Extended Horn Sets in Propositional Logic

V. CHANDRU

Purdue University, West Lafayette, Indiana

AND

J. N. HOOKER

Carnegie Mellon University, Pittsburgh, Pennsylvania

Abstract. The class of Horn clause sets in propositional logic is extended to a larger class for which the satisfiability problem can still be solved by unit resolution in linear time. It is shown that to every arborescence there corresponds a family of extended Horn sets, where ordinary Horn sets correspond to stars with a root at the center. These results derive from a theorem of Chandrasekaran that characterizes when an integer solution of a system of inequalities can be found by rounding a real solution in a certain way. A linear-time procedure is provided for identifying "hidden" extended Horn sets (extended Horn but for complementation of variables) that correspond to a specified arborescence. Finally, a way to interpret extended Horn sets in applications is suggested.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*computational logic, mechanical theorem proving*; G.1.6 [**Numerical Analysis**]: Optimization—*integer programming*; H.2.1 [**Database Management**]: Logical design; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*deduction, resolution*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Horn clauses, propositional logic

1. *Introduction*

Horn clauses are widely used in logical programming languages and knowledge-based systems because inference is easy for them but generally quite hard for larger classes of formulas. Unit resolution solves the inference (or satisfiability) problem for Horn clauses in propositional logic in time that is linear in the number of literals [12, 18], whereas the general satisfiability problem for propositional logic is NP-complete [9].

The research of V. Chandru was supported in part by Office of Naval Research grant N00014-86-K-0689 and by National Science Foundation grant DMC 88-07550.

The research of J. H. Hooker was supported in part by U.S. Air Force Office of Scientific Research grant AFOSR-87-0292.

Authors' addresses: V. Chandru, School of Industrial Engineering, Purdue University, West Lafayette, IN 47907; J. N. Hooker, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213.

It would obviously be desirable to extend Horn clauses to a larger class of propositions for which the inference problem remains easy. Yamasaki and Doshita [21] identified one such class, for which the inference problem can be solved in cubic time; Arvind and Biswas [3] later found a quadratic time algorithm for the class. Whereas Horn clauses have at most one positive literal, Yamasaki and Doshita [21] consider sets of clauses that may have more than one positive literal, provided the positive literals are "nested" in a certain way. Gallo and Scutellà [13] generalized the nested positives idea to obtain a recursively-defined hierarchy of problems $\Gamma_0, \Gamma_1, \ldots$, where Γ_0 is the class of Horn problems, and Γ_1 the class defined by Yamasaki and Doshita. The problems in Γ_k are soluble in $O(Ln^k)$ time, where L is the number of literals and n the number of atomic propositions. It is unclear, however, how these classes of propositions might be more useful in practice than ordinary Horn clauses.

We show below that there is another generalization of Horn clauses for which inference is not only as easy as for Horn clauses but can be accomplished in the same way—by a linear-time unit resolution algorithm. We call sets of such clauses "extended Horn." A set of clauses is extended Horn if its atomic propositions correspond to the arcs of some rooted arborescence (i.e., rooted directed tree in which all arcs are directed away from the root) in such a way that each clause in the set describes an "extended star-chain" flow pattern on the arborescence. That is, if a positive (negative) literal indicates a forward (backward) unit flow on the corresponding arc, then the resulting flow pattern consists of chains each of which carries a unidirectional unit flow into the root, plus at most one such chain that is unconnected to the root. Ordinary Horn sets correspond to arborescences that are stars (i.e., all arcs incident to a central root), so that extended Horn sets represent a substantial generalization of ordinary Horn sets. We also show that the inference problem for "hidden" extended Horn sets, which are sets that become extended Horn when one or more atomic propositions are replaced by their negations, can likewise be solved by unit resolution, thus further enlarging the class of easy inference problems. Finally, we characterize extended Horn sets of rules and suggest one way that they might be interpreted and used in applications for which ordinary Horn sets are inadequate.

We were led to a discovery of extended Horn sets by a theorem of Chandrasekaran [6]. The theorem characterizes sets of linear inequalities for which a 0–1 solution can always be found (if one exists) by rounding a real solution, which can in turn be found by linear programming. We show that a set of inequalities with coefficients in $\{0, 1, -1\}$ that corresponds to an arboresence (or any rooted, directed tree) in the way noted above satisfies the conditions of Chandrasekaran's theorem. In this way, we give a network characterization of a family of 0–1 problems that can be solved by linear programming and rounding. Extended Horn sets (explicit or hidden) are those that, when expressed as a system of linear inequalities, belong to this family of 0–1 problems. It follows that linear programming solves the satisfiability problem for (explicit or hidden) extended Horn sets.

We also expose a remarkable parallel between unit resolution and the rounding process dictated by Chandrasekaran's theorem. This parallel sheds light on why unit resolution solves extended Horn satisfiability problems.

Thus, the ease of solving Horn and extended Horn inference problems is directly related to the fact that they pose 0–1 problems that can be solved by linear programming and rounding. This adds to the collection of interesting mathematical properties of Horn clauses that have recently been demonstrated by Blair et al [5], Jeroslow and Wang [17], and Hooker [16]. See [8] and [15] for surveys.

Although Aspvall [1] has discovered a linear-time algorithm for recognizing hidden Horn sets, we know of no polynomial-time recognition algorithm for hidden (or, for that matter, explicit) extended Horn sets. But once an arborescence and a mapping of atomic propositions to arcs have been specified, one can check in linear time whether a given set of clauses is a hidden extended Horn set corresponding to the arborescence. We state an algorithm for doing so.

2. *Preliminaries*

A *literal* is an atomic proposition x_j or its negation $\neg x_j$; in the former case the literal is *positive*. A *clause* is a disjunction of zero or more literals, such as $x_1 \vee \neg x_2$. A *unit clause* contains exactly one literal, and the *empty clause* contains no literals. Any formula of propositional logic can be transformed to a conjunction of clauses (*conjunctive normal form*) using well-known techniques [14].

A *model* is an assignment of truth values to atomic propositions. A model *satisfies* a clause if it makes at least one literal in the clause true. A set (or conjunction) of clauses is *satisfiable* if some model satisfies all the clauses. (The empty clause is regarded as unsatisfiable.) A set S of clauses *logically implies* a clause C if every model satisfying the clauses in S satisfies C, which is to say that when the negation of every literal in C is added to S, the resulting set if unsatisfiable. Thus, one can check for implication by checking the satisfiability of a set of clauses. A clause C *absorbs* clause D is every literal in C occurs in D. It is clear that C logically implies D if and only if C absorbs D.

A *Horn clause* is a clause containing at most one positive literal. A set of clauses is Horn if it contains only Horn clauses. A set of clauses is *hidden Horn* if it becomes Horn when one or more atomic propositions are replaced by their negations throughout the set.

If there is exactly one atomic proposition x_j that occurs negated in clause C and unnegated in clause D, then C and D are the *parents* of a *resolvent* (on x_j), which is the clause containing every literal in C or D except x_j and $\neg x_j$. In *unit resolution*, at least one parent is a unit clause. It can be shown [19, 20] that a set of clauses is unsatisfiable if and only if repeated application of resolution, each time adding the resolvent to the set, eventually generates the empty clause. The satisfiability of a hidden or explicit Horn set can be so checked using unit resolution only [12].

A clause, such as $x_1 \vee \neg x_2$, can be written as a linear inequality in binary variables, in this case $x_1 + (1 - x_2) \geq 1$, where x_1 and x_2 must take values in $\{0, 1\}$. We say that x_j is true when $x_j = 1$ and false when $x_j = 0$. The inequality can be written $x_1 - x_2 \geq 0$, or in general $ax \geq a_0$, where a is a row vector with components in $\{0, 1, -1\}$, x a column vector $(x_1, \ldots, x_n)$, and a_0 is one reduced by the number of -1's in a. Thus, a set of m clauses can be represented as a linear system $Hx \geq h$, where H is an $m \times n$ matrix and x is binary. Clearly, the set of clauses is satisfiable if and only if the following system has a solution:

$$\begin{aligned} Hx &\geq h \\ -x &\geq -e \\ x &\geq 0,\ x \text{ integral,} \end{aligned} \tag{1}$$

where e is a vector of n ones. The *linear relaxation* of (1) is obtained by removing the integrality constraint on x. Linear programming finds a solution of the linear relaxation if one exists.

3. *Chandrasekaran's Theorem*

Let $\lceil \alpha \rceil$ be the smallest integer greater than or equal to α, and for a vector x let component i of $\lceil x \rceil$ be $\lceil x_i \rceil$. Chandrasekaran's theorem is in part the following:

THEOREM 1 (Chandrasekaran [6]). *Consider the linear system $Ax \geq b$, $x \geq 0$, where A is an $m \times n$ integral matrix and b an integral vector. Let T be an $n \times n$ nonsingular matrix that satisfies the following conditions*:

(*a*) *T and T^{-1} are integral*;
(*b*) *Each row of T^{-1} contains at most one negative entry, and all such entries are -1*;
(*c*) *Each row of AT^{-1} contains at most one negative entry, and all such entries are -1.*

Then if x satisfies the linear system, so does the integral vector $T^{-1}\lceil Tx \rceil$.

Chandrasekaran proves this result, but it is instructive to review here why it is true. It is based on the following lemma:

LEMMA 1 (Cottle and Veinott [10], Chandrasekaran [6]). *A nonempty set of the form $S = \{x : A'x \geq b',\ x \geq 0\}$ has a least element if each row of A' has at most one positive component. (A least element is one such that no other element is smaller in any component.) Furthermore, if A' and b' are integral and every positive component of A' is one, then the least element of S is integral.*

For instance, if $A'x \geq b'$ is the following:

$$-x_1 - x_2 \geq -6, \tag{2}$$
$$x_2 \geq 1, \tag{3}$$
$$x_1 - 2x_2 \geq 2, \tag{4}$$

it is clear in Figure 1 that S has the least element (4, 1). Since (3) and (4) have but one positive coefficient, the slopes of the constraint lines are such that the intersection of the corresponding half planes (and, in general, the positive orthant) has a least element. Furthermore, if x_2 in (3) or x_1 in (4) had an integral coefficient other than one, the least element would be nonintegral.

We can argue in general that S has an integral least element x^*. Let $\bar{S} = \{\bar{x} \mid A'\bar{x} = \bar{b}\}$, where $\bar{x} = x - x^*$ and $\bar{b} = b' - A'x^*$. Initially $x^* = 0$, so that $\bar{b} = b'$ and $\bar{S} = S$ is nonempty. If $\bar{b} \leq 0$, then $\bar{x} = 0$ is a least element of $\bar{S}$, which implies that x^* is a least element of S. Otherwise, pick the largest $\bar{b}_i > 0$. Since $\bar{S}$ is nonempty, row i of A' must contain at least one and therefore exactly one positive component, $a'_{ij} = 1$. This implies $\bar{x}_j \geq \bar{b}_i$, so that we can increase x_j^* by $\bar{b}_i$, observe that $\bar{S}$ is still nonempty, and repeat the process. Since some finite x belongs to S, the process must terminate. Obviously, the minimum element x^* is integral. In the example, eq. (4) implies $x_1 \geq 2$, so that we set $x^* = (2, 0)$ and obtain the new system $-\bar{x}_1 - \bar{x}_2 \geq -4$, $\bar{x}_2 \geq 1$, $\bar{x}_1 - 2\bar{x}_2 \geq 0$. Since $\bar{x}_2 \geq 1$, we get $x^* = (2, 1)$ and the system $-\bar{x}_1 - \bar{x}_2 \geq -3$, $\bar{x}_1 \geq 0$, $\bar{x}_1 - 2\bar{x}_2 \geq 2$. This implies $\bar{x}_1 \geq 2$, and we get $x^* = (4, 1)$, which is the minimal element since the resulting system has a nonpositive right-hand side.

To obtain Theorem 1 from Lemma 1, we reason as follows. Using the change of variable $y = a - x$ and setting $A' = -AT^{-1}$, $b' = b - AT^{-1}a$, Lemma 1 implies that $\{y : AT^{-1}y \geq b,\ y \leq a\}$ has an integral maximal element for integral b and integral a, provided the integral matrix AT^{-1} has at most one negative component per row, namely -1. Thus, if $z \in S' = \{y : AT^{-1}y \geq b\}$, then by setting $a = \lceil z \rceil$ we

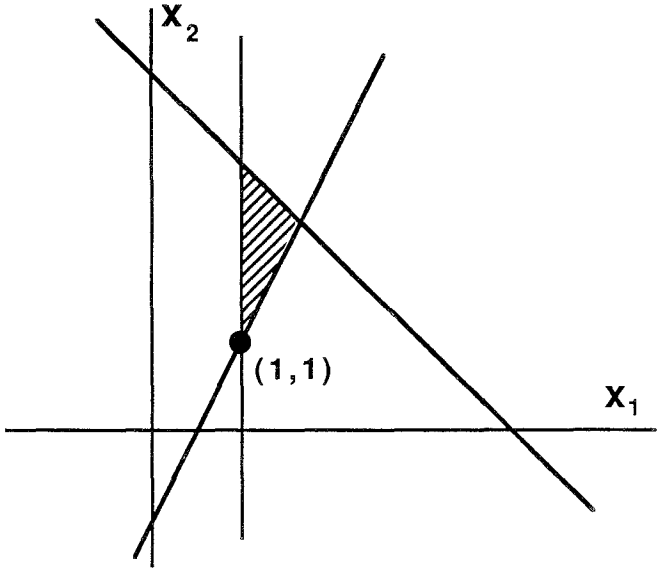


FIGURE 1

see that $\lceil z \rceil \in S'$. This means,

$$AT^{-1}y \geq b \qquad \text{implies} \qquad AT^{-1}\lceil y \rceil \geq b.$$

Setting $x = T^{-1}y$, we get,

$$Ax \geq b \qquad \text{implies} \qquad AT^{-1}\lceil Tx \rceil \geq b. \tag{5}$$

Similarly, by setting $y = a - x$, $A' = -T^{-1}$, and $b' = -T^{-1}a$, Lemma 1 implies that $\{y: T^{-1}y \geq 0,\ y \leq a\}$ has an integral maximal element. From this we infer, as above, that

$$x \geq 0 \qquad \text{implies} \qquad T^{-1}\lceil Tx \rceil \geq 0, \tag{6}$$

provided the integral matrix T^{-1} has at most one negative component per row, namely -1. Theorem 1 follows from (5) and (6).

4. *Trees and Extended Horn Sets*

Recall that a set of clauses is satisfiable if and only if a system of inequalities of the form (1) has a solution. We use Theorem 1 to identify conditions under which (1) has a solution if its linear relaxation does. Sets of clauses that satisfy these conditions will be denoted *extended Horn* (or *hidden extended Horn*).

To apply Theorem 1, we let

$$A = \begin{bmatrix} H \\ -I \end{bmatrix},$$

where H is the matrix in (1). To satisfy condition (c) of Theorem 1, $-T^{-1}$ as well as HT^{-1} must have at most one negative entry in each row, namely -1. We can therefore characterize T as a matrix that satisfies condition (a) and the following:

(b′) Each row of T^{-1} contains at most one $+1$, at most one -1, and no other nonzero entries;

(c′) Each row of HT^{-1} contains at most one negative entry, which must be -1.

Condition (b′) immediately implies that the nonzero rows of T^{-1} may be interpreted as the arc/node incidence matrix of a directed network, with one column removed. In such a matrix, each row corresponds to an arc and each column to a node. If arc k runs from node i to node j, then row k contains a 1 in column i, $a - 1$ in column j, and zeros, otherwise. For instance, the following matrix corresponds to the network in Figure 2.

$$T^{-1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} A & B & C & D & E & F & G \end{array} \\ \left[\begin{array}{ccccccc} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right] \end{array} \begin{array}{c} R \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \end{array}.$$

Vertex R corresponds to a column omitted from T^{-1} but shown to the right of T^{-1} above. In general, the kth entry of the omitted column is 1 if -1 but not 1 appears in row k of T^{-1}, -1 if 1 but not -1 appears, and 0, otherwise.

Furthermore, the nonsingularly of T^{-1} implies that T^{-1} may be interpreted as the arc-node incidence matrix of a directed tree $\mathcal{T}$ on n vertices, due to classical network theory (see [11, Chap. 17]). Figure 2 is again an illustration. It will be convenient to let the *root* of $\mathcal{T}$, which is R in Figure 2, be the node corresponding to the omitted column.

To understand condition (c′), regard any row h^k of H as a vector of flows on the arcs of $\mathcal{T}$. In the example of Figure 2, let $h^k = [-1\ -1\ 1\ 0\ 0\ 1\ 1]$, which indicates the flow marked by heavy arrows. Note that negative flows are in a direction opposite the orientation of the arc. Clearly, h^kT^{-1} indicates the net supply at each node. In the example, $h^kT^{-1} = [1\ 1\ 0\ 0\ 0\ 0\ -1]$, which says that nodes A and B have supply $+1$, G has -1, and all other nodes (except possibly the root R) have supply 0. Condition (c′) states that at most one node (other than the root) can have negative supply, and its supply must be -1. It follows that the flow pattern must consist of chains on which the flow moves in a single direction, and that the downstream end of at most one of the chains can be a node other than the root.

Let an *extended star* be a rooted tree consisting of one or more chains, all incident to the root. If each column of H corresponds to an arc of tree $\mathcal{T}$, let H have the *extended star-chain property* with respect to $\mathcal{T}$ if each row of H represents a flow that consists of the following:

(a) Unit flow moving toward the root of $\mathcal{T}$ on every arc of some (possibly empty) extended star subtree of $\mathcal{T}$;
(b) Unit flow in a single direction on every arc of some (possibly empty) chain in $\mathcal{T}$. The following is now evident:

LEMMA 2. *For a given T satisfying (a) and (b′) and the associated tree $\mathcal{T}$, H satisfies (c′) if and only if H has the extended star-chain property with respect to $\mathcal{T}$.*

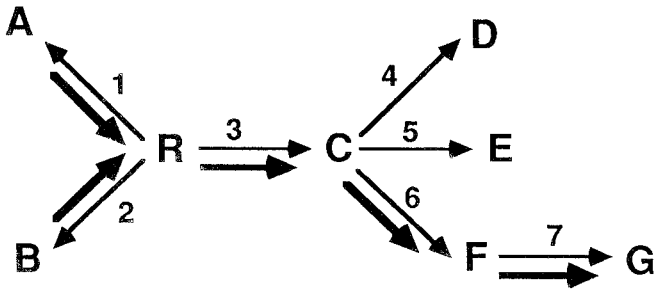


FIGURE 2

For instance, we can set $h^k = [-1\ -1\ 1\ 0\ 0\ 1\ 1]$ because it corresponds to a unit flow along the chain *A–R–C–F–G* and an incoming unit flow on the extended star subtree B–G. (Note that the chain and the extended star need not be disjoint.)

We can now state a condition under which one can use linear programming to determine the 0–1 solubility of any system of linear inequalities with integral right-hand side and coefficients in $\{0, 1, -1\}$. Theorem 1 and Lemma 1 imply the following:

THEOREM 2. *Given any set* (1) *of inequalities with integral h and components of H in* $\{0, 1, -1\}$, *if H has the extended star-chain property with respect to some rooted, directed tree, then* (1) *is soluble if and only if its linear relaxation is soluble.*

A system (1) of the sort described in Theorem 2 represents a set of clauses when each h_i is equal to one reduced by the number of -1's in row i of H. We say that a set of clauses is *extended Horn* if H in the representation (1) has the extended star-chain property with respect to an arborescence. A set is *hidden extended Horn* if H has the extended star-chain property with respect to a rooted, directed tree other than an arborescence. Therefore, linear programming solves the satisfiability problem for both hidden and explicit extended Horn sets.

COROLLARY 1. *An extended Horn set is satisfiable if and only if the linear relaxation of its representation* (1) *has a solution.*

We also note that Horn clauses are a special case of extended Horn clauses. In fact, requiring each row of H to contain at most one $+1$ is equivalent to requiring that H and $T = -I$ satisfy (a), (b′), and (c′). Requiring that the set be explicit or hidden Horn is equivalent to requiring that H and some matrix T that is a diagonal matrix with diagonal elements in $\{1, -1\}$ satisfy (a), (b′), and (c′). The network corresponding to such a matrix T is a rooted, directed star with the root at the center. We therefore have:

COROLLARY 2. *An explicit or hidden Horn set of clauses corresponds to a rooted, directed star* (*with the root at the hub*), *and each clause in the set corresponds to a flow pattern that consists of a unit flow toward the root on some subset of the arcs, plus a unit flow away from the root on at most one arc. An explicit Horn set corresponds to such a rooted star in which all arcs point outward from the root.*

5. *Extended Horn Sets and Unit Resolution*

We can show that unit resolution solves the satisfiability problem for explicit and hidden extended Horn sets, just as it does for ordinary Horn sets.

Suppose that (1) represents an explicit or hidden extended Horn set. We first show how to find a solution of (1)'s linear relaxation, if one exists, using unit resolution. Since by Theorem 1 this solution can be rounded to a solution of (1), it will follow that (1) is soluble if and only if unit resolution solves its linear relaxation.

To solve the linear relaxation of (1), apply the following unit resolution procedure.

Step 1. If there are no unit clauses, assign every variable the value $\frac{1}{2}$, and stop. Otherwise, go to Step 2.

Step 2. Fix the values of the variables in unit clauses so as to satisfy the unit clauses. If this requires setting a variable to both 0 and 1, stop; the linear relaxation is insoluble. Otherwise, go to Step 3.

Step 3. Perform all possible unit resolutions on the current set of clauses, and eliminate all clauses absorbed by the unit clauses in the current set (including the unit clauses themselves). Add the resolvents to the current set, and return to Step 1.

If this procedure finds a contradiction in Step 2, the linear relaxation of (1) clearly has no solution. If it terminates in Step 1, then since all of the remaining clauses have at least two literals, they are satisfied by setting every variable in them to $\frac{1}{2}$. We therefore have,

LEMMA 3. (Blair, et al [5]). *If* (1) *represents a set of clauses,* (1)*'s linear relaxation has a solution if and only if the above algorithm finds one.*

The unit resolution and absorbtion in Step 3 of the above procedure eliminate from the problem all variables that occur in unit clauses. Eliminating a variable is equivalent to deleting the corresponding arc from $\mathscr{T}$ and identifying the endpoints of the arc (*arc contraction*). Note that $\mathscr{T}$ remains a rooted tree, which we may call $\hat{\mathscr{T}}$, after any number of arc contractions. Also, any chain in $\mathscr{T}$ remains a (possibly empty) chain, and the same is true of any extended star subtree of $\mathscr{T}$. Therefore an extended star-chain flow pattern remains an extended star-chain flow pattern after any number of arc contractions. Let $\hat{T}^{-1}$ be the incidence matrix for $\hat{\mathscr{T}}$. Since $x = (\frac{1}{2}, \ldots, \frac{1}{2})$ solves the linear relaxation of (1), we have by Theorem 1 that $\hat{T}^{-1}\lceil \hat{T}x \rceil$ solves (1). Theorem 1 and Lemma 3 imply the following:

THEOREM 3. *An explicit or hidden extended Horn set is satisfiable if and only if the above unit resolution algorithm finds no contradiction.*

6. *The Rounding Pattern*

It is instructive to see how the calculation $\hat{T}^{-1}\lceil \hat{T}x \rceil$ obtains a solution for (1). Let us suppose that after applying the above procedure, we arrive at the tree $\mathscr{T}$ of Figure 2. Let us also suppose that one of the original rows of H has been reduced to the row $\{-1\ -1\ 1\ 0\ 0\ 1\ 1\}$, which represents the flow pattern shown in Figure 2. As guaranteed by the above argument, this flow pattern still has an extended star-chain structure. The variables $x_1, \ldots, x_7$ have been assigned the value $\frac{1}{2}$. To

observe the rounding pattern that yields an integer solution, we note that,

$$T^{-1}\lceil Tx\rceil = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \left\lceil \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 0 & 0 & -1 & -1 & \frac{1}{2} \end{bmatrix} \right\rceil.$$

$$= \begin{bmatrix} -\lceil -\frac{1}{2}\rceil \\ -\lceil -\frac{1}{2}\rceil \\ -\lceil -\frac{1}{2}\rceil \\ -\lceil -\frac{1}{2}\rceil - \lceil -1\rceil \\ -\lceil -\frac{1}{2}\rceil - \lceil -1\rceil \\ -\lceil -\frac{1}{2}\rceil - \lceil -1\rceil \\ \lceil -1\rceil - -\lceil -\frac{3}{2}\rceil \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Thus, the flow of $\frac{1}{2}$ is rounded down on arcs incident to the root, up on arcs once removed from the root, and down on the arc that is twice removed. Rounding down corresponds to setting the associated logical variable x_j to false, and rounding up corresponds to setting it to true.

This suggests that if one regards the arcs as arranged in levels depending on their distance from the root, the rounded solution assigns false to the odd-numbered levels and true to the even-numbered levels. This is in fact the case if the directed tree is an arborescence. When an arc is pointed toward the root, one must reverse the truth value assigned in this pattern.

For a given node i of $\mathscr{T}$, let C_i be the set of arcs in the chain from i to the root. Also let arc k be $(i(k), j(k))$.

THEOREM 4. *Let x be the vector $(\frac{1}{2}, \ldots, \frac{1}{2})$ and T^{-1} be the $n \times n$ arc-node incidence matrix of a rooted, directed spanning tree on n nodes (with the column corresponding to the root omitted). Then the kth component of $T^{-1}\lceil Tx\rceil$ is 0 if $|C_{j(k)}|$ is odd and 1 otherwise.*

PROOF. Let $T^{-1} = (u_{ij})$ and $T = (t_{ij})$. Then for each arc k, $u_{ki(k)} = 1$, $u_{kj(k)} = -1$, and $u_{kl} = 0$ for $l \neq i(k)$, $j(k)$. We can order the rows of T^{-1} so that T^{-1} is triangular, whereupon $u_{kk} = 1$ when arc k is pointed toward the root, and $u_{kk} = -1$, otherwise. For each node i, let $P_i = \{k \in C_i \mid u_{kk} = 1\}$ and $N_i = \{k \in C_i \mid u_{kk} = -1\}$. It is easy to check that t_{ik} is 1 if $k \in P_i$, -1 if $k \in N_i$, and 0, otherwise. Thus, the ith component of $\lceil Tx\rceil$ is $\lceil(\frac{1}{2})(|P_i| - |N_i|)\rceil$. This means that the kth component of $T^{-1}\lceil Tx\rceil$ is

$$\left\lceil\left(\frac{1}{2}\right)(|P_{i(k)}| - |N_{i(k)}|)\right\rceil - \left\lceil\left(\frac{1}{2}\right)(|P_{j(k)}| - |N_{j(k)}|)\right\rceil. \tag{2}$$

When arc k is pointed away from the root, $|P_{i(k)}| = |P_{j(k)}|$ and $|N_{i(k)}| = |N_{j(k)}| - 1$. Thus, if $|C_{j(k)}| = |P_{j(k)}| + |N_{j(k)}|$ is odd, $|P_{i(k)}| - |N_{i(k)}|$ is odd, and (2) has the value 0; otherwise, (2) is 1. When k is pointed toward the root, $|P_{i(k)}| = |P_{j(k)}| + 1$ and $|N_{i(k)}| = |N_{i(k)}|$, and (2) is again 0 when $|C_{j(k)}|$ is odd and 1, otherwise. □

We can now see why this rounding pattern satisfies (1) and, in the process, obtain some insight as to why unit resolution solves the satisfiability problem for explicit and hidden extended Horn sets. For ease of discussion, suppose that the Horn set is explicit, so that the tree $\mathcal{T}$ is an arborescence. For any nonsingleton clause and the corresponding extended star-chain flow pattern on $\mathcal{T}$, we have the following:

(a) When variables associated with arcs on odd-numbered levels are assigned false and the others true, the clause is satisfied. This is because:

(i) a nonempty extended star always contains at least one arc incident to the root, and this arc corresponds to a negated variable that is set to false; and

(ii) if the extended star is empty, the chain (along which flow moves in a single direction) has length at least two and therefore contains either an odd-level arc on which the flow is opposite the orientation of the arc or an even-level arc on which the flow is with the orientation of the arc, and in either case the clause is true.

(b) Any number of arc contractions on $\mathcal{T}$ leaves a (possibly empty) extended star-chain flow pattern or a tree with no flow.

By (b), unit resolution leaves a tree $\mathcal{T}$ with an extended star-chain flow pattern. If the pattern is empty (no flow remains), (1) is satisfiable if and only if unit resolution detected no inconsistency. If flow remains, every clause is nonsingleton, since otherwise unit resolution is not finished. This implies by (a) that (1) is satisfiable. We conclude that (1) is satisfiable if and only if unit resolution detects no inconsistency.

To find other classes of problems for which unit resolution checks satisfiability, one might look for flow patterns that are preserved by arc contraction and for which a prearranged assignment of truth values to arcs satisfies every clause.

7. *Verifying Hidden Extended Horn Sets*

We can describe a linear-time algorithm to determine whether a given set of clauses is hidden extended Horn with respect to a particular arborescence and assignment of variables to arcs. That is, the algorithm checks whether the clauses describe extended star-chain flow patterns on the tree for some orientation of the arcs. The problem of constructing in polynomial time a suitable arborescence for a given set of clauses is open.

We generalize a method proposed by Aspvall [1] for checking, in linear time, whether a set of clauses is hidden Horn. The idea is to transform the verification problem into a specially structured 2-satisfiability problem, and it is best understood in an example. Consider the tree of Figure 2 (without arc orientations), and let H be,

$$\begin{bmatrix} -1 & -1 & 0 & 0 & -1 & 1 & 1 \\ -1 & -1 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & -1 & -1 \end{bmatrix}.$$

As it happens, H is extended Horn for the orientation shown in Figure 2, but we do not know this in advance. Each row of H illustrates a case we must consider. Since for each row the flow pattern is a forest of one or more trees, we refer to each such tree as a *flow tree.*

Row 1. Here there are two flow trees, one of which contains the root (Figure 3). If there had been two or more flow trees excluding the root, H clearly could not have been hidden extended Horn. The sign of each x_j is indicated on the corresponding arc. Note that 5–6–7 must be treated as the chain and the remainder as the extended star. Since variable corresponding to the extended star must be negated, their flows must be contrary to the direction of their arcs. Thus if,

$$y_j = \begin{cases} T & \text{if arc } j \text{ points away from the root,} \\ F & \text{otherwise.} \end{cases}$$

we can assert the propositions y_1 and y_2. Also the flow on 5–6–7 must be unidirectional, and we require $y_5 \equiv y_6 \equiv y_7$, where y_5 is unnegated because x_5 is negated in the clause. This is equivalent to $y_5 \supset y_6 \supset y_7 \supset y_5$. We have the set of propositions,

$$y_1 \qquad y_2 \qquad \neg y_5 \vee y_6 \qquad \neg y_6 \vee y_7 \qquad \neg y_7 \vee y_5.$$

Row 2. Here there is but one flow tree, and only the root has degree greater than two, so that the tree is an extended star (Figure 4). Without loss of generality, we can suppose that the root is not interior to the chain, if there is a chain, and that the flow on the chain is away from the root. Clearly, if the flow on arc 3 is away from the root, it is away from the root on arc 5 as well, so that $\neg y_3 \supset \neg y_5$. It will be convenient to define for row 2 auxiliary variables z_1^2, z_2^2, z_3^2 corresponding to the chains of the extended star beginning with arcs 1, 2, and 3 respectively. Variable z_i^2 will be true when the flow on one or more of the arcs of chain i is directed away from the root. Thus we have $\neg y_1 \supset z_1^2$, $\neg y_2 \supset z_2^2$, $\neg y_5 \supset z_3^2$. Since at most one z_i^2 can be true, we have the following set of propositions:

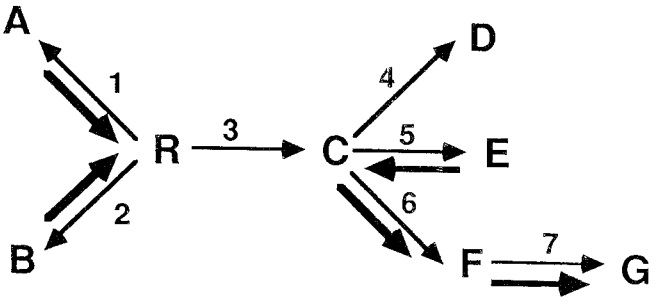
$$y_3 \vee \neg y_5 \qquad \begin{array}{llll} y_1 \vee z_1^2 & \neg z_1^2 \vee \neg z_2^2 & \neg z_2^2 \vee \neg z_1^2 & \neg z_3^2 \vee \neg z_1^2 \\ y_2 \vee z_2^2 & \neg z_1^2 \vee \neg z_3^2 & \neg z_2^2 \vee \neg z_3^2 & \neg z_3^2 \vee \neg z_2^2. \\ y_5 \vee z_3^2 & & & \end{array}$$

Row 3. Here there is again one flow tree (Figure 5), but a node other than the root (node C) has degree greater than two. We must therefore assume that the chain contains node C. Obviously, we must force arcs 2 and 3 to be in the extended star and their flow to be directed toward the root. We can now disregard arcs 2 and 3 and treat the network consisting of 4, 5, 6, and 7 as in the previous case, regarding C as the root. Defining auxiliary variables z_1^3, z_2^3, z_3^3 to correspond to the chains 4, 5, and 6–7, respectively, we have

$$y_2 \quad y_3 \quad y_6 \vee \neg y_7 \qquad \begin{array}{llll} \neg y_4 \vee \neg z_1^3 & \neg z_1^3 \vee \neg z_2^3 & \neg z_2^3 \vee \neg z_1^3 & \neg z_3^3 \vee \neg z_1^3 \\ y_5 \vee \neg z_2^3 & \neg z_1^3 \vee \neg z_3^3 & \neg z_2^3 \vee \neg z_3^3 & \neg z_3^3 \vee \neg z_2^3. \\ y_7 \vee \neg z_3^3 & & & \end{array}$$

The verification algorithm is as follows: We start with an undirected tree $\mathcal{T}$ with n arcs and root r, a set Q of clauses in at most n variables, and a set Y of 2-literal

FIGURE 3



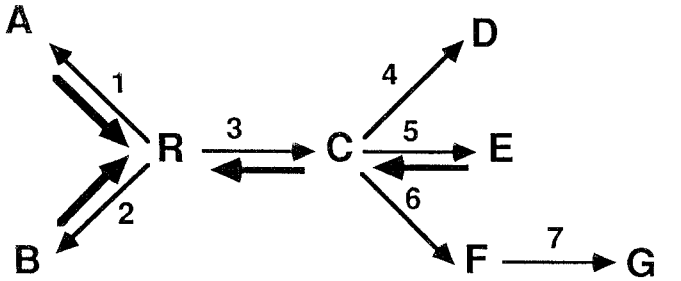
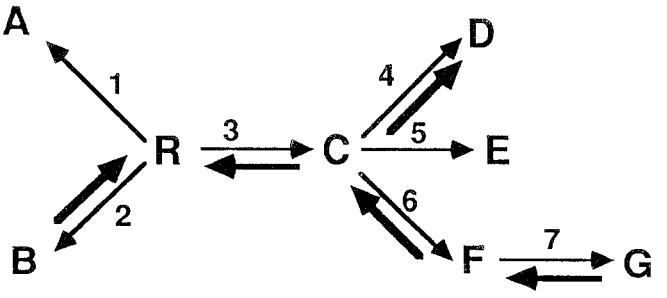


FIGURE 4

FIGURE 5



clauses that is initially empty. We assign each variable x_j occurring in Q to a unique arc of $\mathscr{T}$.

Algorithm
For each clause C of Q **do**:
  Let $\hat{\mathscr{T}}$ be the forest consisting of the arcs of $\mathscr{T}$ that correspond to variables occurring in C.
  If $\hat{\mathscr{T}}$ contains r, $\hat{\mathscr{T}}$ contains at most one node other than r with degree greater than 2, and $\hat{\mathscr{T}}$ contains no node other than r with degree greater than 4, **then**
    if $\hat{\mathscr{T}}$ is a tree, **then**
      if $\hat{\mathscr{T}}$ contains one node i other than r with degree greater than 2, **then**
        perform Procedure 3.
      Else
        perform Procedure 2.
    Else
      if $\hat{\mathscr{T}}$ contains two trees, **then**
        if $\hat{\mathscr{T}}$ contains no node other than r with degree greater than 2, **then**
          let $\mathscr{C}$ be the tree (chain) in $\hat{\mathscr{T}}$ not containing r, and perform Procedure 1.
End do.

If Y is satisfiable, then Q is extended Horn. (An empty Y is unsatisfiable.)

The three procedures correspond to the three cases above. For any node i of degree d in a tree, the tree is the union of d subtrees that are disjoint except for having i as a terminal node; let these be the *i-subtrees* of the tree. If x_j is negated in clause C, let t_j be $\neg y_j$, and otherwise let t_j be y_j.

Procedure 1

For each $j \in \hat{\mathscr{T}} \backslash \mathscr{C}$, add $\neg t_j$ to Y.
  Let $\mathscr{C}$ consist of the subchains of arcs $j_1, \ldots, j_p$ and $k_1, \ldots, k_p$, where j_1 and k_1 are incident to the node of $\mathscr{C}$ closest to r (possibly $p = 0$ or $q = 0$).
  For $s = 1$ to $p - 1$, add $\neg t_{j_s} \vee t_{j_{s+1}}$ to Y.
  For $s = 1$ to $q - 1$, add $t_{k_s} \vee \neg t_{k_{s+1}}$ to Y.
  Add $\neg t_{j_1} \vee t_{k_1}$ and $t_{j_p} \vee \neg t_{k_q}$ to Y.

Procedure 2

For each arc j incident to r, **do**:
  Let the r-subtree of $\mathscr{T}$ at arc j be the chain of arcs $j = j_1, j_2, \ldots, j_p$.
  For $s = 1$ to $p - 1$, add $\neg t_{j_s} \vee t_{j_{s+1}}$ to Y.
  Add $\neg t_{j_p} \vee z_j^c$ to Y.
  For each arc k $(\neq j)$ incident to r, add $\neg z_j^c \vee \neg z_k^c$ to Y.
End do.

Procedure 3

Let $\mathscr{C}$ be the union of the i-subtrees of $\hat{\mathscr{T}}$ not containing r
For each $j \in \mathscr{T} \backslash \mathscr{C}$, add $\neg t_j$ to Y.
For each arc j incident to i, **do**:
  Let the i-subtree at j be the chain of arcs $j = j_1, \ldots, j_p$.
  For $s = 1$ to $p - 1$, add $\neg t_{j_s} \vee t_{j_{s+1}}$ to Y.
  Add $\neg t_{j_p} \vee z_j^c$ to Y.
  For each k $(\neq j)$ in $\mathscr{C}$ that is incident to i, add $\neg z_j^c \vee \neg z_k^c$ to Y.
End do.

In the above algorithm, the number of clauses in Y is quadratic in the number of literals, since quadratically many clauses of the form $\neg z_j^c \vee \neg z_k^c$ are used to ensure that at most one z_j^c is true. We can use a construction of Apsvall [1] to reduce the quadratic to linear size. For each clause C, we can require that at most one of the variables $z_1^c, \ldots, z_q^c$ (renumbered for convenience) is true by introducing auxiliary variables $w_1^c, \ldots, w_{q-1}^c$ and replacing the set of clauses of the form $\neg z_j^c \vee \neg z_k^c$ with the following clauses.

$$\begin{array}{ccccc} & & \neg z_1^c \vee w_1^c, & & \\ \neg w_{i-1}^c \vee \neg z_i^c, & \neg w_{i-1}^c \vee w_i^c, & & \neg z_i^c \vee w_i^c, & i = 2, \ldots, q-1, \\ & & \neg w_{q-1}^c \vee \neg z_q^c. & & \end{array}$$

Since 2-satisfiability problems can be solved in linear time [2], our algorithm runs in linear time.

8. *Extended Horn Sets of Rules*

It is useful to characterize extended Horn sets of rules as well as clauses, since rules are commonly used to build knowledge bases.

We first characterize Horn sets of rules. For our purposes, a *rule* is a proposition of the form,

$$(y_1 \wedge y_2 \wedge \cdots \wedge y_p) \vdash (z_1 \vee z_2 \vee \cdots \vee z_q),$$

where $\vdash$ means "implies." The atomic propositions $y_1, \ldots, y_p$ are the *premises* and their conjunction the *antecedent*. The clause $z_1 \vee z_2 \vee \cdots \vee z_q$ is the *consequent*. Thus, no negations appear. Also there may be no antecedents, or an empty

consequent; "$\vdash z_1$" simply asserts z_1, and "$y_1 \vdash$" denies y_1. We need not consider disjunctions of antecedents or conjunctions of consequents, since $y_1 \vee y_2 \vdash z$ is equivalent to the two rules $y_1 \vdash z$ and $y_2 \vdash z$, and $y \vdash z_1 \wedge z_2$ is equivalent to the two rules $y \vdash z_1$ and $y \vdash z_2$. It is clear that any clause can be written as a rule, by putting positive literals in the consequent and using the atomic propositions in negative literals as antecedents. For instance, the clause $\neg x_1 \vee \neg x_2 \vee x_3 \vee x_4$ can be written as the rule $(x_1 \wedge x_2) \vdash (x_3 \vee x_4)$. A Horn set of rules is one that is one that can be obtained from a Horn set of clauses in this fashion. Thus, a set of rules is Horn if and only if it has at most one proposition in each consequent.

Given a set of rules, an equivalent set can be obtained by complementing one or more variables (the *complement* $\bar{x}_j$ of x_j is a variable interpreted as meaning the negation of x_j). If a premise is complemented, the premise is removed and its complement is added to the consequent. If a variable in the consequent is complemented, it is removed from the consequent and its complement added to the antecedent. A set of rules is hidden Horn if it can be obtained from an equivalent Horn set by complementing variables.

We now characterize extended Horn sets of rules. Any such set must correspond to an arborescence $\mathcal{T}$ in the following way. Since the atomic propositions in the consequent of an extended Horn rule are positive in the associated clause, they must form a chain $\mathcal{T}$ in which every arc is directed away from the root; we can call it the *consequent chain*. To characterize the antecedents, let us also say that a conjunction of premises is a *complete premise* if the individual premises correspond to a chain of $\mathcal{T}$ with all arcs directed toward the root and one endpoint at the root. A conjunction of premises is a *partial premise* if it corresponds to a chain of $\mathcal{T}$ whose arcs are all directed toward the root and that is not incident to the root. A partial premise is *coterminal* with the consequent chain if the two chains have an endpoint in common.

Thus, a set of rules is extended Horn if and only if there is some arborescence $\mathcal{T}$ such that every consequent corresponds to a chain in $\mathcal{T}$, and the antecedent of every rule is a conjunction of zero or more complete premises and at most one partial premise, where the latter is coterminal with the consequent chain. A set of rules is hidden extended Horn if and only if it can be obtained from an extended Horn set by complementing one or more variables.

9. *An Interpretation of Extended Horn Sets*

There are two ways to obtain an extended Horn knowledge base in practice. One is to test an existing knowledge base for whether it is hidden extended Horn, and if it is, complement variables as necessary to make the rules explicitly extended Horn. But this requires a yet undiscovered practical recognition algorithm for extended Horn sets. Another approach is to build an explicitly extended Horn knowledge base from scratch, adding no rule unless it preserves the extended Horn structure. This is the approach now taken to build ordinary Horn knowledge bases, and it is easy to implement because one need only check whether each clause is Horn before adding it to the knowledge base, and the result is a Horn knowledge base. But a knowledge base may fail to be extended Horn even if all of its rules are extended Horn when considered individually. In fact, it is easy to see that *any* rule, considered in isolation, is extended Horn. This raises the problem as to how an extended Horn database can be built in the absence of a practical recognition algorithm. One solution is to provide an interpretation of an extended Horn structure such that, if one keeps the interpretation in mind, one will naturally

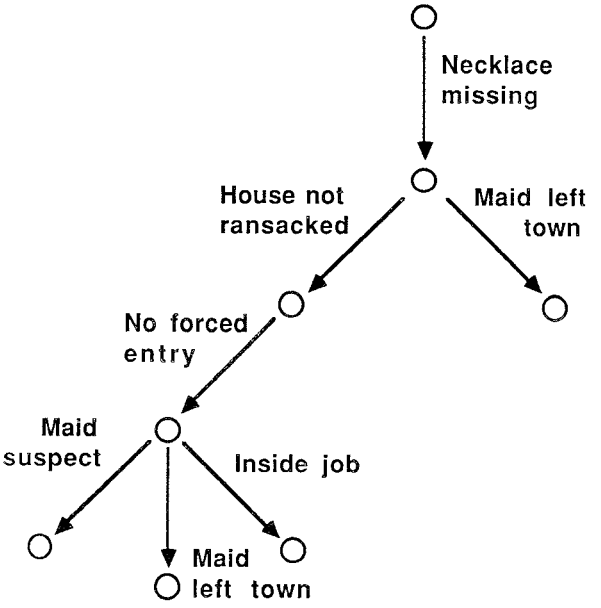


FIGURE 6

construct an extended Horn knowledge base. We suggest here one such interpretation. There may be other equally valid interpretations.

The arborescence that corresponds to an extended Horn set can be viewed as defining paths of inquiry, corresponding to chains leading out from the root. Ascertaining the truth of each proposition leads one to raise certain other questions, represented by the children (immediate successors) of the propositions in the tree. If a certain child is found to be true, one investigates *its* children, and so on. The premises of a typical rule are satisfied when one pursues one or more lines of inquiry and, in each case, finds the relevant propositions to be true. Thus, one would expect the antecedent of a rule to consist of one or more complete premises, each representing a line of inquiry.

Consider for example the arborescence of Figure 6, in which lines of inquiry relate to solving a possible crime, the theft of a necklace. Consider the extended Horn set of rules:

(i) If the necklace is missing, the house was not ransacked, there was no forcible entry, and the maid just left town, then the maid is suspect.
(ii) If the necklace is missing, and the house was not ransacked, then there was no forcible entry or it was an inside job (or both). (In the former case, the necklace could have been lost rather than stolen.)
(iii) If the necklace is missing, it was an inside job and the maid just left town, then the maid is suspect.

(Note that a proposition, in this case "The maid just left town," can occur more than once in the arborescence. The different occurrences are treated as distinct propositions for the purposes of inference.)

The premises of rule (i) are satisfied when one successfully follows a line of an inquiry down through, "The maid just left town." The premises of rule (ii) represent the same line of inquiry taken not quite so far. Also the consequent of rule (ii) is a disjunction. The whole point of using extended Horn sets, after all, is to allow

disjunctions as consequents. Furthermore, one or more of the propositions in a consequent can appear in an antecedent even when they are not part of a complete premise. This occurs in rule (iii), in which "It was an inside job" is a *partial* premise. But note that the consequent, "The maid is a suspect," is coterminal with the partial premise, as required for an extended Horn set. The combination of the three rules permits one to suspect the maid without checking whether there was forcible entry, provided that one verifies that the house was not ransacked and that the maid just left town.

Thus, the extended Horn framework allows for some departure from the restriction that an antecedent represent one or more lines of reasoning followed down from the root. One is permitted to begin with an intermediate step in a line of reasoning, which one presumably establishes by some means external to the reasoning embodied in the rule base. That the necklace was stolen by an insider, for instance, might be learned through an anonymous tip. The propositions verified in this manner form a partial premise. But when external knowledge is used in this way, the *conclusion* drawn must appear as a chain coterminal with the partial premise.

Therefore, one can verify the premises of a rule either by reasoning wholly within the framework of the rule base (i.e., by using complete premises only), or by using external knowledge to establish an intermediate step and reasoning thenceforth as prescribed by the rule base. In the former case, the consequent is a unidirectional chain occurring anywhere in the tree, but in the latter it must be a unidirectional chain that begins at the same node at which the line of reasoning begins. This means that the children of a parent proposition consist not only of propositions one would want to investigate after verifying the parent, but inferences one might draw after beginning a line of reasoning with one of the other children.

10. *Open Problems*

We leave unsolved the general problem of recognizing extended Horn sets in polynomial time. This problem appears to be a variation on the classical graph realization question (e.g., [4]) in combinatorics. Chandru et al. [7] have described a linear-time algorithm for recognizing instances of the generalized Horn sets of Yamasaki and Doshita. Gallo and Scutellà [13] showed that membership in their classes Γ_k can be checked in $O(\mathrm{Ln}^k)$ time. These results could suggest approaches to recognizing extended Horn problems.

ACKNOWLEDGMENT. We are grateful to R. Chandrasekaran for bringing to our attention the connection between Horn systems in logic and Leontief systems in mathematical economics.

REFERENCES

1. APSVALL, B. I. Efficient algorithms for certain satisfiability and linear programming problems. Ph.D. dissertation. Stanford Univ., Stanford, Calif., 1980.
2. APSVALL, B., PLASS, M. F., AND TARJAN, R. E. A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Inf. Proc Lett 8* (1979), 121–123.
3. ARVIND, V., AND BISWAS, S. An $O(n^2)$ algorithm for the satisfiability problem of a subset of propositional sentences in CNF that includes all Horn sentences. *Inf Proc Lett 24* (1987), 67–69.
4. BIXBY, R. E., AND WAGNER, D. K. An almost linear-time algorithm for graph realization. *Math Oper. Res 13* (1988), 99–123.
5. BLAIR, C., JEROSLOW, R. G., AND LOWE, J. K. Some results and experiments in programming techniques for propositional logic. *Comput Oper Res 13* (1988), 633–645.

6. CHANDRASEKARAN, R. Integer programming problems for which a simple rounding type of algorithm works. In W. R. Pulleyblank, ed. *Progress in Combinatorial Optimization*. Academic Press Canada, Toronto, Ontario, Canada, 1984, pp. 101–106.
7. CHANDRU, V., COULLARD, C. R., HAMMER, P. L., AND SUN, X. Horn renameable Horn and generalized Horn functions. *Ann. Math. Artif. Int 1* (1990), 33–47.
8. CHANDRU, V., AND HOOKER, J. N. Logical inference: A mathematical programming perspective. In S. Kumara, A. Soyster, and R. L. Kashyap, Eds. *Artificial Intelligence Manufacturing Theory and Practice*. Institute of Industrial Engineers, 1989, pp. 97–120.
9. COOK, S. A. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing* (Shaker Heights, Ohio, May 3–5). ACM, New York, 1971, pp. 151–158.
10. COTTLE, R. W., AND VEINOTT, A. F. Polyhedral sets having a least element. *Math. Prog. 3* (1972), 238–249.
11. DANTZIG, G. *Linear Programming and Extensions*. Princeton Univ. Press, Princeton, N.J., 1963.
12. DOWLING, W. F., AND GALLIER, J. H. Linear time algorithms for testing the satisfiability of Horn formulae. *J. Logic Prog. 3* (1984), 267–284.
13. GALLO, G., AND SCUTELLÀ, M. G. Polynomially soluble satisfiability problems. *Inf. Proc. Lett. 29* (1988), 221–227.
14. GENESERETH, M. R., AND NILSSON, N. J. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, Calif., 1987.
15. HOOKER, J. N. A quantitative approach to logical inference. *Dec Supp. Syst. 4* (1988), 45–69.
16. HOOKER, J. N. Input proofs and rank one cutting planes. *ORSA J Comput. 1* (1989), 137–145.
17. JEROSLOW, R., AND WANG, J. Dynamic programming, integral polyhedra and Horn clause knowledge bases. *ORSA J. Comput 1* (1989), 7–19.
18. MINOUX, M. LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation, *Inf. Proc. Lett. 29* (1988), 1–12.
19. QUINE, W. V. The problem of simplifying truth functions. *Amer. Math. Monthly* 59 (1955), 521–531.
20. QUINE, W. V. A way to simplify truth functions, *Amer. Math. Monthly* 62 (1955), 627–631.
21. YAMASAKI, S., AND DOSHITA, S. The satisfiability problem for a class consisting of Horn sentences and some non-Horn sentences in propositional logic. *Inf. Cont. 59* (1983), 1–12.

RECEIVED FEBRUARY 1989; REVISED FEBRUARY 1990; ACCEPTED MARCH 1990