PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT.

THEOREM-PROVING WITH RESOLUTION AND SUPERPOSITION: AN EXTENSION OF THE KNUTH AND BENDIX PROCEDURE TO A COMPLETE SET OF INFERENCE RULES.

Michael Rusinowitch

CRIN
Campus Scientifique BP 239
54506 Vandoeuvre les Nancy (FRANCE)
rusi@crin.crin.fr

ABSTRACT

We present a refutation 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 equality units, 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 set of inference rules, at least for the case where a complete simplification ordering is used to compare terms. In (Lankford 1975)(see also Fribourg 1985) a theorem proving procedure (the *derived reduction algorithm*) is described which uses a very restrictive version of paramodulation (Wos Robinson 1970). Given a reduction ordering on terms, assume that some equations are oriented from left to right (large to small, accordingly). A non-variable subterm s within a clause C can be replaced by another one equal to it, only if :

condition a: s matches the left-hand side of a positive equational literal, and

condition b: if s is a subterm of a positive equational literal of C, then it has to belong to the left-hand side of this equation.

The remaining literals of the parent clauses are just added to the inferred clause, as in the classical version of paramodulation. The aim of this paper is to study the refutational completeness of a strategy based on this inference rule.

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 and few is known about their effect on completeness. However, in our case, we are able to incorporate the deletion rules in the same framework as the other inference rules and, to show that completeness is preserved.

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. D. Lankford has proved the completeness of this strategy in the particular 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 Knuth -Bendix algorithm, when an equation that cannot be oriented appears. 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 L.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 1988) 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 1988), 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 is to build a branch which avoids quasi-failure nodes.

2. INFERENCE RULES.

2.1. NOTATIONS.

In this section we review 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 $C \sqsubseteq D$, where C and D are clauses means that the set of literals of C is included in the set of literals of D.

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 $\sigma=\theta\phi$. 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 (see (Peterson 1983) (Hsiang Rusinowitch 1987)).

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 suppose 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:

$P(s_1,...,s_n) < Q(t_1,...,t_m)$ *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.

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. *$\forall A \in C_1'$, $L_1\theta \not\leq A\theta$ and*
3. *$\forall A \in C_2'$, $L_2\theta \not\leq A\theta$ 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. *$\forall A \in C_2-\{L_2\}$, $L_2\theta \not\leq A\theta$ 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 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. *$\forall A \in C_2-\{a=b\}$, $a\theta=b\theta \not\leq A\theta$*

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

If the unit equation s=t is in S and $C_2[s\theta]$ is a clause in S which contains an instance sθ of s, and sθ > tθ, and there is an atom A in $C_2[s\theta]$ such that A> (sθ=tθ), 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 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 $S_0 \to S_1 \to ... \to S_i \to ...$ where S_i is obtained by application of a rule of DRA to S_{i-1} is *fair* if :

$\forall j$, $R \in \cap_{i \geq j} INF(S_i)$ implies that R is subsumed by some clause C ∈

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. This strategy is used in the original Knuth and Bendix completion algorithm. 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. EXAMPLE

The following easy example shows the transitivity of less-or-equal, assuming the associativity of max. The skolemized negation of the theorem is the conjunction of clauses 5,6,7. We use the predicate first ordering, as 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) V LE(y,x).
2. $\neg$LE(x,y) V max(x,y) $\to$ y.
3. $\neg$LE(y,x) V max(x,y) $\to$ x.
4. max(max(x,y),z) $\to$ max(x, max(y,z)).
5. LE(a,b).
6. LE(b,c).
7. $\neg$LE(a,c).

REFUTATION

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

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'∈A(P,F); B'< B }. A left segment is either a set W(B) or the set A(P,F) itsef. Let B+1 be the successor of B within A(P,F).

4.2. DEFINITION: E-INTERPRETATION.

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

E1. I(s=s)=T if (s=s)∈ 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 = $L_1 \vee L_2 \vee ,...,\vee L_k$ be a ground clause whose atoms belong to W(B). We define: I(C) = $I(L_1) \vee I(L_2) \vee ,...,\vee I(L_k)$. The set of equality axioms is denoted by *EQ*.

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.

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 result shows how it is possible to build inductively the E-interpretations.

4.5. THEOREM (Hsiang Rusinowitch 1988). Let I : W(B+1) $\rightarrow$ {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.**

4.6. 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 <|, 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 <| I' if W(B) $\subset$ W(B') and I is the restriction of I' to W(B).

Let us notice that the ordering <| is well founded.

4.7. 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-semantic 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.

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 with the oriented equalities which are valid in R.

4.8. 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.

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 instanciated part of the ground clause. This is the motivation of the next definition :

5.2. 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 θ →(I) θ' if θ is identical to θ' except for one variable, say x, and I(θ(x)= θ'(x))=T and θ(x)>θ'(x). If θ cannot be I-reduced to any substitution we say that θ is I-irreducible.

5.3. THEOREM (Peterson 1983). Suppose I is an E-interpretation , θ a ground substitution, C a clause such that each atom of Cθ belongs to the domain of I. Then there exists a ground I-irreducible substitution ψ, such that I(Cθ) = I(Cψ).

To lift our inferences from the ground case to non-ground case, first we can notice that for every instance Cθ 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 1983). 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):

5.4. Extended superposition lifting lemma. Let C_1 be the clause (s=t) V C and C_2 be the clause (a=b) V D and n be a non-variable position in s. Let SG be the following extended superposition: sθ[n←bθ]=tθ V Cθ V Dθ of the ground instances (s=t)θ V Cθ and (a=b)θ V Dθ 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) and 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 result is:

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

By lack of space, we have not detailed the proof. It can be found in full extent in (Rusinowitch 1987).

Our method can be sketched as follows: given an arbitrary E-unsatisfiable set of clauses S, we want to prove that □ ∈ 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.

7. COMPLETENESS IN PRESENCE OF SIMPLIFICATION AND SUBSUMPTION.

The point of 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 like ITP (Lusk Overbeek 1984) or SEC

(Fribourg 1985), demodulation (Wos et al. 1967), or simplification, is used as a very efficient heuristic. Theoretical foundations for that inference rule were developed through the Knuth and Bendix completion algorithm (Huet 1981). In the general setting of first order calculus, there have been very few investigations 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 : 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 clauses: $\{P(f(x)), \neg P(b), f(g(x)) \rightarrow b\}$. 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: proper subsumption and simplification. We can notice that, as in (Hsiang and Rusinowitch 1987)(Peterson 1983) , unoriented equations can be used as simplifiers; indeed uncomparable terms happen to be comparable when instanciated. Example: $f(x,x,y) = f(x,y,y)$ can simplify $P(f(g(a),g(a),a))$ into $P(f(g(a),a,a))$, notwithstanding the non-orientable equation.

We now consider the full set of rules DRA. 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 have been generated, 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 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 .

7.1. DEFINITION.

Given an initial set of clauses S and a derivation $S_0 \rightarrow S_1 \rightarrow \ldots \rightarrow S_i \rightarrow \ldots$ 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:

7.2. 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.

The proposition is proved by considering the smallest clauses (w.r.t.$<$) in S^* which can label the (quasi)-failure node.

7.3. 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. 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: $C,F \in \cap_{i \geq j} S_i$ *for some* $j \geq 0$

Then $\Gamma \in \cap_{i \geq j} \mathrm{INF}(S_i)$. 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. 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 with complete simplification orderings. This is not a real drawback since all 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 and by extending the notion of *critical pair criteria* to resolution and paramodulation.

9. REFERENCES

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

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

Dershowitz N., Termination. In : Rewriting Techniques and Applications, J.P. Jouannaud, ed., LNCS, vol.202, (1985), pp.180-224.

Fribourg L., A Superposition Oriented Theorem Prover, Theoretical Computer Science,35,(1985) pp.129-164.

Henschen L., Wos L., Unit Refutation and Horn Sets, JACM 21 (1974),pp.295-301.

Hsiang J., Rusinowitch M., A new method for establishing refutational completeness in theorem proving. CADE (1986). Oxford.

Hsiang J., Rusinowitch M., Proving refutational completeness of theorem proving strategies. The transfinite semantic tree method. (1988). To appear in JACM.

Hsiang J. , Rusinowitch M., On word problems in equational theories. ICALP (1987). Karlsruhe.

Huet G., A Complete Proof of Correctness of Knuth-Bendix Completion Algorithm, JCSS,23,(1981) pp.11-21.

Kounalis E., Rusinowitch M., On word problems in Horn theories. Proceedings of 9th CADE(1988).

Knuth D. , Bendix P., Simple words problems in universal algebras, in Computational problems in abstract algebra. Pergamon press, (1970).

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

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

Lusk E.L., Overbeek R.A., A Portable Environment for Research in Automated Reasoning, 7th CADE, Nappa Valley,CA, LNCS 170, (1984).

Paul E., On solving the equality problem in theories defined by Horn clauses. Proc. of EUROCAL 1985.

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

Rusinowitch M., Démonstration automatique par des techniques de réécriture. Thèse. Université de Nancy 1, 1987.

Wos L.T. and Robinson G.A., Paramodulation and theorem proving in first-order theories with equality. Machine intelligence 4, American Elsevier pp135-150 (1969).