First-Order Completion with Ordering Constraints: Some Positive and Some Negative Results

Robert Nieuwenhuis
Albert Rubio

Report LSI–91–43

First-Order Completion with Ordering Constraints: Some Positive and Some Negative Results

Robert Nieuwenhuis, Albert Rubio

Universidad Politécnica de Cataluña
Dept. Lenguajes y Sistemas Informáticos
Pau Gargallo 5, E-08028 Barcelona, Spain
E-mail: roberto@lsi.upc.es rubio@lsi.upc.es

Keywords: Automated theorem proving, programming languages, logic programmimg.

Abstract: We show by means of counter examples that some well-known results on the completeness of deduction methods with ordering constraints are incorrect. The problem is caused by the fact that the usual lifting lemmata do not hold.

In this paper, these problems are overcome by using a different lifting method. We define a completion procedure for ordering constrained first-order clauses with equality, including notions of redundant inferences and clauses, as done in [BG 91] for clauses without constraints. This completion procedure is refutationally complete if the *initial* set of constrained clauses fulfills a property which we have called *pureness*. In particular, clauses without constraints are pure. Pureness is preserved during the completion process. Since the constraints generated during completion reduce the search space considerably, our results allow to do very efficient theorem proving in first-order logic with equality. Moreover, complete sets of axioms (canonical sets of rewrite rules in the equational case) can be obtained in more cases.

1. Introduction

Knuth-Bendix-like completion [KB 70, Rus 87, HR 89, BDP 89, BG 90, NO 91] can be seen as a refutationally complete process that transforms a set of axioms in such a way that, by using the final *complete* set of axioms, efficient *normal form* proofs can be obtained (e.g. *linear* proofs, or the rewrite proofs in the equational case). The possibility to use powerful simplification techniques makes completion methods especially interesting for theorem proving and many other applications.

The idea of applying the constraint paradigm to theorem proving has become more and more popular [Hue 72, Bür 85, KK 89, Pet 90, KKR 90]. In this context, two important advantages are that the search space can be reduced and that complete systems can be obtained in more cases. It is also frequently mentioned that the use of constraints in theorem proving allows to delay the solving of difficult constraints, to represent infinite sets of formulae, and to remove the constraints from the meta-level, including them into the formulae, which provides a more uniform treatment.

In particular, we are interested in the application of *ordering constraints* to completion techniques. We basically use the inference rule of *strict superposition* [BG 90], in which the search space is reduced by selecting only the *maximal* literals to paramodulate upon. Therefore, if a clause is obtained in an inference, we are in fact only interested in those ground instances of it for which the literal selected is really the biggest one. This information is kept in its constraint. Future choices of maximal literals that are incompatible with this constraint can then be shown to be unnecessary. The satisfiability problem for this kind of constraints, i.e. knowing whether a term t can be made bigger than a term s by appropriately instantiating its variables, has recently been shown to be decidable for the lexicographic path ordering by Comon [Com 90] and for the recursive path ordering with status by Jouannoud and Okada [JO 91].

Surprisingly, some well-known results on the completeness of deduction methods with ordering constraints are incorrect. Peterson [Pet 90] claims that every constrained critical pair between equations in R is joinable iff R is a complete set of reductions. In the following counter example, where $f \succ a \succ b \succ c \succ T \succ F$

1. $f(x) = T \quad \text{if} \quad x \succ b$
2. $f(c) = F$
3. $a = c$

there are no critical pairs, because all possible superpositions between the equations produce critical pairs with unsatisfiable constraints. However, there is no rewrite proof for $T = F$, which is a valid consequence, since $T =_1 f(a) =_3 f(c) =_2 F$. In this case, the problem is caused by the fact that the critical pair lemma does not hold for equations with arbitrary ordering constraints. The proof of the critical pair lemma is based on the existence of *all* ground instances of the equations, which is not the case here, since $f(c) = T$ is *not* a ground instance of the first equation.

A similar problem appears in the case of full first-order clauses with equality. In [KKR 90], the authors claim that their inference rules for ordered deduction are refutationally complete in the constrained case. However, under the ordering $f \succ a \succ b \succ c$, the following

inconsistent set of clauses is closed under the inference rules, but does not contain the empty clause:

1. $P(x)$ **if** $x \succ b$
2. $\neg P(c)$
3. $a = c$

In this case, the problem is caused by the fact that the usual lifting lemmata do not hold, since also here smaller ground instances of clauses are supposed to exist, which is not always the case (in the example, $P(c)$ does not exist).

In this paper we overcome these problems by imposing a restriction on the *initial* set of constrained clauses. More precisely, they must fulfil a property which we have called *pureness*. In particular, clauses without constraints are pure. Working with pure sets has allowed us to apply a different lifting method in the proof of correctness of a completion procedure for ordering constrained first-order clauses with equality. This procedure is refutationally complete, even when using some powerful notions of redundant inferences and clauses. When the initial set of axioms has no constraints, these notions include the ones given in [BG 91] for clauses without constraints. Our methods allow to do very efficient theorem proving in first-order logic with equality, since the constraints generated during completion reduce the search space importantly. Moreover, complete sets of axioms (canonical sets of rewrite rules in the equational case) can be obtained more often with these techniques.

This paper is structured as follows. The basic definitions and notations are given in section 2. In section 3 we define a new inference system that is simpler and more restrictive than previous ones. Its inference rules are *strict superposition left* and *right*, *equality resolution* and (equality) *factoring*. The inference systems defined by Bachmair and Ganzinger (1990,1991) are also based on strict superposition left and right, and equality resolution, but they also include, apart from "normal" factoring, inference rules for *merging paramodulation* [BG 90,91] or *equality factoring left* and *equality factoring right* [BG 90].

In section 4, we prove that pureness is preserved during completion and that pure sets of constrained clauses that are closed up to redundant inferences contain the empty clause iff they are inconsistent.

In section 5 it is shown how these complete sets of constrained clauses can be obtained in practice, even when *redundant* constrained clauses are deleted during the process. We also briefly outline some simplification methods included into the notions of redundant inferences and clauses.

2. Basic notions and terminology

We denote the set of first-order terms built from sets of function symbols F and variables V by $\mathcal{T}(F,V)$, and we denote substitutions by σ, σ', etc, and their application to a term t by $t\sigma$. Occurrences in terms are denoted by u, u', etc. If u is an occurrence of a subterm s in a term t, then t/u denotes s, and $t[u \leftarrow t']$ denotes the result of the replacement in t of the subterm at occurrence u by t'. The set of variables of a term t is denoted by $vars(t)$.

By *ordering constraints* we mean quantifier-free first-order formulae built over the binary predicate symbols $\succ$ and $=$ relating terms in $\mathcal{T}(F,V)$. The predicate $=$ denotes syntactic equality, whereas the predicate $\succ$ denotes a given simplification ordering that is total on ground terms. The satisfiability problem for this kind of constraints is decidable [Com 90] for the case in which $\succ$ is interpreted as the lexicographic path ordering. In [JO 91] this result is extended for the recursive path ordering with status.

We sometimes write $\succ_t$ instead of $\succ$ when comparing ground terms in order to emphasize that in this case the ordering is total. We use $\succcurlyeq_t$ to denote the multiset extension of $\succ_t$. In this ordering, the special symbols $\bot$ and *true* will be the smallest symbols, with $true \succ_t \bot$. Note that in fact no negations are needed in ordering constraints, since $t \not\succ_t t'$ is equivalent to $t' \succ_t t \vee t = t'$, and $t \neq t'$ is equivalent to $t' \succ_t t \vee t \succ_t t'$.

An *equation* is a multiset $\{t, t'\}$, denoted by $t \doteq t'$, where t and t' are terms. A first-order clause is a pair of (finite) multisets of equations Γ and Δ, denoted by $\Gamma \rightarrow \Delta$. The sets Γ and Δ are called respectively the *antecedent* and the *succedent* of the clause. In this note, distinct clauses are supposed not to share variables.

For simplicity, we express atoms by equations $P(t_1, \ldots, t_n) \doteq true$, where P is an n-ary predicate symbol, $t_1 \ldots t_n$ are terms, and *true* is a special symbol, i.e. we treat atoms as terms. No significant changes have to be made when distinguishing between atoms and terms e.g. in inference rules or by using orderings with different layers for atoms and for terms.

An equality Herbrand interpretation is a congruence on ground terms. An interpretation I satisfies a ground clause $\Gamma \rightarrow \Delta$, denoted by $I \models \Gamma \rightarrow \Delta$, if $I \not\supseteq \Gamma$ or else $I \cap \Delta \neq \emptyset$. The empty clause $\Box$ is therefore satisfied by no interpretation. Semantic deduction of a ground clause C from a set of ground clauses S, i.e. C is satisfied by every interpretation that satisfies S, will be denoted by $S \models C$.

A first-order clause with ordering constraint is a pair (C, T), denoted $C \,[\![T]\!]$, where C is a first-order clause, and T is an ordering constraint. A first-order clause with constraint $C \,[\![T]\!]$ can be seen as a shorthand for the set $G(C,T)$ of *ground instances* of $C \,[\![T]\!]$: those ground clauses $C\sigma$ such that $T\sigma$ is true under the given ordering. An interpretation I satisfies a clause with ordering constraint $C \,[\![T]\!]$ if it satisfies every clause in $G(C,T)$. *Tautologies* are constrained clauses that are satisfied by every interpretation. Clauses with an unsatisfiable constraint are tautologies. The *constrained empty clause* is an empty clause with a satisfiable constraint. It is satisfied by no interpretation. An interpretation I satisfies a set of (constrained) clauses S, denoted by $I \models S$, if I satisfies every clause in S. Semantic deduction of an equation or clause C from a set of equations and clauses S, i.e. C is satisfied by every model of S, will be denoted by $S \models C$.

A constrained clause $\Gamma \rightarrow \Delta \,[\![T]\!]$ *subsumes* another constrained clause $\Gamma' \rightarrow \Delta' \,[\![T']\!]$ if

there exists a substitution σ s.t. $\Gamma\sigma \subseteq \Gamma'$ and $\Delta\sigma \subseteq \Delta'$, and T' implies $T\sigma$. A constrained clause $C\,[\![T]\!]$ properly subsumes $C'\,[\![T']\!]$, denoted by $C'\,[\![T']\!] \rhd C\,[\![T]\!]$, if $C\,[\![T]\!]$ subsumes $C'\,[\![T']\!]$, but not vice versa, i.e. $\rhd$ is the subsumption ordering on constrained clauses, where more *general* clauses are smaller.

We use ground rewrite rules with the usual definitions. By a ground rewrite rule $t \Rightarrow t'$ we mean an oriented ground equation $t \doteq t'$. If R is a set of ground rewrite rules, then we denote by $\rightarrow_R$ the smallest relation such that $t \rightarrow_R t'$ whenever $t \Rightarrow t' \in R$, and such that $s[t] \rightarrow_R s[t']$ whenever $t \rightarrow_R t'$, for all ground terms s, t and t'. A set of such rules R is ground *confluent* if for every pair of ground terms u and u', $R \models u \doteq u'$ if and only if $u \rightarrow_R^* v$ and $u' \rightarrow_R^* v$ for some ground term v, where $\rightarrow_R^*$ is the transitive reflexive closure of $\rightarrow_R$. Sets of ground rewrite rules R such that there are no infinite sequences $t_1 \rightarrow_R \ldots \rightarrow_R t_n \rightarrow_R \ldots$ are said to have the *termination* property. Systems with the confluence and termination properties are called *canonical*. A ground term t is *irreducible* or in *normal form* w.r.t. R if there is no term t' s.t. $t \rightarrow_R t'$. In canonical systems every ground term has one unique normal form, and it can be decided whether $R \models t \doteq t'$ by reducing the ground terms t and t' to their normal forms and checking whether these normal forms are identical.

3. The inference system $\mathcal{R}$

In the following ordering $\succ_c$ on ground clauses, the terms appearing in antecedents of clauses are slightly more complex than the ones in succedents:

Definition 1: The *multiset expression* of an equation $t \doteq t'$ in a clause $\Gamma \rightarrow \Delta$ is

(i) $\{\{t, \bot\}, \{t', \bot\}\}$ if $t \doteq t'$ belongs to Γ
(ii) $\{\{t\}, \{t'\}\}$ if $t \doteq t'$ belongs to Δ

The ordering $\succ_e$ on equations appearing in ground clauses is defined as the multiset extension of $\gg_t$ on their multiset expressions.
The ordering $\succ_c$ on ground clauses is defined as the ordering $\gg_e$ on the multisets containing the multiset expressions of their equations.

Definition 2: A ground equation e is called *maximal* (resp. *strictly maximal*) w.r.t. $\succ_e$ in a ground clause C if $e \succeq_e e'$ ($e \succ_e e'$) for every other equation e' in C.

Inferences between constrained clauses can be seen as a shorthand for the inferences that can be made between all the ground instances of these constrained clauses. However, we are only interested in those ground inferences that take place in equations of succedents that are strictly maximal and in equations of antecedents that are maximal. Moreover, only the maximal terms in each equation have to be considered. In the following inference system, these facts are included in the constraint of the conclusion of each inference.

Definition 3: The inference rules of $\mathcal{R}$ are the following (we always consider maximality of equations in clauses w.r.t. $\succ_e$, and suppose that no superpositions below variables are made):

1) *strict superposition right:*

$$\frac{\Gamma' \rightarrow \Delta', s \doteq s' \; [\![T']\!] \qquad \Gamma \rightarrow \Delta, t \doteq t' \; [\![T]\!]}{(\; \Gamma', \Gamma \rightarrow \Delta', \Delta, t[u \leftarrow s'] \doteq t' \;)\sigma \; [\![T\sigma \wedge T'\sigma \wedge T''\sigma]\!]} \quad \text{if } \sigma = mgu(t/u, s)$$

where T'' includes:

a) $t \succ t'$, $s \succ s'$, and $t \doteq t' \succ_e s \doteq s'$;
b) $s \doteq s'$ is strictly maximal in $\Gamma' \rightarrow \Delta', s \doteq s'$;
c) $t \doteq t'$ is strictly maximal in $\Gamma \rightarrow \Delta, t \doteq t'$.

2) *strict superposition left:*

$$\frac{\Gamma' \rightarrow \Delta', s \doteq s' \; [\![T']\!] \qquad \Gamma, t \doteq t' \rightarrow \Delta \; [\![T]\!]}{(\; \Gamma', \Gamma, t[u \leftarrow s'] \doteq t' \rightarrow \Delta', \Delta \;)\sigma \; [\![T\sigma \wedge T'\sigma \wedge T''\sigma]\!]} \quad \text{if } \sigma = mgu(t/u, s)$$

where T'' includes:

a) $t \succ t'$ and $s \succ s'$;
b) $s \doteq s'$ is strictly maximal in $\Gamma' \rightarrow \Delta', s \doteq s'$;
c) $t \doteq t'$ is maximal in $\Gamma, t \doteq t' \rightarrow \Delta$.

3) *equality resolution:*

$$\frac{\Gamma, t \doteq t' \rightarrow \Delta \; [\![T]\!]}{\Gamma\sigma \rightarrow \Delta\sigma \; [\![T\sigma \wedge T'\sigma]\!]} \quad \text{where } \sigma = mgu(t, t')$$

where T' includes:

a) $t \doteq t'$ is maximal in $\Gamma, t \doteq t' \rightarrow \Delta$.

4) *factoring:*

$$\frac{\Gamma \rightarrow \Delta, t \doteq s, t' \doteq s' \; [\![T]\!]}{(\; \Gamma, s \doteq s' \rightarrow \Delta, t \doteq s \;)\sigma \; [\![T\sigma \wedge T'\sigma]\!]} \quad \text{where } \sigma = mgu(t, t')$$

where T' includes:

a) $t \succ s$ and $t' \succ s'$;
b) $t \doteq s$ is maximal in $\Gamma \rightarrow \Delta, t \doteq s, t' \doteq s'$.

Note that our inference rule for factoring is a generalization to the equality case of "normal" factoring. For instance, if t and t' are atoms, then both s and s' are the symbol *true* and the equation $true \doteq true$ can be omitted in the antecedent.

4. Complete sets of constrained clauses

Definition 4: A ground substitution σ is *irreducible* w.r.t. a set of ground rewrite rules R if $x\sigma$ is irreducible w.r.t. R, for every variable x in the domain of σ.

Below we define a method for associating to a set of constrained clauses S an interpretation I_S, which will be expressed by a (possibly infinite) canonical set of ground rewrite rules R_S.

Definition 5: Let S be a set of constrained clauses S, let $C\,[\![T]\!]$ be a constrained clause in S, and let $C\sigma$ be $\Gamma \rightarrow \Delta, t \doteq s$ for some ground substitution σ. Then $C\sigma$ *generates* a rule $t \Rightarrow s$ if the following conditions hold:

(1) $T\sigma$ is true
(2) $R_C \not\models C\sigma$
(3) $t \succ_t s$
(4) $t \doteq s$ is maximal (w.r.t. $\succ_e$) in $C\sigma$
(5) $R_C \not\models s \doteq s'$, for every $t \doteq s'$ in Δ
(6) t is not reducible by R_C
(7) σ is not reducible by R_C

where R_C is the set of rules generated by ground instances smaller than C (w.r.t. $\succ_c$) of constrained clauses in S. The set of rules generated by all ground instances of constrained clauses in S is denoted by R_S, and the interpretation I_S is defined as the congruence generated by R_S.

Note that for every set S the set R_S is canonical, since $t \succ_t s$ for every rule $t \Rightarrow s$, and there are no overlappings between left hand sides of rules in R_S. Another fact that will be used below is that rules with a left hand side t cannot be used in rewrite proofs of equations $s \doteq s'$, where s and s' are smaller than t.

As in [BG 90], our technique for completion is based on the notion of *redundant inference:* intuitively, a ground inference is redundant if its conclusion can already be deduced from other clauses which are smaller than the maximal premise. However, in order to deal with the constrained case, in each set S we consider only those ground instances with substitutions σ that are irreducible w.r.t. R_S:

Definition 6: Let S be a set of constrained clauses, and let R_S be the set of rules generated by S. Moreover, let π be an inference of $\mathcal{R}$ with premises $C_1\,[\![T_1]\!], \ldots, C_n\,[\![T_n]\!]$ and conclusion $C\,[\![T]\!]$.

a) A *ground instance* $\pi\sigma$ of π is any inference of $\mathcal{R}$ with premises $C_1\sigma\,[\![T_1\sigma]\!], \ldots, C_n\sigma\,[\![T_n\sigma]\!]$ and conclusion $C\sigma\,[\![T\sigma]\!]$, for a ground substitution σ such that $T\sigma$ is true.

b) The inference π is *redundant* in S if, for every ground instance $\pi\sigma$ of π such that σ is irreducible w.r.t. R_S, there exist ground instances $D_1\sigma_1, \ldots, D_m\sigma_m$ of constrained clauses $D_1\,[\![T'_1]\!], \ldots, D_m\,[\![T'_m]\!]$ in S such that $\{D_1\sigma_1, \ldots, D_m\sigma_m\} \models C\sigma$ and $max_{\succ_c}(C_1\sigma, \ldots, C_n\sigma) \succ_c D_i\sigma_i$ and σ_i is irreducible w.r.t. R_S for $1 \leq i \leq m$.

c) The set S is *$\mathcal{R}$-complete* if every inference of $\mathcal{R}$ with premises in S is redundant in S.

The following lemma states that for any $\mathcal{R}$-complete set of clauses S that does not contain the empty clause, I_S satisfies $C\sigma$, for every ground instance of a clause $C\,[\![T]\!]$ in S such that σ is an irreducible ground substitution. Later on, we will see that if S is *pure* w.r.t. R_S, then I_S satisfies *every* ground instance of the clauses in S, i.e. $I_S \models S$.

Lemma 7: Let S be an $\mathcal{R}$-complete set of constrained clauses that does not contain the constrained empty clause. Then $I_S \models C\sigma$ for every constrained clause $C\,[\![T]\!]$ in S with a ground instance $C\sigma$ such that σ is irreducible w.r.t. R_S.

Proof. Let $C\sigma$ be a minimal (w.r.t. $\succ_c$) ground instance of a constrained clause $C\,[\![T]\!]$ in S with σ irreducible w.r.t. R_S such that $I_S \not\models C\sigma$. We will derive a contradiction from the existence of such a clause. There are several cases to be analyzed, depending on the maximal equation in $C\sigma$:

a) Let $C\sigma$ be a clause $\Gamma\sigma \rightarrow \Delta\sigma, t\sigma \doteq t'\sigma$, with a maximal equation $t\sigma \doteq t'\sigma$, and $t\sigma \succ_t t'\sigma$. Since $I_S \not\models C\sigma$, the clause $C\sigma$ has not generated the rule $t\sigma \Rightarrow t'\sigma$. This must be because one of the conditions 5) or 6) of definition 5 do not hold.

a1) If condition 5) does not hold, then $\Delta\sigma$ must be of the form $\Delta'\sigma, s\sigma \doteq s'\sigma$, where $t\sigma$ is the same term as $s\sigma$ and $R_C \models t'\sigma \doteq s'\sigma$. In this case, consider the following inference π by factoring

$$\frac{\Gamma \rightarrow \Delta', t \doteq t', s \doteq s'\,[\![T]\!]}{(\ \Gamma, t' \doteq s' \rightarrow \Delta', t \doteq t'\ )\sigma''\,[\![T\sigma'' \wedge T'\sigma'']\!]} \quad \text{where } \sigma'' = mgu(t,s)$$

Its conclusion has a ground instance D of the form $\Gamma\sigma, t'\sigma \doteq s'\sigma \rightarrow \Delta\sigma, t\sigma \doteq t'\sigma$ that is not satisfied by I_S. Moreover, D is an instance of this conclusion with a ground substitution σ' such that $\sigma = \sigma''\sigma'$. Therefore, σ' must be irreducible by R_S, as σ is.

Since S is $\mathcal{R}$-complete, π must be redundant in S. But then there exist ground instances $D_1\sigma_1, \ldots, D_m\sigma_m$ of constrained clauses $D_1\,[\![T'_1]\!], \ldots, D_m\,[\![T'_m]\!]$ in S such that $\{D_1\sigma_1, \ldots, D_m\sigma_m\} \models D$, $C\sigma \succ_c D_i\sigma_i$ and σ_i irreducible w.r.t. R_S for $1 \leq i \leq m$. The fact that D is not satisfied by I_S implies that at least one of the $D_i\sigma_i$ is not satisfied in I_S either, which contradicts the fact that $C\sigma$ is a minimal such clause.

a2) If condition 6) does not hold, then $t\sigma$ is reducible by R_C, e.g. with a rule $s\sigma' \Rightarrow s'\sigma'$ generated by a clause $C'\sigma'$ smaller than $C\sigma$. Let C' be a clause $\Gamma' \rightarrow \Delta', s \doteq s'$ in S and $t\sigma/u = s\sigma'$. Now the following inference π by strict superposition right

$$\frac{\Gamma' \rightarrow \Delta', s \doteq s'\,[\![T']\!] \qquad \Gamma \rightarrow \Delta, t \doteq t'\,[\![T]\!]}{(\ \Gamma', \Gamma \rightarrow \Delta', \Delta, t[u \leftarrow s'] \doteq t'\ )\sigma''\,[\![T\sigma'' \wedge T'\sigma'' \wedge T''\sigma'']\!]} \quad \text{where } \sigma'' = mgu(t/u, s)$$

can be made, since all the conditions for its application hold. Its conclusion has a ground instance D of the form $\Gamma'\sigma', \Gamma\sigma \rightarrow \Delta'\sigma', \Delta\sigma, t\sigma[u \leftarrow s'\sigma'] \doteq t'\sigma$, that is not satisfied by I_S. Moreover, D is an instance of this conclusion with a ground substitution τ that is irreducible by R_S. This is easy to see taking into account that σ and σ' are irreducible by R_S. Since S is $\mathcal{R}$-complete, π must again be redundant in S, which, as in the previous case, leads to a contradiction with the minimality of $C\sigma$.

b) If $C\sigma$ is a clause $t\sigma \doteq t'\sigma, \Delta\sigma \rightarrow \Gamma\sigma$, where $t\sigma \doteq t'\sigma$ is maximal in $C\sigma$, and $t\sigma$ is $t'\sigma$, then consider the following equality resolution inference:

$$\frac{\Gamma, t \doteq t' \rightarrow \Delta \; [\![T]\!]}{\Gamma\sigma' \rightarrow \Delta\sigma' \; [\![T\sigma' \wedge T'\sigma']\!]} \quad \text{where } \sigma' = mgu(t, t')$$

The conclusion of this inference has a ground instance $\Gamma\sigma \rightarrow \Delta\sigma$, that is not satisfied by I_S. Since the inference is redundant, as above, a contradiction is obtained.

c) The only remaining case is that $C\sigma$ is a clause $\Gamma\sigma, t\sigma \doteq t'\sigma \rightarrow \Delta\sigma$, where $t\sigma \doteq t'\sigma$ is maximal in $C\sigma$ and $t\sigma \succ_t t'\sigma$. In this case $I_S \models t\sigma \doteq t'\sigma$, because $I_S \not\models C\sigma$. Then $t\sigma$ must be reducible by a rule $s\sigma' \Rightarrow s'\sigma'$ in R_S generated by a clause in S of the form $\Gamma' \rightarrow \Delta', s \doteq s' \; [\![T']\!]$, where $t\sigma/u = s\sigma'$. The following inference π by strict superposition left can then be made:

$$\frac{\Gamma' \rightarrow \Delta', s \doteq s' \; [\![T']\!] \qquad \Gamma, t \doteq t' \rightarrow \Delta \; [\![T]\!]}{(\; \Gamma', \Gamma, t[u \leftarrow s'] \doteq t' \rightarrow \Delta', \Delta \;)\sigma \; [\![T\sigma'' \wedge T'\sigma'' \wedge T''\sigma'']\!]} \quad \text{where } \sigma'' = mgu(t/u, s)$$

For the instance $\pi\sigma$ of the inference, $C\sigma$ is the maximal premise, and, as in case a2), its conclusion is not satisfied by I_S. This implies, as before that, since π is redundant, a contradiction is obtained. ∎

Intuitively, the property of *pureness* of a set of constrained clauses S w.r.t. a set of ground rewrite rules R means that every ground instance of a clause in S is deducible from R and other ground instances* with substitutions that are irreducible w.r.t. R:

Definition 8: Let S be a set of constrained clauses and let R be a set of ground rewrite rules. The set S is *pure* w.r.t. R iff for every ground instance D of a clause in S, there exist ground instances $C_1\sigma_1, \ldots, C_n\sigma_n$ of clauses $C_1 \; [\![T_1]\!], \ldots, C_n \; [\![T_n]\!]$ in S with σ_i irreducible w.r.t. R for $1 \leq i \leq n$, such that $R \cup \{C_1\sigma_1, \ldots, C_n\sigma_n\} \models D$.
The set S is *pure* iff S is pure w.r.t. every set of ground rewrite rules R.

For example, sets of clauses without constraints are pure, since, for every set of ground rewrite rules R, each ground instance $C\sigma$ of a clause C can be deduced from R and the instance $C\sigma'$, where σ' is the "normal form" w.r.t. R of the substitution σ.

The problem with constrained clauses is that $C\sigma'$ may not be a ground instance of a clause $C \; [\![T]\!]$, as $T\sigma'$ may be false. This leads us to another case of constrained clauses that are also pure (w.r.t. every R): the ones with constraints that do not impose lower bounds on the values of variables.

Further work remains to be done on the definition of sufficient conditions for pureness w.r.t. sets R. As the following theorem shows, this will be especially interesting for pureness w.r.t. the sets R_S of $\mathcal{R}$-complete sets S:

Theorem 9: Let S be an $\mathcal{R}$-complete set of constrained clauses that does not contain the constrained empty clause, and that is pure w.r.t. R_S. Then $I_S \models S$.

* Recall that $C\sigma$ is a ground instance of a constrained clause $C \; [\![T]\!]$ iff σ is ground and $T\sigma$ is true.

Proof. We show that $I_S \models C\sigma$, for every clause $C\,[\![T]\!]$ in S and every ground instance $C\sigma$ of $C\,[\![T]\!]$. By pureness, there exist ground instances $C_1\sigma_1, \ldots, C_n\sigma_n$ of clauses $C_1\,[\![T_1]\!], \ldots, C_n\,[\![T_n]\!]$ in S with σ_i irreducible w.r.t. R_S for $1 \leq i \leq n$, such that $R \cup \{C_1\sigma_1, \ldots, C_n\sigma_n\} \models C\sigma$. Moreover, in the previous lemma, it was proved that $I_S \models C_i\sigma_i$ since the σ_i are irreducible w.r.t. R_S. But then also $I_S \models C\sigma$, since I_S is the congruence generated by R_S. ∎

The previous result implies that $\mathcal{R}$-complete sets S that are pure w.r.t. R_S contain the constrained empty clause iff they are inconsistent (since S has the model I_S if it does not contain the empty clause). Note that the sets S of the counter examples given in section 1 are not pure w.r.t. R_S.

In order to use the previous results for obtaining a refutationally complete theorem proving procedure, in the following section we show how these $\mathcal{R}$-complete sets can be effectively computed in practice.

5. Constrained completion procedures

Completion procedures will be algorithms that compute *$\mathcal{R}$-complete* sets S that are pure w.r.t. R_S. These algorithms will be modelled by *theorem proving derivations* in which there are two basic operations on sets of constrained clauses: adding consequences and deleting *redundant constrained clauses.*

Definition 10: A theorem proving derivation is a sequence of sets of constrained clauses $S_1, S_2, \ldots$ with:

(i) $S_i = S_{i-1} \cup \{C\,[\![T]\!]\}$ where $S_{i-1} \models C\,[\![T]\!]$, or
(ii) $S_i = S_{i-1} - \{C\,[\![T]\!]\}$ if $C\,[\![T]\!]$ is *redundant* in S_{i-1}.

where the set S_∞ of *persistent* constrained clauses in $S_1, S_2, \ldots$ is defined as $\cup_j(\cap_{k \succeq j} S_k)$ and where a constrained clause $C\,[\![T]\!]$ is redundant in a set S_j if $C\,[\![T]\!] \triangleright C'\,[\![T']\!]$, for some $C'\,[\![T']\!]$ in S_j, or else for every ground instance $C\sigma$ of $C\,[\![T]\!]$ with σ irreducible w.r.t. R_{S_∞}, there exist ground instances $D_1\sigma_1, \ldots, D_m\sigma_m$ of constrained clauses $D_1\,[\![T'_1]\!], \ldots, D_m\,[\![T'_m]\!]$ in S_j such that $\{D_1\sigma_1, \ldots, D_m\sigma_m\} \models C\sigma$ and $C\sigma \succ_c D_i\sigma_i$ and σ_i is irreducible w.r.t. R_{S_∞} for $1 \leq i \leq m$.

Intuitively, a constrained clause is redundant if every ground instance (with a substitution that is irreducible w.r.t. R_{S_∞}) of it can be deduced from smaller ground instances (with ground substitutions that are also irreducible w.r.t. R_{S_∞}) of other constrained clauses. Obviously, this is a rather theoretical notion of redundancy. How can we compute theorem proving derivations in practice?

Let us first analyze the case in which the initial set of axioms has no constraints. It is then sound to eventually eliminate the constraint of a clause, since the inference rules we use are sound when used without constraints. It is not difficult to see that then in fact the same simplification methods as in the absence of constraints can be used, provided the constraints of the clauses used in the simplification process are eliminated after. Therefore, our methods properly include all the work done for the case without

constraints. Simplification of a clause is done by first adding the clause obtained by simplification, and then deleting simplified clause, which has become redundant. Some known simplification methods are rewriting, contextual rewriting [ZR 85] clausal rewriting [Nie 90, NO 91], etc.

We are currently working on more powerful redundancy notions that allow to keep the constraints of the clauses used to prove the redundancy of other ones. Notions such as the one of constrained simplification by rewriting, as defined in [KKR 90] could be used in that case.

On the other hand, when the initial set of clauses does have constraints, then still some simplification notions such as the elimination of tautologies, the elimination of redundant literals in clauses, subsumption of clauses, etc. fit into our current notion of redundancy.

The following lemma states that pureness is preserved during completion:

Lemma 11: If $S_1, S_2, \ldots$ is a theorem proving derivation where S_1 is pure, then S_j is pure, for every $j \geq 1$.

Definition 12: A theorem proving derivation $S_1, S_2, \ldots$ is *fair* w.r.t. $\mathcal{R}$ if every inference of $\mathcal{R}$ with premises in S_∞ is redundant in some S_j.

Lemma 13: Let $S_1, S_2, \ldots$ be a theorem proving derivation that is fair w.r.t. $\mathcal{R}$. If an inference is redundant in some S_j, then it also is in S_∞. Moreover, S_∞ is $\mathcal{R}$-complete.

Theorem 14: Let $S_1, S_2, \ldots$ be a theorem proving derivation that is fair w.r.t. $\mathcal{R}$, where S_1 is pure. Then S_1 is inconsistent if and only if the constrained empty clause belongs to some S_i.

The previous results state that fair theorem proving derivations are indeed refutationally complete procedures. We think that $\mathcal{R}$-complete systems can be obtained in many cases using our framework, since the use of constraints restricts the number of inferences to be computed importantly. Theorem proving using $\mathcal{R}$-complete systems S can be done very efficiently, since no inferences have to be computed between clauses in S. Moreover, $\mathcal{R}$-complete systems of equations are ground confluent rewrite systems, as we show below.

Definition 15: Let S be a set of constrained equations. A ground term t can be rewritten into a ground term t', denoted $t \rightarrow_S t'$, by one reductive constrained rewrite step with an equation $s \doteq s' \; [\![T]\!]$ of S, if there exists a ground substitution σ s.t.

$t/u = s\sigma$,

$T\sigma \wedge s\sigma \succ_t s'\sigma$ is true and

$t' = t[u \leftarrow s'\sigma]$.

We denote by $\rightarrow_S^*$ the reflexive transitive closure of $\rightarrow_S$. A term t' is a *normal form* of t w.r.t. S if $t \rightarrow_S^* t'$ and there is no term t'' s.t. $t' \rightarrow_S t''$. We say that S is a complete set of reductions if for every pair of ground terms t and t' s.t. s and s' are normal forms of t and t' respectively, the terms s and s' are syntactically equal iff $S \models t = t'$.

Theorem 16: Let S be an $\mathcal{R}$-complete set of constrained equations that is pure w.r.t. R_S. Then S is a complete set of reductions.

Proof. Let t and t' be ground terms and let s and s' be normal forms of t and t' respectively. We prove that s and s' are syntactically equal iff $S \models t = t'$. The only-if-part is trivial. For the if-part, we proceed by induction on the size of $\{t, t'\}$ w.r.t. $\gg_t$. If t or t' is reducible by S, then by the induction hypothesis, for the equation obtained the result holds. Moreover, since $S \models t = t'$, and S is $\mathcal{R}$-complete and pure w.r.t. R_S, it holds that $I_S \models S$, and thus $I_S \models t = t'$, and also $R_S \models t = t'$. Therefore, if t and t' are not syntactically equal, it cannot happen that t nor t' is reducible by S, since, if e.g. $t \succ_t t'$, t must be reducible by R_S, and also by S. ∎

Acknowledgements: We wish to thank Fernando Orejas for his interest and advice on this work, and also Harald Ganzinger and Hélène Kirchner for their E-mailed comments.

6. References

[Bür 85] H-J. Bürckert: Extending the Warren abstract machine to many-sorted Prolog. SEKI-Report 85-VII-KL. Univ. Kaiserslautern (1985).

[BDP 89] L. Bachmair, N. Dershowitz, D. Plaisted: Completion without failure. In H. Ait-Kaci and M. Nivat, editors, Resolution of equations in algebraic structures, vol 2: Rewriting Techniques, pp 1-30, Academic Press, (1989).

[BG 90] L. Bachmair, H. Ganzinger: On restrictions of ordered paramodulation with simplification. In Proc. 10th Int. Conf. on Automated Deduction. Kaiserslautern, 1990. LNCS, pp 427-441.

[BG 91] L. Bachmair, H. Ganzinger: Completion of first order clauses with equality. (final version) 2nd Intl. Workshop on Conditional and Typed Term Rewriting, Montreal (1991). To appear in LNCS.

[Com 90] H. Comon: Solving Symbolic Ordering Constraints. In proc. 5th IEEE Symp. Logic in Comp. Sc. Philadelphia. (June 1990).

[HR 89] J. Hsiang, M. Rusinowitch: Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. Submitted for publication (1989).

[Hue 72] G. Huet: Constrained Resolution: A complete method for higher-order logic. PhD. Thesis. Case Western Reserve University. 1972.

[JO 91] J-P. Jouannaud, M. Okada: Satisfiability of systems of ordinal notations with the subterm property is decidable. To appear in proc. ICALP 1991. Madrid. LNCS (1991)

[KK 89] C. and H. Kirchner: Constraint Equational Reasoning. In Proc. ISSAC 89. Portland, Oregon. (1989)

[KKR 90] C. and H. Kirchner, M. Rusinowitch: Deduction with Symbolic Constraints. Revue Francaise d'Intelligence Artificielle. Vol 4. No. 3. pp. 9-52. Special issue on automatic deduction. (1990).

[KB 70] D.E. Knuth, P.B. Bendix: Simple word problems in universal algebras. J. Leech, editor, Computational Problems in Abstract Algebra, 263-297, Pergamon Press, Oxford, 1970.

[Nie 90] R. Nieuwenhuis: Theorem proving in first order logic with equality by clausal rewriting and completion. PhD thesis, UPC Barcelona, 1990.

[NO 91] R. Nieuwenhuis, F. Orejas: Clausal Rewriting. 2nd Intl. Workshop on Conditional and Typed Term Rewriting, Montreal (1991). To appear in LNCS.

[Pet 90] G.E. Peterson: Complete Sets of Reductions with Constraints. In Proc. 10th Int. Conf. on Automated Deduction. Kaiserslautern, 1990. LNCS, pp 381-395.

[Rus 87] M. Rusinowitch: Theorem-proving with resolution and superposition: an extension of Knuth and Bendix procedure as a complete set of inference rules. Report 87-R-128, CRIN, Nancy, 1987.

[ZR 85] Zhang H., Rémy J.L. Contextual Rewriting, Proc. of the 1st International Conference on Rewriting Techniques and Applications, J.P. Jouannaud ed., L.N.C.S. No. 202, Springer-Verlag, Berlin (1985)