely specifying the conditions on the induction ordering allows us to refine, in the full paper [KKN06], the proof search inference rules and therefore to narrow the search space.

At the proof level, the general framework of deduction modulo is quite relevant to keep at the deduction level only the true deduction steps like modus

ponens and to delegate all computational steps on propositions or terms to specialized provers using equational and rewriting techniques. Then, some parts of the proofs can be deferred to aside computations, while the true skeleton of the proof is being built. At the checking level, the experiences described in [DKKN03] of translating equational and inductive proofs to proof terms for Coq should be quite useful.

If the approach is theoretically fruitful and enlightens the relationship between rewrite based induction methods and Noetherian induction, we are clearly in need of an implementation of the results presented here. Our goal will be to achieve this as a way to mechanize proof search in a proof assistant based on type theory and the rewriting calculus [BCKL03, Wac05]. Moreover our approach provides the ability to use an induction principle based on Noetherian rewrite systems, therefore strongly enhancing over the structural induction principle which is, in practice, used in most of the current proof assistants.

This narrowing based approach opens also new fundamental questions, let us mention three of them. The first one concerns its relationship with the very useful rippling [BBHI05] technique. Indeed, in a way related to rippling, narrowing makes explicit and links with a Noetherian rewrite system what we are in need for inductively proving a goal. This analogy should be deepened and possibly exploited. A second one, that we are currently investigating, concerns the extension of rewrite based inductive theorem proving to class rewriting. This has been explored in particular in [BBR95] for associative-commutative theories. The genericity of narrowing modulo may enlighten and ease the use of such class rewrite systems to base inductive proof search. The third one concerns inductive proof by consistency which is indeed at the source of the use of rewrite techniques for induction [Mus80, GS92, CN98, Ste]. The relationship between deduction modulo and such a consistency technique is worth to be better understood.

Acknowledgments: Many thanks to the members of the Protheo team for stimulating discussions on many of the subjects developed in this paper, and most particularly to Eric Deplagne whose PhD thesis is the initial work on which this paper is based.

References

[BBHI05] A. Bundy, D. Basin, D. Hutter, and A. Ireland. *Rippling: Meta-Level Guidance for Mathematical Reasoning*. Cambridge University Press, 2005. 22

[BBR95] N. Berregeb, A. Bouhoula, and M. Rusinowitch. Extending SPIKE to associative and commutative theories. In *Seminar on Automation of proof by induction*, Dagstuhl seminar, Germany, July 1995. 22

[BCKL03] G. Barthe, H. Cirstea, C. Kirchner, and L. Liquori. Pure Patterns Type Systems. In *Principles of Programming Languages - POPL2003, New Orleans, USA*. ACM, January 2003. 22

[BKR92] A. Bouhoula, E. Kounalis, and M. Rusinowitch. Spike: An automatic theorem prover. In *Proceedings of the 1st International Conference on Logic Programming and Automated Reasoning, St. Petersburg (Russia)*, volume 624 of *Lecture Notes in Artificial Intelligence*, pages 460–462, July 1992. 5

[BN98] F. Baader and T. Nipkow. *Term Rewriting and all That*. Cambridge University Press, 1998. 2

[CN98] H. Comon and R. Nieuwenhuis. Induction = i-axiomatization + first-order consistency. Research report LSV-98-9, LSV, October 1998. To appear in Information and Computation, special issue on RTA'98. 22

[Dep02] E. Deplagne. *Système de preuve modulo récurrence*. Thèse de doctorat, Université Nancy 1, November 2002. 2, 3, 4, 5, 6, 8, 11

[DHK01] G. Dowek, T. Hardin, and C. Kirchner. HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1):21–45, 2001. 2, 3

[DHK03] G. Dowek, T. Hardin, and C. Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, Nov 2003. 2, 3, 5

[DK04] E. Deplagne and C. Kirchner. Induction as deduction modulo. Rapport de recherche, LORIA, Nov 2004. 2, 3, 4

[DKKN03] E. Deplagne, C. Kirchner, H. Kirchner, and Q.-H. Nguyen. Proof search and proof check for equational and inductive theorems. In F. Baader, editor, *Proceedings of CADE-19*, Miami, Florida, July 2003. Springer-Verlag. 2, 22

[Dow99] G. Dowek. *La part du Calcul*. Université de Paris 7, 1999. Mémoire d'habilitation. 5

[DP01] N. Dershowitz and D. A. Plaisted. Rewriting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 9, pages 535–610. Elsevier Science, 2001. 8

[Fer95] M. Ferreira. *Termination of Term Rewriting: Well foundedness, Totality and Transformations*. PhD thesis, Utrecht University, 1995. 8

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989. 4

[GS92] H. Ganzinger and J. Stuber. Inductive theorem proving by consistency for first-order clauses. In *Conditional Term Rewriting Systems*, pages 226–241, 1992. 22

[Hue72] G. Huet. *Constrained Resolution: A Complete Method for Higher Order Logic*. PhD thesis, Case Western Reserve University, 1972. 4

[Hul80] J.-M. Hullot. Canonical forms and unification. In *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, pages 318–334, July 1980. 12

[KK99] C. Kirchner and H. Kirchner. Rewriting, solving, proving. A preliminary version of a book available at `www.loria.fr/~ckirchne/rsp.ps.gz`, 1999. 2, 12

[KKN06] C. Kirchner, H. Kirchner, and F. Nahon. Narrowing based inductive proof search: Definition and optimisations. Research report, LORIA, March 2006. 9, 11, 21

[KKR90] C. Kirchner, H. Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on Automatic Deduction. 4

[KZ95] D. Kapur and H. Zhang. An overview of rewrite rule laboratory (RRL). *J. Computer and Mathematics with Applications*, 29(2):91–114, 1995. 5

[Mus80] D. Musser. On proving inductive properties of abstract data types. In *Proceedings, Symposium on Principles of Programming Languages*, volume 7. Association for Computing Machinery, 1980. 22

[Ste] G. Steel. Proof by consistency - a literature survey. 22

[Wac05] B. Wack. *Typage et déduction dans le calcul de réécriture*. Thèse de doctorat, Université Henri Poincaré - Nancy I, October, 7- 2005. 22