SEMANTIC RESOLUTION FOR HORN SETS

Lawrence J. Henschen
Northwestern University
Evanston, Illinois

Abstract

A new resolution strategy for Horn sets of clauses, each clause of which contains no more than one positive literal, Is presented which requires that in each resolution either one ancestor be a false positive unit or that one ancestor and the resolvent both be false. This strategy emphasizes "relevant" positive units while controlling the explosive non-unit resolution. Some properties of interpretations for Horn sets are reviewed and used to significantly reduce the computation and storage required to Implement semantic resolution for Horn sets. This work was supported in part by the National Science Foundation and Argonne National Laboratory.

I. Introduction

Since the Introduction of resolution for automated theorem proving, one line of effort has been to develop strategies and huertstlcs for special classes of clause sets or special classes of problems. One such class is the class of Horn sets. Recent interest In Horn sets [3,6] stems from the fact that many strategies that are not complete in general are complete for Horn sets. For example, positive unit resolution, which for Horn sets Is just Pi resolution, is complete. Negative resolution Is just set-of-support resolution with the set of support the negative clause. Factoring Is not necessary in order to refute an unsatisftable set of Horn clauses [3], In addition to possessing many useful and interesting properties, Horn sets occur quite often in practice. For example, the axiom systems for groups, rings, and many other algebraic systems have Horn sets as the natural translation Into clause form as do various formal systems (see e.g., Hermes [4]). Horn clauses also have a great deal of intuitive appeal, for the Horn clause $-L_1 \vee \ldots \vee -L_k \vee M$ is logically equivalent to $(L_1 \wedge \ldots \wedge L_k) \rightarrow M$. That is a Horn clause represents an implication in which each of the antecedents is a positive unit (or intuitively, a simple statement of fact or hypothesis) and the consequent is also a positive unit. Many mathematical axioms and theorems take this form. Finally, one can apply the splitting techniques developed in [3] to reduce a non-Horn set of clauses to a class of Horn sets; thus, the theory developed for Horn sets can be applied indirectly to non-Horn sets as well. The purpose of this paper is to present a new strategy (essentially a sharpening of semantic resolution) that is complete for Horn sets and to discuss some interesting properties of Horn sets that greatly simplify the implementation of semantic resolution for those sets.

While many methods for resolution are complete for Horn sets, some desirable combinations of strategies are not complete together even when the class of problems is restricted to Horn sets. Two such strategies are positive-unit resolution and set-of-support resolution [3], Restricting resolution to unit resolution avoids the combinatorial explosion of the number of clauses generated on the various levels that is usually present when non-unit resolution is allowed. For, In the latter case, when forced Into the non-unit section of a theorem prover, a program usually must try resolution on all pairs of literals from a pair of non-unit clauses. (When ordering or lock resolution [2] is used this is not the case; however, these methods are not compatible with many of the more successful strategies like set of support.) Thus, once the program leaves the unit section, even for just one level, relatively large numbers of resolvents are generated. Then in developing a strategy that is not compatible with unit resolution, great care must be taken to control the non-unit resolvents that are allowed to be generated lest the combinatorially explosive non-unit section undo the savings that the strategy yields in the unit section. Of course, semantic resolution [2] has as its main goal to restrict the resolvents generated to those that are highly likely to be relevant to the proof by taking into account the intended meaning of the clauses. In semantic resolution, an interpretation I of the symbols of the language is input to the theorem prover along with the set of clauses to be refuted. Then one parent in each resolution is required to be false in I. The idea Is that two clauses which are both true are less likely to produce a deduction leading to a contradiction than are two clauses one of which Is false. The most widely-used special case of semantic resolution is set-of-support resolution in which the interpretation I satisfies just the axioms and is only Implicitly present. In Section 2 we develop a strategy which is a combination (but not the intersection) of positive-unit resolution and s email tic resolution.

One difficulty with semantic resolution is that if the domain of the interpretation contains more than a few elements the computation required to fully evaluate a non-ground clause may be too time consuming. In addition there is a problem of how to represent for each clause the sets of domain elements for which the clause is true and those for which the clause is false. For these reasons, most applications of semantic resolution have used only very simple interpretations; for example, the interpretation in which all literals are false yields PI resolution. In Section 3 the problem of reducing the amount of computation and storage required for evaluating Horn clauses based on the use of cross products of interpretations will be discussed. Section 4 will contain some concluding remarks.

II. Completeness of Positive-unit Semantic Resolution for Horn Sets

The reader is assumed to be familiar with the resolution principle for first order logic and the terminology related to It. The definitions necessary for the particular theorems to

be proved will be presented; then some lemmas will be recalled and the completeness of the strategy proved.

DEFINITION: A Horn clause is a clause containing not more than one unnegated literal. A Horn set is a set of Horn clauses.

DEFINITION: An interpretation I for a set S of clauses consists of a non-emoty domain D of objects and an assignment of the predicate, function and constant symbols occurring in S such that each n-ary predicate symbol is assigned an n-ary relation on D, each n-ary function symbol is assigned an n-ary function on D into D, and each constant symbol is assigned an element of D.

In the standard inductive way the assignment of values to symbols in S can be extended to an assignment of all ground formulas over the alphabet of S. If ϕ is the assignment above, define ϕ' as follows: let $\phi' = \phi$ on the predicate, function and constant symbols of S; let $\phi'(f(t_1,\ldots,t_n))$ be $\phi(f)[\phi'(t_1),\ldots,\phi'(t_n)]$ for the ground term $f(t_1,\ldots,t_n)$ and $\phi'(P(t_1,\ldots,t_n))$ be $\phi(P)[\phi'(t_1),\ldots,\phi'(t_n)]$ for the ground atom $P(t_1,\ldots,t_n)$; and let ϕ' interpret the logical connectives according to their usual meaning.

DEFINITION: A clause is falsified by the interpretation I if it has a ground instance which is false in I.

DEFINITION: A clause is positive if it contains only unnegated literals. A clause is negative if it contains only negated literals. A clause is mixed if t contains both negated and unnegated literals.

DEFINITION: A unit is a clause containing one literal.

DEFINITION: A resolvent of two clauses A and B is a positive-unit resolvent If one of A and B is a positive unit. A positive-unit refutation of a set S of clauses is a refutation in which each resolvent is a positive-unit resolvent.

DEFINITION: Given an interpretation I, a resolvent of two clauses A and B Is a semantic resolvent with respect to I if one of A and B is falsified by I. A refutation of a set S of clauses is a semantic refutation with respect *to* I if every resolvent in the refutation is a semantic resolvent with respect to I.

In [3] it was shown that any unsatIsflable Horn set has a positive-unit refutation but that for an arbitrary set of support, T, there may not be a T-supported positive-unit refutation. That is, the intersection of the two strategies, positive-unit resolution and set-of-support resolution, is not complete for the class of Horn sets. Since set-of-support resolution can be viewed as a special case of semantic resolution [2], It follows that the intersection of positive-unit resolution and semantic resolution also is not complete for the class of Horn sets. Now, semantic resolution is an attempt to bring the meaning of the symbols into the search effort as an aid in guiding a program to the generation of the empty clause, the idea being that in order to generate the contradiction that produces the empty clause, i.e., falsehood, one should use clauses that are themselves false. Positive-unit resolution has as its goal the elimination of a class of resolutions that produces combinatorially large numbers of new clauses. In addition, there is an intuitive appeal to positive units as conveying more information than other clauses since it is generally more useful to know that a particular property holds than to know just that one of a number of statements hold (see [3] for a more complete discussion of the advantages of unit resolution). A strategy that was based on both of these intuitively appealing ideas would seem to be a good candidate for an efficient theorem prover for Horn sets. Since the intersection of the two cannot be complete conditions must be relaxed and/or altered in such a way as not to lose the advantages of the two constituent strategies. In particular, relaxing the conditions to allow non-unit resolution must be done with extreme care.

In the strategy presented below, this rate of growth will be controlled by requiring that for non-unit resolutions one parent and the resolvent <u>both</u> be falsified by the interpretation. Since the number of false resolvents generated by a false clause will generally be smaller than the total number of resolvents that the false clause could generate, this new restriction should cut down the number of clauses that will be generated by non-unit resolution. With this restriction on the non-unit section, an additional restriction can be placed on the positive-unit section -- namely, that the positive unit must be false. The net effect of these modifications is to produce a strategy that combines the advantages of positive-unit resolution and semantic resolution and that relies especially heavily on the semantics, provided by the user in the form of an Interpretation, to guide it through the dangerous non-unit section. Also, if the interpretation does not falsify any axiom, as would be the usual case, then as in set-of-support pairs of axioms will not be allowed to resolve directly even when one of them is a positive unit, a shortcoming of just positive-unit resolution (and indeed, of hyperresolution In general). Finally, the reader may also note a great similarity between the method used in the completeness proof here and of bilinear resolution of Kuehner [6]. There, of course, there is no restriction on positive-unit resolution. Also the non-positive-section of a bilinear program is based purely on syntactic considerations, i.e., on the use of negative clauses.

Before proving the completeness of the new strategy, some previously proved results [3] on Horn sets are recalled and some lemmas are proved. The ground case is considered first. The following two lemmas from [3] are recalled.

DEFINITION: A set S of clauses is minimally unsatIsflable if S Is unsatIsflable but no proper subset of S Is.

LEMMA 1. (LEMMA 1 from [3]) If S is an unsatIsflable set of Horn clauses then there is a

positive-unit refutation of S.

LEMMA 2. (THEOREM 5 from [3]) If S is a minimally unsatisfiable set of Horn clauses, then S contains exactly one negative clause.

One additional lemma is proved before the completeness theorem is given.

LEMMA 3. Let S be a minimally unsatisfiable set of ground Horn clauses, and let p be an atom occurring in S. Then p occurs unnegated exactly once.

PROOF. Of course if p occurs in S it must occur positively at least once or -p would be a pure literal, and S would not be minimally unsatisfiable. Now suppose S contains the clauses $pA_1,pA_2,\ldots,pA_n$; $-pB_1,-pB_2,\ldots,-pB_m$; $C_1,C_2,\ldots,C_k$ where all occurrences of p and -p are shown explicitly. Then $S_1 = \{A_1,\ldots,A_n,C_1,\ldots,C_k\}$ is also unsatisfiable. Moreover, since each clause pA_i is a Horn clause and p is an atom, each A_i is negative. Now if any A_i is empty, then p occurs in S as a positive unit and subsumes any other clause that contains the positive literal p. In this case, if n>1, S cannot be minimally unsatisfiable. If no A_i is empty and n>1, then by LEMMA 2 above, S_1 is not minimally unsatisfiable for it contains more than one negative clause. Let S_2 be minimally unsatisfiable subset of S_1 and let A_1 be the one negative clause from S_1 that is present in S_2. (Indeed, one A_i must be present or else a subset of the set $\{C_1,\ldots,C_k\}$ would be unsatisfiable contradicting the minimal unsatisfiability of S.) By returning the literal p to A_1, one obtains a deduction of the positive-unit clause p from the clause pA_1 and the clauses C_j, $1\leq j\leq k$. Having deduced this unit clause, all of the clauses pA_i can be discarded because they are subsumed by p and a refutation of S obtained from p and the remaining clauses of S. Thus a refutation exists which uses only one clause containing the positive literal p. But it had been assumed that n>1, and so S could not be minimally unsatisfiable, a contradiction.

QED

Of course, if S is minimally unsatisfiable but <u>not</u> Horn, it may very well contain several occurrences of a positive literal as can be seen by the example {pq,p-q,-pq,-p-q}.

THEOREM 1. Let S be an unsatisfiable set of ground clauses and suppose S is a Horn set. Let I be an interpretation for the language of S. Then there exists a refutation R of S such that each resolution in R satisfies one of the following two conditions: 1. one of the resolvends is a positive unit which is false in I, or 2. one of the resolvends and the resolvent are both false in I.

PROOF. The proof proceeds by induction on the number n of atoms occurring in S. It may be assumed without loss of generality that S is minimally unsatisfiable.

Case 1. n=1. In this case S consists of the two unit clauses p and -p. One of the two clauses is false in I and the resolvent, namely the empty clause is false, so that the one step refutation satisfies condition 2.

Case 2. Assume that the theorem holds for all sets of clauses containing less than n atoms, and suppose S contains exactly n atoms. There are two subcases.

Subcase a. S contains a positive unit p which is false in I. Let S consist of the clauses $p, -pB_1,\ldots,-pB_k,C_1,\ldots,C_m$. Since p is a false positive unit, each resolution $R(p,-pB_i)=B_i$ satisfies condition 1 of the theorem. By LEMMA 3, no other clause of S contains the positive literal p. Thus from S by a sequence of resolutions satisfying condition 1 one can generate the set of clauses $S_1=\{B_1,\ldots,B_k,C_1,\ldots,C_m\}$. S_1 is unsatisfiable and contains strictly fewer than n atoms. Thus by the induction hypothesis there is a refutation R of S_1 satisfying the conditions of the theorem. Then the juxtaposition of the k resolutions involving p and R also satisfies the conditions and is a refutation of S.

Subcase b. S contains no positive units which are false in I. (The reader may wish to consult Example 1 below while reading this portion of the proof.) By an iterative process we will locate a false clause in S which can be used to start a sequence of resolutions satisfying condition 2; that is, each resolvent in the sequence will be false. Finally, the last resolvent will either be the empty clause or a false positive unit. Now, from LEMMA 1 one can conclude that S contains at least one positive unit clause. Let the positive units of S be $p_1,\ldots,p_{n_1}$. Form the set S_1 by excluding these positive units from S and deleting the literals $-p_i$, $1\leq i\leq n_1$ from the remaining clauses. In the usual way one can establish that S_1 is unsatisfiable and Horn. If S_1 does not contain a false positive unit or the empty clause then let $p_{n_1+1},\ldots,p_{n_2}$ be the positive units of S_1. Form the set S_2 by excluding from S_1 these positive units and deleting the literals $-p_i, n_1+1\leq i\leq n_2$ from the remaining clauses. As above S_2 is unsatisfiable and Horn. Continue generating sets S_i, at each step excluding the (true) positive units and deleting their negations from the other clauses, until an S_m is obtained which contains either a false positive unit or the empty clause. Note that since each p_i is a true positive unit, each negative literal that was deleted from a clause in the above process is false. Now suppose that S_m contains a false positive unit, say u. This positive unit corresponds to a clause $U=-q_1-q_2\ldots-q_ju$ in S where each q_g is one of the p_h, $1\leq h\leq n_m$ that was deleted in forming the S_i's. In addition each p_i, $1\leq i\leq n_m$ corresponds to a similar clause C_i in S. Also note that U is false in I since u is false in I and each $-q_j$ is the negation of a true positive unit; in the same way each negative literal of each C_i is false in I. We have now located the desired false clause, U. A chain of resolutions beginning with U, using the C_i and satisfying condition 2 will now be given that will generate the false positive unit u. The clause U contains a literal which is the negation of a positive unit that was excluded in producing S_m from S_{m-1}. That unit p_i in S_{m-1} corresponds to clause C_i in S. The resolvent of U and C_i on p_i consits of the positive literal u and some negative literals from among

the -p . But each of these literals is false so that the resolvent of U and C , call it $D_{m,1}$ is false in I. Therefore, this resolution satisfies condition 2 of the theorem. If $D_{m,1}$ contains another literal, say-p with $n_{m-1} < j \le n_m$ (i.e., a literal that was eliminated in ffirming the last set S) form the resolvent $D_{m,2}$ using $D_{m,1}$ and C .

As above, I) and I) are both false, so condition 2 is satisfied, Continue in this way until all negative literals that were deleted in forming S from S have been resolved away. Call the resulting clause D . Note that each negative literal in a clause C. where n -i^n corresponds to a literal that was eliminated in forming an S Thus, not only is D false in I and contains the positive literal u, it contains only negative literals that were deleted in forming sets S with k<m. The above process is now repeated resolving away negative literals from D that correspond to literals that were eliminated in forming S from S „. Call the result of this step D ™7 As above D is false, contains thr- positive literal u and does not contain a negative literal that participated in forming either S . or S . Continue in this way until the clause™!) is formed. Indeed, D is just the positive unit u which is false in I. Moreover, since u is present in each resolvent in the sequence, all thr intermediate resolvents and U itself can be discarded by subsumption once u is generated. That is, (S-(.U})U|.u} is unsatisftable. Now the false positive unit u can be used to eliminate the literal u altogether as in Case a above; the result is an unsatisfiable Horn set that contains fewer than n atoms. The induction hypothesis now gives the desired result. Tn case S contains the empty clause, the above process can be used with the only modification being that each of the intermediate clauses does not contain a positive literal and the final clause D. is the empty clause; thus when the process terminates, the result is in fact a refutation all of whose resolutions satisfy condition 2 (see Example 2).

This completes the induction step and hence also the proof of the theorem.

The process in the proof of Theorem 1 will now be illustrated with two examples.

EXAMPLE 1. Let S be the set of clauses [r,-rq, -rs,-q-sp,-pt,-tj and let I be the interpretation that assigns the value true to r, q, and s, and false to p and t. Now r is the only positive unit in S and it is true. According to the proof of the theorem the set S.*»[q,s,-q-sp,-pt, -t) would be formed. Then since both the positive units occurring in S are true, the process in the proof would continue to find the set S_2"=[p, -pt ,-t\, S_2 does contain a false positive unit,p. Therefore, according to the proof, the clause -q-sp of the original set S could be used to start a chain of resolutions that would result in the false positive unit p. One would first form the resolvent of -q-sp with, say, -rs to get the clause -r-qp. Note that -q-sp is false and that the resolvent -r-qp is also false and that the only true literal of the clause -rs is the positive literal s. Now the resolvent of -r-qp and -rq is formed. This resolvent is -rp which is also false. Finally, the resolvent of this clause and the clause r is generated. This last resolvent is the positive-unit p, which is a false positive unit clause. This can now resolve with -pt to produce the clause t. This clause resolves with the unit -t to produce the empty clause. Note that a theorem-proving program would not, of course, actually generate the sets S. and S ■ they were discussed here only to illustrate the process of the proof of Theorem 1. A program would simply note that the clause -q-sp was false and that the resolvent of -q-sp and -rq was also false and therefore legal to add to the set of generated clauses. Note that since literals -p might be deleted from several clauses in forming the S sets, one might expect a considerable amount of merging. However, the strategy cannot be strengthened to require merging as can be seen by noting that if the false clause U has only 2 literals, its resolvent with another Horn clause cannot be a merge.

EXAMPLE 2. Let S be as in Example 1, and let the interpretation I assign the value true to all atoms. The process in the proof of the theorem would generate sets S. and S~ as in Example 1. However, the positive unit p of S„ is not false for this I. Thus the set S ~[t,-t} would be generated. This set also does not contain a false positive unit so the set S,»lempty clause] would be generated. Now the empty clause comes from the clause -t in S, . As above one would start with this clause and work back, resolving first with the clause -pt to produce -p, then resolving with -q-sp to produce -q-s and so on. Again, an actual program would not consider any of the sets S.-S,, but simply which clauses were false and had false resolvents. Note that when the interpretation makes all atoms true the refutation produced will be an NI refutation, and that for Horn sets, such refutations are also input refutations [3j.

It is now shown that Theorem 1 does not remain true when the condition of being Horn is removed. Let S be the set (pq,-pq,p-q,-p-q} and let the interpretation I assign true to both p and q. Then one can generate resolvents using the false clause -p-q producing the two clauses -p and -q. However, neither of these can resolve with the clause pq since neither is a false positive unit and in each case the resolvent is not false. Thus no refutation can be produced satisfying the conditions of Theorem 1. Indeed, there is no refutation even if condition 1 is relaxed to read one ancestor is a false positive clause instead of unit. This set of clauses can also be used to show that the converse of Theorem 1 is not true. Let I assign both p and q false. Then the clause pq is false and its resolvent, p, with -qp and its resolvent, q, with -pq are both false positive units so that these resolutions satisfy condition 2. These false positive units can be used one after the other to produce the empty clause from -p-q. However, S is not a Horn set and indeed no renaming of S is Horn. We do have the following result, however.

THEOREM 2. Let S be a set of ground clauses which is minimally unsatisfiable. If for every interpretation 1 there exists a refutation satisfying the conditions of Theorem 1, then S is Horn,

PROOF. Suppose S contains a non-Horn clause, say C=pqC', where p and q are two positive literals. Let I be an interpretation in which both p and q are true. Now consider any refutation R of S. Since S is minimally unsatlsfiable, C must occur in R. Wherever C enters R the literals p and q will enter and will propagate until one of them is resolved away. Suppose p is the first of the two from C to be used as a literal of resolution. Now the clause resolving on p with C or its descendent must contain the literal -p and therefore certainly cannot be a false positive unit. Of course, the resolvend containing p and q also cannot be a positive unit. Thus condition 1 cannot be satisfied. Next, the resolvent must contain the positive literal q, which is true in I. Thus the resolvent cannot be false and condition 2 cannot be satisfied. Therefore R cannot satisfy the conditions of Theorem 1. Since R represented an arbitrary refutation of S we have a contradiction to the hypotheses of this theorem.

It should be pointed out that the conclusion of Theorem 2 is that S is Horn, not just that some renaming of S is (lorn. Indeed, if a set S has a renaming which is Horn but S itself is not Horn, then S will have an interpretation I as in the theorem relative to which there is no refutation satisfying the conditions of Theorem 1.

THEOREM 3. Let S be an unsatisfiable set of Horn clauses. Let I be an interpretation over the language of S. Then there exists a refutation R of S such that each resolution in R satisfies one of the following two conditions: 1. one of the resolvends is a positive unit which is falsified by 1, or 2. one of the resolvends and the resolvent are both falsified by 1.

PROOF. Let S^1 be an unsatisfiable set of ground instances of clauses in S and let R^1 be a refutation of S' satisfying the conditions of Theorem 1 wtth 1 as the Interpretation. Let R be the refutation of S obtained by lifting R' in the usual manner. R satisfies the conditions of this theorem as can be seen by the following. Suppose p is a false positive unit in R' and p is the corresponding clause in R. P cannot contain a negative literal because it has a positive clause, p, as an Instance. Also, P cannot contain more than one positive literal because it is either in a Horn set or is deduced from a Horn set by resolution L3]- Thus P is a positive unit. Moreover, it has p as an Instance, and p is false in I. Therefore, I falsifies P, and the resolution In R using P satisfies condlton 1 of this theorem if the corresponding resolution in R^1 using p satisfies condition 1 of Theorem 1.

In a similar way, If a resolution in R* satisfies condition 2 of Theorem 1, the corresponding resolution in R will satisfy condition 2 of this theorem because one of the resolvends and the resolvent will have instances which are false in 1. Since every resolution in R' must satisfy one of the two conditions, the theorem is proved.

111. Implementation

While semantic resolution has a great deal of intuitive appeal, in actual practice few theorem-proving methods that have been programmed use any but the most trivial of Interpretations for the languages being used and they are usually based on some syntactically recognizable feature. For example, the underlying interpretation for Pi resolution and also for hyporrcsolution is the interpretation in which every predicate symbol is assigned the relation which is identically false, and false clauses are recognized by the lack of negative literals. One would like to see the use of interpretations which satisfy, say, the axioms of the mathematical theory under consideration and thus would have the effect of focusing attention on the special hypotheses and conclusion of the result for which a proof is sought, as in set-of-support resolution. The advantage here over Just set of support would be that the semantic requirement, i.e., that one ancestor be false, would have effect at all levels of the search whereas set-of-support resolution, while quite effective at level 0, allows any level 1 or greater clause to resolve with any other clause. Or perhaps even better, one would like to be able to arrange the interpretation so as to single out particular clauses as being Important for the particular theorem by making those clauses false in the interpretation even If some do happen to be axioms. That is, for a particular problem the user might suspect that a certain axiom was especially relevant; he could arrange that axiom to be false in the interpretation thus allowing it to enter the semantic search at an earlier stage.

There are two reasons why little use has been made of semantic information in theorem provers. First, for interpretations whose domains are not fairly small, determining whether or not a clause containing variables is falsified may be a formidable Job. For example, the standard pair of axioms for associativity in group theory each have 6 variables. In a domain of 100 individuals, the determination that one of these axioms has no false instance, the normal case, would require testing 100 different 6-tuples. Even with highly efficient, optimized methods such a task would not likely be feasible. The second reason has to do with representing, for a given clause, which values falsify that clause. That is, having determined whether a clause possibly has some instances which are false, one would like to save that information. If a clause C has several variables, there may be many tuples that falsify C requiring more storage to save than the information might be worth. The purpose of this section is to present methods by which these two problems might be overcome. These remarks are not meant to be an exhaustive study of this area; in particular, this section will not discuss in great detail the methods and data structures that might be used by a particular program. Rather the purpose is to present some interesting, previously known results about interpretations and Horn sets and to Indicate briefly how these properties can be useful for theorem-proving programs. We begin with a basic definition from mathematics.

DEFINITION; Let L be a first order language, and let I and I be two interpretations over L with domains D. and D respectively. The cross-product of I and I„ is the interpretation I whose domain D is the cross-product of D_1 and D_2 (i.e., the set of ordered pairs whose first coordinate comes from

D_1 and whose second coordinate comes from D_2 and which assigns values to the predicate, constant, *and* function symbols of L as follows: if a is a constant symbol and I_1 assigns a_1 to a and 1_2 assigns a_2 *to a,* then I assigns the ordered pair $(a_1.a_2)$ to a; If f is an n-ary function symbol and if I_1 assigns to f the n-ary function £- on D and I_2 assigns to f the n-ary function f_2 on $D_{„}$, then 1 assigns to f the n-ary function on $D_1 \times D_2$ which maps an n-tuple of ordered pairs ((a ,b), . . .,(a ,b)) onto the ordered pair $(f,(a_1,\ldots,a_n),f_2(b_1,\ldots,b_n))$; if P is a predicate symbol and If I_1. assigns to P the n-ary predicate p_1 on D_1 and I_2 assigns to p the n-ary predicate p_2 on D_2, then 1 assigns to P the n-ary predicate which is true for an n-tuple of ordered pairs $((a_1,b_1),\ldots,(a_n,b_n))$ if and only if both $P_1(a_1,\ldots,a_n)$ and $P_2(b_1,\ldots b_n)$ are true.

McKinsey [7] noted that if a clause containing only one positive literal is true in two interpretations then it is also true in the cross product of the two interpretations. In addition, If a clause *C* is true of each algebra in a class of algebras that is closed under algebraic direct product, then there is subclause C' of C containing at most one positive literal (i.e., C' is Horn) and C' is also true of every algebra in the class. Horn [5] later showed that the class of Horn sets and the class of sets of clauses which are true in a direct product of two interpretations if and only if they are true in each individual interpretation are the same class. (Thus the name Horn clause.) In both cases only clauses containing only the equality predicate were considered; however, their methods easily generalize. Some of these results will be reviewed in the following paragraph; the reader is referred to [7] and [5] for the full presentation of this material as well as other results.

Consider two interpretations I. and I_2 and a ground literal 1,. If I, is an atom, then by definition L is true in 1. x I_2 if and only if it is true in both 1. and I_2 Now if L is the negation of an atom, say L=-M, then L is true in I. x I_2 if and only if it is true in at least one of I_1 and I_2 For M is true in 1. x I_2 if and only if it is true in both 1_1 and 1_2 and so M is false (i.e., L is true) if and only if M is not true in both I_1 and I_2 that is M is false in at least one of tnem. Next, suppose the ground Horn clause $C=-L_1\ldots-L_n M$ is true in both I. and I_2. Now if M is true In both I_1. and I- then M is true in the cross-product, and so C is also true in the cross-product. If M is not true in one of the I's, say M is false in 1., then some negative literal $-I_i$ is true in I . By the above, -I, is true in the cross-product, so again C is true in the cross-product. Thus, any ground Horn clause that is true in two interpretations is true in the cross-product. Of course, a Horn clause would be false in 1. x I_2 if and only if each negative literal was false In both I- and I_2 and the positive literal was false in *at* least one of the two. This analysis applies equally well to the case of determining if a particular instance of a clause with variables is true or false in the cross-product of two interpretations. These results generalize to products of more than just two interpretations in a straightforward way.

Thus, if $I.,\ldots,I_n$ are interpretations and C is a Horn clause which is true in each 1_i, then *C* is true in $I_1x\ldots xI_n$. If C is false in $I_1x\ldots xI_n$ then each negative literal of C is false in each I_i and the positive literal of C is false in at least one I_i .

The above remarks form the basis for reducing the amount of work required to evaluate Horn clauses for semantic resolution and to store information about the tuples for which such clauses are false. Consider again the case of an associativity axiom C with 6 variables. If instead of a single interpretation with a domain of 100 elements we use the cross-product of two interpretations each of whose domains has 10 elements (and thus whose cross-product domain has 100 elements), then to evaluate C in 1 x I_2 one would evaluate C in I_1 and I_2 requiring $2*10^6$*tuples to be considered as opposed to 100^{*12}, with a reduction factor of $5*10^{12}$. If more subInterpretations are used the savings is even more dramatic. For example, by using three domains each with 5 elements, we would be using the equivalent of a single interpretation whose domain had 125 elements, and the evaluation of a clause with 6 variables would require the$_4$ consideration of $3*5^6$, or approximately $5*10^4$, 6-tuples. Moreover, efficient means of organizing the calculations can be expected to reduce to an even greater degree the number of tuples requiring testing for a given clause.

The storage problem can also be reduced by taking advantage of the properties of Horn clauses. For the purposes of this paragraph we use the following notation:

if $\bar{a}=(a_1,\ldots,a_n)$ and $\bar{b}=(b_1,\ldots,b_n)$ are two n-tuples then a x b will indicate the n-tuple of ordered pairs $((a,b),\ldots,(a_n,b_n))$. Recall that a clause C is false in I_1 x I_2 it and only if every negative literal is false in both I1 and I_2 and the positive literal is false in at least one of the two subInterpretations. The amount of space required to store the set of tuples falsifying a clause in the cross interpretation can be greatly reduced by storing instead certain tuples from the subInterpretations and using these to construct tuples from the cross interpretation when (and if) needed. Suppose the tuples of values that make C false in I1, are a.,..,.a and those that make C false in 1~ are b ,...,b . Further, suppose the tuples that make all the negative literals of C _false but the positive literal true in I. are $c_1,.,.,c_n$, and those for I_2 are $d_1,,\ldots,d_m,$. Consider a tuple from

1. x 1_2 of the form a_1 x z. If z is one of the b_i's or 3's then by the above remarks a x z makes C false in the cross interpretation since the first coordinate falsifies every literal of C while the second coordinate satisfies at most just the positive literal. On the other hand, if z is not among the b!s or d's, then it satisfies a negative literal of c and io *"a* x 7 must satisfy C in I. x *1?"*^ ^ ^similar remark holds for values of the_ form z x b . Finally, a value o£ the _ form c x d must satisfy C since both c. and d satisfy the^positive literal of C^ Also, as jjbove^ any value of the form z x y where either z or y is not among the appropriate lists must satisfy C. Thus, the tuples of the cross inter-

preta_tion_that falsify C are a x bi, a_i x d_j , and c_i, x b_j for diffenent combinations of i and j. Thu* by saving, in addition to the values in the subinterpretations that falsify the clause, those values that falsify the negative literals but satistfy the positive literal one can significantly reduce the amount of storage required. To get an estimate of the kind of saving involved, consider again two interpretations of 10 elements each and a clause with three variables. There are 10 3-tuples of values In each domain. Suppose a third falsified the clause in each domain and another third falsified the negative literals but satisfied the positive literal. Then there would be 666 triples in each domain that need to be saved or 1332 altogether. in the cross interpretation there are 333*333*3=332667 triples of ordered pairs that falsify the clause. As before, if more smaller domains are u ed, the reduction is even more drastic. Moreover, efficient notations from set theory can be used to store such information. For example, if several tuples have coordinates in common, this can be indicated by condensing some of the coordinates to be sets. E.g., the six triples (a,b,c), (a,b,d), (a,b,e), (e,b,c), (e,b,d) and(e,b,e) could be represented by ({.a,e} ,b,[c,d,e}). That is, anything from the set [a,e] can be used as first coordinate, b as second coordinate and any of [c,d,e] for the third Such notations have proven useful in other contexts in theorem proving (see, for example, Auguston and Minker [1]).

Finally, the use of cross-products makes it easier for the user to "tailor" interpretations to emphasize particular clauses by giving the various clauses the appropriate properties in the coordinate interpretations. For example, consider the theorem of group theory that if x^2=e for every x the group is commutative and its standard representation as a set of clauses with the conclusion negated. This set contains the three units Pxxe, Pabc, and -Pbac. Suppose one would like to concentrate on Pabc and -Pbac and delay the entry of Pxxe into the search. Then in the cross product interpretation Pabc and -Pbac must be false while Pxxe must be true. Then Pabc must be made false in at least one of the interpretations In the cross-product while the atom Pbac must be true (I.e., -Pbac false) in all the interpretations in the cross product. Also, Pxxe must be true in all interpretations. It would be easier to devise two interpretations of, say, 4 or 5 elements each, that satisfy these conditions, especially the condition on Pxxe, than to verify them for a single interpretation of 20-25 elements. If the above conditions are satisfied and the two subinterpretations are themselves groups, then neither the axioms nor the clause Pxxe can be resolved together, and the refutation search must begin with either Pabc or with -pbac and another clause whose resolvent with -Pbac has a false instance in the cross-product. Indeed, the clause Pxxe would not be able to resolve with any clause C unless C and the resolvent both had false instances.

IV. Concluding Remarks

This paper has presented a new strategy, a strengthening of semantic resolution, which is complete for the class of Horn sets. This strategy is actually a combination of semantic and syntactic approaches—syntactic because the emphasis on positive units is totally independent of any intended meaning of the symbols; semantic because different interpretations supplied by the user will yield different search spaces. The results reviewed in Section 3 will prove most useful in actually implementing the new strategy, or indeed any semantic strategy, for programs that will deal with Horn sets. The time and storage savings that those results will provide may make it possible for the first time to construct truly semantic programs, that is programs which can use the information provided by the user about the intended meaning of the clauses as opposed to a meaning imposed by some syntactic property as in hyperresolution. There are some questions that are raised by these results. First, it may be possible to derive an efficient strategy from Lemma 3 and constrained-variable resolution. Also, it would be very interesting indeed to know if some version of Theorem 1 held in the non-Horn case for hyperresolution, which is the analogue of positive-unit resolution for Horn sets. Finally, it would be most useful to know how many of the results here carry over to paramodulation for equality.

References

1. Auguston, J. and Minker, J., "Clobal parallel unification for large question-answering systems," Technical Report TR-307(1974), University of Maryland.

2. Chang, C. L. and Lee, R.T.C., Symbolic Logic and Mechnical Theorem Proving, Academic Press, New York, 1973.

3. Henschen, L. and Wos, L., "Unit refutations and Horn sets," J. ACM 21,4 (Oct., 1974), 590-605.

4. Hermes, H., Enuroerabi1ity, Decidabi11ty, Computablity, Springer-Verlag, 1965.

5. Horn, A., "On sentences which are true of direct unions of algebras," J. Symbolic Logic 16 (March, 1951), 14-21.

6. Kuehner, D., "Some special purpose resolution systems," Machine Intelligence, Vol. 7, B. Meltzer and D. Michie, Eds., American Elsevier, New York, 1972, pp, 117-128.

7. McKinsey, J. C., "The decision problem for some classes of sentences without quantifiers," J. Symbolic Logic 8 (Sept., 1943), 61-76.