J. Symbolic Computation (1991) **11**, 21–49

Theorem-proving with Resolution and Superposition

MICHAEL RUSINOWITCH

CRIN, Campus Scientifique BP 239, 54506 *Vandoeuvre les Nancy, France*

(*Received* 15 *September* 1987)

We present a refutationally complete set of inference rules for first-order logic with equality. Except for x = x, no equality axioms are needed. Equalities are oriented by a well-founded ordering and can be used safely for demodulation without losing completeness. When restricted to equational logic, this strategy reduces to a Knuth–Bendix procedure.

1. Introduction

The starting point of this work is the following remark in (Peterson 1983): "...,no one has developed a refutation complete set of inference rules for all of first-order logic with equality which reduces to the Knuth-Bendix procedure when restricted to equality units.". We present here one such a set of inference rules when a complete simplification ordering is used to compare terms. Intuitively, when paramodulating between two positive equational literals in two different clauses, our inference rules enable us to only paramodulate between the larger sides of the equalities. One aim of this paper is to prove the refutational completeness of a strategy based on this notion.

A fundamental method to speed up theorem provers is to maintain information under a reduced format and to discard redundancy. This goal is achieved by using deletion inference rules such as demodulation (Wos et al. 1967), subsumption and tautology deletion. In most strategies they are just considered as very efficient heuristics but little is known about their effect on completeness. In our case, we are able to incorporate the deletion rules in the same framework as the other inference rules and to show easily that completeness is preserved.

* A preliminary version of the results in this paper has been presented at the International Conference on Fifth Generation Computer Systems (Tokyo 1988).

0747-7171/91/010049+29 $03.00/0

When all the clauses are orientable equations, the previous strategy reduces to a Knuth-Bendix algorithm (1970). Our result may also be viewed as an extension of the unfailing completion procedures of (Hsiang Rusinowitch 1987) or (Bachmair Dershowitz Plaisted 1987) to the general first order predicate calculus with equality.

We emphasize the fact that this procedure does not use the functional reflexive axioms, and never performs paramodulation into a variable subterm. These restrictions are crucial in order to have an efficient paramodulation-based theorem-prover. Lankford has proved the completeness of this strategy in the special case where the equality predicate does not occur positively in non-unit clauses and the initial set of equations is a complete set of reductions (Lankford 1975). Paul (Paul 1985) has studied the case of Horn clauses. However, his algorithm fails, just like the Knuth -Bendix algorithm, when there is an equation which cannot be oriented. His strategy also has a bigger search space since it does not preclude the replacement of subterms within right-hand sides of equations in non-unit clauses. The same remark is true for the unit strategy for Horn clauses proposed by (Bachmair Dershowitz Plaisted 1987). A very similar procedure described in (Fribourg 1985) allows any orientation of equations (not only reduction orderings). However, the functional reflexive axioms and paramodulation into variables are required to ensure the completeness of the method. Furthermore Fribourg did not show that completeness is maintained when simplification and subsumption rules are added to the system.

Our completeness proof uses the notion of transfinite semantic trees (as in Hsiang Rusinowitch 1986) and an extension of the notion of failure node which we call *quasi-failure node*. A quasi-failure node can be viewed as a partial interpretation J which falsifies a clause reduced by valid rules of J. Quasi-failure nodes are essential for proving that paramodulation in the smallest term of an equation is not needed. For proving completeness of ordered paramodulation (Hsiang Rusinowitch 1986), we show that the rightmost branch of the semantic tree associated with an unsatisfiable set of clauses is empty. If this branch contains a quasi-failure node, the proof does not generalize to our actual set of rules. Therefore, the main point of our proof below is to build a branch which avoids quasi-failure nodes.

2. Inference Rules

2.1. NOTATIONS.

In this section we review some standard concepts and notation. Let F be a set of function symbols graded by an arity function. Let X be a set of variables. The algebra of terms on F and X is denoted by T(F,X). We call T(F) the set of ground terms on F, which is the set of terms with no variables. Let P be a set of predicate (or relation) symbols. The equality symbol "=" is a particular element of P whose arity is 2. The set of atomic formulas (or atoms) is denoted by A(P,F,X), and the set of ground atoms (or atoms with no variables) by A(P,F). An equality is an atom whose predicate symbol is "=". The set of literals is $A(P,F) \cup \neg A(P,F)$, where $\neg$ is the symbol of negation. A clause is a disjunction of

literals. A clause can be identified with the set of its literals. The expression $C \subseteq D$, where C and D are clauses means that the set of literals of C is included in the set of literals of D.

A substitution is a mapping σ from X to T(F,X) with $\sigma(x)=x$ almost everywhere. Substitutions are extended in the usual way to terms, atoms, literals and clauses; the result of applying a substitution σ to an object t is denoted by $t\sigma$. A substitution θ is a unifier of two objects s and t if and only if $s\theta=t\theta$. A unifier θ of s and t is the most general unifier(mgu) iff for every unifier σ of s and t there exists a substitution ϕ such that $\sigma=\theta\phi$ (the mgu is unique up to consistent renaming variables). If C_1 and C_2 are clauses in S such that C_1 has no more literals than C_2 and $C_1\theta \subseteq C_2$ for some substitution θ, then we say that C_1 *subsumes* C_2.

An important feature of our inference system is that any inference step always involves the maximal literal of one of the parent clauses, where the maximality notion is defined relatively to a *complete simplification ordering* $<$ on the Herbrand Universe (Peterson 1983, Hsiang Rusinowitch 1987). Our definition of such an ordering is a little more restrictive than the previous ones, since it requires the extra property O6. This is not a real drawback because, in practice, most simplification orderings satisfy it.

2.2. COMPLETE SIMPLIFICATION ORDERINGS.

A complete simplification ordering $<$ is an ordering on $A(P,F,X) \cup T(F,X)$ such that:

O1. *$<$ is well founded*

O2. *$<$ is total on $A(P,F) \cup T(F)$*

O3. *for every $w,v \in A(P,F,X) \cup T(F,X)$ and every substitution θ : $w < v$ implies $w\theta < v\theta$*

O4. *for every $t,s \in T(F,X)$ $t<s$ implies $w[o \leftarrow t] < w[o \leftarrow s]$*

O5. *for every $t,s,a,b \in T(F,X)$, with $t \leq s$ and $w \in A(P,F,X)$*

   1. *if s is a subterm of w and w is not an equality then $(s=t) < w$.*
   2. *if s is a strict subterm of a or b then $(s=t) < (a=b)$*

O6. *if $(u=w) < A < (u=v)$, $w<u$ and $v<u$, where u,v and w are ground terms, and A is a ground atom then there is a ground term t such that A is equal to the atom $(u=t)$.*

2.2.1. EXAMPLE

We assume that we have a total well-founded ordering $<_p$ on the predicate symbols such that "=" is the smallest element. We further suppose that $<_f$ is a simplification ordering (Dershowitz 1985) on the set of terms which is also total on ground terms. We define the predicate-first ordering $<$ on A(P,F) as follows:

$P(s_1,\ldots,s_n) < Q(t_1,\ldots,t_m)$ *if*

$P <_p Q$ *or*

$P = Q$, P is not the equality predicate and $(s_1,...,s_n) <_f (t_1,...,t_m)$ compared lexicographically, or
$P = Q$, P is the equality predicate, and $\{s_1,s_2\} < <_f \{t_1,t_2\}$, where $< <_f$ is the multiset extension of $<_f$

It is easy to see that < verifies O1,...,O6 and, in general, A(P,F) is not order-isomorphic to **N**. For instance, suppose there are only two predicate symbols = and P, one constant a and one unary function f. Assume that the recursive path ordering (Dershowitz 1982), with a<f is used to order terms. Then, the Herbrand universe is ordered as (atoms s=t and t=s are considered identical):

$$a=a < fa=a < fa=fa < ffa=a < \ldots < Pa < Pfa < Pffa < \ldots$$

2.3. THE SET OF INFERENCE RULES.

Now we give our set of inference rules, which is denoted by DRA. We suppose that < is an ordering that can be extended as a complete simplification ordering.

O-FACTORING

If $L_1,L_2,...,L_k$ are literals of a clause C which are unifiable with mgu θ, and for every atom $A \in C-\{L_1,...,L_2\}$, $L_1\theta \not\leq A\theta$, then $\Gamma = C\theta - \{L_2\theta ,..., L_k\theta\}$ is an O-factor of C.

O-RESOLUTION

If $C_1 = L_1 \vee C_1'$ and $C_2 = L_2 \vee C_2'$ are clauses such that

1. *L_1 and $\neg L_2$ are unifiable with mgu θ and*
2. *for every $A \in C_1'$, $L_1\theta \not\leq A\theta$ and*
3. *for every $A \in C_2'$, $L_2\theta \not\leq A\theta$ and*
4. *if L_1 is an equality literal then C_2 is $x=x$*

then $\Gamma = C_1'\theta \vee C_2'\theta$ is an O-resolvent of C_1 and C_2.

ORIENTED_PARAMODULATION

Let C_1 be a clause $(s=t) \vee C_1'$. Let C_2 be another clause which has a non-variable subterm s' at occurrence n in a literal L_2, such that s' is unifiable with s with mgu θ. We also assume that:

1. *$s\theta \not\leq t\theta$ and*
2. *for every $A \in C_2-\{L_2\}$, $L_2\theta \not\leq A\theta$ and*
3. *L_2 is not a positive equation.*

Then $C = (C_2[n \leftarrow t] \vee C_1')\theta$ is an oriented paramodulant of C_1 into node n of C_2.

EXTENDED_SUPERPOSITION

Let C_1 be a clause $(s=t) \vee C_1'$. Let C_2 be a clause and $a=b$ be a literal of C_2. Let s' be a non-variable subterm of a at occurrence n of C_2, such that s' is unifiable with s with mgu θ. We also assume that:

1. $s\theta \not\leq t\theta$ *and*
2. $a\theta \not\leq b\theta$ *and*
3. *for every* $A \in C_2 - \{a=b\}$, $a\theta = b\theta \not\leq A\theta$

then $C = (C_2[n \leftarrow t] \vee C_1')\theta$ *is an extended superposant of* C_1 *into node* n *of* C_2.

We remark that when C_1 and C_2 are two rewrite rules, an extended superposition of C_1 into C_2 is a superposition as in the Knuth-Bendix algorithm. Let us introduce now some deletion rules which are fundamental as far as efficiency is concerned.

We say that the clause C_1 *properly subsumes* C_2 if C_1 subsumes C_2 and C_2 does not subsume C_1. We shall use the following version of the subsumption rule:

PROPER_SUBSUMPTION

Delete from a given set of clauses S any clause which is properly subsumed by another clause in S.

The simplification rule is slightly more restrictive than the one which is used in completion procedures:

If the unit equation s=t is in S and $C_2[s\theta]$ is a clause in S which contains an instance $s\theta$ of s, and $s\theta > t\theta$, and there is an atom A in $C_2[s\theta]$ such that $A > (s\theta = t\theta)$, then the clause $C_2[t\theta]$ is a *simplification* of $C_2[s\theta]$ by s=t.

SIMPLIFICATION

One may replace in S a clause which has been simplified, by its simplification.

In the case where every clause is an equality or an inequality, the only applicable rules are EXTENDED SUPERPOSITION, RESOLUTION with x=x, PROPER SUBSUMPTION and SIMPLIFICATION. The strategy that we then get coincides with the S-strategy of (Hsiang Rusinowitch 1987). Furthermore, when there is no inequality in the system and every equality is orientable by means of our simplification ordering, the procedure applies the same inferences as in the Knuth and Bendix completion algorithm.

2.4. MAIN RESULT

We state now our main result, whose proof will be postponed to sections 5 and 6. For convenience, we shall call *INF* the subset of DRA made up of the non deletion-rules: O-RESOLUTION, ORIENTED PARAMODULATION, O-FACTORING, EXTENDED SUPERPOSITION. A fairness condition is needed to control an application of these rules, so that no crucial inference is delayed forever:

Given an initial set of clauses S, the derivation $S_0 \rightarrow S_1 \rightarrow ... \rightarrow S_i \rightarrow ...$ where S_i is obtained by application of a rule of DRA to S_{i-1} is *fair* if :

$$\textit{for all } j,\ R \in \cap_{i \geq j} \textit{INF}(S_i) \textit{ implies that}$$
$$R \textit{ is subsumed by some clause } C \in \cup_{i \geq 0} S_i$$

Here is an example of a fair strategy: first, all possible simplifications are performed, then clauses which are subsumed by other ones are deleted, then all resolutions, factorings, paramodulations and superpositions are created. We can now express the completeness of our rules:

2.4.1. THEOREM. Every fair derivation, whose initial set is E-unsatisfiable and contains the axiom x=x, yields the empty clause.

The proof is performed in two steps. First we consider only the inference rules of INF and use the semantic tree method as it is detailed in (Hsiang Rusinowitch 1988). Then we adapt this technique to take the deletion rules into account. Before we give the proofs, we illustrate the inference rules with examples.

3. Examples

The following easy example shows the transitivity of less-or-equal, assuming the associativity of max:

$$\textit{for every } u,v,w\ max(max(u,v),w) = max(u, max(v,w))$$
$$\Rightarrow \textit{for every } x,y,z\ (LE(x,y) \textit{ and } LE(y,z)) \Rightarrow LE(x,z).$$

The skolemized negation of the theorem is the conjunction of clauses 5,6,7. We use the predicate first ordering, as described in Example 2.2.1, with the following precedence on function symbols: max>a>b>c, and on predicate symbols: LE > "=" .

1. LE(x,y) \/ LE(y,x).
2. ¬LE(x,y) \/ max(x,y) → y.
3. ¬LE(y,x) \/ max(x,y) → x.
4. max(max(x,y),z) → max(x, max(y,z)).
5. LE(a,b).
6. LE(b,c).
7. ¬LE(a,c).

REFUTATION

8. max(a,b)→b by res of 5,2.
9. max(b,c)→c by res of 6,2.
10. LE(x,y) \/ max(x,y) → x. by res of 1,3.
11. max(a,c) → a by res of 7,10.
12. max(a,max(b,z))→max(b,z) by super of 8 into 4.
13. max(a,c) → max(b,c) by super of 9 into 12.
14. a→c by simplif of 13 by 11 and 9.
15. ¬LE(c,c) by simplif of 7 by 14.
16. LE(x,x) by fact of 1.
17. □ by res of 16 and 15.

Let us give now an example, borrowed from Brown's thesis (1974). It shows that the quotient of two squares of two numbers without common divisors is not a prime. (Of course, a more general statement is known, but its proof requires the use of induction).

{(for all z, (z divides a and z divides b) ⇒ z=1 or z=-1) and b.b.c=a.a}

⇒ c is prime

We use the following precedence on function symbols: | > . > - > + > b > c > a (the status of binary operators is left-right) and the empty precedence on predicate symbols.

AXIOMS FOR ADDITION AND MULTIPLICATION:

1. $(x+y)+z=x+(y+z)$.
2. $x+y=y+x$.
3. $o+x=x$
4. $x+(-x)=o$.
5. $(x.y).z=x.(y.z)$.
6. $x.y=y.x$.
7. $w.(x+y)=w.x+w.y$.
8. $(-x).y=-(x.y)$.
9. $x.y\neq o \vee x=o \vee y=o$.

PROPERTIES OF THE "DIVIDE" AND "PRIME" PREDICATES:

D1. $\neg D(x,y) \vee (y \mid x).x=y$.

D2. $D(x,y) \vee (y \mid x).x \neq y$.

D3. $\neg P(o)$

D4. $\neg P(1)$.

D5. $\neg P(-1)$.

D6. $\neg P(z) \vee \neg D(x,z) \vee x=1 \vee x=-1 \vee x=z \vee x=-z$.

D7. $\neg P(z) \vee \neg D(z,x.y) \vee D(z,x) \vee D(z,y)$.

D8. $(x.y) \mid y=x \vee y=o$.

NEGATION OF THE THEOREM:

H1. $P(c)$.

H2. $(b.b).c=a.a$.

H3. $\neg D(z,a) \vee \neg D(z,b) \vee z=1 \vee z=-1$.

REFUTATION

P0. y.(-x) = -(y.x) by two successive par of 6 into 8

P1. (a.a) | c=b.b \/ c=o by super H2 into D8

P2. ¬D(c,x.y) \/ D(c,x) \/ D(c,y) by res of D7,H1

P3. ((x.y) | c).c ≠ x.y) \/ D(c,x) \/ D(c,y) by res of P2,D2

P4. ((x.x) | c).c ≠ x.x) \/ D(c,x) by fact of P3

P5. ((x.x) | c).c ≠x.x \/ (x | c).c=x by res of D1,P4

P6. (b.b).c ≠ a.a \/ (a | c).c=a \/ c=o by par of P1 into P5

P7. (a | c).c = a \/ c=o by super of H2 into P6 (and res with x=x)

P8. c.(a | c) = a \/ c=o by par of 6 into P7 (status of . is l-r)

P9. c.((a | c).z)=a.z \/ c=o. by super of P8 into 5

P10. z.x+z.y ≠ o \/ z=o \/ x+y=o by par of 7 into 9

P11. c.x+a.z ≠o \/ c=o \/ x+(a | c).z=o by par of P9 into P10

P12. x.c+a.z ≠o \/ c=o \/ x+(a | c).z=o by par of 6 into P11

P13. a.a+a.z ≠o \/ c=o \/ b.b+(a | c).z=o by par of H2 into P12

P14. a.a+(a.(-w))≠o \/ c=o \/ b.b+(-((a | c).w))=o by par of 0 into P13

The first literal is then simplified by 0 and we get:

a.a+(-(a.w))≠o \/ c=o \/ b.b+(-((a | c).w))=o

P15. a.a+(-(a.w))≠o \/ c=o \/ b.b+((-((a | c).w))+z)=z by super of 1 and P14

and simplification by 3

P16. a.a+(-(a.w))≠o \/ c=o \/ b.b=(a | c).w by super of 2,4 and P15

and simplification by 3

P17. c=o \/ (a | c).a=b.b by par of 2,4 into P16

P18. D(z,x.z) \/ x.z≠x.z \/ z=o by par of D8 into D2

P19. D(c,x.a) \/ c=o by par of P9 into P18

P20. D(c,b.b) \/ c=o by par of P17 into P19

P21. ¬P(c)V D(c,b) \/ c=o by res of D7 and P20

P22. D(c,b) \/ c=o by res of H1 and P21

P23. ¬D(c,a) \/ c=o \/ c=1 \/ c=-1 by res of P22 and H3

P24. (a | c).c ≠a \/ c=o \/ c=1 \/ c=-1 by res of P23 and D2

P25. c=o \/ c=1 \/ c=-1 by par of P7 into P24 and res with x=x

P26. P(o) \/ P(1) \/ P(-1) by successive par of P26 into H1

P27. □ by successive res of P26 and D3 D4 D5.

4. Semantic Trees

In order to prove our main result we shall first recall how to build semantic trees for representing the canonical models for equality theory. For more details, the reader can refer either to (Peterson 1983), (Hsiang Rusinowitch 1988) or (Rusinowitch 1987). Since we want to orient equations with orderings whose ordinality is bigger than ω, we have to build semantic trees which are transfinite. This is done by noetherian induction on A(P,F).

4.1. E-INTERPRETATIONS

Let $<$ be a complete simplification ordering. Let W(B) be the set $\{B' \in A(P,F);\ B' < B\}$. A left segment is either a set W(B) or the set A(P,F) itself. Let B+1 be the successor of B within A(P,F).

4.2. DEFINITION: E-INTERPRETATION.

An E-interpretation on a subset $D \subseteq A(P,F)$ is a mapping $I : D \longrightarrow \{T,F\}$ which satisfies :

E1. $I(s=s)=T$ if $(s=s) \in D$

E2. if $(s=t)$, $B[s],B[t]$ belong to D and $I(s=t)=T$ then $I(B[s])=I(B[t])$.

An E-interpretation is an E-interpretation on A(P,F). One can easily see that an E-interpretation is just a model for the reflexive , symmetric, transitive and substitutive axioms of equality theory. Let I be an E-interpretation whose domain is W(B). Let A be an element of W(B). We define, $I(\neg A) = \neg I(A)$. Let $C = L_1 \vee L_2 \vee \ldots \vee L_k$ be a ground clause whose atoms belong to W(B). We define: $I(C) = I(L_1) \vee I(L_2) \vee \ldots \vee I(L_k)$. The set of equality axioms is denoted by *EQ* and contains:

$$x=x$$

$$x=y \Rightarrow y=x$$

$$(x=y \cap y=z) \Rightarrow x=z$$

$$\textit{Given any } P,\ (x=y \cap P(\ldots,x,\ldots)) \Rightarrow P(\ldots,y,\ldots)$$

$$\textit{Given any } f,\ x=y \Rightarrow f(\ldots,x,\ldots)=f(\ldots,y,\ldots)$$

In order to prove that a set of clauses S containing the equality axioms has no model, it is enough to prove that no E-interpretation can be a model of S. In other words, we have the following :

4.3. THEOREM (see Chang Lee 1973) A set of clauses S is E-unsatisfiable (that is to say, is not valid in any E-interpretation) iff $S \cup EQ$ is unsatisfiable.

4.4. REDUCTION RELATION DEFINED BY AN E-INTERPRETATION.

If I is an E-interpretation on a left segment W(B), it can be used to define a reduction relation $\rightarrow_I$ whose rules are the valid equalities of the model I.

4.5. DEFINITION

Let w and v be elements of $A(P,F) \cup T(F)$. We write $w \rightarrow_I v$ if there is a subterm s of w (we write w=w[s]) and a term t such that $t<s$, $(s=t) < w$, $(s=t) \in W(B)$, $I(s=t)=T$ and v=w[t] . We then say that w is I-reduced to v using s=t. The reflexive transitive closure of $\rightarrow_I$ will be denoted by $\rightarrow_I^*$. The next proposition states that for testing the I-reducibility of an element, we only need to use I-irreducible equalities:

4.6. PROPOSITION (Hsiang Rusinowitch 1988). w is I-reducible iff it is I-reducible using an I-irreducible equality.

The next result shows how it is possible to build inductively the E-interpretations. Its proof can follow (Peterson 1983), since it does not require A(P,F) to have an ordinality smaller than ω.

4.7. THEOREM (Hsiang Rusinowitch 1988). Let I : $W(B+1) \longrightarrow \{T,F\}$ be such that I is an E-interpretation on W(B). Let J be the restriction of I to W(B). Then I is an E-interpretation on W(B+1) iff :

1. **B is J- reducible to some C and I(B)=J(C) or**
2. **B is J-irreducible , of the form t=t and I(B)=T or**
3. **B is J-irreducible and not of the form t=t.**

4.8. TRANSFINITE E-SEMANTIC TREES .

The transfinite E-semantic tree is simply the set TEST made up from all the E-interpretations on left segments of A(P,F), ordered by $\lhd$, the natural extension relation of mappings. To put it more formally, let I and I' be two elements of TEST, with respective domains W(B) and W(B') ; then :

$$I \lhd I' \text{ if } W(B) \subset W(B') \text{ and } I \text{ is the restriction of } I' \text{ to } W(B).$$

The ordering $\lhd$ has the following properties:

T1. $\lhd$ is well founded .

T2. If I belongs to TEST and is defined on $W(B) \neq A(P,F)$ then I has one or two successors (we call a successor or a son a minimal majorant).

T3. If I belongs to TEST and is defined on W(B), where B is a ground atom which has a predecessor for the ordering < , then the restriction of I to W(B-1) is the predecessor of I for the ordering $\lhd$.

EXPLANATIONS

T1 is trivial, since we know that the set $\{W(B) ; B \in A(P,F)\} \cup \{A(P,F)\}$ of the left segments of the well ordered set A(P,F) is also well ordered for the relation of inclusion of sets.

T2 is an easy consequence of the inductive construction of E-interpretations : every E-interpretation on W(B) can be extended (in at most two ways) to W(B+1).

4.9. DEFINITION

When I is an E-interpretation on W(B) which has two successors, the left (resp. the right) successor of I will be the one that assigns the value T (resp. F) to the atom B.

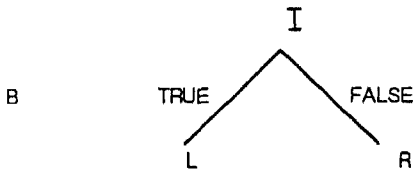


Figure 1

4.10. DEFINITION: MAXIMALLY CONSISTANT SEMANTIC TREES.

If an E-interpretation I on W(B), falsifies a ground instance of a clause C belonging to a set S (i.e. I(Cθ)=FALSE for some ground substitution θ), we call I a *failure node for S*. The *maximally consistant E-semàntic tree* of a set of clauses S, denoted by MCT(S), is the maximal subtree of TEST such that no node I in MCT(S) is a failure node for S. The crucial property of the maximally consistant semantic trees is that they are topologically closed:

4.11. CLOSURE LEMMA (Hsiang Rusinowitch 1988). The limit of an increasing sequence of nodes of MCT(S) belongs to MCT(S).

Let us introduce the notion of *quasi-failure node* which is any E-interpretation R falsifying a clause obtained by reducing a ground instance of a clause of S by $\rightarrow_R$.

4.12. DEFINITION: QUASI-FAILURE NODE.

Let R be a node of MCT(S) whose domain is W(B+1). This node R is a quasi-failure node (for S) if:

1. R(B)=F

2. B is an equality s=t (with s>t)

3. there is a ground instance D of a clause C in S such that every atom in D is strictly smaller than s=s, there is a ground clause D' such that R(D')=F and $D \rightarrow_R^* D'$. We then say that such a clause C *quasi-labels* the node R. We also say that D is *quasi-false* for R.

Let us remark that when 3. is satisfied for some ground clause D', then for any other ground clause D'' such that $D \rightarrow_R^* D''$ and D'' is in the domain of R, we also have R(D'')=F. This is because R can be extended to an E-interpretation.

5. Lifting Lemmas

5.1. IRREDUCIBLE SUBSTITUTIONS AND THE LIFTING PROBLEM.

In order to enable a paramodulation, which is performed into a ground instance of a clause, to be lifted to the clause itself, it is necessary to prevent the replacement of a subterm within the instantiated part of the ground clause.

5.2. EXAMPLE

Let P(x,x,c) be a clause, and c=a another clause. When we paramodulate c=a into P(c,c,c) in the first argument of P, we get P(a,c,a) which is not an instance of a (special) paramodulant of c=a into P(x,x,c). However, if we paramodulate c=a into the third argument we get P(c,c,a) which is an instance of the paramodulant P(x,x,a) of c=a into P(x,x,c). A problem arose in the first case, because the paramodulation step at the ground level did not replace every instance of c, brought by the instantiation of x.

This is the motivation of the next definition :

5.3. DEFINITION

Let I be an E-interpretation and θ, θ' be ground substitutions. We say that θ is I-reducible to θ' and we write $\theta \rightarrow_I \theta'$ if θ is identical to θ' except for one variable, say x, and $I(\theta(x)= \theta'(x))=T$ and $\theta(x)>\theta'(x)$. If θ cannot be I-reduced to any substitution we say that θ is I-irreducible.

5.4. THEOREM (Peterson 1983). Suppose I is an E-interpretation , θ a ground substitution, C a clause such that each atom of $C\theta$ belongs to the domain of I. If $\theta \rightarrow_I \theta'$ then $I(C\theta) = I(C\theta')$.

5.5. COROLLARY (Peterson 1983). Under the same hypothesis there exists a ground I-irreducible substitution ψ, such that $I(C\theta) = I(C\psi)$.

To lift our inferences from the ground case to non-ground case, first we can notice that for every instance $C\theta$ of a clause C in S* which labels or quasi-labels a node I, θ can be assumed to be I-

irreducible. Then we can simply use the classical lifting lemmas for resolution and paramodulation as they are given in (Peterson 83). For lifting the extended superposition rule, let us notice that we can use an argument similar to the one given for paramodulation (or for the critical pair lemma in Knuth and Bendix algorithm):

5.6. Extended superposition lifting lemma. Let C_1 be the clause (s=t) $\vee$ C and C_2 be the clause (a=b) $\vee$ D and n be a non-variable position in s. Let SG be the following extended superposition: $s\theta[n\leftarrow b\theta]=t\theta \vee C\theta \vee D\theta$ of the ground instances $(s=t)\theta \vee C\theta$ and $(a=b)\theta \vee D\theta$ of C_1 and C_2. Then there is an extended superposant S of C_1 and C_2 such that SG is an instance of S.

6. Refutational Completeness of INF

We present here our technique for establishing completeness of the set of inference rules INF. This method is particularly useful for proving the completeness of strategies dealing with equalities as rewrite rules. We have already used it to prove the completeness of the following strategies, where the only equality axiom ever used is x=x - in particular, we never use the functional reflexive axioms- and paramodulation is never performed into variables:

* ORDERED PARAMODULATION (Hsiang Rusinowitch 1986)
* POSITIVE PARAMODULATION (Hsiang Rusinowitch 1988)
* UNFAILING KNUTH-BENDIX-HUET ALGORITHM (Hsiang Rusinowitch 1987)

Let S be a set of clauses. INF(S) denotes the set of clauses obtained by applying some rule in INF to S. Let $INF^0(S)=S$, $INF^{n+1}(S) = INF(INF^n(S))$ and S^* be $\cup_{n\geq 0} INF^n(S)$. The precise statement to be proved is:

6.1. THEOREM. Let S be an E-unsatisfiable set of clauses containing x=x. Then S* contains the empty clause.

Proof: The proof is very similar to the proof of completeness of the unfailing Knuth-Bendix-Huet algorithm (Hsiang Rusinowitch 87). However, since we consider from now on multi-literal clauses, many new difficulties appear. Most of our effort will be spent on dealing with clauses of the type:

$$s=a \vee s=b \vee \ldots$$

Our method can be sketched as follows: given an arbitrary E-unsatisfiable set of clauses S, we want to prove that $\square \in S^*$ which is equivalent to proving that MCT(S*) is empty. Suppose the maximal consistent tree is non-empty. Then we define by induction a particular sequence of nodes in MCT(S*). Since S* has no model, the successors of the last node in the sequence are failure nodes (or quasi-

failure nodes), falsifying some clauses C and D in S*. We apply a proper rule of INF to C and D to get another clause Γ falsified by a node of the sequence. But none of the node in the sequence is a failure node. Hence we get a contradiction.

I. We first show how to define a suitable sequence of nodes in MCT(S).*

We build a sequence Σ of E-interpretations by transfinite induction on the well ordered set A(P,F) which is used for indexing our sequence. First, we define : $I_0 = \Omega$ (empty interpretation). Suppose now that $I_{B''}$ has been defined for all the B'' in the interval [o,B'[with $D(I_{B''}) = W(B'')$. Several situations have to be considered in order to define the element of index B' in the sequence:

B' is not a limit ordinal. Hence, B' has a predecessor in A(P,F), say B. Suppose K' is the last element of the sequence which we have defined so far. Then W(B) is the domain of the interpretation K'. Several cases may occur:

(1) if K' has no successor in MCT(S*) then the sequence is completed.

(2) if K' has exactly one successor J in TEST and J is also in MCT(S*) then it is the next element of the sequence.

(3) if K' has two successors L and R in TEST with L(B)=T and R(B)=F then

(3.1) if R is a quasi-failure node or a failure node and L is in MCT(S*) then the next element will be L.

(3.2) if R is neither a failure node nor a quasi-failure node, it is the next element.

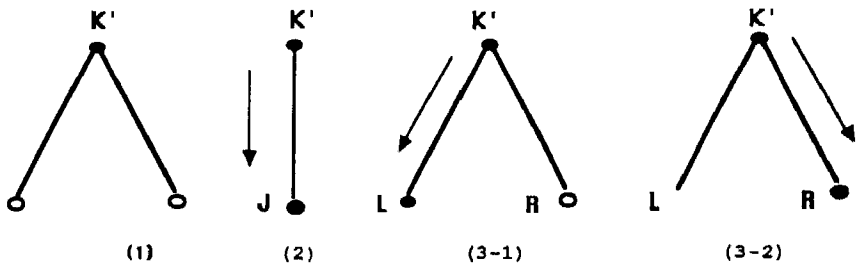


Figure 2

B' is a limit ordinal. We simply define $I_{B'}$ to be the limit of $I_{B''}$ when B'' tends to B'.

The sequence Σ is not empty since MCT(S*) is not empty. Now, our construction cannot "go on forever". Otherwise a model for S would be obtained.

Let B be the smallest atom for which I_B is undefined. We have seen in the second case above that, when B' is a limit ordinal and $I_{B''}$ is defined on [o,B'[, it is always possible to define $I_{B'}$. Hence B cannot be a limit ordinal and,as a consequence, our sequence can only finish in one of the following ways, where K is the last element of the sequence, whose domain is W(B):

case 1: K has exactly one successor I which is a failure node and B is K-irreducible.

case 2: K has two successors L and R which are failure nodes.

case 3: K has exactly one successor I which is a failure node and B is K-reducible.

case 4: K has two successors, L and R (L(B)=T and R(B)=F), with L a failure node and R a quasi-failure node.

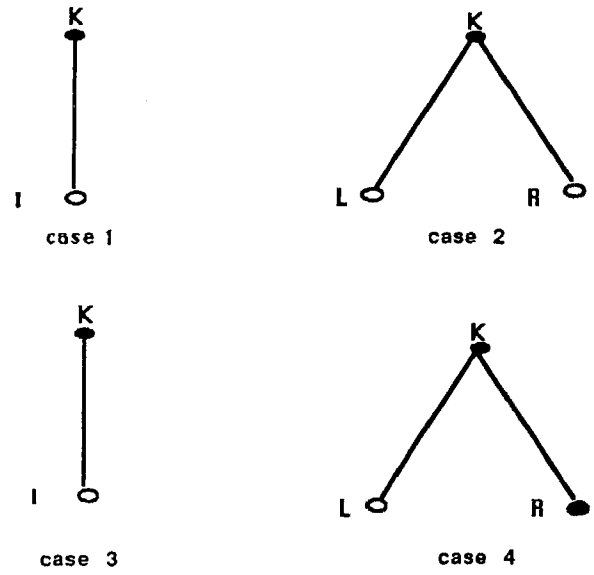


Figure 3

II. Our goal will be achieved by proving that in every case we can find a clause in S that is false in the interpretation K, and meet a contradiction since K was supposed to belong to MCT(S*).*

Before considering every case, we shall prove some technical lemmas which provide some information on the structure of clauses which (quasi-)label the (quasi-)failure nodes.

Lemma A. Let K' be a node in the sequence Σ, which has two successors L and R, such that the domain of K' is W(s=t). and such that R (the right one) is a failure node or a quasi-failure node. Then, for any clause C which (quasi-)labels R, and for any ground instance D of C which is (quasi-)false for R, there is no u such that s≠u is a literal of D.

Proof: let D' be such that R(D')=F and $D \rightarrow_{K'}{}^{*} D'$. Assume that (s=u) is an atom of D, and (s'=u') is the atom of D' which verifies: $(s=u) \rightarrow_{K'}{}^{*} (s'=u')$. Let us notice first that we cannot have s' different from s, otherwise s=t would be reducible by some equality in K': this case has to be excluded since K' has two successors. Thus s=s', and, since (s=u') ≤ (s=t), we also have u'≤t. Let us show now that R(s=u')=F. If u' is t, R(s=u')=F because R is the right successor of K'. If u'<t, then it is because s=t is K'-irreducible and therefore s=u' cannot be used to K'-reduce s=t. Since R(D')=F, R(s'=u') being F forces s'=u' to appear as a positive literal of D'. Therefore, the corresponding literal in D has also to be positive: that is, s=u is a positive literal in D. This concludes the proof of lemma A.

Hence D can be written as :

$$(*)\quad s=u_1 \vee s=u_2 \vee \ldots \vee s=u_m \vee s=u_{m+1} \vee \ldots \vee s=u_k \vee D''$$

where $s=u_i \rightarrow_{K'}{}^{*} s=t$ for $1 \leq i \leq m$, $s=u_i \rightarrow_{K'}{}^{*} s=v_i$ for $m<i \leq k$ and s is not a subterm of D''.

Before going on, we remark that every literal of D'' is strictly smaller than any equality with s on one side. Otherwise if $L \in D''$ verifies L>s=u, then from the hypothesis that L<s=s (recall that K' is a (quasi-)failure node), we derive a contradiction to the hypothesis O6.

Lemma AA. Under the hypotheses of lemma A, there exists an i such that $s=u_i \rightarrow_{K'}{}^{*} s=t$.

Proof: If m=0 let s=v be the maximum of the K'-normal-forms of the atoms $s=u_i$. We note that v<t and K'(s=v)=F. Let K'' be the restriction of K' to the domain W(s=v) and R'' the right successor of K'' in TEST. Since every equality used when K'-reducing D is strictly smaller than s=v, we also have $D \rightarrow^{*}(K'') X$ where X is a clause satisfying R(X)=F. and X is K'-irreducible. Since each of the K'-normal-forms of the atoms of D is smaller than or equal to s=v (recall the remark before lemma AA), we also have R''(X)=F. We thus have proved that K'' satisfy the condition 3.1. However this is impossible because R'' belongs to the sequence (as it is a restriction of K').

The lemma means that m cannot be equal to 0 in the expression of D. With these lemmas, we now discuss the different cases.

CASE 1 AND 2

These cases have been considered in (Hsiang Rusinowitch 1986). One step resolution on clauses of S* which are falsified by the successors of K produces a clause of S* which is falsified by K. Therefore we get a contradiction with the fact that K is in MCT(S*).

CASE 3

We know that I, the successor of the last node K of Σ, is a failure node, which falsifies some ground instance C of a clause of S*. We can suppose that C is minimal with respect to >>, the multiset extension of >.

3.1 : B is not an equality atom.

Let s=t be an I-irreducible equality atom such that s>t, s is subterm of B, and K(s=t)=T. Such an element exists since B is K-reducible. Let K' be the restriction of K to the domain W(s=t), and let J be the right successor of K' in TEST. Since J is not in the sequence we have built, K' satisfies the condition 3-1. Therefore, there is a ground instance D of a clause of S* such that $D \rightarrow_{K'}{}^{*} D'$, J(D') = F and every atom of D is strictly smaller than s=s . We can suppose that D is minimal (w.r.t. the ordering $\ll$, which is by definition, the multiset extension of <).

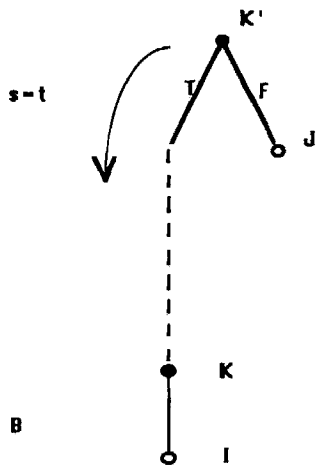


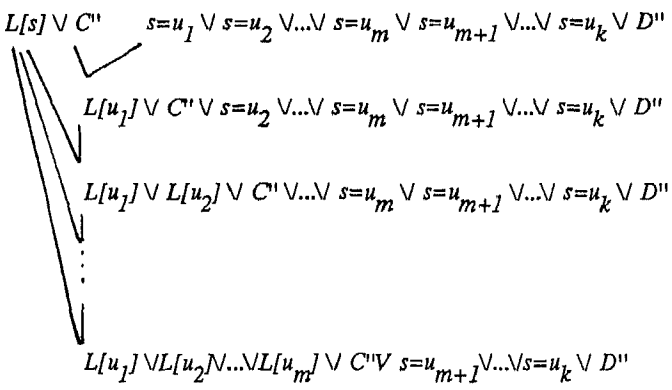
Figure 4

If we apply Lemma A with K', J and D we can derive the expression (*) for the clause D. Let us notice that C can be written as L[s] $\vee$ C'', where L[s] is either the literal B or the literal $\neg$B.

We can obtain after several steps of oriented paramodulation and factoring the following clause P (which is, by definition, in S*):

$$L[u_1] \vee L[u_2] \vee ... \vee L[u_m] \vee s=u_{m+1} \vee ... \vee s=u_k \vee C'' \vee D''$$

The deduction tree of the previous clause is the following, where every inference is a paramodulation from the right parent into the left one, which is always L[s] $\vee$ C'':



The reason why we build such a paramodulant is that we want each of its literals to be false along the interpretation K, so that we can conclude. We have indeed:

Lemma B: K(P)=F

Proof: K(C'')=F since we have K(C)=F. Each of the atoms of D'' is strictly smaller than s=t, therefore D'' is in the domain of K' and K'(D'')=F. But K is an extension of K'; consequently, K(D'')=F. For i>m we have $s=u_i \rightarrow_{K'}{}^* s=v_i$ and K'(s=v$_i$)=F. Every equality atom used to perform a K'-reduction can also be used to perform a K-reduction. So, we can replace the K' above with K. But now s=u$_i$ is in the domain of K since it is strictly smaller than L[s]. Therefore we can derive the equality K(s=u$_i$)=F. For 1≤i≤m we have $s=u_i \rightarrow_K{}^* s=t$. Therefore K(s=u$_i$)=T and I(L[u$_i$]) = I(L[s]) = F. But L[u$_i$] < L[s], so every literal of P is in the domain of K. Since I is an extension of K, by coherence, we have K(L[u$_i$])=F. The lemma is proved.

3.2: B is an equality atom "a=b" with a>b and I(B)=T.

Let s=t be the minimal equality such that I(s=t)=T and s is a subterm of B. If s is a strict subterm of a or s is a subterm of b then we can proceed as before. Now, we show than s cannot be equal to a. If this were the case then I(a=t)=T and I(a=b)=T imply I(b=t)=T. Since b>t, we can use b=t to I-reduce B; but this is impossible, because no equality smaller than s=t may I-reduce B.

3.3: B is an equality atom "a=b" with a>b and I(B)=F.

If there is a strict subterm a' of a such that I(a'=b')=T for some b'≤ a', then we can follow the proof of subcase 3.1. Therefore from now on we assume that no such a' exists.

3.3.1: for each d≤b we have I(a=d)=F.

The hypothesis of 3.3.1 implies that every atom in C of type a=d , appears only as positive literals. Hence, C can be written $a=b_1 \vee a=b_2 \vee ... \vee a=b_m \vee C''$ and a does not appear in the subclause C''. We

shall use the hypothesis (O6) on the ordering < :

If (u=v) > A > (u=w) then there is a ground term x such that A is u=x.

With this hypothesis we can rephrase the previous statement more precisely to be: every atom a=d in C is strictly bigger than any atom of C''.

Let a=c be the maximal (w.r.t <) I-normal-form of the atom $a=b_i$ where $i \leq m$, that is: $c=\sup\{k_i : 1 \leq i \leq m\}$ where $k_i=\inf\{k : I(k=b_i)=T\}$.

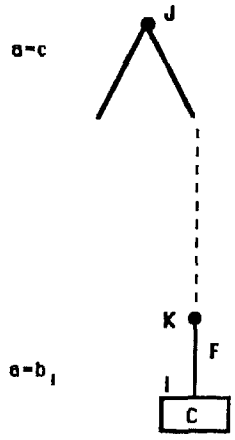


Figure 5

Let J be the restriction of I to W(a=c). Every equality used to I-reduce one of the b_i is necessarily strictly smaller than a=c. Indeed, there is no equality a=z such that I(a=z)=T (hypothesis 3.3.1) and there is no equality s'=t' such that s' is a strict subterm of a and I(s'=t')=T (hypothesis 3.3), therefore we can I-reduce an atom $a=b_i$ only with an equality whose larger side is a subterm of b_i, and such an equality is always smaller than a=c. From these remarks we can assume that there is a ground clause C! such that $C \rightarrow_J^* C!$ and each literal of C! is $\leq$ (a=c). Consequently, J satisfies the condition 3.1. So the right successor of J cannot belong to the sequence of nodes we have defined in MCT(S*). But since K(a=c)=F, K cannot follow J in our sequence. Therefore subcase 3.3.1 never occurs. So under hypothesis 3.3 we always have :

3.3.2: there is a term c such that c<b and I(a=c)=T.

Let us suppose now that c is the smallest term satisfying 3.3.2. Let K' be the restriction of I to the domain W(a=c), and let J be the right successor of K' in TEST. Since J is not in the sequence we have built, K' satisfies the condition 3-1. Therefore, there is a ground instance D of a clause of S* such that $D \rightarrow_{K'}{}^*D'$, J(D')=F and every atom of D is strictly smaller than a=a . We can suppose that D is minimal (w.r.t <).

With Lemma A, and Lemma AA, we have shown that every equality a=u within the clause D never appears within a negative literal.

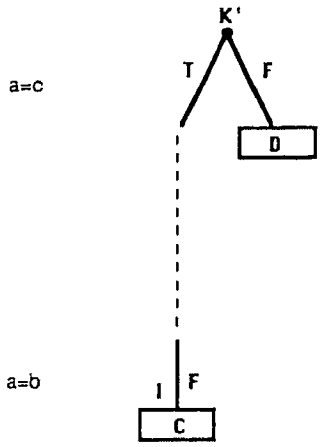


Figure 6

Hence D can be written:

$$a=u_1 \vee a=u_2 \vee ... \vee a=u_m \vee a=u_{m+1} \vee ... \vee a=u_k \vee D''$$

with $a=u_i \rightarrow_{K'}{}^* a=c$ for $1 \leq i \leq m$, $a=u_i \rightarrow_{K'}{}^* a=v_i$ for $m<i \leq k$, $a=v_i$ is K'-irreducible and $v_i < c$ and a is not a subterm of D''.

We know that I is a failure node, which falsifies some instance C of a clause of S* . The hypothesis implies that C (or a factor of C) can be written a=b $\vee$ C'' where each of the atoms in C'' are strictly smaller than a=b.

Let us first suppose that C>D.

A few steps of extended superposition (and factoring) with C and D as input clauses generate the following clause P :

$$[u_1=b] \vee [u_2=b] \vee ... \vee [u_m=b] \vee a=u_{m+1} \vee ... \vee a=u_k \vee C'' \vee D''$$

The deduction tree is identical to the one in subcase 3.1. For $m+1 \leq i \leq k$, $a=u_i \rightarrow_I{}^* a=v_i$ because I is an extension of K'. But $I(a=v_i)=F$. Also $a=u_i$ is in the domain of I, since $a=u_i<a=b$ from the hypothesis "C>D". Therefore we also have $I(a=u_i)=F$.

For $i<m+1$, $(a=u_i) \rightarrow_I{}^* (a=c)$ since I is an extension of K'. Therefore $I(u_i=c)=T$. But I(a=c)=T . Consequently we also have $I(u_i=a)=T$. Then I(a=b)=F implies $I(u_i=b)=F$. As in Lemma B, it is easy to see that I(C'' $\vee$ D'')=F and conclude with I(P)=F. Since each of the atoms of C is strictly smaller than a=b, we also have K(P)=F. That means that K is a failure node: this is contrary to the fact that K belongs to the sequence.

Let us suppose now that C≤D:

In order to point out the atoms containing the subterm a, we can write C as the following expression : $L_1[a] \vee L_2[a] \vee ... \vee L_r[a] \vee CC$ where $L_h[a]$ is either $a=b_h$ or $a \neq b_h$ and a never occurs within CC. We can also suppose that $L_1[a] > L_2[a] >, ..., > L_r[a]$. From each equality $a=u_i$ ($i \leq m$) of D we paramodulate or superpose successively into each occurrence of a within C to get the following clause LP:

$$L_1[u_1] \vee L_1[u_2] \vee ... \vee L_1[u_m] \vee L_2[u_1] \vee L_2[u_2] \vee ... \vee L_2[u_m] \vee ..... \vee$$
$$L_r[u_1] \vee L_r[u_2] \vee ... \vee L_r[u_m] \vee a=u_{m+1} \vee ... \vee a=u_k \vee CC \vee D''$$

which can be obtained by the following deduction:

$L_1[a] \vee ... \vee L_r[a] \vee C''$

$a=u_1 \vee a=u_2 \vee ... \vee a=u_m \vee a=u_{m+1} \vee ... \vee a=u_k \vee D''$

$L_1[u_1] \vee ... \vee L_r[u_1] \vee C'' \vee a=u_2 \vee ... \vee a=u_m, a=u_{m+1} \vee ... \vee a=u_k \vee D''$

$L_1[u_1] \vee ... \vee L_r[u_1] \vee$
$L_1[u_2] \vee ... \vee L_r[u_2] \vee C'' \vee ... \vee a=u_m, a=u_{m+1} \vee ... \vee a=u_k \vee D''$

.
.
.

$L_1[u_1] \vee ... \vee L_r[u_1] \vee$
$L_1[u_2] \vee ... \vee L_r[u_2] \vee$
,.....,
$L_1[u_m] \vee ... \vee L_r[u_m] \vee C'' \vee a=u_{m+1} \vee ... \vee a=u_k \vee D''$

We prove that there is a clause LP' such that $LP \rightarrow_{K'} {}^*LP'$ and J(LP')=F.

Proof: For $i<m+1$, $I(u_i=c)=T$. But $I(a=c)=T$. Consequently we also have $I(u_i=a)=T$. Then $I(L_j[a])=F$, for $j \leq r$, implies $I(L_j[u_i])=F$ and also $K'(L_j[u_i])=F$. Since CC and D'' are smaller than (a=c), the subclause ($a=u_{m+1} \vee ... \vee a=u_k \vee CC \vee D''$) can be (K')-reduced to some clause X such that J(X)=F. Consequently LP itself is (K')-reducible to some clause falsified by J.

Since m is not 0, LP is strictly smaller than D. This is a contradiction with the fact that D is a minimal clause satisfying the condition 3.1 at node K'.

CASE 4

The last node K of the sequence has two successors L and R: L is a failure node and R is a quasi-failure node. Hence, there is an equality s=t such that the domain of K is W(s=t). Lemma A and Lemma AA applied to K and R imply the existence of a clause D which has the expression :

$$s=u_1 \vee s=u_2 \vee ... \vee s=u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee D''$$

with $s=u_i \rightarrow_K^* s=t$ for $1 \leq i \leq m$, $s=u_i \rightarrow_K^* s=v_i$ for $m<i \leq k$, $s=v_i$ K-irreducible and $v_i < t$ and s is not a subterm of D''.

Since L is a failure node but not K, there is a clause C which can be written $s \neq t \vee C''$ and which is a ground instance of a clause of S* such that $L(s \neq t \vee C'')=F$. In order to eliminate some occurrences of s we perform successive paramodulations from D and C, and get the clause LP:

$$t \neq u_1 \vee t \neq u_2 \vee ... \vee t \neq u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee C'' \vee D''$$

For $i \leq m$ we have $K(u_i=t)=T$ (recall that $s=u_i \rightarrow_K^* s=t$). Up to some factoring , we can suppose that C'' and D'' are in the domain of K and therefore satisfy $K(C'')=K(D'')=F$. For $i>m$ $(s=u_i) \rightarrow_K^* (s=v_i)$ with $v_i<t$. Therefore $LP \rightarrow_K^* LP'$ and K(LP')=F. Since every literal in LP' is smaller than s=t, we also have R(LP')=F. However, no literal of LP can be K-reduced to s=t. Here we get a contradiction with Lemma AA, because K belongs to the sequence of nodes and satisfies condition 3.1 .

Since every case has been considered, the proof is complete.

7. Horn Clauses

A Horn clause is a clause which contains at most one positive literal. When we restrict our inference rules to Horn clauses, they can be further refined. In particular, we just need to infer on the maximal literals of the clauses and factoring is not needed. This can be proved simply by examining the proof above when S only contains Horn clauses:

In case 1, let $A \vee \neg(a=a)$ be the minimal clause that is falsified by I. Since $\neg(a=a)$ is a maximal literal of the clause, we can resolve with x=x and obtain the clause A which is smaller and is still falsified by I. This rises a contradiction.

In case 2, let $A \vee \neg B$ and $B \vee C$ be two minimal clauses which are falsified by L and R respectively. Since $B \vee C$ is Horn, it contains a unique occurrence of B. Then, one step resolution on B generates $A \vee C$ which is smaller than $A \vee \neg B$ and is still falsified by L. Note that it was not needed to ensure that $A \vee \neg B$ contains only one occurrence of $\neg B$. Therefore factoring was not required.

In case 3.1 and 3.2, when dealing with Horn clauses, we just need one paramodulation step into the maximal literal of C to get the clause P. Moreover, no factoring is needed. In case 3.3.2, the clause C can be written $a=b \vee C''$ with $C'' < (a=b)$ without factoring since it admits only one positive literal. A single step of extended superposition is enough to get a clause which is smaller than C and which is

falsified by I.

In case 4, a single paramodulation step of D into C generates a clause LP which is smaller than C and falsified by L.

We can conclude:

7.1. THEOREM. The rules of O-RESOLUTION, ORIENTED PARAMODULATION EXTENDED SUPERPOSITION are refutationally complete for Horn clauses even when we restrict ORIENTED PARAMODULATION and EXTENDED SUPERPOSITION to be applied to the maximal literals of clauses.

This theorem is the basis for the conditional completion procedure which has been proposed in (Kounalis and Rusinowitch 1988). Note also that more refinements could be obtained. For instance, it is not needed to paramodulate **from** a Horn clause $s=a \vee C$ when s occurs in C. We can also suppose that paramodulation into a negative equational literal $\neg(a=b)$, is always performed into a member of a=b which is maximal in {a,b}.

8. Completeness in the Presence of Simplification and Subsumption

The purpose for using deletion rules is to get rid of redundancies and tautologies and to keep the system as small as possible. In many equality theorem-provers such as ITP (Lusk Overbeek 1984) or SEC (Fribourg 1985), demodulation (Wos et al. 1967), or simplification, is used as a very efficient heuristic. Theoretical foundation for this inference rule was developed through the Knuth and Bendix completion algorithm (Huet 1981). In the general setting of first order calculus, there have been very little investigation about how completeness is preserved in presence of a "deletion" rule such as simplification. Everybody agrees that in general simplification leads to shorter refutations; however this is not always the case. For example, the unsatisfiable set of clauses $\{ P(f(x)), \neg P(f(g(a))), f(g(x)) \rightarrow b\}$ admits a straightforward one step refutation by resolution. If we first apply the equation as a demodulator, we get the following normalized set of clause: $\{P(f(x)), \neg P(b), f(g(x)) \rightarrow b\}$. The shortest refutation we can get now uses two steps: one paramodulation step in the first clause followed by one resolution step. Our goal is to show that the normalization of clauses does not push the empty clause out of reach of our theorem-prover. This goal has been achieved; the proof of that result heavily relies on the noetherian feature of the demodulators.

Subsumption, as simplification, is a rule which decreases the search space. It was studied carefully by D.Loveland (1978) from the proof theoretic point of view which is much harder to handle than the semantic one. A nice aspect of our approach is that it allows a common treatment of subsumption and simplification.

In this chapter, an inference rule is a rule for replacing a set of clauses by an equivalent set of clauses. With this new definition, we consider now two other inference rules: subsumption and simplification. Let us recall their precise definitions:

8.1. DEFINITIONS

The classical subsumption rule causes problems, since the relation "is subsumed by" is not well-founded, even when it is quotiented by the variable renaming relation: P(f(x)) \/ P(f(z)) and P(u) \/ P(f(w)) subsume each other, but they are not variants. Therefore, we shall need a slight restriction of the subsumption rule in order to ensure that completeness is preserved. Let us recall that a clause C_1 properly subsumes C_2 if C_1 subsumes C_2 and C_2 does not subsume C_1.

PROPER_SUBSUMPTION

One may delete from S any clause which is properly subsumed by another clause in S.

The advantage of proper subsumption appears in the next lemma:

8.1.1. LEMMA (Loveland 1978). There is no infinite sequence $C_0,C_1,...,C_i...$ such that C_{i+1} properly subsumes C_i.

For the next rule, the symbol < represents a complete simplification ordering. This ordering has to be the one which is used for defining the other inference rules. If the unit equation s=t is in S and $C_2[s\theta]$ is a clause in S which contains an instance $s\theta$ of s, and $s\theta > t\theta$, and there is an atom A in $C_2[s\theta]$ such that A> ($s\theta$=$t\theta$), then the clause $C_2[t\theta]$ is a simplification of $C_2[s\theta]$ by s=t.

SIMPLIFICATION

One may replace in S a clause which has been simplified, by its simplification.

The restricted format of the simplification rule is needed in order to apply our completeness proof. The restriction on the atom A is probably not necessary as noticed in (Peterson 1983).

We can also notice that, as in (Hsiang and Rusinowitch 1987, Peterson 1983) , our definition allows unoriented equations to be used as simplifiers: indeed uncomparable terms happen to be comparable when instantiated. For instance, f(x,x,y) = f(x,y,y) can simplify P(f(g(a),g(a),a)) into P(f(g(a),a,a), notwithstanding the non-orientable equation.

8.2. PROOF OF COMPLETENESS

We now consider the full set of rules DRA, that is: {O-RESOLUTION, O-FACTORING, ORIENTED-PARAMODULATION, EXTENDED SUPERPOSITION, SIMPLIFICATION, PROPER SUBSUMPTION}. Let DRA(S) be the set we obtain by application of one of the inference rules of DRA to the set of clauses S. Let $DRA^0(S)$=S and $DRA^{n+1}(S)$ = $DRA(DRA^n(S))$. We shall denote $DRA^n(S)$ by S_n.

Because from now on we are dealing with deletion inference rules, we cannot assume any more monotonicity of the process. The problem is that we cannot ensure anymore that clauses which appear in some S_n remain available throughout the inference process, and may always take part in a refutation. Some clauses might be simplified or subsumed during the process. Suppose for instance that $C \in S_i$, $D \in S_j (j>i)$ and C and D can be resolved. This resolvent will perhaps never be generated, since C or D may not be simultaneously present in the system due to the deletion inference rules. What is enough to prove in order to avoid this problem is that clauses involved in a refutation can be chosen in such a way that they will never be simplified or subsumed later on .

Given an initial set of clauses S and a derivation $S_0 \rightarrow S_1 \rightarrow ... \rightarrow S_i \rightarrow ...$ where S_0 is equal to S and S_i is obtained by application of a rule of DRA to S_{i-1}, S* denotes, from now on, $\cup_{i \geq 0} S_i$. A clause C of S* is *persisting* (w.r.t. the derivation $(S_i)_{i \geq 0}$) if there is a $k \in N$ such that C belongs to every S_i , for $i \geq k$. The crucial proposition is:

8.2.1. PROPOSITION. Every failure node of S* can be labelled by a persisting clause. Every quasi-failure node can be quasi-labelled by a persisting clause.

Proof: the proposition is proved by considering the smallest clauses (w.r.t.$\ll$) in S* which can label the (quasi)-failure node. Let GR be the application which associates to a subset of clauses of S* the set of its ground instances. Let I be a failure node for S* and let Σ be the set of clauses labelling I, namely: $\{G;\ G \in S^*$ and there is $G1 \in GR(G)$ such that $I(G1)=false\}$

The set of minimal elements of $GR(\Sigma)$ w.r.t. $\ll$ which are falsified by I is denoted by TG. The set of clauses in Σ which have an instance in TG is $GR^{-1}(TG)$; it will be denoted by T. The subset of clauses of T which are minimal for the proper subsumption ordering will be denoted by T' (i.e. the clauses of T which are not properly subsumed by another member of T). We can notice that GR(T)=GR(T')=TG.

8.2.2. Lemma. If C belongs to T' then C is persisting.

Proof: let us first prove that C will never be simplified. Otherwise,there exists a j such that $C \in S_j$, $(s \rightarrow t) \in S_j$, C can be written $C[s\sigma]$ and $S_{j+1} = (S_j - \{C\}) \cup \{C[t\sigma]\}$. Since $C \in T$, there is a ground substitution θ such that: $I(C\theta)=false$ and $C\theta$ is minimal in $GR(\Sigma)$.

By definition of the simplification rule , $(s=t)\sigma < C$; therefore, by stability of <, we have $(s=t)\sigma\theta < C\theta$. However we cannot have $I((s=t)\sigma\theta)=false$ because I is a failure node of S*. Hence $I((s=t)\sigma\theta)=true$. Since I is an E-interpretation we can derive: $I(C[t\sigma]\theta) = I(C[s\sigma]\theta) = false$. But, $C[t\sigma]\theta < C[s\sigma]\theta$. This rises a contradiction with the assumption that $C\theta$ is minimal in $GR(\Sigma)$.

Let us prove now that C is never subsumed. Assume that Γ and C belong to S_j, Γ properly subsumes C and $S_{j+1} = S_j - \{C\}$. By definition, C does not subsume Γ. One can notice that Γ belongs to Σ. Since

every ground instance of C contains a ground instance of Γ, and C belongs to T, Γ also belongs to T. However C is in T' and therefore cannot be properly subsumed by another member of T.

We can prove with the same technique that a quasi-failure I node can be quasi-labelled by a persisting clause. The only change is to define Σ as the set of clauses quasi-labelling I. Afterwards, we have to prove that when simplifying (or subsuming) a clause C which quasi-labels a node I we get another clause Γ with the same property.

We can now express the completeness in presence of subsumption and simplification.

8.2.3. THEOREM. Every fair derivation, whose initial set is E-unsatisfiable and contains the axiom x=x, yields the empty clause.

Proof: let S be an E-unsatisfiable set of clauses containing x=x. We assume that MCT(S*) is not empty. Let K be the last node of the right branch of MCT(S*). We first suppose that K has two successors L and R in TEST, which are failure nodes. Let C be a clause of S* labelling L and F a clause of S* labelling R.

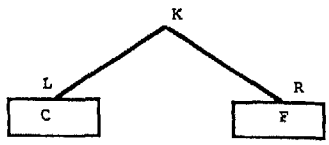


Figure 7

We know that there is a clause $\Gamma \in INF(\{C,F\})$ falsified by K. This Γ can be obtained by resolution between C and F. With the proposition above, we can suppose that:

$$C,F \in \cap_{i \geq j} S_i \quad \text{for some } j \geq 0$$

Then $\Gamma \in \cap_{i \geq j} INF(Si)$. Now, the fairness assumption ensures that Γ is subsumed by some clause Γ' of S*. We derive a contradiction with the fact that K belongs to MCT(S*), as usual, by showing that K falsifies the clause Γ' of S*. When K has only one successor, the proof is quite similar but uses paramodulation or superposition instead of resolution.

8.3. OTHER DELETION RULES.

We give now other deletion rules that do not destroy completeness of the previous strategy, as soon as fairness is ensured:

TAUTOLOGY_DELETION

A tautology is any clause which contains a subclause of the following type: $A \vee \neg A$. The tautology deletion rule states that we can delete the tautologies.

Completeness in presence of the tautology deletion rule is trivial: since a tautology never labels a failure node, it is never needed to perform an inference step on such a clause.

CLAUSAL_SIMPLIFICATION

If the unit literal L is in S, then we can replace any clause in S which contains a negated instance of L, by the same clause where this instance has been deleted.

The "clausal_simplification rule" can be simulated by one resolution step followed by one subsumption step. Completeness is preserved when adding this deletion rule: as above, we prove that a minimal clause C of S* which labels a failure node can be chosen to be persisting. We have the same remark for the next rule:

FUNCTIONAL_SUBSUMPTION

If a clause C in S contains the literal $g[s]=g[t]$, and an equation $l=r$ in S verify $l\sigma=s$ and $r\sigma=t$, then C can be removed of S.

9. Concluding Remarks

Using the powerful tool of transfinite-semantic trees, we have been able to prove the completeness of a set of inference rules which extend the Knuth-Bendix completion procedure. The only restriction is that equations are oriented according to a complete simplification ordering. This is not a real drawback since most of the orderings that are used in the context of term-rewriting systems are of that type. The strategy described above can be refined when we deal with Horn clauses. For instance we can restrict the paramodulation or superposition rules to be performed only into the maximal literals of any clause. The clauses can then be interpreted as conditional rewrite rules. This is detailed in (Kounalis Rusinowitch 1988). It is also possible to obtain a complete unit strategy, as in (Henschen Wos 1974, Paul 1985, Bachmair et al.1987). We think that we should gain more efficiency by incorporating axioms like

associativity and commutativity in the unification algorithm (Plotkin 1972) and by extending the notion of *critical pair criteria* to resolution and paramodulation, (see, for instance, (Küchlin 1985)).

Acknowledgements. The author is very grateful to Jean-Luc Rémy, Leo Bachmair and Jieh Hsiang for their comments.

References

Bachmair, L., Dershowitz, N., Plaisted, D. (1987). Completion without failure, Proc. Coll. on Resolution of Equations in Algebraic Structures, Lakeway, Texas.

Brown, T. (1974). A Structured Design Method for Specialized Proof Procedures, Ph.D. Thesis, Cal. Tech.

Chang, C. L., Lee, C. T. (1973). *Symbolic Logic and Mechanical Theorem Proving*, London: Academic Press.

Dershowitz, N. (1982). Orderings for term rewriting systems. *Theoretical Computer Science* **17**, 279–301.

Dershowitz, N. (1985). Termination. In: Rewriting Techniques and Applications, Jouannaud, J. P. (ed.). *Lecture Notes in Computer Science* **202**, 180–224. Also in *J. Symbolic Comp.* (1987) **3**, 69–115.

Fribourg, L. (1985). A superposition oriented theorem prover. *Theoretical Computer Science* **35**, 129–164.

Henschen, L., Wos, L. (1974). Unit refutation and horn sets. *J ACM* **21**, 295–301.

Hsiang, J., Rusinowitch, M. (1986). A new method for establishing refutational completeness in theorem proving. In: Proceedings of 8th Conference on Automated Deduction, Siekmann, J. (ed.). *Lecture Notes in Computer Science* **230**, 141–152.

Hsiang, J., Rusinowitch, M. (1988). Proving refutational completeness of theorem proving strategies. The transfinite semantic tree method. *J ACM* (to appear).

Hsiang, J., Rusinowitch, M. (1987). On word problems in equational theories. In: Proceedings of 14th International Colloquium on Automata, Languages and Programming, Karlsruhe, Germany, Ottmann, Th. (ed.). *Lecture Notes in Computer Science* **267**.

Huet, G. (1981). A complete proof of correctness of Knuth-Bendix completion algorithm. *JCSS* **23**, 11–21.

Kounalis, E., Rusinowitch, M. (1988). On word problems in Horn theories. In: Proceedings of 9th Conference on Automated Deduction, Lusk, E., Overbeek, R. (eds.). *Lecture Notes in Computer Science* **310**, 527–537. Also in *J. Symbolic Comp.* (1991) **11**, 113–127.

Knuth, D., Bendix, P. (1970). Simple words problems in universal algebras. In: *Computational Problems in Abstract Algebra*. Pergamon Press.

Küchlin, W. (1985). A confluence criterion based on the generalised Newman lemma. In: Proceedings of EUROCAL, Caviness, B. (ed.). *Lecture Notes in Computer Science* **204**, 390–399.

Lankford, D. S. (1975). Canonical inference. Report ATP-32. DMCS University of Texas at Austin.

Loveland, D. (1978). *Automated Theorem Proving: A Logical Basis*. Amsterdam: North-Holland.

Lusk, E. L., Overbeek, R. A. (1984). A portable environment for research in automated reasoning. In: Proceedings of 9th Conference on Automated Deduction, Nappa Valley, CA. *Lecture Notes in Computer Science* **170**.

Paul, E. (1985). On solving the equality problem in theories defined by Horn clauses. In: Proceedings of EUROCAL. *Lecture Notes in Computer Science* **204**, 363–377.

Peterson, G. E. (1983). A technique for establishing completeness results in theorem proving with equality. *SIAM J. of Computing* **12**, 82–100.

Plotkin, G. (1972). Building-in equational theories. In: *Machine Intelligence*, Meltzer, B., Mitchie, D. (eds.), Vol. 7, New York: American Elsevier, pp. 73–90.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *J ACM* **12**, 32–41.

Rusinowitch, M. (1987). Démonstration automatique par des techniques de réécriture. Thèse. Université de Nancy 1. Published in: (G. Huet, director) Collection Science Informatique, Inter Editions, 1989.

Wos, L. T., Robinson, G. A. (1969). Paramodulation and theorem proving in first-order theories with equality. In: *Machine Intelligence*, Vol. 4, New York: American Elsevier, pp. 135–150.

Wos, L., Robinson, G. A., Carson, D. F., Shalla, L. (1967). The concept of demodulation in theorem proving. *J ACM* **14**, 698–709.