Unit Refutations and Horn Sets

L. HENSCHEN

Northwestern University, Evanston, Illinois

AND

L. WOS

Argonne National Laboratory, Argonne, Illinois

ABSTRACT. The key concepts for this automated theorem-proving paper are those of Horn set and strictly-unit refutation. A Horn set is a set of clauses such that none of its members contains more than one positive literal. A strictly-unit refutation is a proof by contradiction in which no step is justified by applying a rule of inference to a set of clauses all of which contain more than one literal. Horn sets occur in many fields of mathematics such as the theory of groups, rings, Moufang loops, and Henkin models. The usual translation into first-order predicate calculus of the axioms of these and many other fields yields a set of Horn clauses. The striking feature of the Horn property for finite sets of clauses is that its presence or absence can be determined by inspection. Thus, the determination of the applicability of the theorems and procedures of this paper is immediate.

In Theorem 1 it is proved that, if S is an unsatisfiable Horn set, there exists a strictly-unit refutation of S employing binary resolution alone, thus eliminating the need for factoring; moreover, one of the immediate ancestors of each step of the refutation is in fact a *positive* unit clause. A theorem similar to Theorem 1 for paramodulation-based inference systems is proven in Theorem 3 but with the inclusion of factoring as an inference rule. In Section 3 two reduction procedures are discussed. For the first, Chang's splitting, a rule is provided to guide both the choice of clauses and the way in which to split. The second reduction procedure enables one to refute a Horn set by refuting but one of a corresponding family of simpler subproblems.

KEY WORDS AND PHRASES: theorem-proving, resolution, paramodulation, factoring, first-order logic, Horn set

CR CATEGORIES: 3.60, 5.21

1. *Introduction*

One of the earliest strategies formulated and implemented to affect the proof search in automated theorem-proving programs was that of unit preference [16]. This strategy places extremely heavy emphasis on unit resolution [2, 16]. A unit resolution is one in which at least one of the clauses involved is a unit—contains but one literal. (Throughout this paper resolution means binary resolution, i.e. exactly one literal in each clause is unified or matched.) Many in the field of automated theorem-proving have devoted much effort and interest to the advantages, properties, and implementation of techniques which give strong preference and often exclusion to unit inference (Chang [2], Luckham [9],

Copyright © 1974, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

This work was performed under the auspices of the US Atomic Energy Commission and supported in part under NSF Grants GJ32717 and GU-3851.

Authors' addresses: L. Henschen, Computer Sciences Department, Northwestern University, Evanston, IL 60201; L. Wos, Argonne National Laboratory, Argonne, IL 60439.

Wos et al. [16]). This interest stems, in part, from the desire to produce shorter clauses (since proofs generally terminate when conflicting units are found), from the desire to avoid the proliferation of various literals, and from the desire to avoid the generation of both longer resolvents and a larger number of resolvents from a given pair of clauses.

Some procedures have been devised with the intention of yielding, for any set S' of clauses, a corresponding family S of sets S_i of clauses such that all S_i are unsatisfiable if and only if S' is, and such that there exists a unit refutation of each S_i if and only if S_i is unsatisfiable. (A unit refutation means that no inference rule is applied therein to a pair of nonunit clauses.) Two attempts at such a procedure are the tautology adjunction of Wos et al. [17] and the splitting theorem or transformation of Chang [3, 4]. For any such attempt, the difficult problem is determining whether or not the S_i can be refuted when all nonunit inferences are excluded. More generally, one would like to have conditions which, if satisfied, would guarantee that a given set of clauses can be refuted without recourse to nonunit inference. This paper gives in Theorem 1 a condition, namely, that of being a set of Horn clauses, under which a strictly-unit refutation (factoring is unnecessary) is guaranteed for unsatisfiable sets S. A Horn set or set of Horn clauses is a set in which no clause contains more than one positive literal. Theorem 1 shows, in fact, that a strictly-*positive*-unit refutation exists for such sets. Similar results are also proven for paramodulation-based inference systems. It is shown herein that the converse of Theorem 1 although true on the ground level for minimally unsatisfiable sets is not true in general. Theorem 2 shows that if S is an unsatisfiable set of Horn clauses, there exists an input refutation of S without factoring. Since, for finite sets S of clauses, one can decide by inspection whether or not S is a Horn set, Theorem 1 is quite useful for automated theorem-proving. The following remarks show not only in what way the theorem is of value but also illustrate an important difference between Theorem 1 and similar theorems concerning the existence of refutations possessing specific properties.

One of the most undesirable theoretical properties of automated theorem-proving programs, and of proof procedures in general, is that of semidecidability. Semidecidability means the following: given any (fixed) algorithm A and a program based on A, there (always) exists a formula F such that no amount of time will be sufficient for the program to ascertain whether or not F is a theorem. For the subset of formulas which are in fact theorems, there are, of course, many algorithms such that the corresponding program would, given enough time and memory, eventually find a proof—correctly ascertain theoremhood. A difficulty similar to that of the undecidability of theoremhood occurs in connection with, for example, the interesting theorem, Theorem 2, of Chang [2]. The theorem states that, for any given set S of clauses containing its unit factors, a unit refutation exists if and only if an input refutation exists. Now if a program starts with a set S of clauses and searches for a unit proof, no amount of time is sufficient to guarantee in general the removal of either of the two relevant uncertainties. First, no amount of time is sufficient to establish whether or not the formula corresponds to a theorem. Second, even if the formula is known to correspond to a theorem, no amount of time is sufficient to establish whether or not a unit refutation exists. In dealing with the second instance of undecidability, Theorem 1 has an important advantage, say, over Chang's theorem and the set of support theorem of Wos et al., for one can determine by inspection that Theorem 1 is applicable while such is not the case for the other two results. (The existence of an input proof is as undeterminable, in general, as the existence of a unit proof. Determining that $S - T$ is satisfiable is often as difficult as the seeking of a refutation of S.) Either in seeking new proofs of known theorems or in the testing of new features of a program, knowledge of the existence of a unit refutation is often quite valuable.

Another use to which Theorem 1 can be put is in connection with the splitting theorem of Chang [4]. According to Chang, splitting can materially improve proof search efficiency both with respect to time and memory. One would like to have a general rule to decide which clauses to split and a rule for deciding how to split them. One such rule would be: choose clauses and split them so that the resulting family of clause sets has the property that each member of the family is a Horn set. For each member of the family, one then need only search the unit section for a refutation.

Horn sets occur in many fields of mathematics such as elementary group theory, ring theory, the theory of Moufang loops, Henkin models, and Boolean algebras. Thus the results of this paper are relevant to a number of automated theorem-proving experiments.

2. *Definitions, Theorems, and Corollaries*

In this section we prove an inclusion relation between the class of unsatisfiable clauses which have the Horn property and the class of *strictly*-unit refutable by binary resolution sets of clauses. Since factoring or its equivalent is no longer required for the refutation completeness of binary resolution based programs for Horn sets, we have a sharpening of the work of Kuehner [8] in which proofs of Lemma 1 and Theorem 4 can be found. (The Kuehner paper was not yet available and was unknown to the authors at the time this research was completed.) Perhaps more important than the theoretical aspects of the sharpening of Lemma 1 are its consequences for implementation and for possible improvement in the efficiency of theorem-proving programs. In a similar way, Theorem 2 is stronger than the corresponding statement for vine or input refutations in that, for many authors, the latter concepts include explicit factoring or implicit factoring by permitting resolution to involve more than one literal from a clause.

Definition. A set S of clauses has the Horn property if every clause in S contains at most one positive literal. Such an S is called a Horn set. A clause containing at most one positive literal is a Horn clause. (See Horn [7]; Slagle [15] calls a clause $-L_1 \vee \ldots \vee -L_n \vee M$ with exactly one positive literal M an implication because it is equivalent to $(L_1 \,\&\ldots\&\, L_n) \to M$.)

Definition. A unit refutation $D_1, D_2, \ldots, D_n$ of the set S of clauses with respect to the inference system Ω is a refutation of S such that, if D_i is obtained (justified) by application of a rule from Ω other than factoring, then at least one of the parents of D_i is a unit clause. If at least one parent for each such D_i is a positive unit, the refutation is a positive-unit refutation. A set S is (positive-) unit refutable with respect to Ω if there exists a (positive-) unit refutation of S with respect to Ω. If factoring does not occur, then we have, respectively, a strictly-unit refutation or a strictly-positive-unit refutation.

Definition. An input refutation $D_1, D_2, \ldots, D_n$ of the set S of clauses with respect to the inference system Ω is a refutation of S such that, if D_i is obtained (justified) by application of a rule from Ω, then at least one of the parents of D_i is a member of S or a factor of a member of S. If factoring does not occur, we have a strictly-input refutation.

For finite sets of clauses, the property of being Horn is decidable and, in fact, is decidable by inspection. It will be seen in Section 3 that the existence of the simple test, inspection, is both significant and useful. When S itself is not Horn, there often exists a renaming of S which yields a set S' which is Horn. Since by the renaming theorem of Meltzer [10] S' is unsatisfiable if and only if S is, both the results of this section and the techniques of the next with the appropriate modifications can be applied to S' to yield information about S.

Definition. A negative clause is a clause containing only negative literals. A positive clause is a clause containing only positive literals. A mixed clause is a clause containing both positive and negative literals. The empty clause is, therefore, both a negative and positive clause.

The results of this section are not to be viewed as an endorsement of strategies emphasizing positive-unit refutation. For many problem domains such strategies would concentrate on the axioms of the system regardless of the theorem under consideration. In such a strategy, the negation of the theorem (usually a negative clause) would often serve the sole purpose of terminating the proof search, thereby decreasing rather than increasing program efficiency. The results and proofs herein are to provide the necessary background for the more important practical considerations, such as the techniques of Section 3, which may stem from them.

LEMMA 1. *Let Ω consist of the inference rules resolution and factoring. If S is an unsatisfiable Horn set, then there exists with respect to Ω (1) a positive-unit refutation of S, and (2) an input refutation of S.*

PROOF. Let S be an unsatisfiable Horn set. For the first part of the lemma, apply J. A. Robinson's P_1-deduction theorem [13] and note that the only positive clauses at any level are the positive units. For the second part of the lemma, apply the set-of-support strategy, choosing as set of support the subset T of S consisting of the negative clauses. In this case, all supported clauses on any level are negative so that the positive literal in any resolution must come from a clause in $S - T$.

The following example shows that Lemma 1 cannot be strengthened to guarantee the existence of a refutation D which is simultaneously (positive-) unit and input for an arbitrary Horn set.

$$S = \{r,\ -r\ q,\ -r\ s,\ -p\ -q\ -s,\ p\ -q\ -s\}.$$

The set of level 1 unit resolvents is $UR^1 = \{q, s\}$. The set of level 2 unit resolvents is $UR^2 = \{-p\ -s, p\ -s, -p\ -q, p\ -q\}$. Up to this point, all resolutions have been input resolutions. However, no new input resolvents can be generated by unit resolution. The reader can easily see how to complete a (positive-) unit refutation as well as an input refutation.

Lemma 1 can, however, be sharpened in a rather significant way; namely, it will be shown in Theorems 1 and 2 that the inference rule of factoring is not needed to refute unsatisfiable Horn sets. In order to prove this, we first prove a general lemma about Horn deductions, that is, deductions in which no clause contains more than one positive literal. For the proof of this lemma it should be remembered that in automated theorem-proving it is often necessary to view a deduction as a sequence of ordered pairs, (D_i, J_i), $i = 1, n$, where D_i is a clause and J_i the corresponding justification, rather than just the sequence $D_1, D_2, \ldots, D_n$. The justification J_i of D_i gives the immediate ancestor(s) of D_i and the inference rule by which D_i was inferred.

The J_i are needed, for example, to test for set of support or to test for the presence of factoring. To see the former, consider the deduction

Pa
Pb
$-Px\ Qy$
Qy.

Assume that Pa has T-support for some given T and that Pb does not. One justification of the inference producing the clause Qy results in the subdeduction having T-support while the other obvious justification yields a subdeduction without T-support. To see that the J_i are needed to test for the presence of factoring, consider the subdeduction

Pab
$-Pxy\ -Pab$
$-Pab$.

The third clause can be obtained by resolving the first two or by factoring the second.

LEMMA 2. *If D is a deduction with respect to Ω of the clause A, if Ω consists of the inference rules resolution and factoring, and if no clause in D contains more than one positive literal, then there exists a deduction E of a clause B such that E employs resolution alone and B subsumes A.*

PROOF. Because of the nature of this proof, it has been placed in Appendix A.

Of course, factoring cannot be eliminated in general. The difficulty is that some refutations involve resolutions where both literals of resolution are factored; this situation cannot occur when all the clauses are Horn.

THEOREM 1. *If S is an unsatisfiable Horn set, and if Ω' consists solely of resolution, then there exists a strictly-positive-unit refutation with respect to Ω'.*

PROOF. Let Ω' and S satisfy the hypotheses of this theorem. Let Ω consist of the inference rules, resolution and factoring. In the proof of Part 1 of Lemma 1, a positive-unit refutation D_1 with respect to Ω was shown to exist. Since all clauses deducible from a Horn set by any combination of factoring and resolving are themselves Horn clauses, all elements of D_1 are Horn clauses. Since D_1, the empty clause, and Ω all satisfy the hypotheses of Lemma 2, we can apply the procedure in the proof of that lemma. Let E be the deduction obtained by the procedure. In order to see that E is a strictly-positive-unit refutation, first note that none of the positive units in D_1 can arise by factoring since all clauses in S and all clauses deducible from S with the rules in Ω are Horn. There are two classes of resolutions that occur in E: those between pairs of clauses which subsume their counterpart in D_1, and those in subdeductions whose purpose is to remove extra literals. For the first class, since the elements of E can be seen to be Horn clauses, since the clause in E containing the positive literal of resolution subsumes its correspondent in D_1, and since the correspondent is a positive unit, these resolutions must be positive-unit resolutions. For the second class, the clauses containing the positive literal of resolution are, for a given subdeduction, identical and in fact all correspond to a clause in D_1 containing a positive literal of resolution. By the argument given for the first class, these clauses must all be positive units. So E is a strictly-positive-unit deduction. That E is a refutation follows from the fact that D_1 is a refutation, the only clause which subsumes the empty clause is the empty clause, and the conclusion of Lemma 2. The proof is complete.

THEOREM 2. *If S is an unsatisfiable Horn set and Ω' consists solely of resolution, then there exists a strictly-input refutation with respect to Ω'.*

PROOF. Let S and Ω' satisfy the hypotheses of this theorem, and let Ω consist of resolution and factoring. Lemma 1 can be applied to S and Ω to yield an input refutation D_2 with respect to Ω. All clauses in D_2 are Horn clauses since all clauses inferred with respect to Ω from a Horn set are themselves Horn. As in the proof of Lemma 1, D_2 has the additional valuable property that the only clauses therein containing positive literals are clauses in $S - T$ or factors of clauses in $S - T$. Because the hypotheses of Lemma 2 are satisfied, we can apply the procedure used for the proof of that lemma. The deduction E yielded by the procedure has in place of factors of input clauses the corresponding input clauses themselves. Each subdeduction in E, whose purpose is to remove superfluous literals in order to yield a clause subsuming its correspondent in D_2, utilizes a sequence of resolutions all involving a common clause. Each of these common clauses contains a positive literal. Since the only clauses in D_2 containing a positive literal are from $S - T$ or are factors of such, and since in E the factors in question have been replaced by the original clauses from $S - T$, the above-mentioned common clauses are from $S - T$. Thus the only clauses in E containing a positive literal are from $S - T$. Since all resolutions require such a clause, E is a strictly-input deduction. Since D_2 is a deduction of the empty clause, and since E is a deduction of a clause subsuming that clause, E is a refutation. Since factoring does not occur in E, the proof is complete.

Remark. Of course, if a Horn set contains no positive units, by Theorem 1 it is satisfiable.

For finite unsatisfiable sets S of clauses, Theorems 1 and 2 have a distinct advantage over similar results such as Chang's theorem [2] on the equivalence of input and unit proofs and the set-of-support theorem of Wos et al. [17]. For such an S the theorems respectively state:

1. If S is Horn, S is strictly-positive-unit refutable;
2. If S is input refutable and contains its unit factors, S is unit refutable; and
3. If $T \subseteq S$ such that $S - T$ is satisfiable, S is T-support refutable.

The advantage consists in the fact that in the first of these three the hypothesis is verifiable by inspection, while in the second and, to a lesser extent, the third verification of the hypothesis is often as difficult as the central problem of determining theoremhood. For example, the difficulty of establishing the existence of an input refutation is equal to that in general of establishing the existence of a unit refutation. Knowledge as to the satisfiability of $S - T$, on the other hand, may be lacking for new fields of mathematics under study or when $S - T$ corresponds to the data base in a question-answering problem.

It is perhaps natural to attempt to generalize the concept of Horn set to that of being Horn relative to an interpretation. A clause A would be Horn relative to an interpretation I if the intersection of each ground instance of A with I contains at most one element. Unfortunately, Theorem 1 no longer holds for this generalization since, for the following example, factoring is necessary to obtain the refutation. Let S consist of the two clauses $Pa, x\ Px, a$ and $-Pa, x\ -Px, a$, and let I consist of Pa, a.

Before stating Theorem 3 we remind the reader of the meaning of R-unsatisfiability, paramodulation, and functional reflexive axioms [18]. A set S of clauses is R-unsatisfiable if the set S', obtained by adjoining to S the clauses corresponding to the appropriate equality axioms for S, is unsatisfiable. Paramodulation is the inference rule for "built-in equality" yielding inferences in a single step which are equivalent to applying universal instantiation followed by equality substitution. The functional reflexive axioms for a set S are the unit clauses $Rf(x_1, x_2, \ldots, x_n), f(x_1, x_2, \ldots, x_n)$ for every n-ary function for $n \geq 1$ occurring in S. The unit factors of S are the unit clauses which can be obtained by factoring completely the clauses in S.

THEOREM 3. *If S is an R-unsatisfiable set of Horn clauses containing Rx, x and its functional reflexive axioms, and if Ω consists of the inference rules of paramodulation, resolution, and factoring, then there exists an input refutation D_1 and a unit refutation D_2.*

PROOF. Let S and Ω satisfy the hypotheses, and let T be the set of negative clauses of S. As in the proof of Part (2) of Lemma 1, every T-supported resolution is an input resolution. Similarly, since all positive equality literals occur in clauses in $S - T$, every T-supported paramodulation is an input paramodulation. To see that $S - T$ is R-satisfiable, consider a domain of one object and the interpretation over that domain consisting of all (positive) Herbrand atoms as the R-model. By the set-of-support theorem for Ω in Wos and Robinson [18], there is a T-supported refutation D_1 with respect to Ω. This refutation, by the remarks above, must be an input refutation. Since factoring is present in Ω, by Theorem 2 of Chang and Slagle [5], the existence of D_1 implies the existence of a unit refutation D_2 with respect to Ω, which completes the proof.

It is desirable to extend Theorem 3 as Lemma 1 was extended. The correspondents of Theorems 1 and 2 could be proved with the aid of the analogues of Lemma 2 and J. A. Robinson's P_1-deduction theorem for the inference system consisting of paramodulation, resolution, and factoring, if such analogues could be proved.

Theorems 1–3 give the major theoretical results of this paper. However, to illustrate some of the difficulty in attempting to prove results about Horn sets, we show that the converse of Theorem 1 for minimally unsatisfiable sets is only true in general in the ground case. Thus properties of Horn sets often do not lift.

Definition. A set S of clauses is minimally unsatisfiable if S is unsatisfiable and no proper subset of S is unsatisfiable.

THEOREM 4. *Let S be a set of clauses which is minimally unsatisfiable and ground. If there is a positive-unit refutation of S, then S is a Horn set.*

PROOF. The proof proceeds by induction on the number n of atoms occurring in S.

Case 1. $n = 1$. In this case, S consists of exactly two clauses, p and $-p$. The theorem is obviously true in this case.

Case 2. Assume by induction that the theorem is true for all i such that $1 \leq i \leq n$, and assume that S has exactly $n + 1$ atoms. If S has a positive-unit refutation P, then S must contain at least one positive unit, say u. Since S is minimally unsatisfiable, the clauses of S must be u, $-u\ B_1, \ldots, -u\ B_p$, $C_1, \ldots, C_q$, where all occurrences of u and $-u$ are shown explicitly. Let S' be the set of clauses $B_1, \ldots, B_p, C_1, \ldots, C_q$. Then S' is also unsatisfiable and has a positive-unit refutation as can be seen by the following. Consider the refutation P and the occurrences of u in P. Form the deduction P' from P by deleting the occurrences of the unit u and deleting the literal $-u$ from the remaining clauses of P. P' is a refutation of S'. Furthermore, it is easy to see that P' must also be a positive-unit refutation since each resolution step in P' also occurs in P. Finally, S' is minimally unsatisfiable. For, if some proper subset, say T', of S' is unsatisfiable then the set T formed by returning $-u$ to the appropriate clauses of T' and adding the unit u would be an unsatisfiable proper subset of S, contradicting the minimal unsatisfiability of S. Now by the induction hypothesis, S' is a Horn set. In particular then, each B_i is a Horn clause, and so $-u\ B_i$ is also a Horn clause. Since the unit clause u is Horn, S is a Horn set.

COROLLARY 1. *If S is a minimally unsatisfiable, unit refutable set of ground clauses, then there exists a set S' such that S' is a renaming of S and S' is a Horn set.*

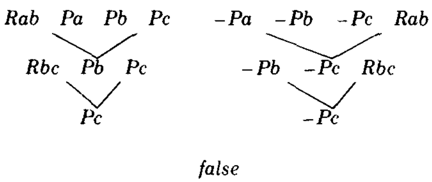
PROOF. An induction proof similar to the proof of Theorem 4 will suffice.

Remark. From Corollary 1 and Theorem 1, if a ground set S is unit refutable, there is a renaming which is positive-unit refutable.

Note that the induction step of the proof of Theorem 4 fails if P includes paramodulation steps. For, if u is an equality unit, it is not necessary for u to interact only with literals $-u$ in other clauses. Moreover, in the set of clauses inferable from S by paramodulation, the number of atoms may be preserved or increased. Consider, for example, the set of clauses, $\{Rab, Rbc, Pa, -Pc\}$, which contains four atoms. Upon paramodulating Rab into Pa we get Pb, an entirely new atom. That Corollary 1 does not hold for R-unsatisfiable ground sets and inference systems containing both paramodulation and resolution can be seen from the following example:

$$S = \{Rab, Rbc, Pa\ Pb\ Pc, -Pa\ -Pb\ -Pc\}.$$

S is minimally R-unsatisfiable and has the following unit refutation:



No renaming can transform S into a set of Horn clauses. In fact, when equality is present, one cannot arbitrarily rename literals without possibly violating the meaning of equality. The difficulty lies in the fact that different literals may express the same fact in the presence of equality. In the above example, it is asserted that $a = b = c$; thus the three literals Pa, Pb, and Pc are actually logically equivalent. Then it would be inconsistent with the meaning of equality to rename Pa to $-Pa$ without also renaming Pb and Pc to be negative. Thus the technique indicated in Kuehner [8] would not be applicable to sets of clauses involving equality.

We now show by example that neither of the conditions on S of Theorem 4 can be relaxed in general. Consider the following set S of clauses which is not minimally unsatisfiable: $\{p, q, -p\ -q, p\ q, -p\ q, p\ -q\}$. One can construct a positive-unit refutation from the first three clauses. However no renaming of S will be a Horn set. Thus in Theorem 4 we cannot drop the condition of minimal unsatisfiability.

For the other condition on S, that of being ground, consider the following example. Let S be the set of clauses

1. $-Qx\ -Qy\ Px\ Py$
2. $Qb\ -Pa$

3. Qa
4. $-Qb$

The following is a positive-unit refutation of S:

5. From 3 and 1 $-Qy\ Pa\ Py$
6. From 3 and 5 Pa
7. From 6 and 2 Qb

 7 and 4 are conflicting units.

This set of clauses is minimally unsatisfiable. However, no renaming of S is a Horn set.

The reason that properties concerning Horn sets do not in general lift is that two positive literals in a clause with variables may map onto the same literal in a ground instance of the clause. To see this suppose S is a minimally unsatisfiable non-Horn set of clauses with a positive-unit refutation P. The set of substitutions used in P often yields a minimally unsatisfiable set S' of ground instances of clauses in S. Let P' be the refutation of S' corresponding to P. P' must also be positive-unit, and, when S' is in fact minimally unsatisfiable, by Theorem 4 S' is Horn. Now the only way for a non-Horn clause C in S to have an instance C' which is Horn is for C to contain two or more positive literals that map onto the one positive literal of C'. For the above example the ground instances determined by P are

1. $-Qa\ Pa$
2. $Qb\ -Pa$
3. Qa
4. $-Qb$

This ground set is a Horn set. However, no renaming of the original set S is Horn. This kind of situation can prevent theorems about Horn sets from being lifted from ground sets to nonground sets.

3. *Horn Sets and Strategies*

Whenever introducing a new rule of inference, strategy, or problem domain restriction, A, it is generally profitable to study the compatibility of A with already existing strategies. In this connection we consider some of the more widely used strategies.

Luckham [9] discusses three refinements of resolution, R_1, R_2, R_3:

(1) R_1 is resolution with respect to an interpretation M; resolvents of A and B are generated only if one of A and B is not satisfied in M (Luckham uses the term model in place of interpretation);

(2) R_2 is resolution with merging; a resolvent C of A and B is generated only if (a) one of A and B is an input clause, or (b) C is a resolvent of $A\lambda$ and $B\beta$ and one of $A\lambda$ and $B\beta$ is a merge;

(3) R_3 is resolution relative to ancestry filter; a resolvent C of A and B is generated only if one of A and B is an input clause or is in the deduction tree of the other.

(The reader should take note of the fact that for Luckham resolution admits the possibility of unifying more than one literal from each clause, i.e. factoring is implicit.) Luckham showed that both $R_1 \cap R_2$ and $R_1 \cap R_3$ are not refutation complete. However, when the problem domain is restricted to the class of unsatisfiable Horn sets and the interpretation M is the set of (positive) Herbrand atoms, the intersection of all three strategies is refutation complete. To see this, first recall that it was shown in the proof of Theorem 2 that, if S is an unsatisfiable Horn set, there exists an input refutation of S in which every resolution step has a negative clause as one ancestor. Since each of the negative clauses in this input refutation is not satisfied in the interpretation M consisting of the (positive) Herbrand atoms, with respect to the given M, $R_1 \cap R_2 \cap R_3$ is refutation complete for the domain of Horn sets. Two comments are relevant for this combination of strategies. First, it is not claimed that, even for the restricted problem domain of Horn sets, the combination is in general refutation complete, for the interpretation cannot be arbitrarily chosen. Second, if all clauses in a subset T of a Horn set S are negative, any T-supported refutation of S using resolution alone will automatically satisfy the conditions for the interpretation M and strategy combination of R_1, R_2, and R_3.

We now turn our attention to the main topic of this section, unit resolution. In this section, binary resolution will be the main inference rule underlying the discussion. Theorems 1 and 2 obviate the need to consider factoring for Horn sets.

Section 2 gives a sufficient condition for the existence of a positive-unit refutation—a condition whose presence can, for finite sets, be tested for by inspection. For any Horn set S, the subset T of all positive units is a valid set of support since, by definition, no clause in $S - T$ can be a positive clause. Now if positive-unit resolution is used, the effect is stronger than that of just straightforward set of support corresponding to the above choice of T. It has been noted that set of support is essentially a level 1 strategy, that is, the sharp reduction in clause generation is at level 1. That this in turn leads to further reduction at higher levels is not denied, but, since the retained level 1 clauses are all T-supported and therefore allowed to resolve with all clauses, the effect beyond level 1 is relatively less dramatic. If A is a mixed clause with T-support on level 1, then resolution of A with all clauses B must be attempted in the usual set-of-support strategy, while, with unit resolution, B must be a unit clause. If A is a negative clause with T-support, with the usual set-of-support strategy resolution is attempted for A and all B, while, with positive-unit resolution, B must be a positive unit clause. So, a level 1 clause may be prevented from resolving with many other clauses even though it has T-support. Thus positive-unit resolution is a restriction of T-supported resolution, where T is the set of positive unit clauses. The effect of this restricted set of support is equally felt at all levels and not just at level 1. The authors are not expressing any opinion concerning the relative merit of positive-unit resolution versus the usual set of support approach with regard to program efficiency. Actual experimentation with a theorem-proving program is the only way to see whether a useful combination of the two strategies exists.

Although the positive-unit T-support strategy for the above choice of T is refutation complete for the problem domain of Horn sets, the replacement of T by a proper subset of itself may result in incompleteness. For example, let $S = \{-p\ q\ -u,\ -p\ -q\ -u,\ p\ -r,\ r,\ u\}$ and let T be replaced by $T' = \{u\}$. S is unsatisfiable while $S - T'$ is satisfiable. There is, of course, a T'-supported refutation, but it requires the resolution of two nonunit clauses. With the added restriction that resolutions must be positive-unit resolutions, the only T'-supported clauses deducible are $-p\ q$ and $-p\ -q$.

For sets that do not satisfy the Horn property, one can often apply the splitting techniques of Chang to derive subproblems which do satisfy the Horn property. In splitting one starts with a set of clauses $S \cup \{A\}$, where $A = C \cup D$. From this problem, two subproblems are generated—$S \cup \{C\}$ and $S \cup \{D\}$. If refutations for both of these subproblems can be derived and in a consistent way, then one can show that the original problem also has a refutation (see Chang [4]). For the refutations to be consistent the substitutions that are applied to the variables that are common to C and D must themselves be unifiable. The lack of consistency in the refutations of the subproblems and/or the lack of the carryover of unsatisfiability to the subproblems is generally traceable to the requirement of more than one ground instance of the split clause for a refutation of the original problem. This situation can lead to the necessity of an additional instance of the split clause being present in one or both of the subproblems, and hence possible further splitting. The original goal of splitting was to eliminate long clauses since these may take many steps to reduce and also yield many intermediate clauses which pollute the search space. However, our work suggests that an alternate goal in splitting is to transform a set of non-Horn clauses into subsets of clauses which are Horn. For example, if S contains one clause C with two positive literals and no other clause contains more than one positive literal, then one could split C into two subclauses each containing one positive literal. Then each of the two subproblems would satisfy the Horn condition. In this way a problem which may not have a unit refutation can be transformed into subproblems each of which may have a positive-unit refutation. The possibility of such a reduction gives a rule, previously lacking, for deciding which clauses to split and how to split them. Since many theorem-proving programs do rely solely on unit resolution, this technique may prove valuable.

Example. If a subgroup H of a group G has index 2, then H is a normal subgroup, i.e. $xyx^{-1} \in H$ for any $x \in G$ and $y \in H$. The authors know of no unit refutation for this theorem. However, by splitting into Horn subproblems we obtain a problem set, each member of which has a positive-unit refutation. (Since this example is hand computed and is only intended to be illustrative, the authors make no claim about program efficiency. Only a number of experiments with theorem-proving programs can ascertain the value of any technique.) The clauses for the negation of this theorem are in Appendix B. If we rename the predicate O to $-O$, the only non-Horn clause is the correspondent of 23, $-Ox\ -Oy\ -Px,y,z\ Oz$. (Note: We do not actually perform the renaming in the following. However, if we had done the renaming, then the correspondents of Ob and $Ox\,Oy\,Oi(x, y)$, for example, would be negative while the correspondent of $-Od$ would be a positive unit.) The non-Horn clause is split into two clauses $-Ox\ -Px,y,z\ Oz$ and $-Oy$. The first of these yields the following positive-unit refutation relative to the renaming.

23′	split portion of 23	$-Ox\ -Px,y,z\ Oz$
31	29 and 23′	$-Oa\ Od$
32	30 and 31	$-Oa$
33	32 and 23′	$-Ox\ -Px,y,a$
34	2 and 33	$-Oe$
	34 and 21 are conflicting units.	

Note again that the clauses $-Oa$, $-Oe$, and $-Od$ correspond to positive units when O is renamed to $-O$. In the above refutation, the two constants c and a were used in 23′ for y. If we were using constrained variables as in Chang [3], we would now have to find refutations of $-Oy$ that were compatible with each of these substitutions. But, clearly the only such refutations will correspond to refutations of $-Oc$ and $-Oa$, respectively. Thus, we must find refutations of each of these. For $-Oc$ it is necessary to again include a copy of the split clause. This time 23 is split into $-Oy\ -Px,y,z\ Oz$ and $-Ox$ leading to the following refutation:

23″	split portion of 23	$-Oy\ -Px,y,z\ Oz$
31	subcase hypothesis	$-Oc$
32	28 and 23″	$-Og(a)\ Oc$
33	31 and 32	$-Og(a)$
34	33 and 22	$-Oa$
35	30 and 26	$Ox\ Px,i(x,d),d$
36	34 and 35	$Pa,i(a,d),d$
37	29 and 19	$-Pa,x,d\ Rx,c$
38	36 and 37	$Ri(a,d),c$
39	31 and 24	$-Ox\ -Rx,c$
40	38 and 39	$-Oi(a,d)$
41	30 and 25	$Ox\ Oi(x,d)$
42	34 and 41	$Oi\ (a,d)$
	42 and 40 are conflicting units.	

In this refutation, 23″ was used once with b being substituted for x. Thus, we would have to find a refutation for the subcase $-Ob$ of the subcase $-Oc$. This sub-subcase is trivial since axiom 27 is the unit Ob. The next subcase is the refutation of $-Oa$. This case also requires a copy of 23, which is split into $-Oy\ -Px,y,z\ Oz$ and $-Ox$.

23″	split portion of 23	$-Oy\ -Px,y,z\ Oz$
31	subcase hypothesis	$-Oa$

32	30 and 26	$Ox\ Px,i(x,d),d$
33	31 and 32	$Pa,i(a,d),d$
34	29 and 19	$-Pa,x,d\ Rx,c$
35	33 and 34	$Ri(a,d),c$
36	28 and 6	$-Px,b,u\ -Pu,g(a),w\ Px,c,w$
38	4 and 36	$-Pe,g(a),x\ Pg(b),c,x$
39	2 and 38	$Pg(b),c,g(a)$
40	39 and 23″	$-Oc\ Og(a)$
41	31 and 24	$-Ox\ -Rx,a$
42	18 and 41	$-Og(g(a))$
43	42 and 22	$-Og(a)$
44	43 and 40	$-Oc$
45	44 and 24	$-Ox\ -Rx,c$
46	35 and 45	$-Oi(a,d)$
47	46 and 25	$Oa\ Od$
48	31 and 47	Od
	30 and 48 are conflicting units.	

In this refutation, 23″ was used with $g(b)$ being substituted for x. Thus, we have the final sub-subcase for the two units $-Oa$ and $-Og(b)$. This has the following refutation:

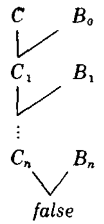
31	subcase hypothesis	$-Oa$
32	subcase hypothesis	$-Og(b)$
33	32 and 22	$-Ob$
	33 and 27 are conflicting units.	

An examination of the $-Oc$ and $-Oa$ cases above indicates that use of common subgoals from the different subcases may improve performance of a program. For example, in the $-Oc$ case the clause $-Oa$ is generated. When in the $-Oa$ case $-Oc$ is generated, a program could recognize that this subproblem had, in effect, already been solved.

The reader may notice that in each subproblem only one clause which would become negative by renaming O to $-O$ was required for each refutation. This is a general property of unsatisfiable Horn sets and leads to a different type of reduction technique. In the splitting reduction, one starts with a problem S and produces a set of subproblems $S_1, \ldots, S_k$ *each* of which must be solved. In addition, the solutions to the subproblems must be compatible. We now give a method by which some Horn sets may be reduced to a set of subproblems in such a way that the original set is unsatisfiable if and only if at least one of the subproblems is unsatisfiable. We first prove the following theorem.

THEOREM 5. *Let S be an (R-) unsatisfiable set of Horn clauses (containing its unit factors). If S contains more than one negative clause, S is not minimally (R-) unsatisfiable.*

PROOF. Let T be the set of negative clauses of S. By previous theorems, there is a T-supported input refutation



where each B_i is an input clause or a factor of an input clause and $C \in T$ or C is a factor of a clause in T. Now the first positive literal of resolution or paramodulation is contained in B_0 and is not present in C_1. Since all other literals in B_0 and C are negative, C_1 is also negative. Similarly, each C_i is negative. Since both resolution and paramodulation require a positive literal, each B_i must come from $S - T$. Thus no clause of T other than C, or the clause of which C is a factor, is necessary to complete a refutation, which completes the proof.

Theorem 5 leads to the following reduction technique. Let $S = S' \cup \{C_1, \ldots, C_k\}$, where the C_i, $1 \leq i \leq k$, are the negative clauses of S. The reduction produces k subproblems $S' \cup \{C_i\}$, $1 \leq i \leq k$. By Theorem 5, if S is (R-) unsatisfiable, then one of the subproblems is (R-) unsatisfiable. Conversely, since each subproblem is contained in the original set of clauses, the (R-) unsatisfiability of any one of them implies the (R-) unsatisfiability of S. Thus in applying this technique, it is necessary to refute only one of the subproblems. This technique would no doubt be more valuable for programs which do not emphasize the role of positive unit clauses.

Of course, the two techniques for reduction can be used together. One may start with a non-Horn set of clauses and apply the first reduction technique to produce Horn subproblems. Then the technique based on Theorem 5 for negative clauses can be applied to each individual subproblem.

We make a final comment concerning the emphasis on units. By the above theorem, only one negative clause is needed for a positive-unit refutation of a set of Horn clauses. Those refutations not employing paramodulation give rise to corresponding refutations based on hyperresolution. Each hyperresolvent, except the empty clause, is a positive unit obtained by starting with a mixed clause and stripping away the negative literals with positive units. Thus the emphasis is on positive units, and not just units. Now if units convey relatively more information than clauses with more than one literal, then, intuitively, positive units will convey even more useful information than just units. For suppose a property (or predicate) P is defined; in general it is more important or useful to know that certain objects satisfy this property than to know that they do not satisfy the property. When paramodulation is used in addition to resolution, positive equality units play a special role.

4. *An Additional Remark*

The undecidability of theoremhood arises from the possibility of the generation of an infinite sequence of clauses. There are essentially two ways in which resolution can produce an unending sequence of clauses, by producing clauses with increasing numbers of literals and by producing clauses with longer terms. This follows from the fact that, given a bound on the number of literals in a clause and a bound on the length of the terms, there are only a finite number of clauses (except for alphabetic variants) deducible by resolution from a given finite set of clauses. Now consider the class of Horn sets. The theorems of Section 2 allow the elimination of the first of the above problems for this class. For neither unit resolution nor unit paramodulation can produce a clause longer than the longest ancestor. While it is well known that the satisfiability of Horn sets is undecidable (see Reynolds [11] or Hermes [6]), the above remarks naturally raise the question of whether or not the second problem, term length, can also in some way be at least controlled or analyzed. The existence of a very special kind of refutation for any given unsatisfiable Horn set, namely a positive-unit refutation without factoring, provides one possible means of analysis of term structure needed for a refutation. Two examples will illustrate the kind of process we have in mind.

Example 1. Consider the set of clauses

Pa	$R_1\, g(g(a))$	$-Px\ -R_1x$
$-Px\ Q_1g(x)$	$-R_1x\ R_2g(x)$	
$-Q_1x\ Pg(g(x))$	$-R_2x\ R_1g(x)$	

From the clauses on the left, we can produce units of the form $Pg^{3n}(a)$ for any $n \geq 0$ while the clauses on the right produce units of the form $R_1g^{2m}(a)$ for arbitrary $m \geq 1$.

For these to resolve with $-Px\ -R_1x$ to produce the empty clause, we must have $3n = 2m$ for some n and m. We do have equality for $n = 2$ and $m = 3$ and so we know that there is a refutation without actually performing all the resolutions.

The idea of the analysis is this: We know the deductions start with positive units and produce positive units by peeling off the negative literals from mixed clauses. These new positive units will have different term structures. They can be used again to produce still newer positive units, but with essentially the same mixed clauses as before. Thus, while the term structure is changing, there will be a definite pattern to the change. By identifying each of these patterns, we can possibly determine if a set of positive units will ever have terms unifiable with all the terms of all of the negative literals of some negative clause. The following example shows how such an analysis can be used to establish satisfiability:

Example 2.

Pa	$R_1g(a)$	$-Px\ -R_1x$
$-Px\ Qg(x)$	$-R_1x\ R_2g(x)$	
$-Qx\ Pg(x)$	$-R_2x\ R_1g(x)$	

The clauses on the left produce units of the form $Pg^{2n}(a)$ while those on the right produce units of the form $R_1g^{2m+1}(a)$. Thus the empty clause can never be produced. It should be pointed out that we did *not* give a bound on the term length.

Of course, these examples are very simple. In more complex examples the patterns would be much more numerous and complex. In that event, one might be simply forced to submit the problem to a theorem-proving program. Moreover, when the splitting technique of Chang is used to produce Horn subproblems which can be analyzed as above, it may still be necessary to generate the refutations in order to examine their consistency. In such cases, the kind of analysis above may be useful as a heuristic in guiding the theorem-proving process.

5. *Summary*

For a finite set S of clauses, a quick perusal of the clauses therein will establish either the presence or the absence of the Horn property—that no clause contain more than one positive literal. If S is not a Horn set, with very little additional effort one can examine all of the predicate renamings in search of one which would map S into a Horn set. If S is or a renaming of S yields a Horn set, the theorems of Section 2 and the techniques of Sections 3 and 4 can be applied. Such is often the case since the clauses corresponding to the axioms of a number of fields of mathematics are Horn clauses. For example, the axioms of groups, rings, Henkin models, and Boolean algebras under the usual translation into first-order predicate calculus will have as correspondents sets of Horn clauses. Many theorems of mathematics can, therefore, be proven in the unit section of a theorem-proving program. Theorem 1 establishes that for Horn sets, positive-unit binary resolution without the need of factoring is sufficient for refutation completeness. Theorem 3 guarantees the existence of a unit refutation for any R-unsatisfiable Horn set when the inference system consists of paramodulation, binary resolution, and factoring.

In Section 3 two reduction procedures are discussed, the splitting procedure of Chang and the proper subset reduction procedure originated by the authors. The first procedure is provided with a rule, previously lacking, to guide the choice of clauses to split and to direct the splitting itself. The second procedure enables one to reduce a given problem, when the corresponding set of clauses is Horn, to a set of simpler subproblems in such a way that one need refute but one of the subproblems to establish unsatisfiability of the original problem.

Since, for many applications and experiments in automated theorem-proving, the corresponding set of clauses is itself Horn or is reducible to a set of Horn clauses, and since the Horn property can be tested for so simply, the results of this paper should provide valuable information and techniques for the field of automated theorem-proving.

Appendix A

LEMMA 2. *If D is a deduction with respect to Ω of a clause A, if Ω consists of the inference rules resolution and factoring, and if no clause in D contains more than one positive literal, then there exists a deduction E of a clause B such that E employs resolution alone and B subsumes A.*

PROOF. Assume that D, A, and Ω satisfy the hypotheses. If factoring is not present in D, then let $E = D$ and $B = A$. Otherwise let $D = D_1, D_2, \ldots, D_n$ and consider the following inductive construction of a deduction E:

1. Set $k = 1$;
2. If the justification of D_k is that it is an input clause, let $E_k = D_k$; the literal in E_k corresponding to a literal M in D_k is that same literal M; go to step 5;
3. If the justification of D_k is that it is a factor of D_j with $j < k$, then $E_k = E_j$; the set of literals in E_k corresponding to a literal L in D_k is the set $M = M_1 \cup \ldots \cup M_m$ of literals in E_k ($= E_j$) such that M_i is the set of literals in E_j corresponding to a literal $L_i \in D_j$ and L_i is mapped onto L by θ where $D_j\theta = D_k$; go to step 5;
4. If the justification of D_k is that it is a resolvent of D_i and D_j with $i, j < k$, then possibly perform a sequence of resolutions; assume without loss of generality that the negative literal of resolution M occurs in D_i; let $M' = \{M_1, \ldots, M_m\}$ be the set of literals in E_i corresponding to M; if M' is empty, let $E_k = E_i$; then the correspondence of literals between E_k and D_k is the same as between E_i and D_i; otherwise let K be the positive literal of resolution in D_j and K' the corresponding literal in E_j (there can be only one); let F_1 be the resolvent of E_i and E_j on M_1 and K', F_2 the resolvent of F_1 and E_j on the descendant of M_2 and K', $\ldots$, F_p the resolvent of F_{p-1} and E_j on the descendant of M_m and K'; E_k is F_p; the set of literals in E_k corresponding to a literal L in D_k are those literals descended from $L_0 \cup L_1 \cup \ldots \cup L_p$ where L_0 is the set of literals in E_i corresponding to L' in D_i and L' is mapped onto L in the resolution if D_i and D_j and for each q, $1 \leq q \leq p$, L_q is the set of literals in E_j corresponding to a literal L'' in D_j where L'' is mapped onto L in the resolution of D_i and D_j; there are p sets $L_1, \ldots, L_p$ because there are p resolutions each using E_j, and thus, each possibly introducing alphabetic variants of the literals in question; go to step 5;
5. If $k = n$, stop; otherwise set $k = k + 1$ and go to step 2.

In review, if D_k is an input clause, then E_k is that same input clause. If D_k is a factor of D_j, E_k is E_j. (While E_j may or may not equal D_j, E_k clearly will not equal D_k.) If D_k is a resolvent of D_i and D_j, the corresponding E_i and E_j are resolved to produce, depending on the number of literals in E_i corresponding to the negative literal of resolution in D_i, a one-step subdeduction or to begin a p-step subdeduction with $p > 1$. The extra literals may be present (i.e. $p > 1$) because a factor step was eliminated or because at an earlier stage another resolution in D required more than one corresponding resolution in E. Clearly, E does not involve the inference rule of factoring. Also, as is the case in all deductions involving only resolution starting with Horn clauses, no E_k contains more than one positive literal. In fact, E_k contains 1 or 0 positive literals depending on whether D_k contains 1 or 0 positive literals, respectively. Finally, note that the sets of literals in E_k corresponding to two distinct literals in D_k are disjoint.

Lemma 2 will now be proved by showing that the conclusion holds for each E_k and D_k. When $k = 1$, step 2 must apply; thus $E_1 = D_1$ and the conclusion holds trivially. For $k > 1$, assume the result holds for $1 \leq i \leq k - 1$. If step 2 or step 3 applies or if step 4 applies and no resolution is produced, the result is again trivial. So assume D_k is the resolvent of D_i and D_j and E contains one or more corresponding resolutions. Let the negative literal of resolution in D be M occurring in D_i and the positive literal of resolution in D be K occurring in D_j. Then $D_k = ((D_i - M) \cup (D_j - K))\theta$ where θ is the MGU of M and $-K$. By the induction assumption there exist (most general) substitutions α_i and α_j such that $E_i\alpha_i \subseteq D_i$ and $E_j\alpha_j \subseteq D_j$. Let M' be the set of literals in E_i corresponding to M and K' the literal in E_j corresponding to K. Let $E_{j_1}, \ldots, E_{j_p}$ be the p copies of E_j with variables separated used for the p resolutions in E with K'_m the copy of K' in E_{j_m}. Let α_{j_m} be the substitution corresponding to α_j such that $E_{j_m}\alpha_{j_m} \subseteq D_j$. Then $\beta = \alpha_i \cup (\bigcup_{m=1}^{p} \alpha_{j_m})$ is a most general substitution such that $E_i\beta \subseteq D_i$ and $E_{j_m}\beta \subseteq D_j$, $1 \leq m \leq p$. The p resolutions given in step 4 eliminate all the literals in M'. Clearly $\beta\theta$ is a unifier of the set of atoms occurring in M' and the atoms K'_m so that the sequence of resolutions prescribed in step 4 is admissible. Let δ be an MGU of this set so that $\beta\theta = \delta\lambda$ for some λ. The clause E_k obtained by the sequence of resolutions satisfies

$$E_k \subseteq ((E_i - M') \cup (\bigcup_{m=1}^{p} (E_{j_m} - K'_m)))\delta.$$

Then,

$$E_k\lambda \subseteq ((E_i - M') \cup (\bigcup_{m=1}^{p} (E_{j_m} - K'_m)))\delta\lambda$$

$$= ((E_i - M') \cup (\bigcup_{m=1}^{p} (E_{j_m} - K'_m)))\beta\theta \subseteq ((D_i - M) \cup (D_j - K))\theta = D_k.$$

By induction, the lemma is proved.

It would appear that the conclusion could be strengthened to read "$B\lambda = A$ for some substitution" by sharpening the procedure in step 4.

Appendix B

The following is the list of clauses expressing the negation of the theorem in the example in Section 3.

THEOREM. *If a subgroup H of a group G has index 2, then $xyx^{-1} \in H$ for any $x \in G$ and $y \in H$.*

The clauses for the negation are (R is the equality predicate):

1.	$Pxyf(xy)$	(closure)
2.	$Pexx$	(left identity)
3.	$Pxex$	(right identity)
4.	$Pg(x)xe$	(left inverse)
5.	$Pxg(x)e$	(right inverse)
6.	$-Pxyu\ -Pyzv\ -Puzw\ Pxvw$	(associativity)
7.	$-Pxyu\ -Pyzv\ -Pxvw\ Puzw$	
8.	Rxx	(reflexivity of =)
9.	$-Rxy\ Ryx$	(symmetry of =)
10.	$-Rxy\ -Ryz\ Rxz$	(transitivity of =)
11.	$-Pxyu\ -Pxyv\ Ruv$	(product well defined)
12.	$-Ruv\ -Pxyu\ Pxyv$	(substitution for =)
13.	$-Ruv\ -Pxuy\ Pxvy$	
14.	$-Ruv\ -Puxy\ Pvxy$	
15.	$-Ruv\ Rf(xu)f(xv)$	
16.	$-Ruv\ Rf(uy)f(vy)$	
17.	$-Ruv\ Rg(u)g(v)$	
18.	$Rg(g(x))x$	
19.	$-Pxuy\ -Pxvy\ Ruv$	(left cancellation)
20.	$-Puxy\ -Pvxy\ Ruv$	(right cancellation)
21.	Oe	
22.	$-Ox\ Og(x)$	
23.	$-Ox\ -Oy\ -Pxyz\ Oz$	
24.	$-Ox\ -Rxy\ Oy$	
	$-Ruv\ Ri(xu)i(xv)$	
	$-Ruv\ Ri(ux)i(vx)$	
25.	$Ox\ Oy\ Oi(xy)$	(subgroup has index 2)
26.	$Ox\ Oy\ Pxi(xy)y$	
27.	Ob	
28.	$Pbg(a)c$	
29.	$Pacd$	
30.	$-Od$	

REFERENCES

(Note. References [1, 12, 14] are not cited in the text.)

1. Allen, J., and Luckham, D. An interactive theorem-proving program. In *Machine Intelligence, Vol. 5*, B. Meltzer and D. Michie, Eds., American Elsevier, New York, 1970, pp. 321–336.
2. Chang, C. L. The unit proof and the input proof in theorem proving. *J. ACM 17*, 4 (Oct. 1970), 698–707.
3. Chang, C. L. Theorem proving with variable-constrained resolution. *Inform. Sci. 4* (1972). 217–231.
4. Chang, C. L. The decomposition principle for theorem proving systems. Proc. Tenth Annual Allerton Conference on Circuit and System Theory, U. of Illinois, Oct. 1972, pp. 20–28.
5. Chang, C. L., and Slagle, J. R. Completeness of linear refutation for theories with equality. *J. ACM 18*, 1 (Jan. 1971), 126–136.
6. Hermes, H. *Enumerability, Decidability, Countability*. Springer-Verlag, New York, 1965.
7. Horn, A. On sentences which are true of direct unions of algebras. *J. Symbolic Logic 16* (1951), 14–21.
8. Kuehner, D. Some special purpose resolution systems. In *Machine Intelligence, Vol. 7*, B. Meltzer and D. Michie, Eds., American Elsevier, New York, 1972, pp. 117–128.
9. Luckham, D. Refinement theorems in resolution theory. Proc. IRIA Symposium on Automatic Demonstration, Springer-Verlag, New York, 1970, pp. 163–190.
10. Meltzer, B. Theorem proving for computers: Some results on resolution and renaming. *Computer J. 8* (1966), 341–343.
11. Reynolds, J. C. Transformational systems and the algebraic structure of atomic formulas. In *Machine Intelligence, Vol. 5*, B. Meltzer and D. Michie, Eds., American Elsevier, New York, 1970, pp. 135–152.
12. Robinson, G., and Wos, L. Paramodulation in first-order theories with equality. In *Machine Intelligence, Vol. 4*, B. Meltzer and D. Michie, Eds., American Elsevier, New York, 1969, pp. 135–150.
13. Robinson, J. A. Automatic deduction with hyper-resolution. *Internat. J. Computer Math. 1* (1965), 227–234.
14. Slagle, J. R. Heuristic search programs. In *Theoretical Approaches to Non-Numerical Problem Solving*, R. B. Banerji and M. D. Mesarovic, Eds., Springer-Verlag, New York, 1970, pp. 246–273.
15. Slagle, J. R., and Koniver, D. Finding resolution proofs using duplicate goals in AND/OR trees. *Inform. Sci. 4* (1971), 313–342.
16. Wos, L., Carson, D. F., and Robinson, G. A. The unit preference strategy in theorem proving. Proc. AFIPS 1964 FJCC, Vol. 26, Pt. 1, Spartan Books, New York, pp. 615–621.
17. Wos, L., Robinson, G. A., and Carson, D. F. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM 12*, 4 (Oct. 1965), 536–541.
18. Wos, L., and Robinson, G. Paramodulation and set of support. Proc. IRIA Symposium on Automatic Demonstration, Springer-Verlag, New York, 1968, pp. 276–310.

RECEIVED FEBRUARY 1973