Paramodulation (Some Notes)

Krysia Broda

March 2, 2004

1 About Models of the Equality Axioms

a Not every model of the equality axioms (*i.e.* reflexivity, symmetry, transitivity and substitutivity axioms) treats = as identity.

For example, consider a structure with domain (A, B, C), for a language including just the equality predicate = and constants $\{a, b, c\}$, which interprets a, b, c as A, B, C respectively and = as the relation $\{(A, B), (B, A), (A, A), (B, B), (C, C)\}$. *i.e.* $= (A, B), = (B, A)$ *etc.* are all true and other atoms involving =, such as $= (B, C)$, are false. This is a model of the equality axioms (show it !) and it also satisfies the formula $a = b$ (for example). However, a and b do not denote the same element in the domain, so equality is not being treated as identity here.

b If S is a set of clauses and has a Herbrand model which satisfies the equality axioms (called an E-interpretation), then it also has a normal model (*i.e.* one in which = *is* treated as identity). See (c) below for an outline proof. (Note that S does not have a Herbrand model that treats = as identity; for example, if $a = b$ (say) is a clause in S, then a and b map to unique elements of the HU, in contradiction to = being identity.)

The contrapositive states that if S has no normal model, then S has no Herbrand model which satisfies the equality axioms. By an earlier theorem, if $S \cup \{EQAX\}$ has no Herbrand models then it has no models at all. Moreover, if $S \cup \{EQAX\}$ has no model, then in particular S has no normal model, for a normal model of S would certainly satisfy the EQAX as well. Therefore,

$$\begin{gathered} S \cup \{EQAX\} \text{ is unsatisfiable (or } S \text{ is E-unsatisfiable)} \\ \leftrightarrow S \text{ has no normal model} \end{gathered}$$

For refutations it is therefore sufficient to work with E-unsatisfiability. (*i.e.* to show S together with the equality axioms is unsatisfiable).

c Proof outline (of b): Let I be H-interpretation of $S \cup \{EQAX\}$, then if the interpretation I^* defined as follows is a normal model of S. The elements of the domain of I are collapsed to their equivalence classes under the predicate $=$ and these equivalence classes form the domain of I^*. The denotations of function symbols and the truth values of predicates are re-evaluated using the new domain in an obvious way (see example below). This can be done consistently because I satisfies the equality axioms.

As an example, take the non-normal model given in (a) and make it into a Herbrand model by changing the domain to $\{a, b, c\}$ and interpreting the predicates in a corresponding way. We'll call it I. So $=(a, b)$, $=(b, a)$, $=(a, a)$, etc. are true as before. In addition, suppose the signature is extended to include monadic predicate symbols P, Q and let S be the set $\{P(a), Q(b), Q(c), a = b\}$. Suppose $P(a)$, $P(b)$, $Q(a)$, $Q(b)$, $Q(c)$ are true and $P(c)$ is false in I, then I satisfies S and the equality axioms. (Any relation satisfying EQAX is an equivalence relation on HU.)

Now form I^* : Domain $= \{\| a \|, \| c \|\}$, where $\| a \| = \| b \| = \{a, b\}$, $\| c \| = \{c\}$, and $\| x \|$ names the equivalence class of x for the equivalence relation induced by $=$ and implicit in I.

We have $\| a \| = \| b \|$. If the denotation of P under I^* is $P(\| a \|)$, $Q(\| a \|)$, $Q(\| c \|)$ all true, $P(\| c \|)$ false, with $=$ mapping to the identity relation, then I^* is a model of S and equality is identity in this model.

Generally, let the new domain of I^* be the equivalence classes induced by $=$ in I. If $P(x)$ was assigned true before, then $P(\| x \|)$ is assigned true in I^*. This is a sound asssignment as for all elements u and v in $\| z \|$, $P(u)$ and $P(v)$ had the same truth value in I (by virtue of the substitutivity axioms). The arguments can be extended to cope with any function symbols that may be involved. In I^* $=$ is now identity: if $a = b$ is to be satisfied then a and b map to the same domain element, namely $\| a \| = \| b \|$.

2 Completeness Proofs for Paramodulation

Completeness proofs for paramodulation (and refinements) are generally difficult. Often, the simpler the refinement the more difficult the proof. However, some proofs first find a refutation using resolution and equality axioms and then reconstruct this refutation into one in which the parts using equality axioms can be replaced by paramodulation steps. This is particularly easy in the case of hyper-resolution as we show. Some proofs of completeness use a generalisation of semantic trees. See [1] and [2].

Another problem with paramodulation is that even if a ground level refutation within some particular refinement is obtained, finding it at the general level can be very difficult (indeed impossible sometimes, unless axioms such as $f(x) = f(x)$ are added explicitly). (*e.g.* the clauses $\{a = b, b = c, P(u, u, f(c)), \neg P(f(a), v, v)\}$ have a ground level refutation using paramodulation and resolution using the instances $P(f(a), f(a), f(c))$ and

$\neg P(f(a), f(c), f(c))$, but this particular refutation cannot be found at the general level without the additional use of the equation $f(a) = f(a)$. Another refutation using different instances *can* be found, however. This is another example of the two sorts of completeness, namely (a): *every* refutation can be found (a strong completeness), or (b): at *least one* refutation can be found (a weaker completeness).)

One refinement is obtained by restricting paramodulation to those steps which are necessary in order to resolve two literals. So one or more paramodulation steps can be applied to one or more literals in order to make them unifiable, but otherwise no paramodulation is carried out. Detection of such sequences of paramodulation steps is difficult enough at ground level. It is undecidable in general whether such a sequence exists, so detection is very difficult. Hence syntactic restrictions such as hyper-paramodulation, derived from hyper-resolution restrictions imposed on EQAX, and oriented paramodulation, in which equalities are oriented, have proved most useful in practice. (See Otter.)

3 Paramodulation and Various Refinements

3.1 Paramodulation and Hyper-resolution

The paramodulation step, if rewritten in terms of its basic resolution steps is a combination of the use of the substitutivity axioms for functions and predicates, using hyper-resolution.

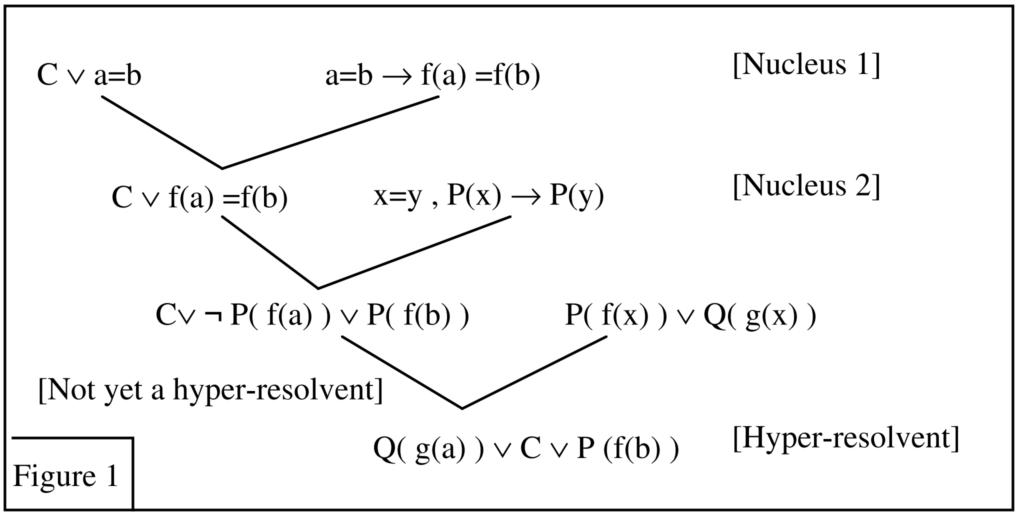

Figure 1:

Consider a typical example in which $C \vee a = b$ is paramodulated into $P(f(x)) \vee Q(g(x))$ which is shown in full in Figure 1; the last resolvent $C \vee P(f(b)) \vee Q(g(a))$ is the result of one (of several) paramodulation steps between $C \vee a = b$ and $P(f(x)) \vee Q(g(x))$. Notice that the two clauses involved in the paramodulation step are both positive, which is enforced by the restrictions of the hyper-resolution strategy. We have:

Completeness of hyper-paramodulation

If S is an E-unsatisfiable set of clauses, then there is a hyper-refutation from S using paramodulation and hyper-resolution from $S \cup \{x = x\} \cup F$, where F is the set of functionally reflexive axioms for S. (*i.e.* $f(x) = f(x), g(x) = g(x)$ *etc.* for function symbols in S. needed for the lifting step.)

Proof

Since S is E-unsatisfiable, by completeness of hyper-resolution S plus the equality axioms have a hyper-refutation P. We show that each clause in P can be derived using paramodulation and hyper-resolution from $S \cup \{x = x\} \cup F$.

For each hyper-resolvent generated in P, the electrons must either be in S, or be $x = x$, or belong to F, or be derived, as no other clause in the equality axioms is all positive. If the nucleus is also in S there is no further consideration of the step, which is a hyper-resolution step. If not, it must be from the equality axioms and can be one of four kinds: (i) symmetry, (ii) transitivity, (iii) substitutivity in predicates, (iv) substitutivity in functions. [Cases (i) and (ii) are special cases of (iii)].

i N (the nucleus) is $x \neq y \vee y = x$. There is just one electron, of the form $t1 = t2 \vee C$. The hyper-resolvent is $(t2 = t1 \vee C)\tau$ which is a paramodulant of $x = x$ and $t1 = t2 \vee C$. ($t1$ is matched with the first x in $x = x$ and replaced by $t2$, the second x (by unification) being bound to $t1$. τ is the unifying substitution (not really needed here).)

ii N is $x \neq y \vee y \neq z \vee x = z$. There are two electrons, $t1 = t2 \vee C$, $s1 = s2 \vee D$ and the hyper- resolvent is $(t1 = s2 \vee C \vee D)\tau$, where $t2\tau = s1\tau$. This is also a paramodulant of the two electrons.

iii N has the form $r \neq s \vee \neg P(x1 \ldots r \ldots x_n) \vee P(x1 \ldots s \ldots x_n)$ and the electrons have the form $t1 = t2 \vee C$ and $P(s1 \ldots t3 \ldots s_n) \vee D$. The hyper-resolvent is $(P(s1 \ldots t2 \ldots s_n) \vee C \vee D)\tau$ where $t1\tau = r\tau = t3\tau$, $t2\tau = s\tau$. This is the result of paramodulating $t1 = t2 \vee C$ and $P(s1 \ldots t3 \ldots s_n) \vee D$.

iv N has the form $r \neq s \vee f(x1 \ldots r \ldots x_n) = f(x1 \ldots s \ldots x_n)$ and the electron is $t1 = t2 \vee C$. The hyper-resolvent is $(f(x1 \ldots t1 \ldots x_n) = f(x1 \ldots t2 \ldots x_n) \vee C)\tau$. This is a paramodulant of $t1 = t2 \vee C$ and $f(x1 \ldots x_n) = f(x1 \ldots x_n)$ (match $t1$ with the appropriate argument (say x_j) in the second $f(x1 \ldots x_n)$ and replace it by $t2\tau$).

In practice the steps (iii) and (iv) are combined into one paramodulation step and it is here that the functionally reflexive axioms may be required explicitly, if the result of the intermediate step is required to mkae a hyper-resolution step. For example, look again at Figure 1. Suppose, after deriving $C \vee f(a) = f(b)$, the hyper-refutation had resolved with the nucleus clause $f(a) \neq f(b)$ to give the final electron C. That is, the two clauses involved were $f(a) \neq f(b)$ and $C \vee a = b$. The corresponding step using hyper-paramodulation has to be to paramodulate $a = b$ into $f(x) = f(x)$ to yield $C \vee f(a) = f(b)$ and then to resolve with $f(a) \neq f(b)$ as before. This is because paramodulation is not allowed into a negative literal.

It is also possible to incorporate paramodulation into hyper-resolution when the predicates are ordered. Then, paramodulation can be given priority over resolution, or otherwise, by the position of $=$ in the ordering.

3.2 Paramodulation and ME

Consider now incorporating paramodulation into ME tableaux, although to date there have been very few practical combinations of oriented paramodulation and the ME-refinement.

Positive equality literals may arise either as a selected literal or be in an input clause. In the former case, a natural use would seem to be to generate new ancestors from old using paramodulation. This is more like rewriting (see later). When an equation is in an input clause one natural use is to use it to facilitate resolution between a leaf literal and some other. For example, one could use the clause $c = d \vee P$ in a branch that includes $C(c)$ and $\neg C(d)$. Paramodulation between $C(c)$ and $c = d$ gives $C(d) \vee P$, which ancestor resolves with $\neg C(d)$. Exactly which combinations of the above possibilities are either necessary or sufficient for completeness is not known. In addition, the orientation of equalities might also be incorporated. Simple counter examples show that paramodulation cannot be restricted to those equalities appearing as selected literals if *any* top clause (in the set of support) is to be allowed.

In the example in Figure 2, which uses the particular approach described in the slides, there is a paramodulation step between $el([1,2],[2,1]) = 1$ and $\neg el([1,2],[2,1]) \in [1]$, replacing $el([1,2],[2,1])$ by 1.

Additionally, an equality might be considered similar to a rewrite rule and applied (everywhere) to previous literals in a branch in this guise (if appropriate), and even to input clauses, as long as unification is simple matching and variables in the rewritten literals are not bound. The completeness of this latter strategy would seem to be provable in a way similar to that used in[1].

4 Oriented Paramodulation

Definition (*Oriented Paramodulation*)

$l = r \vee C$ paramodulates into $s[u]$ if $l\sigma = u\sigma$ and $\neg(l\sigma \leq r\sigma)$, where σ is the m.g.u. of l

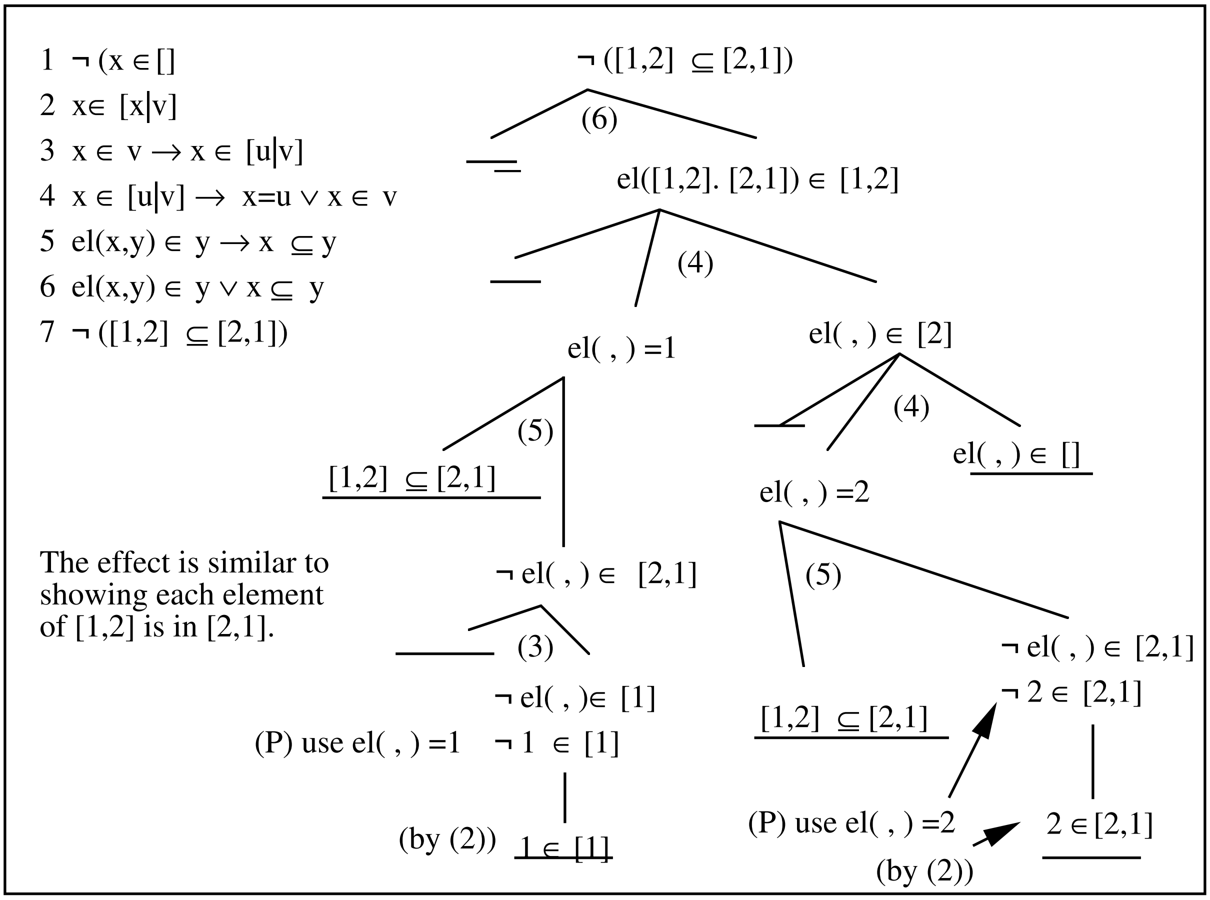
1 ¬ (x ∈ []
2 x∈ [x|v]
3 x ∈ v → x ∈ [u|v]
4 x ∈ [u|v] → x=u ∨ x ∈ v
5 el(x,y) ∈ y → x ⊆ y
6 el(x,y) ∈ y ∨ x ⊆ y
7 ¬ ([1,2] ⊆ [2,1])
¬ ([1,2] ⊆ [2,1])
(6)
el([1,2]. [2,1]) ∈ [1,2]
(4)
el(,) =1
el(,) ∈ [2]
(5)
(4)
[1,2] ⊆ [2,1]
el(,) ∈ []
el(,) =2
The effect is similar to showing each element of [1,2] is in [2,1].
¬ el(,) ∈ [2,1]
(5)
(3)
¬ el(,) ∈ [2,1]
¬ el(,)∈ [1]
¬ 2 ∈ [2,1]
(P) use el(,) =1
¬ 1 ∈ [1]
[1,2] ⊆ [2,1]
(P) use el(,) =2
2 ∈ [2,1]
(by (2))
1 ∈ [1]
(by (2))

Figure 2:

ad u.

(See [1]. Note: The notation $C2[r]$ means clause $C2$ contains r as a subterm, and may contain r more than once. Moreover, the notation $C2[t\sigma]$ indicates that one or more occurrences of $r\sigma$ have been replaced by $t\sigma$. Not all occurrences need to be replaced and usually just one occurrence is replaced at a time. Terms are ordered (as in rewriting) and paramodulation is restricted so that the equation is used to replace a large term by a smaller term. The restriction $\neg(l\sigma \leq r\sigma)$ ensures that certainly a small term is not replaced by a larger one. The particular type of ordering used has to be total on ground terms and also be monotonic (i.e. respect contexts, so that, for example, if $x < y$ then $f(x) < f(y)$) and satisfy the requirement that a subterm of $t < t$.

For the next example, a possible ordering is the recursive path ordering based on $b < a < f$. Moreover, in the refutation only ground occurrences of equations are used in paramodulation steps.

Example: (from Gallier)

(1) $f(f(f(y))) = y \leftarrow f(y) = f(b)$	(2) $a = b$
(3) $P(a)$	(4) $f^5(x) = x \leftarrow Q(x)$
(5) $Q(a) \leftarrow f^3(a) = a$	(6) $R(a) \leftarrow f(a) = a \wedge P(f(a))$
(7) $\neg R(f(z)) \vee \neg P(z)$	

Use (2), to derive (3'), (5'), (6') by paramodulation - (simplification here.) (2) can be deleted as there need be no more occurrences of a generated. (It is not so easy in general to be sure of this kind of property when function symbols are involved. Consider, for example, how you might be sure that no more occurrences of $f(a)$ (say) would be involved, having replaced those $f(a)$ that do occur by some other term.) We have,

(3') $P(b)$
(5') $Q(b) \leftarrow f^3(b) = b$
(6') $R(b) \leftarrow f(b) = b \wedge P(f(b))$

There are several possible strategies next; for example, perform resolution first or paramodulation first.

Here, let's resolve (5') with (1) and the axiom $x = x$ to derive

(8)	$Q(b)$	
(9)	$f^5(b) = b$	(from 8, 4)
(10)	$f^2(b) = b \leftarrow f(b) = f(b)$	Paramodulate (1) into (9) ($y = = b$)
	$\Rightarrow f^2(b) = b$	
(11)	$f(b) = b$	Paramodulate (10) into (1) ($y = = b$)
(12)	$R(b) \leftarrow b = b \wedge P(b)$	Resolve (11) with (6')

	$\Rightarrow R(b)$	by (3')
(13)	$\neg R(f(b))$	Resolve (7) with (3')
(14)	$\neg R(b)$	Paramodulate (11) into (13)
	$\Rightarrow \Box$	Resolve with (12).

By making use of this paramodulation refinement, the Knuth Bendix procedure can be simulated. Superposition corresponds to (oriented) paramodulation. (Equations which cannot be ordered into a rewrite rule according to the ordering are left as equations which can be used in either direction; the others can be turned into rewrites.)

In addition to oriented paramodulation it is possible to incorporate resolution refinements such as hyper-resolution or ordering of predicate symbols. If rewriting is applied to terms whenever possible, the clause prior to rewriting can be removed from the set of clauses. This would not be the case with narowing, when the paramodulation steps binds variables in the clause being rewritten. So $P(x, f(x))$ can be replaced by $P(x, x)$ after rewriting by $f(x) = x$ but not after narrowing by $f(a) = a$ to yield $P(a, a)$, since the original clause may still be needed with some substitution other than a.

Knuth-Bendix completion can also be partially simulated by avoiding the use of equality and using predicates to represent funtion symbols or operators. To simulate the procedure the use of a functionality axiom is often required together with (in that case) substitutivity. (So equality is not really eliminated it would seem .)

Example

(a)	$Plus(x, 0, x)$	(1)	$x + 0 = x$
(b)	$Plus(0, x, x)$	(2)	$0 + x = x$
(c)	$Plus(x, i(x), 0)$	(3)	$x + i(x) = 0$
			$i.e.\ x + -x = 0$
(d)	$(Plus(a, z, r) \leftrightarrow Plus(x, b, r))$	(4)	$(x + y) + z = x + (y + z)$
	$\leftarrow Plus(x, y, z) \wedge Plus(y, z, b)$		
(e)	$Plus(x, y, z) \wedge Plus(x, y, w)$		(functionality of $+$)
	$\rightarrow z = w$		

The completion procedure generates :

(5)	$i(0) = 0$, (or $-0 = 0$)	(from 2,3)
(6)	$x + (-(x) + z) = z$	(from 3,4)
(7)	$- - x = x$	(from 3,6)
(8)	$-x + x = 0$	
(9)	$-x + (x + z) = z$	*etc.*

Using the predicate *Plus* and ordinary resolution you can obtain:

(from (c),(d) with $\{y == x, z == i(x), b == 0\}$)
(f) $(Plus(0, z, r) \leftrightarrow Plus(x, b, r)) \leftarrow Plus(i(x), z, b)$
and then (from (f) and (b))
(g) $Plus(x, b, z) \leftarrow Plus(i(x), z, b)$ (equivalent to (6)).
Again, using (c) : $Plus(x, 0, i(i(x))$
By (e) $i(i(x)) = x$ (equivalent to 7).

The funtionality (e) is used to generate equalities which can be used in rewriting steps although it is not always needed.

Otter has many user restrictions that can be imposed on paramodulation and you are encouraged to try them out. It does not incorporate a full form of oriented paramodulation yet (or so I believe).

5 Other Approaches: Extensions and Alternatives to Paramodulation

5.1 E-resolution

E-resolution was first described in 1969 (Morris) and is still a popular method - in the sense that researchers have refined it by extending the idea to the use of theories other than equality. E-resolution allows two literals to be resolved if they can be shown to be complementary by using equality literals. Gallier has investigated thoroughly this idea within the realm of Horn clauses in order to incorporate equality into logic programming. (See [3].).

To extend the ideas to ME-resolution would, I think, be a very interesting enterprise. Dershowitz also has done much work in the general area of E-resolution ([4]). Because of the (in general) undecidability of the problem of E-unification tractable subcases are usually considered. I think the orderings introduced into rewriting systems to guarantee termination of the Knuth Bendix procedure are valuable for this work.

5.2 RUE-resolution

Digricoli has described the RUE resolution rule (see [5]) and more recently in [6] as another approach. Given two literals $P(s1 \dots s_n)$ and $P(t1 \dots t_n)$ the RUE resolvent is a disjunction of inequalities specified by the "disagreement set" of the terms $s1 \dots s_n$ and $t1 \dots t_n$.

For example, $P(a, f(x, x))$ and $P(b, f(a, c))$ has the disagreement set, derived from $\{a : b, f(x, x) : f(a, c)\}$. It is $\{a : b, x : a, x : c\}$ and hence the corresponding RUE resolvent is $a \neq b \vee x \neq a \vee x \neq c$ and $a \neq b \vee a \neq c$.

Procedurally, this rule says "$P(s_i)$ and $\neg P(t_i)$ will resolve to the empty clause if $s_i = t_i$ can be shown (*i.e.* if the goal $\bigvee s_i \neq t_i$ is unsatisfiable)". Thus the sequence of paramodulation steps applied to a literal to transform it into another with which it is complementary is performed after, rather than before, the match.

(Notice that for the example given earlier, as a difficulty for paramodulation, namely $\{a = b, b = c, P(u, u, f(c)), \neg P(f(a), v, v)\}$ RUE resolution will eventually generate the clause $f(a) \neq f(c)$, which is unsatisfiable using the clauses $a = b$ and $b = c$. But there are other solutions that cannot be found easily using RUE that can be found by ordinary paramodulation.)

The relationship betweeen this and other techniques such as rewriting have not been investigated to my knowledge and would be interesting. S. Reeves ([8]) has explored a similar method to incorporate equality into tableau procedures and Beckert ([7]) developed ideas on similar lines for tableaux. The use of equality in tableaux methods is thoroughly surveyed in the *Handbook of Tableau Methods* edited by Hahnle et al. Dimitris Raptis, [9], investigated RUE resolution very thoroughly as an alternative to paramodulation in a practical theorem prover.

The idea used by Digricoli, namely "Pa and $\neg Pb$ can be resolved if $a = b$ can be shown" (or $a \neq b$ refuted) is extended by Manna & Waldinger to other relations. For example,

> Pa and $\neg Pb$ can be resolved if $a < b$ and also $Pa \rightarrow Px$, $\forall x > a$, for then $Pa \rightarrow Pb$, and $Pb, \neg Pb$ can be resolved.

The relation '$<$' could be any binary relation. Manna and Waldinger call this extended paramodulation, for it uses axioms similar to the substitutivity axioms for equality. These axioms are also like the definitions of increasing and decreasing functions: $x < y \rightarrow f(x) < f(y)$. See [10]. Additionally, they reported on the completeness of their methods in [11].

References

[1] J. Hsiang and M. Rusinowitch CADE - 8

[2] Peterson, 1983 Siam J. of Computing

[3] J. Gallier, Symposium of logic programming 1987

[4] Dershowitz, Symposium of logic programming 1987

[5] Digricoli, IJCAI-81

[6] Digricoli, CADE 12 (94)

[7] B. Beckert, 2nd Workshop on tableau and related methods, Marseille, 1993

[8] S. Reeves, J. Aut. Reasoning 1987

[9] D. Raptis, PhD, DOC, IC, 1996

[10] Z. Manna and R. Waldinger, Vol2 of The Logical Basis for Computer Programming

[11] Z. Manna and R. Waldinger, CADE 11 1992.