INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Proving Negative Conjectures on Equational Theories using Induction and Abstract Interpretation

Thomas Genet , Valérie Viet Triem Tong

N°4576

Octobre 2002

THÈME 2

Rapport de recherche

ISSN 0249-6399

Proving Negative Conjectures on Equational Theories using Induction and Abstract Interpretation

Thomas Genet , Valérie Viet Triem Tong

Thème 2 — Génie logiciel
et calcul symbolique
Projet Lande

Rapport de recherche n$^\circ$ 4576 — Octobre 2002 — 17 pages

Abstract: In this paper we present a method to prove automatically initial negative properties on equational specifications. This method tends to combine induction and abstract interpretation. Induction is performed in a classical way using cover sets and rewriting. Abstract interpretation is done using an additional set of equations used to approximate the initial model into an abstract one. Like in the methods dedicated to the proof by induction of positive properties, the equational specification is supposed to be oriented into a terminating, confluent and complete term rewriting system.

Key-words: Equational theories, proof by induction, abstract interpretation, rewriting

(Résumé : tsvp)

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Preuve de conjectures négatives dans les théories équationnelles par induction et interprétation abstraite

Résumé : Dans cet article, nous présentons une méthode dédiée à la preuve de propriétés négatives dans le modèle initial de spécifications équationnelles. Cette méthode combine à la fois l'induction et l'interprétation abstraite. L'induction est effectuée de façon classique en utilisant des ensembles couvrants. L'interprétation abstraite est effectuée à l'aide d'un ensemble d'équations supplémentaires dont le rôle est d'approcher le modèle initial en un modèle abstrait. Comme dans les méthodes dédiées à la preuve de conjectures positives, la spécification équationnelle est orientée en un système de réécriture terminant, confluent et complet.

Mots-clé : Théories équationnelles, preuve par induction, interprétation abstraite, réécriture

1 Introduction

In the field of automatic deduction, the proof of (positive) inductive theorems on equational specifications has already been widely investigated [Red90, KZ95, BR95, Com94, CN98, GS92]. Starting from an equational specification $\mathcal{E}$ and from an equation $s = t$ those techniques try to prove that $s = t$ is true in the initial model of $\mathcal{E}$ and thus prove that $s = t$ is true in every Herbrand model. This implies that $s = t$ is an inductive theorem of $\mathcal{E}$. As far as we know, the proof of negative theorems has not been investigated [1]. What we intend to do here is to prove that a disequality $s \neq t$ is true in the initial model of E, using induction and rewriting. Furthermore, it is also possible to combine induction with abstract interpretation to simplify some proofs. In this particular setting, abstract interpretations can be defined in a very simple, general and sound way with some 'approximating' additional equations.

Like positive ones, negative theorems are of great interest in verification. In particular, they permit to prove some unreachability properties in equational specifications. This has already been investigated in [GVTT01, GK00] with the Timbuk tool for proving security properties on cryptographic protocols. However, in these works, all the proofs are achieved thanks to abstract interpretation only. Now, we aim at combining abstraction with induction. This work is a first step in that direction.

In section 2, we recall the definitions of equational theories. In section 3, we recall the definitions of rewriting systems, Herbrand models and, complete equational specifications and cover sets. In section 4, we present the deduction system for negative theorems as well as an example. In section 5, we show how to use abstract interpretation with the same deduction system. Finally, in section 6, we conclude on the combination of induction and abstract interpretation and some tracks for possible enhancements.

2 Equational Theory

2.1 Σ-algebra

Definition 1 (sorted signature) *A sorted signature denoted by $\Sigma = (\mathcal{S}, \mathcal{F})$ is given by a non-empty set of sorts $\mathcal{S}$ and a set of ranked functions symbols $\mathcal{F}$. A function symbol f with an arity n is denoted by*

$$f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1} \qquad (n \leq 0,\, \mathcal{S}_i \in \mathcal{S}\, (0 \leq i \leq n)).$$

$\mathcal{S}_1 \times \ldots \times \mathcal{S}_n$ is the domain of f and $\mathcal{S}_{n+1}$ its co-domain. A symbol of constant is a symbol with arity 0*.*

We suppose that there exists at least one element of each sort.

[1] Except in Musser's approach where disequalities of the form $s \neq t$ are encoded into of the form $eq(s,t) = false$ [Mus80].

Definition 2 (Σ-algebra) *For a given signature $\Sigma = (\mathcal{S}, \mathcal{F})$, a Σ-algebra is a set A such that for every function symbol f:*

$$f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1}$$

There exists a function $\overline{f}$ in $A_{\mathcal{S}_1} \times \ldots \times A_{\mathcal{S}_n} \rightarrow A_{\mathcal{S}_{n+1}}$, $\overline{f}$ is called interpretation of f in A.

Definition 3 (Terms) *Let $\mathcal{X}$ be a countable set of variables, and $\Sigma = (\mathcal{F}, \mathcal{S})$ a signature, $\mathcal{T}(\mathcal{F}, \mathcal{X})$ denotes the set of well-formed and well-typed terms over $(\Sigma, \mathcal{X})$ and is defined by induction: A variable is a term, if $t_1 : \mathcal{S}_1, \ldots, t_n : \mathcal{S}_n$ are some terms and $f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1}$ is a symbol of rank n then $f(t_1, \ldots, t_n)$ is a term of sort $\mathcal{S}_{n+1}$.*

For a term t the set of variables that occurs in t is noted $Var(t)$. A ground term is a term such as $Var(t) = \emptyset$. The set of ground terms is noted $\mathcal{T}(\mathcal{F})$. A term is linear if each variable occurs only one time. A substitution σ is a mapping from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F}, \mathcal{X})$, its domain $dom(\sigma)$ is $\{x \in X | x\sigma \neq x\}$.

$\mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\mathcal{T}(\mathcal{F})$ are $\Sigma - algebra$ called $\Sigma - algebra$ of terms with variables and $\Sigma - algebra$ of ground terms.

Definition 4 (equation) *An equation is a pair of terms (l, r) denoted by $(l = r)$. The variables are assumed to be universally quantified.*

2.2 theory of equality

Definition 5 (equational theory) *For a given set of equation $\mathcal{E}$ over a sorted signature $\Sigma = (\mathcal{F}, \mathcal{X})$, the equational theory of $\mathcal{E}$ is the set of equality that can be deduced from $\mathcal{E}$ and the theory of equality:*

1. *Reflexivity*
   $\vdash (u = u)$
2. *Symmetry*
   $(u = v) \vdash (v = u)$
3. *Transitivity*
   $(t = u), (u = v) \vdash (t = v)$
4. *Congruence*
   $\forall f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1}$ *and* $\forall u_1 \in Sort_1, \ldots u_n \in \mathcal{S}_n$, $v_1 \in Sort_1, \ldots, v_n \in \mathcal{S}_n$,
   $(u_1 = v_1), \ldots, (u_n = v_n) \vdash (f(u_1, \ldots u_n) = f(v_1, \ldots v_n))$
5. *Substitutivity*
   $(u = v) \vdash (u\sigma = v\sigma)$ *for every substitution* σ

3 Rewriting systems and equational logic

Definition 6 *A position p for a term t is a word over $\mathbb{N}$, the set of positions is denoted by $Pos(t)$ and is defined by:*

- $Pos(t) = \epsilon$ *if* $t \in \mathcal{X}$
- $Pos(f(t_1, \ldots, t_n)) = \{\epsilon\} \cup \{i.p \mid 1 \leq i \leq n$ *and* $p \in Pos(t_i)$

A position p of a term t is strict if the subterm on position p is not a variable.

For a term t and a position p in t, the depth of p is the length of p, the size of t is equal to the maximum depth of the positions of t.

Definition 7 (Context) *A context is a term $C[]$ in $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{X})$ where the new constant symbol $\Box$ appear only one time. For every context $C[]$, and every term t, $C[t]$ denotes the term obtained by replacing $\Box$ by t in $C[]$.*

Definition 8 (Rewriting system) *A rewriting system is a pair $(\Sigma, \mathcal{R})$ where Σ is a signature and $\mathcal{R}$ a set of rules of the form $l \rightarrow r$ where , $r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, l is not a variable and $Var(r) \subseteq Var(l)$. For a given rewriting system $\mathcal{R} = \{l_i \rightarrow r_i\}_{i \in I}$, we denote $=_{\mathcal{E}}$ the equational theory induced by the set of equations $\mathcal{E} = \{l_i = r_i\}$.*

A substitution σ is a mapping from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F}, \mathcal{X})$, its domain $dom(\sigma)$ is $\{x \in X | x\sigma \neq x\}$. For two terms t, t', a position p of t, a rule $l \rightarrow r$ and a substitution σ, we say t is rewritten in t' (which is denoted by $t \rightarrow_{\mathcal{R}} t'$) iff $t|_p = l\sigma$ and $t' = t[r\sigma]_p$. $\mathcal{R}$ induces a rewriting relation $\rightarrow_{\mathcal{R}}$ on terms whose reflexive transitive closure is noted $\rightarrow^*_{\mathcal{R}}$.
A term t is said to be in normal form (with respect to a given rewriting system $\mathcal{R}$) if there exist no term t' such that $t \rightarrow_{\mathcal{R}} t'$, it is also called *irreducible*, otherwise t is said *reducible*.

Theorem 1 (Birkhoff) *Let $\mathcal{R} = \{l_i \rightarrow r_i\}_{i \in I}$ be a rewriting system and $\mathcal{E} = \{l_i = r_i\}$ the equational theory induced. For all terms u and v, $u =_{\mathcal{E}} v$ iff there exists an finite sequence of terms $t_0, \ldots t_n$ such that $t_0 = u$, $t_n = v$ and for every $1 \leq i \leq n-1$, we have either $t_i \rightarrow_{\mathcal{R}} t_{i+1}$ or $t_i \leftarrow_{\mathcal{R}} t_{i+1}$.*

3.1 Herbrand model

Definition 9 (Σ-congruence) *A Σ-congruence over a Σ-algebra A is a equivalence relation $\equiv$ over A such that $\forall f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1}$ and $\forall a_1 \in Sort_1, \ldots a_n \in \mathcal{S}_n$, $b_1 \in Sort_1, \ldots, b_n \in \mathcal{S}_n$, $\in A$,*

$$(a_1 \equiv b_1 \wedge \ldots \wedge a_n \equiv b_n) \Rightarrow (f(a_1, \ldots a_n) \equiv f(b_1, \ldots b_n)$$

For a given rewriting system $\mathcal{R}$ and $\mathcal{E}$ the equational theory induced, the smallest Σ-congruence $\equiv_{\mathcal{E}}$ over $\mathcal{T}(\mathcal{F})$ is the smallest congruence such that $u \equiv_{\mathcal{E}} v$ if and only if there is a ground substitution σ and an equation $l = r$ of the equational theory of $\mathcal{E}$ verifying $u = l\sigma$ and $v = r\sigma$. The quotient of the set of terms is denoted by $\mathcal{T}(\mathcal{F})/_{\equiv_{\mathcal{E}}}$.

Definition 10 *A Herbrand model of a given equational theory $\mathcal{E} = \{u_1 = v_1, \ldots, u_n = v_n\}$, is a model $\mathcal{M}$ of theory of equality (def. 5) whose domain is the set of terms $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $\mathcal{M}$ checks the formula $u_1 = v_1 \wedge \ldots \wedge u_n = v_n$.*

Let $\mathcal{E}$ be a theory, an equation $(u = v)$ is an inductive consequence (or inductive theorem) of $\mathcal{E}$ iff every Herbrand model $\mathcal{M}$ of $\mathcal{E}$, $\mathcal{M}$ checks $u = v$. In particular, such an equation holds true in all Herbrand models iff $\mathcal{T}(\mathcal{F})/_{\equiv_\mathcal{E}}$ checks it. The set of all inductive consequences of $\mathcal{E}$ is called *inductive theory* of $\mathcal{E}$.

3.2 Specification

Definition 11 *Let $\Sigma = (\mathcal{S}, \mathcal{F})$ be a signature, a subset $\mathcal{C}$ of $\mathcal{F}$ is called set of constructor symbols for a given theory $\mathcal{E}$ if for every term $t \in \mathcal{T}(\mathcal{F})$, there is a term $u \in \mathcal{T}(\mathcal{C})$ such that $t \equiv_\mathcal{E} u$. A set of constructors is called free iff for all distinct terms $s, t \in \mathcal{T}(\mathcal{C})$, $s \not\equiv_\mathcal{E} t$.*

A constructor is free if there are no rules like $c(c_1, \ldots c_n) \rightarrow d$ in $\mathcal{R}$. We interest here to theories where all constructors are free.
A specification is a pair $(\Sigma, \mathcal{R})$ where Σ is a signature and $\mathcal{R}$ a rewriting system over Σ. A specification is said *sufficiently complete* regards to a set D of symbols called *defined symbols* if $\mathcal{C} = \Sigma \setminus D$ is a set of constructors for $\mathcal{E}$. This is generally undecidable except when $\mathcal{R}$ is linear, confluent and terminating and $\mathcal{R}$ protect constructor: if there is a rule $l \rightarrow r$ such that $l \in \mathcal{T}(\mathcal{C}, \mathcal{X})$ then $r \in \mathcal{T}(\mathcal{C}, \mathcal{X})$. A sufficiently complete specification generated a congruence $\equiv_\mathcal{E}$ over $\mathcal{T}(\mathcal{F})$ whose set of representatives is $\mathcal{T}(\mathcal{C})$.

3.3 Cover

Definition 12 (cover set) *A finite set C of irreducible terms of sort s is a cover set for s with regards to $\mathcal{R}$ iff for all ground term g of sort s there is a term $t \in C$ and a substitution τ such that $g \equiv_\mathcal{E} t\tau$.*

A *cover-substitution* is a substitution that associates to each free variable an element of a cover set.

Definition 13 (Cover) *Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and x a variable, $Cover(t, x)$ is the set of terms obtained by applying every cover-substitution to x, x is said "expanded" in $Cover(t, x)$ and is denoted by $\overline{x}$ and every other variable is duplicated. If x does not appear in t, $Cover(t, x) = \{t\}$*
More precisely:

1. *Σ_s is ordered, every element is associated to a number corresponding to its order.*
2. *$Cover(t, x) = \{t\{x \mapsto c(\overline{w_1}^i, \ldots, \overline{w_n}^i)\} \mid c \in \Sigma_s$ where c is numbered by i and $c : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \rightarrow \mathcal{S}_{n+1}\}$. In $c(\overline{w_1}^i, \ldots, \overline{w_n}^i)$, every w_j $(j = 1 \ldots n)$ is a fresh variable and every expanded variable $\overline{y}^k$ is replaced by $\overline{y}^{k.i}$.*

By extension, if $C = \{t_1 \cong t_2, \ldots t_n \cong t_{n+1}$ (where $\cong$ denotes $=$ or $\neq$), $Cover[C, x] = \{Cover[t_1 \cong t_2, x], \ldots, Cover[t_n \cong t_{n+1}, x]\}$.

Example 1 *Consider the specification over* $\{Plus : nat \rightarrow nat,\ Minus : nat \rightarrow nat,\ S : nat \rightarrow nat,\ 0 : nat\}$*, where* $Plus, Minus$ *are defined symbols and* $S, 0$ *constructor symbol-s.*

$$\begin{aligned} Minus(0, x) &\rightarrow 0 && (3.1)\\ Minus(x, 0) &\rightarrow x && (3.2)\\ Minus(S(x), S(y)) &\rightarrow Minus(x, y) && (3.3)\\ Plus(0, y) &\rightarrow y && (3.4)\\ Plus(S(x), y) &\rightarrow S(Plus(x, y)) && (3.5) \end{aligned}$$

Σ_{nat} *is ordered in the following way* $\{(0, 1), (S, 2)\}$*. The set* $Cover(Plus(x, Minus(y, \overline{z}^{1.2})), x)$ *is:*
$\{Plus(0, Minus(0, \overline{z}^{1.2.1})),\ Plus(S(\overline{x}^{2}), Minus(y, \overline{z}^{1.2.2}))$

Definition 14 (recurrence position) *Let* f *be a defined symbol, the set of recurrence position for* f *is the set of positions* u *verifying:*

- *either there is a rule* $l \rightarrow r$ *in* $\mathcal{R}$ *such that* f *is the top symbol of* l *and the subterm of* l *on position* u *is not a variable*
- *or there is a rule* $l \rightarrow r$ *in* $\mathcal{R}$ *such that* f *is the top symbol of* l *and the subterm of* l *on position* u *is a variable* x *such that* l *is not linear on* x*.*

4 Initial properties

For a theory $\mathcal{E}$ sufficiently complete, where all constructors are free, we are interested in properties P of $\mathcal{T}(\mathcal{F}, \mathcal{X})/_{\equiv_{\mathcal{E}}}$, with P on the form $t_1 \neq t_2$ universally quantified ($t_1, t_2 \in \mathcal{T}(\mathcal{F}, \mathcal{X})$). Let x be a variable of sort s such that x appears in t_1 and $\exists u\, t_1|_u = x$ and u is a recurrence position for t_1, we denote t_1 by $t_1[x]$. We divide the set of elements of sort s in Σ in two parts $\Sigma_{s_{base}}$ and $\Sigma_{s_{rec}}$:

- if there is a finite set of ground irreducible terms of sort s then $\Sigma_{s_{base}}$ contains all elements of $\Sigma \cap \mathcal{C}$ of sort s and $\Sigma_{s_{rec}} = \emptyset$
- else $\Sigma_{s_{base}}$ contains elements of $\Sigma \cap \mathcal{C}$ of sort s whose arity is 0 and $\Sigma_{s_{rec}}$ contains elements $\Sigma \cap \mathcal{C}$ of sort s whose arity is $n \geq 0$.

Proving $\forall \sigma\, t_1\sigma \neq t_2\sigma$, classically consists in

- First proving that $\{t_1[x\sigma_{base}] \neq t_2[x\sigma_{base}]\}$

- Assuming that $\{t_1[x] \neq t_2[x]\}_i$ holds for any terms x and proving that $\{t_1[x\sigma_{rec}] \neq t_2[x\sigma_{rec}]\}$

where $\{t[x\sigma_{base}]\}$ denotes the set of terms obtained by replacing x in $t[x]$ with every elements in $\Sigma_{s_{base}}$ and $\{t[x\sigma_{rec}]\}$ denotes the set of terms obtained by replacing x in $t[x]$ with all possible $f(u_1, \ldots, u_n)$ such that $f : \mathcal{S}_1 \times \ldots \times \mathcal{S}_n \to \mathcal{S}_{n+1} \in \Sigma_{s_{rec}}$ and one term u_i of sort $S_i = S$ is replaced by $\overline{x}$ and others by fresh variables. The induction variable x is expanded into $\overline{x}$ in order to prevent case reasoning on x after induction.

> For the general case of induction, we propose here to use the dual reasoning: we prove that $t_1[x\sigma_{base}] \neq t_2[x\sigma_{base}]$, and
> we show that if ever $t_1[x\sigma_{rec}]=t_2[x\sigma_{rec}]$ then it was already true for $t_1[x]=t_2[x]$.

4.1 deduction system

We present our deduction system (figure 4.1) based on a set $\mathcal{I}$ of inference rules over the tuple (L, HR, C_-, C_+) where L is a set of rules $l \to r$ such that $l = r$ is an inductive theorem, HR contains the recurrence hypothesis and starts empty, C_+ is a set of positive conjectures and C_- a set of negative conjectures.

Definition 15 *An $\mathcal{I}$-derivation is a sequence*

$$(L_0, HR_0, C_{0,-}, C_{0,+}) \vdash_{\mathcal{I}} (L_1, HR_1, C_{1,-}, C_{1,+}) \vdash_{\mathcal{I}} \ldots \vdash_{\mathcal{I}} (L_n, HR_n, C_{n,-}, C_{n,+})$$

A I-derivation is successful if it ends with $(L_n, HR_n, \emptyset, \emptyset)$.

4.2 Correction

Theorem 2 *Let $\mathcal{R}$ be a rewriting system, and $\mathcal{E}$ the equational theory associated, over a sorted signature Σ such that all constructors are free and $\mathcal{R}$ is sufficiently complete. C_- a set of conjectures of the form $t_1 \neq t_2$ (universally quantified), and C_+ a set of conjectures of the form $t_1 = t_2$ (universally quantified). If there exists a successful $\mathcal{I}$-derivation starting with $(L, \emptyset, C_-, C_+)$ for any set L of rules $l \to r$ such that $l = r$ is an inductive theorem of $\mathcal{E}$ then C_- is a set of negatives properties of $\mathcal{T}(\mathcal{F})/_{\equiv_\mathcal{E}}$ and C_+ a set of equational theorem.*

***Proof* 1** *Assume that there exists n such that $(L, HR, C_-, C_+) \vdash^n_{\mathcal{I}} (L_n, HR_n, \emptyset, \emptyset)$, we prove by induction on n that C_- is a set of negatives properties of $\mathcal{T}(\mathcal{F})/_{\equiv_\mathcal{E}}$ and C_+ a set of equational theorem under induction hypothesis HR.*

1. *If $(L,\ HR,\ C_-,\ C_+) \vdash_{\mathcal{I}} (L,\ HR,\ \emptyset,\ \emptyset)$ then*
   - *either $(L,\ HR,\ \{f(\overrightarrow{t_1}) \neq g(\overrightarrow{t_2})\},\ \emptyset) \vdash_{\mathcal{I}} (L,\ HR,\ \emptyset,\ \emptyset)$ with $f,\ g \in \mathcal{C}$, $f \neq g$ and the rule **elim -** have been applied. Since all the constructors are free, we know that $f(\overrightarrow{t_1}) \not\equiv_{\mathcal{E}} g(\overrightarrow{t_2})$ in $\mathcal{T}(\mathcal{F}, \mathcal{X})/_{\equiv_\mathcal{E}}$. Hence, $f(\overrightarrow{t_1}) \neq g(\overrightarrow{t_2})$ is a negative property of $\mathcal{T}(\mathcal{F})/_{\equiv_\mathcal{E}}$.*

1. rules for simplification by rewriting:
   (= and $\neq$ are supposed to be commutative)

   simplify1

   $$(L, HR, C_- \cup \{t_1 \neq t_2\}, C_+) \vdash_{\mathcal{I}} (L, HR, C_- \cup \{t'_1 \neq t_2\}, C_+) \ \ if\, t_1 \rightarrow_{\mathcal{R}} t'_1$$

   simplify2

   $$(L, HR, C_-, C_+ \cup \{t_1 = t_2\}) \vdash_{\mathcal{I}} (L, HR, C_-, C_+ \cup \{t'_1 = t_2\})$$

   $if\, t_1 \rightarrow_{\mathcal{R} \cup L} t'_1 or (t_1 = t'_1) \in HR$

   simplify3

   $$(L, HR \cup \{t_1 = t_2\}, C_-, C_+) \vdash_{\mathcal{I}} (L, HR \cup \{t'_1 = t_2\}, C_-, C_+) \ (if\, t_1 \rightarrow_{\mathcal{R} \cup L} t'_1)$$

2. a rule of elimination for negatives conjectures:

   elim -

   $$(L, HR, C_- \cup \{f(\overrightarrow{t_1}) \neq g(\overrightarrow{t_2})\}, C_+) \vdash_{\mathcal{I}} (L, HR, C_-, C_+) \ (f,\, g\, \in \mathcal{C},\, f \neq g)$$

3. a rule for tautology:

   elim +

   $$(L, HR, C_-, C_+ \cup \{t = t\}) \vdash_{\mathcal{I}} (L, HR, C_-, C_+)$$

4. a rule for simplifying positive conjectures:

   simp +

   $$(L, HR, C_-, C_+ \cup \{f(\overrightarrow{t_1}) = f(\overrightarrow{t_2})\}) \vdash_{\mathcal{I}} (L, HR, C_-, C_+ \cup \{\overrightarrow{t_1} = \overrightarrow{t_2}\}) \, (f \in \mathcal{C})$$

5. a rule about free constructors symbols:

   constr

   $$(L, HR \cup \{f(\overrightarrow{t_1}) = f(\overrightarrow{t_2})\}, C_-, C_+) \vdash_{\mathcal{I}} (L, HR \cup \{f(t_1) = f(t_2)\} \cup \{\overrightarrow{t_1} = \overrightarrow{t_2}\}, C_-, C_+) \ (f \in \mathcal{C})$$

6. a rule for case reasoning

   case

   $$(L, HR, C_-, C_+) \vdash_{\mathcal{I}} (L, Cover[HR, x], Cover[C_-, x], Cover[C_+, x])$$

   (for x a non-expanded variable)

7. a rule for applying recurrence reasoning (x induction variable)

   rec

   $$(L, HR, C_- \cup \{t_1[x]) \neq t_2[x])\}, C_+)$$
   $$\vdash_{\mathcal{I}}$$
   $$(L, HR \cup \{t_1[x\sigma_{rec}] = t_2[x\sigma_{rec}]\}, C_- \cup \{t_1[x\sigma_{base}] \neq t_2[x\sigma_{base}]\}, C_+ \cup \{t_1[\overline{x}] = t_2[\overline{x}]\})$$

Figure 1: Inference system $\mathcal{I}$

- *or* $(L, HR, \emptyset, \{t = t\}) \vdash_{\mathcal{I}} (L, HR, \emptyset, \emptyset)$*, in that case* C_+ *is a set of tautology and in particular a set of equational theorem.*

2. *We assume that for all sets of conjectures* C_- *and* C_+*, if* $(L, HR, C_-, C_+) \vdash^n_{\mathcal{I}} (L, HR_n, \emptyset, \emptyset)$ *then* C_- *is a set of negatives properties of* $\mathcal{T}(\mathcal{F})/_{\equiv_{\mathcal{E}}}$ *and* C_+ *a set of equational theorem under induction hypothesis* HR*. We show that the property hold true for* $n+1$*. We reason by case over the first rule used: there is only two non-trivial case:*

   case $(L, HR, C_-, C_+) \vdash_{\mathcal{I}} (L, Cover[HR, x], Cover[C_-, x], Cover[C_+, x]) \vdash^n_{\mathcal{I}} (L, HR_{n+1}, \emptyset, \emptyset)$ *(for* x *a non-expanded variable) by definition of* $Cover$*, for any term* t*,* $Cover(t, x)$ *is the set of terms obtained by applying every possible cover-substitution to* x*. First case: if* x *does not appear in* HR*,* C_- *or* C_+ *then the sets do not change. Hence* C_- *and* C_+ *are respectively sets of negative and positive properties of* $\mathcal{E}$*. Second case: if* x *appear in* HR*,* C_- *or* C_+*, by definition of* $Cover[t, x]$*, if* $g \in \mathcal{T}(\mathcal{F})$ *such that* $\exists \sigma\, g = t\sigma$ *then* $\exists \sigma' \exists t' \in Cover[t, x]$ *such that* $g = t'\sigma'$*. This means that for every equation in* HR*,* C_- *or* C_+*, there exists a set of equivalent equation in in* $Cover[HR, x]$*,* $Cover[C_-, x]$ *or* $Cover[C_+, x]$*. Clearly if* $(L, Cover[HR, x], Cover[C_-, x], Cover[C_+, x]) \vdash^n_{\mathcal{I}} (L, HR_{n+1}, \emptyset, \emptyset)$ *then* C_- *is a set of negatives properties of* $\mathcal{T}(\mathcal{F})/_{\equiv_{\mathcal{E}}}$ *and* C_+ *a set of equational theorem under induction hypothesis* HR*.*

   rec $(L, HR, C_- \cup \{t_1[x]) \neq t_2[x])\}, C_+ \vdash_{\mathcal{I}} (L, HR \cup \{t_1[\overline{x}\sigma_{rec}] = t_2[\overline{x}\sigma_{rec}]\}, C_- \cup \{t_1[\overline{x}\sigma_{base}] \neq t_2[\overline{x}\sigma_{base}]\}, C_+ \cup \{t_1[\overline{x}] = t_2[\overline{x}]\}) \vdash^n_{\mathcal{I}} (L, HR_{n+1}, \emptyset, \emptyset)$*. We have* $(L, HR \cup \{t_1[\overline{x}\sigma_{rec}] = t_2[\overline{x}\sigma_{rec}]\}, C'_-, C_+ \cup C'_+) \vdash^n_{\mathcal{I}} (L, HR_{n+1}, \emptyset, \emptyset)$ *hold true by induction hypothesis. This correspond to dual induction reasoning described before.*

4.3 Proof example

Example 2 *Consider the theory* $\mathcal{E}$ *defined on example 1 over natural number, and the conjecture* $\mathcal{P} = Plus(x, S(y)) \neq Minus(x, y)$*.*
First note that this conjecture is not true in all herbrand model: consider the Herbrand model $\mathcal{M}_1$ *of* $\mathcal{E}$*, verifying* $S(S(0)) = 0$*, for* $x = 0$ *and* $y = S(0)$ *we have:*
$Plus(x, S(y)) = Plus(0, S(S(0))) = S(S(0)) = 0$
$Moins(x, y) = Moins(0, S(0)) = 0$
$\mathcal{P}$ *is not verified !*

We prove that $\mathcal{P}$ *is an initial property of* $\mathcal{T}(\mathcal{F})/_{\equiv_{\mathcal{E}}}$ *using the inductive theorems* $Minus(u, S(v) = Minus(Minus(u, v), S(0)$ *and the commutativity of Plus:* $Plus(u, v) = Plus(u, v)$
We start from

- $L = \{Minus(u, S(v)) \leftrightarrow Minus(Minus(u, v), S(0), Plus(u, v) \leftrightarrow Plus(u, v)\}$
- $HR = \emptyset$

- $C_- = \{\mathcal{P}\}$
- $C_+ = \emptyset$

According to the definition 14, the set of recurrence position for $Plus(x, S(y)$ *is reduce to* $\{\epsilon.1\}$, $Plus(x, S(y))|_{\epsilon.1} = x$. *This variable is of sort nat,* Σ_{nat} *is ordered in* $\Sigma_{nat_{base}} = \{0\}$ *and* $\Sigma_{nat_{rec}} = \{S\}$.

Starting with $(L, \emptyset, \{Plus(x, S(y)) \neq Minus(x, y)\}, \emptyset)$

We apply ***rec***

- $HR = \{Plus(x\sigma_{rec}, S(y)) = Minus(x\sigma_{rec}, y)\} = \{Plus(S(\overline{x}^2), S(y)) = Minus(S(\overline{x}^2, S(y)))\}$
- $C'_- = \{Plus(x\sigma_{base}, S(y)) \neq Minus(x\sigma_{base}, y)\} = \{Plus(0, S(y)) \neq Minus(0, y)\}$
- $C_+ = \{Plus(\overline{x}^2, S(y)) = Minus(\overline{x}^2, y),\}$

We apply ***case*** *on* y

- $HR = \{Plus(S(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0), Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^2))) = Minus(S(\overline{x}^2), S(\overline{y}^2))\}$
- $C'_- = \{Plus(0, S(0) \neq Minus(0, 0),\ Plus(0, S(S(\overline{y}^2)) \neq Minus(0, S(\overline{y}^2))\}$
- $C_+ = \{Plus(\overline{x}^{2.1}, S(0)) = Minus(\overline{x}^{2.1}, 0), Plus(\overline{x}^2, S(S(\overline{y}^2))) = Minus(\overline{x}^{2.2}, S(\overline{y}^2))\}$

using the rule ***simplify1*** C'_- *can be reduced in* $\{0 \neq S(0), 0 \neq S(\overline{y}^2)\}$ *which is reduced in* $\emptyset$ *thanks to the rule* ***elim-***.

- $HR = \{Plus(S(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0), Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^2))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^2))\}$
- $C'_- = \emptyset$
- $C_+ = \{Plus(\overline{x}^{2.1}, S(0)) = Minus(\overline{x}^{2.1}, 0), Plus(\overline{x}^{2.2}, S(S(\overline{y}^2))) = Minus(\overline{x}^{2.2}, S(\overline{y}^2))\}$

Now we use ***simplify3*** *for reducing the first part of* HR:
$Plus(S(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0) \rightarrow_{\mathcal{R}} S(Plus(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0)$
$S(Plus(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0) \rightarrow_{L} S(Plus(S(0), \overline{x}^{2.1})) = Minus(S(\overline{x}^{2.1}), 0)$
$S(Plus(S(0), \overline{x}^{2.1})) = Minus(S(\overline{x}^{2.1}), 0) \rightarrow_{\mathcal{R}} S(S(Plus(0, \overline{x}^{2.1}))) = Minus(S(\overline{x}^{2.1}), 0)$
$S(S(Plus(0, \overline{x}^{2.1}))) = Minus(S(\overline{x}^{2.1}), 0) \rightarrow_{\mathcal{R}} S(S(\overline{x}^{2.1}) = Minus(S(\overline{x}^{2.1}), 0))$
and $S(S(\overline{x}^{2.1}) = Minus(S(\overline{x}^{2.1}), 0)) \rightarrow_{\mathcal{R}} S(S(\overline{x}^{2.1})) = S(\overline{x}^{2.1})$

and at the same time we use ***simplify2*** *for reducing* C_+:
$(Plus(\overline{x}^{2.1}, S(0)) = Minus(\overline{x}^{2.1}, 0)) \rightarrow_{\mathcal{R},L} (S(\overline{x}^{2.1}) = \overline{x}^{2.1})$

- $HR = \{S(S(\overline{x}^{2.1})) = S(\overline{x}^{2.1}), Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^2))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^2))\}$
- $C'_- = \emptyset$
- $C_+ = \{S(\overline{x}^{2.1}) = \overline{x}^{2.1}, Plus(\overline{x}^{2.2}, S(S(\overline{y}^2))) = Minus(\overline{x}^{2.2}, S(\overline{y}^2))\}$

now we can apply the rule ***constr***

- $HR = \{S(\overline{x}^{2.1}) = \overline{x}^{2.1}, Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^{2}))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^{2}))\}$
- $C'_{-} = \emptyset$
- $C_{+} = \{S(\overline{x}^{2.1}) = \overline{x}^{2.1}, Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2}))) = Minus(\overline{x}^{2.2}, S(\overline{y}^{2}))\}$

*the rule **simplify2** allows us to conclude on the first part of C_{+}: $S(\overline{x}^{2.1}) = \overline{x}^{2.1}$ corresponds to the induction hypothesis: this means that if ever there exists $\overline{x}^{2.1}$ such that $Plus(S(\overline{x}^{2.1}), S(0)) = Minus(S(\overline{x}^{2.1}), 0)$ ($\in HR$) then $S(S(\overline{x}^{2.1})) = S(\overline{x}^{2.1})$, $S(\overline{x}^{2.1}) = \overline{x}^{2.1}$ and $Plus(\overline{x}^{2.1}, S(0)) = Minus(\overline{x}^{2.1}, 0)$ ($\in C_{+}$)*

- $HR = \{S(\overline{x}^{2.1}) = \overline{x}^{2.1}, Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^{2}))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^{2}))\}$
- $C'_{-} = \emptyset$
- $C_{+} = \{Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2}))) = Minus(\overline{x}^{2.2}, S(\overline{y}^{2}))\}$

*using the following rewriting sequence, which corresponds to the application of the rule **simplify2***

$Minus(\overline{x}^{2.2}, S(\overline{y}^{2})) \to_{L} Minus(Minus(\overline{x}^{2.2}, \overline{y}^{2}), S(0)) \to_{HR} Minus(Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^{2}))), S(0)) \to Minus(S(Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2})))), S(0)) \to Minus(Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2}))), 0) \to Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2})))$
we finish with

- $HR = \{S(\overline{x}^{2.1}) = S(\overline{x}^{2.1}), Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^{2}))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^{2}))\}$
- $C'_{-} = \emptyset$
- $C_{+} = \{Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2}))) = Plus(\overline{x}^{2.2}, S(S(\overline{y}^{2})))\}$

*and the rule **elim+** allows us to conclude the second case:*

- $HR = \{S(\overline{x}^{2.1}) = S(\overline{x}^{2.1}), Plus(S(\overline{x}^{2.2}), S(S(\overline{y}^{2}))) = Minus(S(\overline{x}^{2.2}), S(\overline{y}^{2}))\}$
- $C'_{-} = \emptyset$
- $C_{+} = \emptyset$

We succeed in reducing $(L, \emptyset, \{Plus(x, S(y)) \neq Minus(x, y)\}, \emptyset)$ into $(L, HR, \emptyset, \emptyset)$ thanks to theorem 2, $\mathcal{P}$ is a property of $\mathcal{T}(\mathcal{F})/_{\equiv_{\mathcal{E}}}$.

5 Using abstract interpretation

One of the most interesting property of the negative conjectures is that if they hold in a Herbrand model 'bigger' than the initial model then they also hold in the initial model. Indeed, if $\mathcal{E}' = \mathcal{E} \cup A$ where A is a non empty set of 'approximating' equations, then if $s \neq t$ is valid in the initial model of $\mathcal{E}'$ then $s \neq t$ is also valid in the initial model of $\mathcal{E}$. This property is of great use for proving inductive theorems by abstract interpretation using the same set of deduction rules $\mathcal{I}$. Note that, in this setting, every abstract interpretation given as a set of equations is necessarily sound.

Example 3 *Let E be the following equational specification:*

$$\begin{array}{l} Plus(0,x) = x \\ Plus(S(x),y) = S(Plus(x,y)) \\ Mult(0,x) = 0 \\ Mult(S(x),y) = Plus(y, Mult(x,y)) \end{array}$$

Let $Mult(x,x) \neq S(S(0))$ be the negative theorem we want to prove, i.e. prove that there exist no natural number x such that $x^2 = 2$. This can be shown exactly in the same way as it is done in Example 2, but it can also be shown using abstract interpretation of this specification on the abstract domain $\{0, one, two, sup2\}$ where $sup2$ represent $\{3, 4, \ldots\}$. In order to do this, it is enough to complete the previous specification together with the following set A of additional 'approximating' equations for the abstract interpretation:

$$\begin{array}{l} S(0) = one \\ S(S(0)) = two \\ S(sup2) = sup2 \\ S(S(S(x))) = sup2 \end{array}$$

as well as two other equations in order to make completion converge: $\{Plus(x,0) = x, Plus(x,S(y)) = S(Plus(x,y))\}$. Note these last equations are inductive theorems of the specification [2]*. On all those equations, the completion* [3] *converges and produces a terminating and confluent term rewriting system: $\mathcal{R}_{app}$.*

[2] but we do not need to prove it, for the same reason as adding A is correct.
[3] we used either CiME [CMMU] and Waldmeister [HL] for our experimentations.

$S(0) \to one$
$S(sup2) \to sup2$
$Plus(0, x) \to x$
$Plus(x, 0) \to x$
$s(one) \to two$
$Mult(0, x) \to 0$
$S(S(S(x))) \to sup2$
$s(two) \to sup2$
$Plus(S(x), y) \to S(Plus(x, y))$
$Plus(x, S(y)) \to S(Plus(x, y))$
$Plus(one, x) \to S(x)$
$Plus(two, x) \to S(S(x))$
$Plus(sup2, x) \to sup2$
$Plus(x, one) \to S(x)$
$Plus(x, two) \to S(S(x))$
$Plus(x, sup2) \to sup2$
$Mult(S(x), y) \to Plus(y, Mult(x, y))$
$Mult(one, x) \to x$
$Mult(two, x) \to Plus(x, x)$
$Mult(sup2, x) \to Plus(x, Plus(x, x))$
$Plus(x, Plus(x, Plus(x, x))) \to Plus(x, Plus(x, x))$
$Plus(x, Plus(x, Plus(x, Mult(y, x)))) \to Plus(x, Plus(x, x))$
$Mult(x, 0) \to 0$

Note that in this system, the definitions of functions $Plus$ and $Mult$ on the abstract domain $\{0, one, two, sup2\}$ have been automatically computed by the completion. In particular, the term rewriting system is also complete with regards to the abstract version of sort nat and its new set of free constructors $\{0, one, two, sup2\}$. Now, we can start the deduction process from the following initial state:

- $L = \emptyset$
- $HR = \emptyset$
- $C_- = \{Mult(x, x) \neq S(S(0))\}$
- $C_+ = \emptyset$

We have a unique recurrence variable x which is of sort nat and $\Sigma_{nat_{base}} = \{0, one, two, sup2\}$ and $\Sigma_{nat_{rec}} = \emptyset$. Thus starting with $(\emptyset, \emptyset, \{Mult(x, x) \neq S(S(0))\}, \emptyset)$, we apply ***rec*** *and we obtain:*

- $L = \emptyset$

- $HR = \emptyset$
- $$\begin{array}{rcl} C_- & = & \{Mult(x\sigma_{base}, x\sigma_{base}) \neq S(S(0))\} \\ & = & \{Mult(0,0) \neq S(S(0)), \\ & & Mult(one, one) \neq S(S(0)), \\ & & Mult(two, two) \neq S(S(0)), \\ & & Mult(sup2, sup2) \neq S(S(0))\} \end{array}$$
- $C_+ = \emptyset$

Then by several applications of rule ***simplify1****, the equation of C_- are deterministically normalized by $\mathcal{R}_{app}$ and the proof state becomes:*

- $L = \emptyset$
- $HR = \emptyset$
- $$\begin{array}{rcl} C_- & = & \{0 \neq two, \\ & & one \neq two, \\ & & sup2 \neq two, \\ & & sup2 \neq two\} \end{array}$$
- $C_+ = \emptyset$

Finally several applications of rule ***elim*** *- retrieve one by one all those equations since their constructors are free and we end on proof state $(\emptyset, \emptyset, \emptyset, \emptyset)$, proving the initial theorem.*

6 Further work

In this work we have proposed a technique for proving negative theorems on equational specifications by induction and abstract interpretation. The use of both induction and abstract interpretation should be investigated further in the case where there are several induction variables in a disequation to prove. In this case, it is also possible to combine abstract interpretation and induction in the same proof, i.e. instead of performing two inductions on the two variables, use one induction on the first variable and use abstract interpretation on the second one. This can already be done using our technique but only in some particular cases. In particular, the abstract interpretation should not involve induction variables. For instance, assume that we want to prove a property on some lists of naturals, this could be done by using an induction on the list structure and an abstract interpretation on the naturals. Conversely, using an abstract interpretation for the list structure and an induction on the naturals, may also lead to an abstract interpretation of the naturals invalidating the induction. We are still investigating this aspects in order to relax those constraints and use the combination of the two techniques in every cases.

What should be also investigated is if proving negative properties of equational specification may offer a way to find axiomatisation for the smallest Herbrand model of a theory. This would be useful in proof techniques like ”inductionless induction” [CN98].

References

[BR95] Adel Bouhoula and Michael Rusinowitch. SPIKE: A system for automatic inductive proofs. In *Algebraic Methodology and Software Technology*, pages 576–577, 1995.

[CMMU] E. Contejean, C. Marché, B. Monate, and X. Urbain. Cime. `http://cime.lri.fr/`.

[CN98] H. Comon and R. Nieuwenhuis. Induction = I-axiomatization + first-order consistency. *Information and Computation*, 1998.

[Com94] H. Comon. Inductionless induction. In René David, editor, *2nd Int. Conf. in Logic For Computer Science: Automated Deduction. Lecture notes*, Chambéry, 1994. Univ. de Savoie.

[GK00] T. Genet and F. Klay. Rewriting for Cryptographic Protocol Verification. In *Proceedings 17th International Conference on Automated Deduction, Pittsburgh (Pen., USA)*, volume 1831 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2000.

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

[GVTT01] T. Genet and Valérie Viet Triem Tong. Reachability Analysis of Term Rewriting Systems with timbuk. In *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Havana (Cuba)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 691–702. Springer-Verlag, 2001.

[HL] T. Hillenbrand and B. Löchner. Waldmeister. `http://www.mpi-sb.mpg.de/ hillen/waldmeister/`.

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

[Mus80] D. R. Musser. On proving inductive properties of abstract data types. 1980.

[Red90] U. Reddy. Term rewriting induction. In *Procedings of the Tenth International Conference on Automated Deduction*. Springer–Verlag, 1990.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399