A New Hyperparamodulatlon Strategy for the Equality Relation*

Younghwan Lim and Lawrence J. Henschen

Department of EE/CS
Northwestern University, Evanston

Abstract

Equality is an Important relation and many theorems can be easily symbolized through it's use. A proposed Inference rule called HL-resolution Is intended to have the benefits of hyper steps while controlling the application of paramodulation. It generates a resolvent by building a paramodulation and demodulation link between two terms using a pre pro ceased plan as a guide. The rule Is complete for E-unsatisfiable Horn sets. The linking process makes use of an equality graph which is constructed once at the beginning of the run. Once a pair of candidate terms for HL-resolutlon is chosen in the search, potential linkages can be found and tested for compatibility efficiently by looking at the paths in the graph. The method was implemented on an existing theorem-proving system. A number of experiments were conducted on problems in abstract algebra and a comparison with set-of-support paramodulation was made.

1. Introduction

Equality is an important relation and many theorems can be easily symbolized through it's use. Important research with respect to the equality relation has been carried out in several directions by many authors. Darlington [2] used a second-order equality substitution axiom, and Robinson and Wos [13,11] proposed demodulation and paramodulation to handle equality. Along this line, Wos, Overbeek and Henschen[14] proposed a refinement of paramodulation called HYPERPARAMODULATION and McCune[8] proposed Horn semantic paramodulation. Along another line, there is the E-resolution system by Morris[9] for the treatment of equality. Later, DIgricoli [31 proposed the RUE-NRF rule of inference following the lines of research proposed by Morris in E-resolution and by Harrison and Rubin[4j In generalized resolution. The Connection Graph Procedure introduced by Kowalski[5] represents all possible resolution steps by links between the complementary unifiable literals. In [12], the ideas of the Connection Graph Proof Procedure are extended to handle paramodulation. On the other hand, Knuth and Bendix created a procedure for deriving consequences from equality units using a reduction.

We remind the reader of the following problems that occur in handling equality. First, equality of two terms with respect to a given set of equations is in general undeoidable. Second, few effective control mechanisms for the search and application of equality derivation steps have been developed. Third, equality proof procedures seem not to make use of any high level planning. Fourth, heuristic information does not seem to be easily incorporated into existing equality proof procedures.

A proposed inference rule called HL-resolution is intended to have the benefits of hyper steps while controlling the uses of paramodulation. It generates a resolvent by building a paramodulation and demodulation link between two given terms using a preprocessed plan as a guide. This linking process makes use of an equality graph which is constructed once at the beginning of the run. Once a pair of candidate terms for HL-resolution is chosen in the search, potential linkages can be found and tested for compatibility efficiently by looking at paths in the graph. Furthermore, using the properties of links, pairs of end terms for inner level linking can be found easily. The method was Implemented on an existing theorem-proving system and a number of experiments were conducted on problems in abstract algebra.

2. **Definitions**

In this section, we give the basic definitions for HL-resolution. Any definition which is used and not defined will follow the standard terminology in (equality) theorem proving.

Definition Let P be a set of paramodulators and D be a set of demodulators. A clause C' is called a **k-para/demod link** (k-pd link) of a clause C relative to P ∪ D if and only if there exists a sequence of clauses A_0, A_1, ... , A_k such that

1) $A_0 = C$ and $A_k = C'$
2) A_i, for $0<i\leq k$, is a paramodulant or demodulant of A_{i-1} and a clause in P ∪ D under the restriction that the into-terms of A_{i-1} and A_i are from the same literal.
3) For each paramodulation/demodulation, the into term of A_i is <u>not</u> properly contained in the replacement of the into term of A_{i-1}.
4) If there is j such that $0<j\leq k$ and A_j is a demodulant of A_{j-1}, then each A_i, for all $j\leq i\leq k$, is a demodulant of A_{i-1}.

The k is called the **length** of the link. The definition implies that a clause is a 0-pd link of

• Supported in part by the National Science Foundation under Grant Number MCS-8306637

itself. The sets of equalities P and D need not be disjoint. Their choice, in practice, is heuristic but at present, for the theory, P needs to be all of the equalities in S. Condition 3), which will be called the into-term-containment restriction, will be of specific importance in plan formation to be discussed in Section 4.

Definition. Let C' be a k-para/demod link of a clause C relative to P U D and $A_0, A_1, \ldots, A_k$ as above. If, for $0<i<=k$, A_i is a paramodulant of A_{i-1} then let E_i be the unifier of the paramodulator and the into term in A_{i-1}. Otherwise, i.e. A_i is a demodulant of A_{i-1}, let E_i be the empty substitution. Then k-E = $(\ldots ((E_1{*}E_2){*}E_3) \ldots E_k)$ is called a **k-linked unifier** of C' from C, where * is the composition operator.

Definition. A **partial unifier** of terms/literals t_1 and t_2 having the same function/predicate symbol is a substitution which unifies t_1 and t_2 from left to right, skipping over any pair of ununifiable arguments.

Definition. A **function substitution link** of a clause of the form $f(t_1,\ldots,t_n)<>f(s_1,\ldots s_n)$ v A is a clause D v A*E, where A is a set of literals, E is a given substitution to be applied to $f(t_1,\ldots,t_n)$ and $f(s_1,\ldots,s_n)$, and D is a disjunction of inequalities formed by the pairs of arguments not unified in $f(t_1,\ldots,t_n){*}E$ and $f(s_1,\ldots,s_n){*}E$.

Definition. A **predicate substitution link** of a pair of clauses of the form $P(t_1,\ldots t_n)$ v A and $-P(s_1,\ldots,s_n)$ v B, where A and B are sets of literals, is D v A*E v B*E, where E is a substitution to be applied to $P(t_1,\ldots t_n)$ and $P(s_1,\ldots s_n)$, and D is a disjunction of inequalities formed by the pairs of arguments not unified in $P(t_1,\ldots,t_n){*}E$ and $P(s_1,\ldots s_n){*}E$.

The substitution used in a predicate or a function substitution link may be the empty substitution, a partial unifier or a full most general unifier. In the last case, of course, D will be empty. In fact, in the experiments discussed in Section 5 we always used partial unifiers, and our program obtained proofs for all the problems tried. The role of the substitution link is to simplify a clause by stripping off the outer function/predicate symbol of one of its literals. The soundness of rules of inference which generate function/predicate substitution links can be derived directly by the use of the function and predicate substitution axioms. In our equality-reasoning system, the above two rules of inference will replace the use of function and predicate substitution axioms. This, in effect, restricts the use of those axioms by not allowing the generation of clauses corresponding to arbitrary resolutions from substitution axioms. However, unlike previous attempts in this direction (e.g., [3,4,9]), we will propose a system in which the rules themselves will be used in a very restricted way, further cutting down on the number of clauses they are allowed to generate.

We now define a new inference rule called HL(Henschen-Lim)- resolution.

Definition. Let S be a set of clauses, P be a set of paramodulators and D be a set of demodulators. Let N be the transitivity clause {x<>y y<>z x=z} in the equality axioms, A_1 be a positive unit clause in S, and A_2 v B be a clause in S, where A_2 is a negative literal and B is the set of the remaining literals. Let the variables in these clauses all be separated. Suppose that the set of clauses {A_1, A_2 v B, N} satisfies one of the following conditions:

1) (forward) There exists a most general unifier(MGU) E_1 of A_1 and the literal x<>y and a MGU E_2 of A_2 and L_1', where L_1' is a k-pd link of $\{x=z\}{*}E_1$ relative to P U D with a k-linked unifier k-E.
2) (backward) There exists a MGU E_1 of A_2 and the literal x=z and a MGU E_2 of A_1 and L_2', where L_2' is a k-pd link of $\{x<>y\}{*}E_1$ relative to P U D with a k-linked unifier k-E.

Then the clause (y<>z v B)*(k-E)*E_2*E_1 is called an **HL-resolvent** of the set. N is called **the nucleus clause**, and A_1 and A_2 v B are **the satellite clauses**.

The terms of an (in)equality in A_1, A_2, P or D are allowed to be flipped if necessary to match. In particular, paramodulation proceeds from either side of any equality in P. Further this definition can be extended in such a way that A_1 is an arbitrary clause in S. We can also allow A_2 to be positive and link to y<>z in N, or A_1 to link to y<>z, etc. For the rest of this paper, however, we use the definition as given above.

Definition. Given a set S of clauses, a deduction from S is called an **HL-deduction** if and only if each clause in the deduction is a clause in S, an HL-resolvent, a regular resolvent or a function or predicate substitution link of an HL-resolvent.

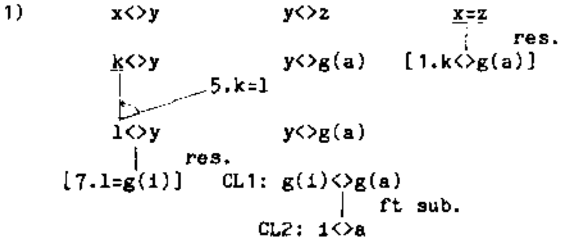
Definition. An HL-deduction of the empty clause from S is called an **HL-refutation** of S.

Example 1

Consider the E-unsatisfiable set of clauses:

1. k<>g(a)	2. f(h(b),c)=a
3. d=h(b)	4. e=c
5. k=l	6. i=f(d,e)
7. l=g(i)	

An HL-refutation looks like:



```
2)       x<>y      y<>z       x=z
           |      res.
   [6.f(d,e)=i]  i<>z   f(d,e)=z
                          |     2.d=h(b)
                          |   /
                 i<>z   f(h(b),e)=z
                               |   4.e=c
                               |  /
                 i<>z   f(h(b),c)=z
                                   2.f(h(b),c)=a
                               |  /
                 i<>z         a=z
                               |  res.
                 i<>i   [CL2: a<>i]
                  |   unit conflict
                 proof
```

Example 2
Consider the theorem if x*x=e then f(y,z)=f(z,y) in the group theory.

```
    1) f(e,x)=x                    2) f(x,e)=x
    3) f(g(x),x)=e                 4) f(x,g(x))=e
    5) f(f(x,y),z)=f(x,f(y,z))     6) f(x,x)=e
    7) f(a,b)<>f(b,a)
```

An HL-refutation looks like;

```
1)      u<>v          v<>w              u=w
         | res.
       [5.]  f(x,f(y,z))<>w  f(f(x,y),z)=w
                                  |     6.f(u,u)=e
                                  |   /
             f(x,f(x,z))<>w     f(e,z)=w
                                        1.f(e,u)=u
                                  |   /
             f(x,f(x,z))<>w       z=w
                                  |  res.
     f(x,f(x,f(b,a)))<>f(a,b)   [7'.]
                    |        ft sub. link
      CL1:f(a,f(b,a))<>b

2)      u<>v          v<>w              u=w
         | res.
      [5'.]  f(f(x,y),z)<>w   f(x,f(y,z))=w
                                    |    6.f(u,u)=e
                                    |  /
             f(f(x,y),y)<>w     f(x,e)=w
                                   |     2.f(u,e)=u
                                   |  /
             f(f(x,y),y)<>w       x=w
                                   |  res.
     f(f(b,y),y)<>f(a,f(b,a))    [CL1']
                 |        ft sub. link
   CL2:f(b,f(b,a))<>a

3)      u<>v          v<>w              u=w
         | res.
      [5.]   f(x,f(y,z))<>w   f(f(x,y),z)=w
                                  |      6.f(u,u)=e
                                  |   /
             f(x,f(x,z))<>w     f(e,z)=w
                                  |      1.f(e,u)=u
                                  |   /
             f(x,f(x,z))<>w      z=w
                                  |  res.
     f(x,f(x,a))<>f(b,f(b,a))   [CL2']
                 |   unit conflict
               proof
```

Comments:

1). We do not allow function/predicate substitution to be used during the formation of the HL-resolvent, but only to the end result. We believe this corresponds to human-like approaches to function/predicate stripping. In any case, it severely limits the way in which these rules are allowed to generate new clauses. Similarly, transitivity is used only as an HL-nucleus and only when two terms have been chosen for linkage. Symmetry is built in, so we don't need to include xOy y=x. But note that flipping equalities is allowed only if it will make some larger deduction sequence work. So we have very tight restrictions on the uses of equality axioms.

2). As illustrated in the above example, we plan to use HL-resolution with the set-of-support strategy, reasoning from the denial of the theorem. Simple paramodulation with set-of-support and without function reflexive axioms is known to be incomplete as the following example shows:

```
1. g(a,x,x)<>g(b,x,x)
2. c=d
3. g(a,f(c),f(d))=g(b,f(c),f(d))
```

with only clause 1 supported. Because the linkage in HL-resolution is allowed from either A_j or A_2, we can often get around this problem, and the reader can verify that there is a supported HL-resolvent

```
g(b,f(c),f(d))<>g(b,f(c),f(c))
```

using a 1-pd link from clause 3 which generates the function substitution link cOd. The idea is that we reason from a supported clause and another clause, and the direction of the linkage shouldn't matter.

3) Although the above is a cooked up counterexample, the situation is totally different when demodulation is used. Then most normal problems are not refutable by paramodulation using only the denial as support. For example, the first backward reasoning paramodulation step in the x*x=e problem is to generate f(e,f(b,a))Of(a,b), which immediately demodulates back to f(b,a)Of(a,b). Since HL-resolution is a hyper rule, demodulation is blooked on all the intermediate steps. A corallary of this observation is that if a k-pd link starts out with a number of equalities used backwards, somewhere in the linkage a non-demodulator equality must be used or else the target term must rearrange the built-up term. Otherwise the entire HL-resolvent will Just redemodulate. We believe this feature also matches that of human-style equality reasoning in which people often purposely make complicating substitutions into terms with the goal of being able to reassociate, distribute or some other such.

4). As with other hyper methods, the idea is to keep only the end result, which hopefully will be a more meaningful one. For equality we believe this may also lead to the possibility of some high level planning. For example, in the k<>g(a) problem above, there are several HL-resolvents starting from k<>g(a). However, only the given one leads to function stripping. Thus, one heuristic in choosing A_1 and A_2 for HL-resolution is to pick terms with the same outer function symbol when possible. Further in the given example, we can note that there is linkage between the arguments, a and 1, of the two terms that were resolved, further indicating that k<>g(a) and l=g(i) might be a profitable choice. Such analysis will be facilitated by the use of the equality connection graph to be described below. We have only begun to consider these ideas, but believe they may lead to interesting results. We believe the idea of choosing two terms and attempting to link them through the graph makes HL-resolution perhaps more amenable to such heuristic analyses than other equality systems. In any case the generation of paramodulants is, again, controlled and directed by the choice of A_1 and A_2, which choice could be made by some high level planning process or even human interaction.

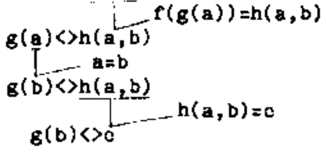
5). Note that we could pose the rule strictly in terms of paramodulation without recourse to a nucleus and resolution at the last step. However our approach emphasizes the notion that we are looking for OUTERLEVEL linkages, and especially those that lead to potentially useful function/predicate stripping. Furthermore, it provides target terms to be linked.

3. Completeness

The basic idea is that, from the existance of an unrestricted paramodulation deduction, we construct an equivalent HL-deduction. Proofs of the following Lemmas and Theorems are found in [6].

Lemma 1. If S is an E-unsatisfiable set of UNIT, EQUALITY clauses including x=x and functional reflexive axioms, then S has a refutation with a negative clause as top clause by paramodulation and resolution whose paramodulators are unit clauses in S

Lemma 1 shows that there always exists an unrestricted link between the negative clause and the last clause of the proof in the unit case. The only problem is the restriction of into term containment. We know that a paramodulation deduction tree can be rearranged into subsequences of deduction in such a way that all paramodulations in a subsequence are into one side of the inequality. For example, the deduction can be rearranged so that the two paramodulations into the right side are done first. Note, it cannot be rearranged so that the paramodulation into the left side is done first unless a new step using g(y)=g(y) is used. Note, however, that each subsequence into the same side will correspond to one or more HL-resolvents.



Lemma 2. A given clause t<>t' and a paramodulation deduction

$$T[t_1[r_1],u_1=v_1,t_2[r_2],u_2=v_2,\ldots,t_{n-1}[r_{n-1}],u_{n-1}=v_{n-1}:t_n],$$

which generates a paramodulant t_n<>t", where

1) The paramodulation sequence in T can be arranged according to the order of argument positions in the outer term of r_i, 1<=i<=n-1,
2) The terms t and t1 are unifiable,
3) t"=t'*E(T), where E(T) is the composition of unifiers in T, and
4) The last paramodulation only is paramodulation into the outer term itself whose from term is not a variable,

can be transformed into an HL-deduction which generates an HL-resolvent that is the same as t_n<>t" within the alphabetic variance.

Theorem 1. If S is an E-unsatisfiable set of UNIT, EQUALITY clauses including x=x and functional reflexive axioms, then S has an HL-refutation with the choice that the sets of paramodulators contains all the positive equality clauses in S.

Theorem 2. If S is an E-unsatisfiable Horn set of EQUALITY clauses including x=x and the functional reflexive axioms FR, then S has an HL-refutation with choice of the set of paramodulators as above.

Corollary 1. If S is an E-unsatisfiable Horn set of clauses including x=x and the functional reflexive axioms FR, then S has an HL-refutation with the same choice of the set of paramodulators as above.

While completeness is important to know about, it is more important in our view to develop effective proof procedures. Thus, for example, we would not recommend using FR(Function Reflexive) axioms. We have seen in Section 2, Comment 2 that for some cases where FR is required in simple paramodulation, HL-resolution proofs exist without FR because we are allowed to link either forward or backward. Whether HL-resolution is complete without FR (perhaps with some other restrictions relaxed) remains open. However, in practice we do not recommend using them.

Also we believe that in practice P and D will not be chosen to be both all of S. Again, we do not have a theory as to how they may be restricted and still maintain completeness. Indeed we are only beginning to consider practical aspects of choosing P and D. This is an area for considerable further study and experimentation.

Finally, we note that the use of HL-resolution (or any paramodulation rule) with only the negative clause as support eliminates the generation of positive equalities. This may not be advantageous in view of the effective use of new demodulators in [10]. We have not experimented with the generation of HL-resolvents in which A_1 and A_2 are both positive. This also is an area for considerable further investigation.

4. Plans for k-pd Linking

Clearly a major part of HL-resolution is to determine if there is one or more k-pd links between the chosen target terms. The restrictions on and properties of k-pd links suggest the use of an equality graph to aid in finding links, very much like regular connection graphs are used in finding resolutions. The basic idea is to form a graph at the beginning of the run in which terms that could potentially paramodulate at the outer level are connected and the corresponding unifiers are formed. Then two candidate terms for HL-resolution can be attached to the graph. Paths of length less than the bound for k which connect both the outer terms and inner terms can then be easily found and the corresponding set of unifiers tested for compatibility as in [1].

We now present some definitions leading up to such a graph mechanism and the formation of HL-resolution plans for a pair of terms.

Definition. Let P be a set of positive unit equality clauses. An **equality graph** (EG) is a graph such that

1) To every left or right term of equality, there corresponds a node whose label is the term.
2) Two nodes are connected if their terms are unifiable after renaming variables so that different clauses contain different variables. The most general unifier is the label of the edge.
3) Nodes corresponding to terms which belong to the same equality clause are grouped together in the graph. The clause number is the label of the group.

Since a group of nodes in an equality graph consists of only two nodes, the notation i+ (or i-) will be used to represent the right (or left) node in the group i, -i+ (or -i-) represents the node i- (or i+) which is the other term in the group i, and i* denotes i+ or i-.

Definition. . A **linking path** between two terms t_0 and t_n is a path in the equality graph of the form

$$t_0 \text{ -- } t_1=s_1 \text{ -- } t_2=s_2 \text{ -- } \ldots \text{ -- } t_{n-1}=s_{n-1} \text{ -- } t_n$$

where, $t_i=s_i$ is a clause and t_0 links to t_1 and each s_i links to t_{i+1}. The length of this linking path is n.

Definition. Let the linking path between two terms s_1 and t_{n+1} be

$$s_1 \overset{E_1}{\text{ -- }} t_2=s_2 \overset{E_2}{\text{ -- }} t_3=s_3 \overset{E_3}{\text{ -- }} \ldots \overset{E_n}{\text{ -- }} t_{n+1}$$

where the variables in all clauses have been separated and E_i, $1<=i<=n$, is a MGU of s_i and t_{i+1}. If $E=E_1*E_2*\ldots*E_n$ is defined, where * is the operation of compatible composition, then the linking path is said to be **link compatible** and E is called a link compatible unifier.

Definition. Let a term $f(t_1, t_2, \ldots, t_n)$ have a link compatible unifier E_i for each argument t_i to a k-pd link t_i', $1<=i<=n$. If $E=E_1*E_2*\ldots*E_n$ is defined, then the term $f(t_1, t_2, \ldots, t_n)$ is said to be **term compatible** and E is called a term compatible unifier.

Definition. An **augmented equality graph** of a term t for a term s, denoted by AEG(t,s), is an equality graph as above with the two extra groups of nodes {t} and {s}, where,

1) All nodes in EG whose labels are unifiable with t or s are connected to t or s, respectively, with a labeled link labeling the unifier.
2) All nodes in EG whose labels have the same outer function symbol as t but are not unifiable with t are connected to t with an unlabeled link.

The labeled link is used for checking compatibility in finding a paramodulation sequence and the unlabeled link is for finding inner level target terms, which will be described below.

Definition. A plan Plan(t,s) of a term t for a term s is a set of tuples of the form $(\langle I_1*, I_2*, \ldots, I_k*\rangle, k\text{-}E, r)$, where,

1) In the sequence $\langle I_1*, I_2*, \ldots, I_k*\rangle$, I_j, $1<=j<=k$, is a group number and I_j* represents a left or right node (from term) depending on the sign of *.
2) k-E is the composition of the labels (unifiers) of the linking path $s \text{ -- } I_1 \text{ -- } I_2 \text{ -- } \ldots \text{ -- } I_k$ from s to I_k.
3) r is the term formed by paramodulating the paramodulators in the sequence $\langle I_1*, I_2*, \ldots, I_k*\rangle$ into the term s in their order.

Rather than trying to build k-pd links of a term t which are to be resolved with a term s in an ad hoc way, a systematic method like target-driven search can be devised using the restrictions on the k-pd link. Since a k-pd link has the restriction of into-term containment, any into term should not be properly contained in any proceeding into-term.

Suppose we try to find all k-pd links of a term t which are to be resolved with a term s.

1) case 1: t is a constant or variable.

In this case, the linking process is simple due to the into-term-containment restriction. In fact, all the position vectors of the into-terms in the sequence are the same, i.e., the outer level position vector.

2) case 2: t is complex term

Let t be a complex term $f(t_1,t_2,\ldots,t_n)$. There will in general be many into term candidates at the beginning of the linking process. Furthermore, it seems difficult to know when to terminate an inner level linking sequence. We propose a method, which we call target-driven search, that works backwards from s rather than forwards from t.

Step 1)

Try to find outer level links of s which are either unifiable with t or have the same function symbol as t and are of length no greater than the bound on k.

Step 2)

Let s' be one of the outer level links of s.

Subcase 1): s' is unifiable with $f(t_1,t_2,...,t_n)$. Then, by the compatibility of unifiers on the k-pd link, there exists a k-pd link of t on the path t--s'--...--s. Therefore we have found a k-pd link.

Subcase 2): s' is not unifiable with t but has the same function symbol f. Let s' be of the form $f(s_1,s_2,...,s_n)$. Now we can break down the linking process into sub-linking processes of finding k-pd links of t_1 to s_1, t_2 to s_2, ..., and t_n to s_n. The sum of the lengths of these links and the length of the link from s to s' must be bounded, which narrows the search considerably. Assume that all sub-link paths with compatible compositions E_1, E_2, ..., E_n respectively are found. If E_1, E_2, and E_n are compatible, then $f(s_1',s_2',...,s_n')$, where s_i', 1<=i<=n, are k-pd links of s_i, is checked to see if it is unifiable with t. If they are unifiable, there exists a k-pd link of t on the path $f(t_1,t_2,...,t_n)$--$f(s_1',s_2',...,s_n')$-- ... --$f(s_1,s_2,...,s_n)$--...--s.

Note that if s' is neither of those two cases, there is no link between t and s' because of the restriction of into-term containment.

5. Implementation and Experimentation

We have implemented HL-resolution on NUTS (Northwestern University Theorem-proving System). NUTS is a programmable, interactive theorem proving system based on LMA (Logic Machine Architecture) [7]. The main part of the additions to NUTS centers on a pair of algorithms, based directly on the comments in Section 4, which generate first the set PLAN(t,s) of all plans for the two terms t and s and second the set PDLINK(t,s) of all k-pd links.

A primary purpose is to compare HL-resolution and paramodulation, so we didn't try any open problems yet but included problems from group theory and ring theory.

In the experiments reported on below, we made several restrictions. In linking process, no paramodulation was allowed from or into variables, as is the standard in most paramodulation experiments. Since HL-resolution may generate an HL-resolvent using 0-pd link, in the case that one of from-term or to_term happens to be a variable, the HL-resolvent is, in fact, a paramodulant genetated by paramodulation from or into a variable. But that is not a severe problem because it is allowed only from or into outer level term. We placed a bound on k in such a way that we did't allow more than 1 paramodulation at a position. An interesting restriction is to not allow the same paramodulator to be used at the same position more than once in the link. Of course, we did not use the functional reflexive axioms. As remarked above, it is an open question as to the effect of these restrictions on completeness. However, they are necessary for effectiveness in both HL and regular paramodulation.

In all experiments, we picked the clause with the fewest number of symbols for the next step. In the oase of HL-resolution, we always worked backward. For the paramodulation runs, we picked some positive olause as set-of-support, usually a clause from the special hypotheses and all paramodulators were applied only from the left to the right side of equalities. The clause used in a particular experiment are indicated in the tables below.

An important comment is that in the HL experiments we did not make use of any heuristics or human intervention in ohoosing a target term to link to or in filtering the HL-resolvents for retention except in the ring problems. There we used a very simple heuristic - if the outer function symbol of the HL-resolvent did not also occur as an outer function symbol in some input clause, the resolvent was not kept. This gave extra emphasis to the notion of working on outside terms. The importance of this comment is that the HL format provides first a pair of target terms and second an end result that is much more significant and much more like a human level inference than ordinary paramodulation. We intend that HL-resolution be used with heuristics for better selection of target terms and "interesting" results. It is possible that some problems would admit good heuristics for selecting the target terms; certainly there are more intelligent possibilities than to just take the one with fewest symbols or to take any target for which there is a link as was done in our simple experiments. We also feel that there could be better heuristics developed for deciding to keep a clause or not based on the fact that an HL-resolvent is a larger, more human-like step. In fact, in this last regard, one might even consider using HL-resolution in an interactive mode since the number of clauses presented to the user would be significantly less than in ordinary resolution or paramodulation. A user might be able to digest an analyse the limited number of these clauses and help direct the program's effort.

Legend: In the following report of experiments, the experiment **h** and **p** mean HL-resolution experiment and set-of-support paramodulation experiment, respectively. Further, A, P, D, S, and N represent Axiom set, Paramodulator set, Demodulator set, Supported clause set and Non-supported clause set, respectively. Here the axiom set is the set of clauses which can be used as satellite clauses.

Group Theory Experiments

Set of clauses;

1. f(e,x)=x   2. f(x,e)=x
3. f(g(x),x)=e   4. f(x,g(x))=e
5. f(f(x,y),z)=f(x,f(y,z))
6. f(a,e)<>a ; (^{V}x) f(x,e)=x
7. f(a,y)<>e ; (^{V}x)(Ey) f(x,y)=e
8. g(g(a))<>a ; (^{V}x) g(g(x))=x
9. f(g(a),g(b))<>g(f(b,a))
; $(^{V}x^{V}y)$ g(f(x,y))=f(g(y),g(x))
10. f(x,x)=e ; (^{V}x) f(x,x)=e
11. f(a,b)<>f(b,a) ; --> $(^{V}y^{V}z)$ f(y,z)=f(z,y)

Experiments

g1hd1: A:1,3,5,6	P:1,3	D:1,3
g1hd2: A:5,6	P:1,3	D:1,3
g1pd1: S:5	N:1,3,6	D:1,3
g2hd1: A:1,3,5,7	P:1,3	D:1,3
g2hd2: A:5,7	P:1,3	D:1,3
g2pd1: S:5	N:1,3,7	D:1,3
g3hd1: A:1,2,3,4,5,8	P:1,2,3,4	D:1,2,3,4
g3hd2: A:5,8	P:1,2,3,4	D:1,2,3,4
g3pd1: S:5	N:1,2,3,4,8	D:1,2,3,4
g4hd1: A:1,2,3,4,5,9	P:1,2,3,4	D:1,2,3,4
g4hd2: A:5,9	P:1,2,3,4	D:1,2,3,4
g4pd1: S:5	N:1,2,3,4,9	D:1,2,3,4
g5hd1: A:1-5,10,11	P:1,2,3,4,10	D:1,2,3,4,10
g5hd2: A:5,11	P:1,2,3,4,10	D:1,2,3,4,10
g5pd1: S:5	N:1-4,10,11	D:1,2,3,4,10

gihj and gipj are similar to the experiments gihdj and gipdj, respectively, except that no demodulation is applied to inferred clauses. In the tables below, the number of paramodulants used in the k-pd links of an HL run is given in parentheses under the paramod column.

Results

	proof	HL-	para-	kept	gen.	proc.
	found	res.	mod.		time	time
g1hd1	yes	78	(178)	47	132	38
g1h1	yes	115	(266)	51	177	49
g1hd2	yes	67	(169)	44	111	35
g1h2	yes	116	(269)	51	150	41
g1pd1	yes	na	240	63	30	31
g1p1	no	na	1000	263	969	389
g2hd1	yes	45	(105)	26	61	13
g2h1	yes	63	(139)	30	79	17
g2hd2	yes	44	(107)	29	57	15
g2h2	yes	54	(133)	48	64	16
g2pd1	yes	na	133	48	15	18
g2p1	no	na	600	192	276	104
g3hd1	yes	46	(122)	30	128	29
g3h1	yes	46	(122)	30	129	25
g3hd2	yes	33	(93)	22	54	14
g3h2	yes	33	(93)	22	54	13
g3pd1	yes	na	55	15	13	8
g3p1	no	na	600	132	104	67
g4hd1	yes	44	(122)	20	102	24
g4h1	yes	44	(122)	20	101	23
g4hd2	yes	40	(118)	20	59	23
g4h2	yes	40	(118)	20	59	21
g4pd1	no	na	600	101	58	94
g4p1	no	na	600	116	85	58
g5hd1	yes	139	(376)	57	288	79
g5h1	yes	139	(376)	57	288	74
g5hd2	yes	127	(365)	57	149	78
g5h2	yes	127	(365)	57	149	72
g5pd1	yes	na	104	26	8	15
g5p1	no	na	600	110	63	54

Boolean Algebra Experiments

Set of input clauses

1. s(x,y)=s(y,x)
2. p(x,y)=p(y,x)
3. s(x,0)=x
4. s(0,x)=x
5. p(x,1)=x
6. p(1,x)=x
7. s(x,n(x))=1
8. s(n(x),x)=1
9. p(x,n(x))=0
10. p(n(x),x)=0
11. s(p(x,y),p(x,z))=p(x,s(y,z))
12. p(s(x,y),s(x,z))=s(x,p(y,z))
13. s(a,1)<>1 ; (∀x) s(x,1)=1
14. s(a,a)<>a ; (∀x) s(x,x)=x
15. s(a,p(a,b))<>a ; (∀x∀y) s(x,p(x,y))=x

Experiments

b1hd1: A:3,5,7,9,11-13	P:1-12	D:3-12
b1hd2: A:12,13	P:1-11	D:3-10
b1pd1: S:12	N:1-11,13	D:3-10
b2hd1: A:3,5,7,9,12,14	P:1-12	D:3-10
b2pd1: S:12	N:1-11,14	D:3-10
b3hd1: A:3,5,7,9,11,12,15	P:1-12	D:3-10
b3hd2: A:5,11,12,15	P:1,2,3,5,7,9	D:3-10
b3pd1: S:12	N:1-11,15	D:3-10

bihj and bipj are similar to the experiments bihdj and bipdj, respectively, except that no demodulation is applied to inferred clauses.

Results

	proof	HL-	para-	kept	gen.	proc.
	found	res.	mod.		time	time
b1hd1	yes	9	(29)	8	15	6
b1h1	yes	9	(29)	8	15	5
b1hd2	yes	9	(29)	8	15	6
b1h2	yes	9	(29)	8	15	5
b1pd1	yes	na	34	25	4	15
b1p1	no	na	500	181	156	191
b2hd1	yes	9	(29)	7	15	6
b2h1	yes	9	(29)	7	15	5
b2pd1	yes	na	36	27	5	15
b2p1	no	na	600	178	103	89
b3hd1	no	110	(440)	86	194	113
b3h1	no	200	(816)	168	335	165
b3hd2	yes	44	(130)	43	54	41
b3h2	yes	44	(130)	43	57	43
b3pd1	yes	na	99	45	13	36
b3p1	no	na	600	193	221	104

Ring Theory Experiments

Set of input clauses

1. s(x,y)=s(y,x)
2. s(s(x,y),z)=s(x,s(y,z))
3. s(x,0)=x
4. s(0,x)=x
5. s(x,i(x))=0
6. s(i(x),x)=0
7. p(x,p(y,z))=p(p(x,y),z)
8. s(p(x,y),p(x,z))=p(x,s(y,z))
9. s(p(y,x),p(z,x))=p(s(y,z),x)
10. p(a,0)<>0 ;(∀x) p(x,0)=0
11. p(a,i(b))<>i(p(a,b)) ;(∀x∀y)p(x,i(y))=i(p(x,y))
12. p(x,x)=x ;(∀x) p(x,x)=x
13. s(a,a)<>0 ; --> (∀y) s(y,y)=0

Experiments

r1hd1:	A:1-9,10	P:3,5	D:3-6
r1hd2:	A:2,5,8,10	P:1,3,5,7,8,9	D:3,4,5,6,8,9
r1pd1:	S:2,8	N:1,3-7,9,10	D:3,4,5,6,8,9
r2hd1:	A:2-9,11	P:1-9	D:2-9
r2hd2:	A:2,4,5,8,11	P:1,3,5,7,8,9	D:3,4,5,6,8,9
r2pd1:	S:2,8	N:1,3-7,9 11	D:3,4,5,6,8,9
r3hd1:	A:2,6,8,13	P:4,6,9,12	D:3-9,12
r3pd1:	S:2,8	N:1,3-7,9,12,13	D:3-6,8,9,12

rihj and ripj are similar to the experiments rihdj and ripdj, respectively, except that no demodulation is applied to inferred clauses.

Results

	proof found	HL-res.	para-mod.	kept	gen. time	proc. time
r1hd1	yes	114	(310)	27	124	68
r1h1	yes	114	(310)	27	125	64
r1hd2	yes	19	(68)	8	27	11
r1h2	yes	19	(68)	10	27	11
r1pd1	no	na	600	112	119	206
r1p1	no	na	600	152	188	170
r2hd1	yes	84	(432)	31	289	66
r2hd2	yes	38	(143)	21	76	24
r2h2	no	240	(794)	146	1050	326
r2pd1	no	na	600	112	119	204
r2p1	no	na	600	152	191	169
r3hd1	yes	197	(605)	98	543	446
r3h1	yes	200	(598)	116	670	538
r3pd1	no	na	600	112	117	213
r3p1	no	na	600	152	187	170

6. Conclusion

We proposed a new inference rule called HL-resolution for the equality relation that is Intended to have the benefits of hyper steps and to control the uses of paramodulation. It generates a resolvent by building a paramodulation/demodulation link between two terms using a preprocessed plan as a guide. We proved completeness for Horn sets and suggested an efficient method for implementation. A number of experiments were conducted on problems in abstract algebra and the results are encouraging. But many problems remain untouched. Completeness without Function Reflexive axioms possibly with some other restrictions relaxed remains open. And we do not have a theory as to how to restrict the choice of the sets of paramodulators and demodulators and still maintain completeness or effectiveness. We have not, as yet, considered what strategies for choosing pairs of target terms might be effective nor experimented with different target strategies. Equally important is the question of whether or not a program might be able to select only the profitable links from the set PDLINK(t,s). In our experiments, we simply generated all HL-resolvents possible within the bound on k. As mentioned earlier, we believe HL-resolvent has more potential for developing effective heuristics because of the format - there could be heuristics for picking target terms, and heuristics for selecting k-pd links. Further, an HL step is a larger, potentially more significant step; we feel that it could be easier to predict the utility of such a larger step than to do the same for a series of shorter steps. Whether or not this potential can be really developed remains to be seen.

7. References

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

2) Darlington, J. L., "Automated Theorem Proving with Equality Substitutions and Mathematical Induction," in Machine Intelligence$_f$ Vol.3 (B. Meltzer and D. Michie, eds.), American Elsevier, New York, pp.113-127.

3) Digricoli, V. J., "Resolution by Unification and Equality," Proceedings of 4th Workshop on Automated Deduction, 1979, Texas.

4) Harrison, M. and N. Rubin, "Another Generalization of Resolution," J. ACM, Vol.25, No.3, July 1978, pp34l-351.

5) Kowalski, R., "A Proof Procedure Using Connection Graph," JACM 22,4, 1975.

6) Lim, Younghwan, "A New Hyperparamodulation Strategy for the Equality Clauses", Ph.D Dissertation, Northwestern University, 1985.

7) Lusk, E., W. McCune and R. Overbeek, "Logic Machine Architecture: Kernel Functions," Proceedings of the 6th Conference on Automated Deduction, Springer-Verlag Lecture Notes in Computer Science, Vol. 138, 1982.

8) McCune, W., Semantic Paramodulation for Horn Sets, Ph.D. Dissertation at Northwestern University, 1984.

9) Morris, J., "E-resolution: an Extension of Resolution to Include the Equality Relation," IJCAI, 1969.

10) Overbeek, R., J. McCharen and L. Wos, "Complexity and Related Enhancements for Automated Theorem-proving Program," Comp. and Maths. with Appls. Vol.2, No.1-A, 1976, pp.1-16.

11) Robinson, G. A. and L. Wos, "Paramodulation and Theorem Proving in First Order Theories with Equality," in Machines. Intelligence Vol.4 (B. Meltzer and D. Michie, eds), American Elsevier, New York, 1969, pp.135-150.

12) Siekmann, J., and G. Wrightson, "Paramodulated Connectiongraphs," Acta Informatica, 1980.

13) Wos, L., G. A. Robinson, D. F. Carson and L. Shalla, "The Concept of Demodulation in Theorem Proving," JACM, Vol.14, No.4, October 1967, pp.698- 709.

14) Wos, L, R. Overbeek and L. Henschen, "HYPERPARAMODULATION: A Refinement of Paramodulation," Proceedings of the 5th Conference on Automated Deduction, 1980.