A SUPERPOSITION ORIENTED THEOREM PROVER

L. Fribourg

L.I.T.P
2, Place Jussieu
75251 PARIS Cedex 05
France

ABSTRACT

A theorem proving procedure is described which combines the approach of locking resolution with that of rewriting systems. Indeed, both the binary resolution and a complete restriction of paramodulation are embodied by an extension of the rewriting operation called superposition. Experimental results are reported and compared with literature automated proofs.

I - INTRODUCTION

We describe in this paper a theorem-proving procedure for first-order logic with equality which combines the approach of locked resolution with that of Term Rewriting Systems .

In 1971, R.S. Boyer introduced a restriction of resolution (without naranodulation) called "locking" which involves an index ordering of literals within clauses [Bo] This index ordering restriction was somehow refined and extended to paramodulation under the name of 0 E-resolution by J.A. Loveland [Lo]. Independently in some other works, paramodulation was specifically controlled by favoring one direction in the substitution of equals : a subterm u within a clause can be replaced by an equal only if u matches with a certain side of an equation alpha=betanstance alpha and not B The concept of demodulation and the equation marking one(atom term locking) have thus been introduced [WR] [La].

In the procedure described herein, we use this notion of unidirectional paramodulation but with an increased selectivity : indeed, the subterm u of an equation y=6 can be replaced only if u belongs to a certain side of that equation, for instance y and not 6. The restriction is developed on the basis of the 0JE-resolution and has the completeness property.

Its strongly oriented character induces us to consider the restricted paramodulation as a form of superposition (the rewriting operation). We achieve this in using a new formalism of clauses that we call "equational clauses", each literal being converted into an equation. We will show that superposition on to equational clauses embodies not only paramodulation but also binary resolution and will so constitute our major rule of inference.

II - I-ORDERING

The reader is assumed to be familiar with the notions of E-insatisfiability and classical resolution. In this section, we will summarize the definitions and results attached to the index ordering deduction by resolution and paramodulation, which is named 0J E-deduction in [Lo], and here simply noted as I-deduction.

In I-deduction, a positive integer is assigned to each literal occurrence of the given set S of clauses. Within a clause, literals are disposed from left to right in non - decreasing index order. Such ordered clauses are called I-clauses. Within an I-clause, only the leftmost occurrence of identical literals is retained (merging low rule) and positions among literals of like index are interchangeable.

Def : given an I-clause C, if the rightmost literal L and other literals having the same index are unifiable with the most general unifier σ, an I-factor of C is the I-clause Cσ obtained from C by instanciating literals, indexing them as in C and merging them low.

Def : if $C_1 : (C'_1, L_1)$ and $C_2 : (C'_2, L_2)$ are I-clauses such that the rightmost literal L_1 of C_1 and the complementary one $\neg L_2$ of C_2 are unifiable with m.g.u.σ, an I-binary resolvent C of C_1 and C_2 is the I-clause such that :

(1) the set of literals is $\{C'_1\sigma, C'_2\sigma\}$

(2) the literals of C are indexed as their parent literals and are merged low.

Def : if $C_1 : (C'_1, s=t)$ and $C_2 : (C'_2, L_2)$ are I-clauses such that the rightmost literal L_2 of C_2 contains : either

(i) a subterm s' at occurrence u which is unifiable with the left-hand side s of the rightmost literal of C_1 (with m.g.u.σ)

or (ii) a subterm t' at occurrence v which is unifiable with the right-hand side t of the rightmost literal of C_1 (with m.g.u η), then an I-paramodulant C of C_1 into C_2 is the I-clause such that :

(1) the set of literals of C is

$\{C'_1\sigma, C'_2\sigma, L_2\sigma[u \leftarrow t]\}$ (resp. $\{C'_1\eta, C'_2\eta, L_2\eta[v \leftarrow s]\}$)

(2) the literals of C are indexed as their parent literals - except the descendant of L_2 which receives the index $N+1$, where N is the highest index assigned to an I-clause of S - , and they are merged low.

Def : for a given set S of I-clauses, the set of functionnally reflexive axioms is the set defined as $S^F = \{f(x_1,\ldots,x_n) = f(x_1,\ldots,x_n)$ for all n-adic function letters f occurring in S $\}$

Theorem [Lo] : if S is a finite E-unsatisfiable set of I-clauses, then there exists a deduction of the empty clause (D) from $S \cup \{x=x\} \cup S^F$ by I-resolution, I-factoring and I-paramodulation.

III - EQUATION MARKING

In order to control even more the I-paramodulation rule, we shall mark the equations nearly the same as Lankford does in term locking I La J.But unlike [La], marking does not affect each equation and descendant clause marking is not induced by parent clause marking.

Def : an I-clause is marked when its rightmost literal, being an equation, has one and only one underscored side.

In our procedure, starting with a given set of (marked) I-clauses, the descendant clauses are marked through a binary resolution or factoring when an equation becomes the rightmost literal or through a paramodulation when the paramodulated literal is an equation. In every case, the side to be marked is freely chosen. For instance, the I-clause (b=c,Pa) can be resolvedagainst the I-clause (Pa) into either (b=c) or (b=c) .

The equation marking enables us to distinguish four types of paramodulation. First, two cases of paramodulation can be defined, depending on whether the paramodulant is obtained (i) by a matching with the marked side of the active equation or (ii) by a matching with the unmarked side. Then for each of these cases, two new subcases can be defined, depending on whether (j) the paramodulated literal is not an equation, or is an equation whose marked side contains the matched subterm ; (jj) the paramodulated literal is an equation whose unmarked side contains the matched subterm. A paramodulation satisfying both cases (i) and (j) is said to be a)-typed.

IV - COMPLETENESS OF a)-TYPED PARAMODULATION

Theorem : If S is a finite E-unsatisfiable set of marked I-clauses, there exists a deduction of D from $S \cup \{x=x\} \cup s^F$ by I-factoring, I-binary resolution and I-paramodulation of type a).

The proof is given in [Fr].

V - LINK WITH REWRITING SYSTEMS

The a)-typed paramodulation appears as a directed kind of paramodulation where the equations are treated as a one way-replacement from the marked side to the unmarked one. Now in Term Rewriting Systems [KB] [HO], equations are also used unidirectionnally, but then from left to right. In order to imitate Rewriting, we may have to invert the sides of an equation so that the side to be marked becomes the left-hand one. We are entitled to do so, provided that we extend the resolution and factoring rules so that they are no longer submitted to the order of the sides within equations (thus we can factorize (a=b,a=b) into (a=b) and resolve (b=a) against (a not = b)). Henceforth, marking an I-clause will only consists of choosing a left-right orientation for the righmost equation.

Since the marked side of the equation is on the left, the a)-typed paramodulation of an equation E1 into an other one E2 involves the matching of the left-hand side of E1 with a subterm r in the left-hand side of E2 . In Rewriting language, this is a superposition of E1 on E2 - provided that r is not a variable.

Let us consider now the a)paramodulation of an equation $s=t$ into a literal P which is not an equation. P is either (1) an atom A or (2) a negation $\neg A$. In order to continue with the comparison between superposition and a)-paramodulation, we have to superpose s=t on P , which implies assimilating the matched term in P to a subterm of an equation left-hand side. This is achieved by writing P in the form of(1')A=true or (2')A-false, where "true" and "false" *are* new constant symbols. This leads us to define a new formalism (noted [EC I) of special clauses, called equational clauses, in which every literal has an equational form.

Def : given an I-clause $C : (L_1, L_2, \ldots, L_n)$, the associated equational clause is the index ordered set $D : (E_1, E_2, \ldots, E_n)$ where, for $1 \leq k \leq n$, E_k is :

- $s=t$, if L_k is the equation $s=t$
- $E(s,t)$=false, if L_k is the inequation $s \neq t$ (E is a new equality symbol)
- $P(t_1,\ldots,t_n)$=true, if L_k is the atom $P(t_1,\ldots,t_n)$ where P is any n-adic relation symbol except "="
- $P(t_1,\ldots,t_n)$=false, if L_k is $\neg P(t_1,\ldots,t_n)$

For $1 \leq k \leq n$, E_k is indexed as L_k .

Thus in [EC] formalism, the a)-paramodulation can be viewed as a rewriting operation of superposition extended to the variable subterms and involving the rightmost literal of equational clauses.

Furthermore the transposition into [EC] of the binary resolution also appears as a kind of superposition ! Let us consider two resolvable I-clauses $C : (C_1, A)$ and $C' : (C_2, \neg A)$ inferring (C_1, C_2) . The associated equational clauses are $D : (D_1, A=\text{true})$ and $D' : (D_2, A=\text{false})$. The superposition of D on D' infers the equational clause $(D_1, D_2, \text{true}=\text{false})$ or $(D_1, D_2, \text{false}=\text{true})$. If we add a rule enforcing the deletion of each literal true=false or false=true (trivial removal rule) , then we can retrieve the associated resolvent (D_1, D_2).

The [EC] transposition of the a)-paramodulation completeness theorem is :

Theorem : if S is a finite E-unsatisfiable set of oriented equational clauses, there exists a deduction of D, from S u {E(x,x)->true} u S by superposition, factoring and trivial removal.

So far, the orientation of term equalities has been used as a means of restricting paramodulation. Yet, the original idea behind orientation is to simplify terms by applying equality units [KB,SI].

This idea has been incorporated in [EC] by authorizing term simplifications within the rightmost literals. Our procedure then has the features of a Knuth-Bendix algorithm running on equational clauses ; still it is a form of locking resolution when restricted to clauses without term equality [Gr et al].

VI - EXPERIMENTAL RESULTS

The implemented program of Superposition on Equational Clauses (SEC) is written in LISP and runs on INRIA's HB 68. It has been developed as an extension to the system KB written by J.M. Hullot [HH,HO], now a part of the FORMEL system. We present the most noteworthy examples with, for each one, the equivalent literature results of a theorem prover chosen for its specific competitiveness. By comparison, SEC proofs are often impressively shorter - with regard to the number of generated clauses. The last example - which fails - is included to point out that so far, for very rich sets of initial axioms, SEC runs out of space.

	SEC		COMPETITOR		
ex	n° clauses generated	n° clauses retained	ref.of competitor	n°clauses generated	n° clauses retained
1	39	27	[Gr]	52	
2	15	15	[Fl]	206	99
3	112	43	[Br]	154	
4	48	48	[HR]	597	80
5	80	39	[WR]	1144	383
6	32	32	[Nv]	245	
7	>250		[Nv]	960	

examples

1 : if S is a subset of group such that $xy^{-1} \in S$ for every $x,y \in S$, then $x^{-1} \in S$

2 : in a group, if $x^2=e$ for every x, the group is commutative

3 : in an ordered field, if $x > 0$ then $x^{-1} > 0$

4 : in a ring $-x.-y = x.y$ for every x and y

5 : a subgroup of index 2 is normal

6 : Grau's three axioms are sufficient to define a ternary Boolean algebra

7 : if H and K are subgroups of G, then HK is a subgroup of G iff HK = KH

VII - CONCLUSION

The reduced length of our experimental proofs convinces us that the described approach constitutes a progress in the handling of equality in resolution oriented systems. Our procedure efficiency is due to :

1) the combination of locking resolution with a strong, new and complete restriction of paramodulation

2) the use of a new formalism of equational clauses which unifies the two research processes of paramodulants and binary resolvents into the one of superposants.

3) the use of equality units as simplifiers.

REFERENCES

[Bo] Boyer, R.S., "Locking : A Restriction of Resolution", Ph. D., U. of Texas, 1971

[Br] Brand, D., "Analytic Resolution in Theorem Proving", Artificial Intelligence 7 (1976)

[Fl] Fleisig, S., Loveland, D., Smiley,A.K. & Yarmush, D.L., "An Implementation of the Model Elimination Proof Procedure", J.ACM 21: 1 (1974) 124-139

[Fr] Fribourg, L., "A Superposition Oriented Theorem Prover", Report 83-11, LITP, 1983

[Gr et al.] Greenbaum, S., Nagasaka, A., O'Rorke P., & Plaisted, D., "Comparison of natural deduction and locking resolution implementations", 6th CAD, N.Y., 1982, pp. 159-171

[HH] Huet, G.& Hullot, J.M., "Proofs by induction in equational theories with constructors", 21st FOCS, 1980

[HO] Huet, G. & Oppen, D., "Equations and Rewrite Rules : A Survey", Formal Languages : Perspectives and Open Problems. Ed. Book R. Academic Press, 1980

[HR] Harrison, M.C. & Rubin, N., "Another generalization of resolution", J. ACM 25:3 1978 341-351

[KB] Knuth, D. & Bendix, P., "Simple word Problems in Universal Algebras", Computational Problems in Abstract Algebras. Pergamon Press, 1978

[La] Lankford, D., "Equality atom term locking", Ph. D., U. of Texas, 1972

[Lo] Loveland, D., "Automated Theorem Proving : A logical basis", Fundamental Studies in Computer Science. North Holland, 1978

[Nv] Nevins, A.J., "A human oriented logic for automatic theorem proving", JACM 21:4 (1974) 606-621

[RW] Robinson, G. & Wos, l; "Paramodulation and theorem proving in first order theories with equality", Machine Intelligence 4, Meltzer & Michie, eds, 1969

[SI] Slagle, J.R., "Automated Theorem Proving for Theories with simplifiers", J. ACM 21:4 (1974)

[WR] Wos, L., Robinson, G., Carson, D. & Shalla,L., "The concept of demodulation in theorem proving", JACM 14:4 (1967) 698-709