SLAC-PUB-482 September 1968 (MISC)

Paramodulation and Theorem-proving in First-order

Theories with Equality\*

George Robinson Stanford Linear Accelerator Center, Stanford, California

and

Lawrence Wos Argonne National Laboratory, Argonne, Illinois

# July 1968

Revised August 1968

\*Work supported by U.S. Atomic Energy Commission

(To appear in Machine Intelligence Vol IV, ed. by D. Michie.)

(Presented at the Fourth Annual Machine Intelligence Workshop, Edinburgh, Scotland, August 12-21, 1968)

#### Introduction

A <u>term</u> is an individual constant or variable or an n-adic function letter followed by n terms. An <u>atomic formula</u> is an n-adic predicate letter followed by n terms. A <u>literal</u> is an atomic formula or the negation thereof. A <u>clause</u> is a set of literals and is thought of as representing the universally-quantified disjunction of its members. It will sometimes be notationally convenient<sup>1</sup> to distinguish between the empty clause  $\Box$ , viewed as a clause, and "other" empty sets such as the empty set of clauses even though all these empty sets are the same set-theoretic object  $\phi$ . A <u>ground</u> clause (term, literal) is one with no variables. A clause C' (literal, term) is an <u>instance</u> of another clause C (literal, term) if there is a uniform replacement of the variables in C by terms that transforms C into C'.

The <u>Herbrand universe</u>  $H_S$  of a set S of clauses is the set of all terms that can be formed from the function letters and individual constants occurring in S (with the proviso that if S contains no individual constant, the constant 'a' is used). An <u>interpretation</u> I of a set S of clauses is a set of literals such that for each atomic formula F that can be formed from an n-adic predicate letter occurring in S and n terms from  $H_S$  exactly one of the literals F or  $\overline{F}$  (the negation of F) is in I.

For any set J of literals,  $\tilde{J}$  is the set of negations of members of J. The set J <u>satisfies</u> a ground clause C if  $J \cap C \neq \emptyset$  and <u>condemns</u> C if C -  $\tilde{J} = \emptyset$ .

- 1 -

<sup>&</sup>lt;sup>1</sup> Note, for example, that the empty set is a satisfiable set of clauses but at the same time is an unsatisfiable clause.

J <u>satisfies</u> a non-ground clause C if it satisfies every instance of C and <u>condemns</u> C if it condemns some instance of C. A clause (possibly ground) that is neither satisfied nor condemned by J is said to be <u>undefined</u> for J; otherwise it is <u>defined</u> for J. J <u>satisfies</u> a set S of clauses if it satisfies every clause in S and <u>condemns</u> S if it condemns some clause in S.

An R-<u>interpretation</u> of a set S of clauses is an interpretation I of S having the following properties: Let  $\alpha$ ,  $\beta$ , and  $\gamma$  be any terms in H<sub>S</sub> and L any literal in I. Then

1.  $(\alpha = \alpha) \in \mathbb{I}$ 

2. If  $(\alpha = \beta) \in I$  then  $(\beta = \alpha) \in I$ 

3. If  $(\alpha = \beta) \in I$  and  $(\beta = \gamma) \in I$ , then  $(\alpha = \gamma) \in I$ .

4. If L' is the result of replacing some one occurrence of  $\alpha$  in L by  $\beta$  and  $(\alpha = \beta) \in I$ , then L'  $\in I$ .

An (R-)model of S is an (R-)interpretation of S that satisfies S.

A set S of clauses is (R-)<u>satisfiable</u> if there is an (R-)model of S; otherwise it is (R-)<u>unsatisfiable</u>.

If S is a set of clauses or a single clause and T is a set of clauses or a single clause, S (R-)<u>implies</u>. T(abbreviation  $S \nvDash T$  or  $S \nvDash_R T$ ) if no (R-)model of S condemns T.

A deductive system W is (R-) <u>deduction-complete</u> if  $S \vdash_W T$  (T is deducible from S in the system W) whenever  $S \models T$  ( $S \models_R T$ ). W is (R-) <u>refutation-complete</u> if  $S \vdash_W D$  whenever S is (R-) unsatisfiable.

## Equality in Automatic Theorem-Proving

The methods for dealing with the concept of equality in theorem proving can be grouped roughly into three classes: (1) those which employ a set of

- 2 -

first-order axioms for equality, for example, the following set (which we shall call E(K), where K is the set of first-order sentences under study):

(i)  $(x_1) (x_1 = x_1)$ 

(ii)  $(x_1)...(x_n)(x_0)$   $(x_j \neq x_0 \vee Fx_1...x_j...x_n \vee Fx_1...x_0...x_n)$  (j=1,...,n) (iii)  $(x_1)...(x_n)(x_0)$   $(x_j \neq x_0 \vee f(x_1...x_j...x_n) = f(x_1...x_0...x_n))$  (j=1,...,n) where n axioms of the form (ii) are included for each n-adic (n>0) predicate letter P occurring in K and n axioms of the form (iii) are included for each n-adic (n>0) function letter in K<sup>2</sup>; (2) those which employ a smaller set of second-order axioms for equality; and (3) those which employ a substitution rule for equals as a rule of inference.

### Some Desirable Properties for Theorem-proving Algorithms

In addition to the logical properties of soundness and completeness, two sets of somewhat more elusive properties are of interest in judging the usefulness of the inference apparatus for automatic theorem proving.

The first set of properties (<u>efficiency</u>, <u>brevity</u>, and <u>naturalness</u>) are global properties in that they deal with the entire proof or proof-search and are of interest in themselves. <u>Efficiency</u> refers to the ease or dispatch with which the search procedure locates a proof. <u>Brevity</u> refers to the lengths of proofs found. <u>Naturalness</u> refers to being in the spirit of what a human mathematician might write in a proof. Other factors being equal, a briefer proof might be considered more natural, but naturalness goes beyond this. For example, among proofs of roughly the same length, a unit resolution proof<sup>3</sup> might be considered more natural than a non-unit proof.

- <sup>2</sup> Note that an interpretation I of K is a R-interpretation of K iff it satisfies E(K).
- <sup>3</sup> In effect one that is free from simultaneous case-analysis type reasoning and which prefers modus ponens to syllogism -- formally, one in which nonunit clauses are never resolved against each other.

-3-

The second set of properties (<u>immediacy</u>, <u>convergence</u>, and <u>generality</u>) are local properties in that they focus on only a small part of the proof or proof-search and are of interest primarily because they contribute to other properties such as efficiency.

Immediacy is rather easily grasped. One inference apparatus  $\mathcal{A}$  is said to be more immediate than another apparatus  $\mathcal{B}$  (at least for the case in question) when  $\mathcal{A}$  enables one to deduce a given conclusion from a given set of hypotheses in fewer steps than  $\mathcal{B}$ . For example (see Figure 1) if to infer F from D and E by  $\mathcal{B}$  one first had to infer G from D and only then infer F from E and G while  $\mathcal{A}$  allowed the inference of F directly from D and E in one step without recourse to G, then  $\mathcal{A}$  would (for this case) be more immediate than  $\mathcal{B}$ .

<u>Convergence</u> is a slightly subtler but, for automatic theorem-proving, perhaps more important property. Consider the clause G in the example above. Often such an intermediate result will seriously detract from proof search efficiency by interacting with other clauses to produce unnecessary "noise" in the proof search space, either by generating successive generations of less than helpful clauses, or somewhat less seriously, by requiring additional machine time to determine that no interesting clauses can be inferred from G. Freedom from this generation of "side-effect" clauses we call <u>convergence</u>. Thus in the example,  $\alpha$  is both more immediate and more convergent than  $\mathcal{B}$ .

-4-



- Figure l

<u>Generality</u> refers to choosing to infer a clause C rather than a proper instance of C when either inference could be made from the premises without loss of soundness. For example, inferring from f(xa) = g(x) and Qf(xa) the conclusion Qg(b), although sound, would be less general than inferring Qg(x).

It is not difficult to see the advantage of inferring a clause rather than a proper instance of that clause, since the more general clause, being stronger, has greater potential for future inferences. Perhaps even easier to see is the problem of deciding which proper instance to select if a proper instance were to be preferred to the more general clause. Usually there is an infinite set of proper instances. For example, from h(xyy) = g(x) and Qh(zww)a, we can infer Qg(x)a by substitution. There is, however, an infinite set of proper instances of Qg(x)a which could also be legitimately inferred. Among these are Qg(a)a, Qg(g(a))a, Qg(g(g(a)))a... We shall apply the phrase <u>most general</u> to a clause (or term) C with respect to some given condition when C satisfies the condition and no clause (term) which satisfies the condition has C as a proper instance.

Of the approaches to equality given earlier, approach 1 has three obvious disadvantages. One has to do with length of deduction chains in the proof. In order to infer from

- (1) Qa and
- (2) a = b
- the result
  - (3) දැං

one must first infer from the axiom

(4)  $x \neq y \lor \overline{Q}x \lor Qy$ 

and, say (1), the intermediate result

(5) a≠y V Qy

-6-

before passing from (5) and (2) to (3). By contrast, approach 3 would allow us to go directly from (1) and (2) to (3) without ever inferring the intermediate result (5). Thus approach 3 contributes to <u>brevity of proofs</u>. More important for proof search, it contributes (by means of immediacy) to <u>brevity</u> of deduction chains within proofs.

A second, and perhaps more serious disadvantage of approach 1 as compared to approach 3, is that the intermediate debris such as step (5) tends to spawn increasingly larger generations of generally useless offspring, polluting the search space badly. We describe this difference by saying that approach 3 tends to be more <u>convergent</u> than approach 1. (Presence of various subsidiary strategies such as set of support may possibly in some cases tend to mitigate the severity of such non-convergence effects.)

The third disadvantage of approach l is perhaps the least important although superficially the most obvious: the equality axioms E(K) must be present. The clerical chore of writing them all down could be eliminated merely by incorporating into the theorem-prover a program to generate them. Alternatively they may be specified by means of a schema (we shall call this variation approach lb), or in approach 2 by means of a few second-order axioms. We feel that this third disadvantage is so superficial and trivial (since one can simply place E(K) outside the set of support as is done in the standard set of support variant of approach 1) as to be quite spurious.

The method given by Darlington(1968)whether it be classed as approach lb or as approach 2 can be taken as typical of methods which avoid the third disadvantage (greater number of explicit axioms) but fail to dent the first and second disadvantages (longer deduction chains and non-convergence). In effect Darlington infers (5) from (1) and

(4')  $x \neq y \vee \overline{\phi}(x) \vee \phi(y)$ 

- 7 -

which is thought of either as a schema defining a set of first-order axioms including (4) or as a single second-order axiom having (4) as an instance.

## Paramodulation

Since our automatic theorem-proving environment consists exclusively of clauses, we should like our rule of inference for equality to operate on two clauses and yield a clause. Furthermore we should like it to apply to units and non-units alike<sup>4</sup> and to yield a most general clause that can be R-soundly inferred. We shall now describe the inference rule for paramodulation, which is asserted to have these properties. Examples of paramodulation are given in Figure 2.<sup>5</sup>

<u>Paramodulation</u>: Given clauses A and  $\alpha'=\beta'\vee B$  (or  $\beta'=\alpha'\vee B$ ) having no variable in common and such that A contains a term  $\delta$  with  $\delta$  and  $\alpha'$  having a most general common instance  $\alpha$  identical to  $\alpha'[s_i/u_i]$  identical to  $\delta[t_j/w_j]$ , where A' is obtained by replacing in  $A[t_j/w_j]$  some single occurrence of  $\alpha$ (resulting from an occurrence of  $\delta$ )<sup>6</sup> by  $\beta'[s_i/u_i]$ , infer A'  $\vee B[s_i/u_i]$ .<sup>7</sup>

- <sup>4</sup> Consider for example the set S = {c=d v Qc, g(c)≠g(d) v Qc, a=b v Qc, g(a)≠g(b) v Qc, x=x}. If the rule applied only to units, it would not be possible to refute this R-unsatisfiable set.
- <sup>5</sup> These examples are primarily to give an intuitive idea of how paramodulation works. A comparison of the length and complexity of paramodulation proofs vs. resolution proofs can be obtained by considering the proofs of the theorem from group theory to the effect that  $x^3=e$  implies ((x,y),y)=e. The resolution proof is 136 steps long while the paramodulation proof is 47 steps long. These proofs appear in the appendix.
- <sup>6</sup> Without this restriction one could infer from a=b and Qxa v Px the clause Qab v Pa(a proper instance of the paramodulant Qxb v Px), resulting in a loss of generality.
- <sup>7</sup> Since every non-trivial immediate modulant (see Wos <u>et.al</u>, (1967b)) of a clause is a paramodulant, any clause obtained by demodulation can be obtained by repeated paramodulation.

From a superficial point of view, paramodulation might be described as "a substitution rule for equality". Indeed, the motivation given above for studying the rule has dwelt principally on that aspect of paramodulation. But to consider it as only substitution of equals for equals would be to make a mistake analogous to characterizing resolution as merely a syllogistic inference akin to that employed by Davis and Putnam (1960). The property of maximum generality provided by paramodulation must not be overlooked if the process is to be fully understood. Consider the following example:

From  $f(xg(x)) = e \lor Qx$  and  $Pyf(g(y)z)z \lor Wz$  one can infer  $Pyeg(g(y)) \lor Qg(y) \lor Wg(g(y))$  by paramodulating with  $f(xg(x)) as \alpha'$  and  $f(g(y)z) as \delta_{\circ}$ 

Example 1:	Example 2:	Example 3:	Example 4:
l. a = b	l. a = b	1. $a = b$	1. $a = b$
2. Qa	2. Qx	2. $Qx \vee Px$	2. $Qx \vee Px$
3 Qb	3. ∴ Qb	3. $\therefore Qb \vee Pa$	3. $\therefore Qa \vee Pb$

Example 5:	Example 6:	Example 7:
l. $x = h(x)$	$l \cdot a = b$	1. $f(xg(x)) = e$
2. $Qg(y)$	2. Qf(g(h(j(a))))	2. $Pyf(g(y)z)z$
3. : $Qh(g(y))$	3. :. $Qf(g(h(j(b))))$	3. : Pyeg(g(y))

Example 8: If  $x^2 = e$  for all x in a group, the group is commutative.

1.	f(ex) = x	
2.	f(xe) = x	
3.	f(xf(yz)) = f(f(xy)z)	)
4.	f(xx) = e	
5.	f(ab) = c	
6.	c ≠ f(ba)	
7.	f(xe) = f(f(xy)y)	4 into 3 with $\delta$ : f(yz)
8.	x = f(f(xy)y)	2 into 7 on f(xe)
9.	a = f(cb)	5 into 8 on f(xy)
10.	f(yf(yz)) = f(ez)	4 into 3 on f(xy)
11.	f(yf(yz)) = z	l into 10 on f(ez)
12.	f(ca) = b	9 into ll on f(yz)
13.	c = f(ba)	12 into 8 on f(xy)
14.		13 resolved with 6 $ m '$

Figure 2.

#### Completeness of Paramodulation for Basic Group Theory

Al	Pxyf(xy)	closure
A2	Pexx	left identity
A3	Pg(x)xe	left inverse
A <sup>1</sup> 4	Pxyu V Pyzv V Puzw V Pxvw	associativity (case 1)
A5	$\overline{P}xyz  v  \overline{P}xyu  v  z = u$	uniqueness of product
Аб	z≠u ∨ Pxyz ∨ Pxyu	substitution (3rd position)
A7	z 🗲 u 🗸 Pxzy 🖌 Pxuy	substitution (2nd position)
8A	z≠u v Pzxy v Puxy	substitution (1st position)
A9	$\mathbf{x} = \mathbf{x}$	reflexivity
Alo	$x \neq y \lor y = x$	symmetry
All	$x \neq y \lor y \neq z \lor x = z$	transitivity
A12	$x \neq y \lor f(xz) = f(yz)$	f-substitution (1st position)
Al3	$x \neq y \lor f(zx) = f(zy)$	f-substitution (2nd position)
A14	$x \neq y \vee g(x) = g(y)$	g-substitution

Consider the following clauses from the first-order theory of groups:

Let us define a <u>basic</u> set S of clauses of group theory to be a set over the vocabulary of Al-Al4 and such that  $S \vdash \{Al, \dots, A5\}$ . We then have the following completeness result for the special case of basic sets.

Theorem: If S is a satisfiable fully paramodulated fully factored basic set of clauses of group theory, then S is R-satisfiable.

<u>Proof</u>: Let M be a maximal model <sup>8</sup> of S. Suppose that  $\alpha = \beta$  and  $P\gamma \delta \alpha$  are both in M. By the maximality of M, there must be clauses A and B in S having instances A':  $\alpha = \beta \vee K$  and B':  $P\gamma \delta \alpha \vee L$  with  $K \cap M = \phi = L \cap M$ . Then factors of A and B can be paramodulated on the arguments corresponding to  $\alpha$  to give a clause in S having  $P\gamma \delta \beta \vee K \vee L$  as an instance. Since M satisfies S,  $(P\gamma \delta \beta \vee K \vee L) \cap M \neq \phi$ 

<sup>&</sup>lt;sup>o</sup> The concept of maximal model is defined and the pertinent existence theorem proved in Wos and Robinson(1968a). For the present purpose a maximal model of S may be thought of as a model M such that for each positive literal x in M there is an instance C' of some C in S with C' ∩ M = { x } .

But  $(K \vee L) \cap M = \emptyset$ . Hence Py $\delta\beta \in M$ . Thus M satisfies A6. It can be shown <sup>9</sup> that Al-A6  $\vdash$  A7-Al4. Hence M satisfies A6-Al4 and is therefore an R-model of S.

This result is generalized to the case of what will be called functionally-reflexive systems in the next section.

## Completeness of Paramodulation for Functionally-Reflexive Systems

Paramodulation is intended to be utilized, along with resolution, for theorem proving in first-order theories with equality.

We first give an algorithm for generating a refutation (of a finite set of clauses) employing paramodulation and resolution if such a refutation exists.

<u>Full Search Algorithm</u> (FSA): Let  $S_0$  be the set of all factors of the given set S of clauses.<sup>11</sup> For odd i > 0 let  $S_i$  be formed from  $S_{i-1}$  by adding all clauses that can be obtained by paramodulating two clauses in  $S_{i-1}^{\circ}$ . For even i > 0 let  $S_i$  be formed from  $S_{i-1}$  by adding all factors of clauses that can be obtained by resolving two clauses in  $S_{i-1}^{\circ}$ . Since each deduction from S is contained in  $S_n$  for some n, each refutation of S must be contained in  $S_n$  for some n. Each  $S_j$  is finite. If  $S_j$  contains  $\Box$ , a refutation has been found, so stop. Otherwise form  $S_{i+1}^{\circ}$ .

<sup>9</sup> Robinson and Wos (1967c)

10

The earliest formulations of paramodulation were designed to operate without resolution and could be shown to subsume resolution as a special case. It is felt, however, that the processes can be better understood if the inference apparatus not involving equality is isolated from the apparatus for equality, even if this means that some of the completeness theorems cannot be stated in quite as pat a fashion.

11 Every elause is a factor of itself as in G. Robinson et.al. (1964b). For further definitions of factoring and resolution see Wos et.al.(1964a) and J. Robinson (1965). Now, to prove that paramodulation and resolution are complete for theorem-proving in first-order theories with equality, we would like to show that FSA is a semi-decision procedure for R-unsatisfiability. The difficult part is to show that, for R-unsatisfiable sets of clauses, there <u>exists</u> a refutation, i.e., that paramodulation plus resolution is R-refutation complete. It will suffice to show that an unsatisfiable set can be deduced from an R-unsatisfiable set, since (due to the refutation-completeness of resolution) FSA will generate a refutation if it ever generates an unsatisfiable set.

For functionally-reflexive systems S (theories such that  $S \models \alpha = \alpha$  for  $\alpha \equiv x_1$  and for  $\alpha \equiv f(x_1, \dots, x_n)$  for each n-adic function letter occurring in S -- there are h+l such unit clauses where h is the number of function letters in the vocabulary of S), we prove refutation completeness in (1968c)<sup>12</sup>. From that result we can obtain the following corollary: If S is a finite functionally-reflexive set of clauses, FSA is a semidecision procedure for R-unsatisfiability.

Even for theories that do not happen to be functionally reflexive, this result shows that adding the h+l functional-reflexivity units before applying FSA gives a general semi-decision procedure for R-unsatisfiability.

#### Further Completeness Results for Paramodulation

Since first-order theories are not usually functionally-reflexive when the only rules are resolution and paramodulation, and since adding the functional-reflexivity units to the theory may detract somewhat from proofsearch efficiency, we should like if possible to show that some weaker assumption than functional-reflexivity will suffice for completeness. It seems that at least  $S \vdash x = x$  will be needed. (Consider the case where  $S = \{a \neq a\}$ .

#### 12

A weaker version of this result was given in the earlier (1968b) paper.

-13-

S is R-unsatisfiable but cannot be refuted without some sort of help from reflexivity.) This is not surprising, since the standard texts on logic that use the substitution rule or schema approach to equality consistently supply a separate reflexivity axiom.<sup>13</sup>

But is simple reflexivity (x = x) enough? We think so, <sup>14</sup> although a proof of this is not yet available.

To see where the difficulty arises in generalizing the proof given in Wos and Robinson (1968c) beyond the functionally-reflexive case, we examine the relation between deductions and refutations based on a given set S and those based on proper instances of clauses from S.

<u>Capturing lemma</u>:  $\stackrel{\downarrow \downarrow}{\rightarrow}$  Let S by a fully paramodulated and fully resolved set of clauses such that S  $\vdash$  x = x, and let A' and B' be instances of clauses A and B in S and let C' be the result of paramodulating from a term  $\alpha$ ' in A' into an occurrence  $\delta_{\alpha}$  of a term in B'. Then

Strong subterm form: there is a clause C in S with C' as an instance. Restricted subterm form: If B has a term in the same position as that of  $\delta_0$  in B', then there is a clause C in S with C' as an instance. (Occurrences of terms in two literals are said to be in the <u>same position</u> if each is the i<sub>1</sub>-st argument of the i<sub>2</sub>-nd argument of ... of the i<sub>n</sub>-th argument of its literal.)

Argument form: If  $\delta$  is an argument of B' (as opposed to a proper subterm of an argument), then there is a clause C in S with C' as an instance.

<sup>13</sup> See, e.g., Church (1956) or Quine (1963).

<sup>14</sup> In the two years that paramodulation has been under study, no counterexample has been found to the R-refutation completeness of paramodulation and resolution for simply-reflexive systems.

<sup>15</sup> The analogue of this capturing lemma for resolution alone plays a basic role in proving the refutation-completeness of resolution (see J. Robinson(1965) and Slagle (1967)) and of set-of-support (Wos, <u>et-al</u>. (1965)).

-14-

When the strong subterm form of the capturing lemma holds and

S + x = x, every maximal (with respect to positive literals) model of S is an R-model, and since every satisfiable set S has a maximal model, it follows that either  $\mathbf{D} \in S$  or S is R-satisfiable. Thus the strong subterm form of the capturing lemma and simple reflexivity imply R-refutation completeness. The line of proof given for R-refutation-completeness in functionally-reflexive systems in (1968c) depends (at least indirectly) on the strong subterm form, which happens to hold in such systems.<sup>16</sup> The following example will suffice to show however that the strong subterm form is not universally true:

S: {x=x,a=b,b=a,a=a,b=b, Qxg(x), Qag(a), Qbg(b), Qag(b), Qbg(a)}

- A: a=b
- A': a=b
- B: Qxg(x)
- B': Qg(a)g(g(a))
- C': Qg(b)g(g(a))

S is fully-paramodulated and (vacuously) fully-resolved. A' and B' paramodulate on a into the first occurrence of a in B' to give C'. But C' is an instance of no clause in S. (The restricted subterm form of the lemma is not violated since B has no term in the same position as the first occurrence of a in B'. Neither is the argument form of the lemma, since a is not an argument of B'.) Functional-reflexivity of S, if present, would dispose of the difficulty since if g(x)=g(x) were in S, so would g(a)=g(b) be in S if it were fully paramodulated; and hence the result Qg(b)g(g(a)) of paramodulating g(a)=g(b)and Qxg(x) would be in S and serve as C.

Weakening the strong subterm capturing lemma in a different fashion leads to the

<sup>&</sup>lt;sup>10</sup> Alternatively, one can view the difficulty as resulting from the fact that it is not always possible to satisfy the hypotheses of the restricted subterm form.

<u>Refutation capturing lemma</u>: If there exists a refutation of a set of instances of clauses in a set S by means of paramodulation and resolution, then there exists a refutation of S itself by means of paramodulation and resolution.

For functionally-reflexive S, this lemma may be proved by noting that the refutability of a set of instances of S and R-soundness of paramodulation and resolution yield the R-unsatisfiability of S; so that the refutationcompleteness of paramodulation and resolution for functionally-reflexive systems establishes the refutability of S itself.

Given the refutation capturing lemma one could prove the following: <u>General refutation-completeness</u>: If S is a fully-paramodulated and fully resolved R-unsatisfiable set and if  $S \rightarrow x=x$ , then  $\mathbf{c} \in S$ .

<u>Corollary</u>: FSA is a semi-decision procedure for R-unsatisfiability for finite sets S of clauses such that  $S \rightarrow x=x$ .

Conversely, given general refutation-completeness, one can prove the refutation capturing lemma (at least for systems S such that  $S \mapsto x=x$ ). In view of this equivalence, proof of the refutation capturing lemma can be considered the most pressing unsolved problem in the theory of paramodulation. Alternatively, one might seek a proof of general refutation completeness based on the restricted subterm form of the capturing lemma, which holds even when the assumption of functional reflexivity is suppressed.

- 16 -

Church, A. (1956), Introduction to mathematical logic I. Princeton.

Darlington, J. L. (1968), Automatic theorem proving with equality substitutions and mathematical induction. <u>Machine Intelligence</u> 3 (ed. D. Michie). Edinburgh: Oliver and Boyd.

- Davis, M., and Putnam, H., (1960), A computing procedure for quantification theory, J. Assn. Comput. Mach. 7, pp. 201-215.
- Quine, W. V. O. (1963), <u>Set theory and its logic</u>. Harvard University Press, Cambridge, Massachusetts.
- Robinson, J. A. (1965), A machine-oriented logic based on the resolution principle, J. Assn. Comput. Mach. 12, pp. 23-41.
- Slagle, J. R. (1967), Automatic theorem proving with renamable and semantic resolution, J. Assn. Comput. Mach. 14, pp. 687-697.
- Wos, L., Carson, D., and Robinson, G. (1964a), The unit-preference strategy in theorem proving. AFIPS Conference Proceedings <u>26</u>, Spartan Books, Washington D.C., pp. 615-621.
- Robinson, G. A., Wos, L. T., and Carson, D. (1964b), Some theorem-proving strategies and their implementation. AMD Tech. Memo No. 72, Argonne National Laboratory.
- Wos, L., Robinson, G. A., and Carson, D. F. (1965), Efficiency and completeness of the set of support strategy in theorem proving, <u>J. Assn. Comput. Mach.</u> 12, pp. 536-541.
- Robinson, G. A., Wos, L., and Shalla, L. (1967a), Two inference rules for first-order predicate calculus with equality. AMD Tech. Memo No. 142, Argonne National Laboratory.
- Wos, L., Robinson, G. A., Carson, D. F., and Shalla, L. (1967b), The concept of demodulation in theorem proving. <u>J. Assn. Comput. Mach.</u> <u>14</u>, pp. 698-709.
- Robinson, G., and Wos, L. (1967c), Dependence of equality axioms in elementary group theory - Comp.Group Tech.Memo No.53, Stanford Linear Accelerator Center
- Wos, L., and Robinson, G. (1968a), The maximal model theorem, Spring 1968 meeting of Assn. for Symbolic Logic. Abstract to appear in J. Symb. Logic.

- Robinson, G., and Wos, L. (1968b), Completeness of paramodulation, Spring 1968 meeting of Assn. for Sympolic Logic. Abstract to appear in J. Symb. Logic.
- Wos, L., and Robinson, G. (1968c), Maximal models and refutation completeness (unpublished).

#### APPENDIX

#### PARAMODULATION VERSUS RESOLUTION

Problem: 
$$x^3 = e$$
 implies  $((x,y),y) = e$ .

Reference: Group Theory by Marshall Hall, page 322, 18.2.8.

## Refutation by Paramodulation

ŧ

- 1. f(ex) = x
- 2. f(xe) = x
- 3. f(g(x)x) = e
- 4. f(xg(x)) = e
- 5. f(xf(yz)) = f(f(xy)z)
- 6. x = x
- 7. f(f(xx)x) = e
- 8. h(xy) = f(f(f(xy)g(x))g(y))
- 9.  $h(h(ab)b) \neq e$
- 10. f(xe) = f(f(xy)g(y)), f(xg(x)) of 4 into f(yz) of 5
- 11. x = f(f(xy)g(y)), f(xe) of 2 into f(xe) of 10
- 12. x = f(eg(g(x))), f(xg(x)) of 4 into f(xy) of 11
- 13. x = g(g(x)), f(ex) of 1 into f(eg(g(x))) of 12
- 14. f(f(xx)f(xz)) = f(ez), f(f(xx)x) of 7 into f(xy) of 5
- 15. f(f(xx)f(xz)) = z, f(ex) of 1 into f(ez) of 14
- 16. f(f(xx)e) = g(x), f(xe(x)) of 4 into f(xz) of 15
- 17. f(xx) = g(x), f(xe) of 2 into f(f(xx)e) of 16
- 18. f(f(xy)f(g(y)z)) = f(xz), f(f(xy)g(y)) of ll into f(xy) of 5
- 19. f(f(xy)f(g(y)g(x))) = e, f(xg(x)) of 4 into f(xz) of 18
- 20. f(we) = f(f(wf(xy))f(g(y)g(x))), f(f(xy)f(g(y)g(x))) of 19 into f(yz) of 5
- 21. w = f(f(wf(xy)f(g(y)g(x))), f(xe) of 2 into f(we) of 20

g(f(xy)) = f(ef(g(y)g(x))), f(g(x)x) of 3 into f(wf(xy)) of 2122. g(f(xy)) = f(g(y)g(x)), f(ex) of l into f(ef(g(y)g(x))) of 22 23. g(h(xy)) = f(g(g(y))g(f(f(xy)g(x)))), f(f(f(xy)g(x))g(y)) of 8 into f(xy) of 2324. g(h(xy)) = f(yg(f(f(xy)g(x)))), g(g(x)) of 13 into g(g(y)) of 2425. g(h(xy)) = f(yf(g(g(x))g(f(xy)))), g(f(xy)) of 23 into g(f(f(xy)g(x))) of 2526. g(h(xy)) = f(yf(xg(f(xy)))), g(g(x)) of 13 into g(g(x)) of 2627. g(h(xy)) = f(yf(xf(g(y)g(x)))), g(f(xy)) of 23 into g(f(xy)) of 2728.  $f(f(f(h(ab)b)g(h(ab)))g(b)) \neq e, h(xy) \text{ of } 8 \text{ into } h(h(ab)b) \text{ of } 9$ 29.  $f(f(f(f(f(f(ab)g(a))g(b))b)g(h(ab)))g(b)) \neq e, h(xy) \text{ of } 8 \text{ into } h(ab) \text{ of } 29$ 30.  $f(f(f(f(f(ab)g(a))f(g(b)b))g(h(ab)))g(b)) \neq e, f(f(xy)z)$  of 5 into 31. f(f(f(ab)g(a))g(b))b) of 30  $f(f(f(f(ab)g(a))e)g(h(ab)))g(b)) \neq e, f(g(x)x) \text{ of } 3 \text{ into } f(g(b)b) \text{ of } 31$ 32.  $f(f(f(ab)g(a))g(h(ab)))g(b)) \neq e, f(xe) \text{ of } 2 \text{ into } f(f(f(ab)g(a))e) \text{ of } 32$ 33.  $f(f(f(ab)g(a))f(bf(af(g(b)g(a))))g(b)) \neq e, g(h(xy))$  of 28 into g(h(ab)) of 33 34.  $f(f(f(ab)f(aa))f(bf(af(g(b)g(a))))g(b)) \neq e, g(x) \text{ of } 17 \text{ into } g(a) \text{ of } 34$ 35.  $f(f(f(f(f(ab)f(aa))b)f(af(g(b)g(a))))g(b)) \neq e, f(xf(yz)) \text{ of 5 into}$ 36. f(f(f(ab)f(aa))f(bf(af(g(b)g(a))))) of 35  $f(f(f(f(f(f(ab)f(aa))b)a)f(g(b)g(a)))g(b)) \neq e, f(xf(yz)) \text{ of 5 into}$ 37. f(f(f(ab)f(aa))b)f(af(g(b)g(a)))) of 36  $f(f(f(f(f(f(f(ab)a)a)b)a)f(g(b)g(a)))g(b)) \neq e, f(xf(yz))$  of 5 into 38. f(f(ab)f(aa)) of 37  $f(f(f(f(f(ab)a)f(ab))a)f(g(b)g(a)))g(b)) \neq e, f(f(xy))z)$  of 5 into 39. f(f(f(ab)a)a)b) of 38  $f(f(f(f(f(ab)a)f(f(ab)a))f(g(b)g(a)))g(b)) \neq e, f(f(xy)z)$  of 5 into 40. f(f(f(ab)a)f(ab))a) of 39  $f(f(f(ab)a)f(f(ab)a))f(f(g(b)g(a))g(b))) \neq e, f(f(xy)z)$  of 5 into 41. f(f(f(f(ab)a)f(f(ab)a))f(g(b)g(a)))g(b)) of 40

-20-

42.  $f(f(f(ab)a)f(f(ab)a))f(g(f(ab)))g(b)) \neq e, f(g(y)g(x))$  of 23 into f(g(b)g(a)) of 41

43.  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)f(ab))g(b))) \neq e, g(x) \text{ of } 17 \text{ into } g(f(ab)) \text{ of } 42$ 

44.  $f(f(f(ab)a)f(f(ab)a))f(f(f(f(ab)a)b)g(b))) \neq e, f(xf(yz)) \text{ of 5 into}$ 

f(f(ab)f(ab)) of 43

- 45.  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)a)f(bg(b)))) \neq e, f(f(xy)z)$  of 5 into f(f(f(f(ab)a)b)g(b)) of 44
- 46.  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)a)e)) \neq e, f(xg(x))$  of 4 into f(bg(b)) of 45

ł

47.  $f(f(f(ab)a)f(f(ab)a))f(f(ab)a)) \neq e, f(xe) \text{ of } 2 \text{ into}$ 

f(f(f(ab)a)f(f(ab)a))f(f(f(ab)a)e)) of 46

7 contradicts 47.

## PARAMODULATION VERSUS RESOLUTION

Problem:  $x^3 = e$  implies ((x,y),y) = e where  $(x,y) = xyx^{-1}y^{-1}$ 

## Refutation by Resolution

- 1. f(ex) = x
- 2. f(xe) = x
- 3. f(e(x)x) = e
- 4. f(xg(x)) = e
- 5. f(xf(yz)) = f(f(xy)z)
- 6. x = x
- 7. x≠y y=x
- 8.  $x \neq y \quad y \neq z \quad x = z$

9.  $u \neq w f(ux) = f(wx)$ 

- 10.  $u \neq w f(xu) = f(xw)$
- 11.  $u \neq w g(u) = g(w)$
- 12. f(f(xx)x) = e
- 13. h(xy) = f(f(f(xy)g(x))g(y))
- 14. h(h(ab)b) ≠ e
- 15.  $x \neq f(ew) = w$ , 1 and 8<sub>2</sub>
- 16. f(f(xg(x))w) = f(ew), 4 and 9
- 17.  $f(f(xy)z) \neq w$  f(xf(yz)) = w, 5 and  $8_1$
- 18. f(xf(g(x)z)) = f(ez), 16 and 17
- 19. f(xf(g(x)z)) = z, 18 and 15<sub>1</sub>
- 20. f(uf(yg(y))) = f(ue), 4 and 10<sub>1</sub>

f(ue) = f(uf(yg(y))), 20 and 7<sub>1</sub>. 21.  $f(uf(yg(y))) \neq z$  f(ue) = z, 21 and 8<sub>1</sub> 22.  $f(xe) = g(g(x)), 19 \text{ and } 22_1$ 23. x = f(xe), 2 and 7<sub>1</sub> 24.  $f(xe) \neq z \quad x = z, 24 \text{ and } 8_1$ 25.  $x = g(g(x)), 23 \text{ and } 25_1$ 26. f(f(f(uu)u)y) = f(ey), 12 and 9 27. 28. f(f(f(uu)u)y) = y, 27 and 15<sub>1</sub> f(f(xx)f(xy)) = y, 28 and 17<sub>1</sub> 29.  $f(f(xx)e) = g(x), 29 \text{ and } 22_1$ 30. f(xx) = g(x), 30 and 25<sub>1</sub> 31. 32.  $f(xe) = f(f(xy)g(y)), 5 \text{ and } 22_1$  $x = f(f(xy)g(y)), 32 \text{ and } 25_1$ 33. f(xz) = f(f(f(xy)g(y))z), 33 and 9<sub>1</sub> 34. f(f(f(xy)g(y))z) = f(xz), 34 and 7<sub>1</sub> 35. 36. f(f(xy)f(g(y)z)) = f(xz), 35 and 17  $x \neq f(ug(u))$  x = e, 4 and 8<sub>2</sub> 37.  $f(f(xy)f(g(y)g(x))) = e, 36 \text{ and } 37_1$ 38. 39. e = f(f(xy)f(g(y)g(x))), 38 and 7<sub>1</sub>  $u \neq f(xf(yz))$  u = f(f(xy)z), 5 and 8<sub>2</sub> 41. 43.  $u = f(f(uf(xy))f(g(y)g(x))), 42 \text{ and } 25_1$  $f(f(g(x)x)u) = f(eu), 3 \text{ and } 9_1$ 44. 45.  $z \neq f(f(g(x)x)u)$  z = f(eu), 44 and 8<sub>2</sub>

-23-

40. 
$$f(we) = f(wf(f(xy)f(g(y)g(x))))$$
, 39 and 10<sub>1</sub>

42. 
$$f(ue) = f(f(uf(xy))f(g(y)g(x)))$$
, 40 and 41<sub>1</sub>

Ł

-24-

 $f(f(f(uw)z)g(h(xy))) = f(f(f(uw)z)f(yf(xf(g(y)g(x))))), 71 \text{ and } 70_1$ 72.  $f(f(f(xy)z)g(h(xy)))u) = f(f(f(f(xy)z)f(yf(xf(g(y)g(x)))))u), 72 \text{ and } 9_1$ 73. 74. f(h(xy)z) = f(f(f(f(xy)g(x))g(y))z), 13 and 9<sub>1</sub>  $u \neq f(f(xy)z)$  u = f(xf(yz)), 69 and 8<sub>2</sub> 75.  $f(h(xy)z) = f(f(f(xy)g(x))f(g(y)z)), 74 \text{ and } 75_1$ 76. 77. f(uf(g(x)x)) = f(ue), 3 and 10, 78.  $z \neq f(uf(g(x)x))$  z = f(ue), 77 and 8<sub>2</sub> 79. f(h(xy)y) = f(f(f(xy)g(x))e), 76 and 78, 80.  $u \neq f(xe)$  u = x, 2 and 8,  $f(h(xy)y) = f(f(xy)g(x)), 79 \text{ and } 80_1$ 81. f(f(h(xy)y)z) = f(f(f(xy)g(x))z), 81 and 9<sub>1</sub> 82. f(f(f(h(xy)y)z)w) = f(f(f(f(xy)g(x))z)w), 82 and 9<sub>1</sub> 83. 84.  $h(h(ab)b) \neq y \quad y \neq e, 14 \text{ and } 8_2$  $f(f(f(h(ab)b)g(h(ab)))g(b)) \neq e, 13 \text{ and } 84_1$ 85.  $f(f(f(h(ab)b)g(h(ab)))g(b)) \neq y \quad y \neq e, 85 \text{ and } 8_3$ 86.  $f(f(f(ab)g(a))g(h(ab)))g(b)) \neq e, 83 \text{ and } 86_1$ 87.  $f(f(f(ab)b(a))b(h(ab)))g(b)) \neq y \quad y \neq e, 87 \text{ and } 8_3$ 88.  $f(f(f(ab)g(a))f(bf(af(g(b)g(a))))g(b)) \neq e, 73 \text{ and } 88_1$ 89. 9Q. g(x) = f(xx), 31 and 7<sub>1</sub>  $f(wg(x)) = f(wf(xx)), 90 \text{ and } 10_1$ 91. 92.  $f(uf(wg(x))) = f(uf(wf(xx))), 91 \text{ and } 10_1$ 93.  $f(uf(wg(x))) = f(f(uw)f(xx)), 92 \text{ and } 41_1$ 94. f(f(uw)g(x)) = f(f(uw)f(xx)), 93 and 70,f(f(f(uw)g(x))y) = f(f(f(uw)f(xx))y), 94 and 9<sub>1</sub> 95. f(f(f(f(uw)g(x))y)z) = f(f(f(f(uw)f(xx))y)z), 95 and 9<sub>1</sub> 96.  $f(f(f(ab)g(a))f(bf(af(g(b)g(a))))g(b)) \neq y \quad y \neq e, 89 \text{ and } 8_3$ 97.

-25-

 $f(f(f(ab)f(aa))f(bf(af(g(b)g(a))))g(b)) \neq e, 96 \text{ and } 97_1$ 98.  $f(f(f(f(ab)f(aa))f(bf(af(g(b)g(a))))) \neq y \quad y \neq e, 98 \text{ and } 8_3$ 99. 100.  $f(f(xf(yz))u) = f(f(f(xy)z)u), 5 \text{ and } 9_1$ 101.  $f(f(f(f(f(ab)f(aa))b)f(af(g(b)g(a))))g(b)) \neq e, 100 \text{ and } 99_1$  $f(f(f(f(ab)f(aa))b)f(af(g(b)g(a))))g(b)) \neq y y \neq e, 101 and 8,$ 102. 103.  $f(f(f(f(f(f(ab)f(aa))b)a)f(g(b)g(a)))g(b)) \neq e, 100 \text{ and } 102,$ f(f(f(xf(yz))u)v) = f(f(f(xy)z)u)v), 100 and 9, 104. 105.  $f(f(f(f(xf(yz))u)v)w) = f(f(f(f(xy)z)u)v)w), 104 \text{ and } 9_1$  $f(f(f(f(f(xf(yz))u)v)w)t) = f(f(f(f(f(xy)z)u)v)w)t), 105 \text{ and } 9_1$ 106.  $f(f(f(f(f(f(ab)f(aa))b)a)f(g(b)g(a)))g(b)) \neq y y \neq e, 103 and 8)$ 107.  $f(f(f(f(f(f(ab)a)a)b)a)f(g(b)g(a)))g(b)) \neq e, 106 \text{ and } 107_1$ 108.  $f(f(f(f(f(xy)z)u)v)w) = f(f(f(f(xf(yz))u)v)w), 105 \text{ and } 7_1$ 109.  $f(f(f(f(f(f(f(ab)a)a)b)a)f(g(b)g(a)))g(b)) \neq y y \neq e, 108 and 8,$ 110.  $f(f(f(f(f(f(ab)a)f(ab))a)f(g(b)g(a)))g(b)) \neq e, 109 \text{ and } 110_1$ 111. 112.  $f(f(f(f(xy)z)u)v) = f(f(f(xf(yz))u)v), 104 \text{ and } 7_1$  $f(f(f(f(f(f(ab)a)f(ab))a)f(g(b)g(a)))g(b)) \neq y \quad y \neq e, 111 \text{ and } 8_3$ 113. 114.  $f(f(f(f(f(ab)a)f(f(ab)a))f(g(b)g(a)))g(b)) \neq e, 112 \text{ and } 113_1$  $f(f(f(ab)a)f(f(ab)a))f(f(g(b)g(a))g(b)) \neq e, 114 \text{ and } 70_2$ 115. 116.  $f(f(f(ab)a)f(f(ab)a))f(f(g(b)g(a))g(b)) \neq y \quad y \neq e, 115 \text{ and } e_{y}$ f(g(y)g(x))g(f(xy)), 47 and 7, 117. 118. f(f(g(y)g(x))z) = f(g(f(xy))z), 117 and 9<sub>1</sub>  $f(uf(f(g(y)g(x))z)) = f(uf(g(f(xy))z)), 118 \text{ and } 10_1$ 119.  $f(f(f(ab)a)f(f(ab)a))f(g(f(ab))g(b))) \neq e, 119 \text{ and } 116_1$ 120. 121. f(g(x)z) = f(f(xx)z), 90 and 9, 122.  $f(uf(g(x)z)) = f(uf(f(xx)z)), 121 \text{ and } 10_1$ 

123.  $f(f(f(ab)a)f(f(ab)a))f(g(f(ab))g(b))) \neq y \neq e$ , 120 and  $8_3$ 

 $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)f(ab))g(b))) \neq e, 122 \text{ and } 123_1$ 124. f(wf(f(xf(yz))u)) = f(wf(f(f(xy)z)u)), 100 and 10,125.  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)f(ab))g(b))) \neq y \quad y \neq e, 124 \text{ and } 8_{3}$ 126.  $f(f(f(ab)a)f(f(ab)a))f(f(f(f(ab)a)b)g(b))) \neq e, 125 \text{ and } 126_1$ 127. 128.  $f(uf(f(xy)z)) = f(uf(xf(yz))), 69 \text{ and } 10_1$ 129.  $f(f(f(ab)a)f(f(ab)a))f(f(f(f(ab)a)b)g(b))) \neq y \quad y \neq e, 127 \text{ and } 8_3$  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)a)f(bg(b)))) \neq e, 128 \text{ and } 129_1$ 130.  $f(zf(uf(yg(y)))) = f(zf(ue)), 20 \text{ and } 10_1$ 131.  $f(f(f(ab)a)f(f(ab)a))f(f(f(ab)a)f(bg(b)))) \neq y \quad y \neq e, 130 \text{ and } 8_3$ 132.  $f(f(f(ab)a)f(f(ab)a))f(f(f(a)b)a)e)) \neq e, 131 \text{ and } 132_1$ 133. f(uf(xe)) = f(ux), 2 and  $10_1$ 134.  $f(f(f(f(ab)a)f(f(ab)a))f(f(f(a)b)a)e)) \neq y \quad y \neq e, 133 \text{ and } 8_3$ 135.  $f(f(f(ab)a)f(f(ab)a))f(f(ab)a)) \neq e, 134 \text{ and } 135_1$ ,136.

12 contradicts 136

-27-