DIMACS Technical Report 93–37
June 1993

Quasi-Acyclic Propositional Horn Knowledge Bases: Optimal Compression[1]

by

Peter L. Hammer[2]
RUTCOR • Rutgers University
P.O. Box 5062 • New Brunswick, NJ 08903-5062
e-mail: hammer@rutcor.rutgers.edu

and

Alexander Kogan[2]
RUTCOR and Graduate School of Management
Rutgers University • 92 New Street, Newark, NJ 07102
e-mail: kogan@rutcor.rutgers.edu

[1]The authors gratefully acknowledge the partial support of AFOSR (Grant F49620-93-1-0041) and ONR (Grants N00014-92-J1375 and N00014-92-J4083).
[2]Permanent member of DIMACS

ABSTRACT

Horn knowledge bases are widely used in many applications. This paper is concerned with the optimal compression of propositional Horn production rule bases – one of the most important knowledge bases used in practice. The problem of knowledge compression is interpreted as a problem of Boolean function minimization. It was proved in [14] that the minimization of Horn functions, i.e. Boolean functions associated to Horn knowledge bases, is **NP**-complete.

This paper deals with the minimization of *quasi-acyclic* Horn functions, the class of which properly includes the two practically significant classes of quadratic and of acyclic functions. A procedure is developed for recognizing in quadratic time the quasi-acyclicity of a function given by a Horn CNF, and a graph-based algorithm is proposed for the quadratic time minimization of quasi-acyclic Horn functions.

1 Introduction

The growing complexity of knowledge incorporated in the modern expert systems has led to a rapid increase in the size of their knowledge bases (see e.g. [5, 6, 10, 16, 20]). One consequence of this development is the increase of the response time, due to the dependence of the computational complexity of answering queries on the size of the knowledge base.

Taking into account that a particular knowledge base is just one possible representation of knowledge, interpreted as the set of models of the knowledge base, it was observed in [14] that other, logically equivalent, representations of the same knowledge exist, and they can have significantly smaller size.

It was remarked in [14] that the production rule $\bigwedge_{i=1}^{n} q_i \rightarrow \bigvee_{j=1}^{m} r_j$ and the clause $(\bigvee_{i=1}^{n} \neg q_i) \bigvee (\bigvee_{j=1}^{m} r_j)$ are logically equivalent, i.e. they have the same set of models (satisfying truth assignments). Therefore, production rule knowledge bases are in fact sets of clauses (conjunctive normal forms). To a set of models of a knowledge base we associate a Boolean function, which assumes the value 1 exactly on the 0-1 vectors corresponding to the models, and which assumes the value 0 elsewhere:

$$f(\alpha) = \begin{cases} 1, & \text{if } \alpha \text{ is a model;} \\ 0, & \text{otherwise.} \end{cases}$$

Since the Boolean function corresponding to a knowledge base may admit other equivalent conjunctive normal form (CNF) representations which may be shorter than the initial one, the transformation of a given knowledge base to a logically equivalent knowledge base of minimum size is a Boolean function minimization problem.

It is well known that Horn knowledge bases are among the most widely used ones, and different aspects of them are researched actively (see e.g. [2,8,7,18]. Therefore, we concentrate in this paper on the problem of optimal compression of propositional Horn clause knowledge bases, i.e. the problem of minimizing those Boolean functions, which are given by Horn conjunctive normal forms. It was proved in [14] that the minimization of Horn functions is **NP**-complete.

Many knowledge bases used in practice possess special structural properties. Among the frequently encountered properties we mention those of *acyclicity* and of *quadraticity* (see e.g. [1, 10, 11, 16]). In this paper we generalize both of this properties by defining the class of *quasi-acyclic* Horn functions. The main results of this paper consist of quadratic time algorithms for the recognition and the optimal compression of Horn knowledge bases belonging to this class.

The first sections of the paper introduce the basic concepts, review some of the results on which the paper is built, and provide the technical tools needed for the development of the main results. In Section 2 we introduce the main definitions and review the results of [12] which are used in this paper. These results define a canonical decomposition of an arbitrary Horn function into a negative part and a definite Horn one (Theorem 2.5). A further important tool needed in this paper is the "forward chaining" procedure, which is described in Section 3, where we also present a linear time implementation of it. In Section 4 we associate a directed graph to a definite Horn CNF, and establish a correspondence between the prime implicates of the function and certain directed paths in this graph.

In Section 5 we define the concept of acyclic Horn functions. We show that *any* prime CNF of such a function is acyclic, implying the possibility of recognizing acyclicity in quadratic time. The main result of this section (Theorem 5.5) proves that every acyclic Horn function has a unique irredundant and prime CNF. A qudratic time minimization algorithm for acyclic Horn functions follows immediately from this result.

In Section 6 we introduce the *2-component* of an arbitrary Horn function as the conjunction of all its quadratic definite Horn prime implicates. The 2-component induces an equivalence relation between certain pairs of variables. The function obtained by identifying all the equivalent variables of the initial function will be called its *2-condensation*. Section 6 is devoted to the investigation of relations between the Horn CNFs of the initial function and the Horn CNFs of its 2-condensation. The results of this section provide the foundation for the study of the class of quasi-acyclic Horn functions in Section 7.

In Section 7 we define quasi-acyclic Horn functions, as being those Horn functions whose 2-condensations are acyclic. We provide a quadratic time algorithm for the recognition of the quasi-acyclicity of a function given by a Horn CNF. We characterize all the irredundant prime, as well as all the minimum CNFs of quasi-acyclic Horn functions. A quadratic time minimization algorithm for quasi-acyclic Horn functions follows immediately.

2 Background Information

In this section we introduce some basic Boolean terminology and review those results of [12] which are used in this paper. These results introduce a canonical decomposition of an arbitrary Horn function into a negative part and a definite Horn one (Theorem 2.5).

2.1 Basic Definitions

A *Boolean function* f of n variables $x_1, \ldots, x_n$ is a mapping $\{0,1\}^n \to \{0,1\}$. The variables $x_1, \ldots, x_n$ and their negations $\neg x_1, \ldots, \neg x_n$ are called positive and negative literals respectively, or simply *literals*. If $f(x_1, \ldots, x_n) = 1$ implies $g(x_1, \ldots, x_n) = 1$ for every 0-1 vector, then we write $f \leq g$.

A *clause* is a disjunction of literals in which each variable appears at most once. We shall say that a clause C *subsumes* a clause C' iff $C \leq C'$ (e.g. the clause $x \bigvee \neg y$ subsumes the clause $x \bigvee \neg y \bigvee z$). A clause C is called an *implicate* of a function f iff $f \leq C$. An implicate C of a function is called *prime* iff there is no distinct implicate C' subsuming C (in other words, an implicate of a function is prime iff dropping any literal from it produces a clause which is not an implicate of the function).

A *conjunctive normal form* (CNF) is a conjunction of clauses. Sometimes we will not distinguish between a CNF and the set of its clauses. The *length* of a CNF is the number of literals in it. It is well known that every Boolean function can be represented by a CNF. Any two CNFs representing the same Boolean function are called *equivalent*.

A CNF representing a function is called *prime* iff each clause of the CNF is a prime implicate of the function. On the other hand, a CNF representing a function is called *irredundant* iff dropping any clause from it produces a CNF which does not represent the same function.

A prime implicate of a Boolean function is called *essential* iff it is contained in every prime CNF of the function. A prime implicate of a Boolean function is called *redundant* iff no irredundant prime CNF representing the function contains it.

A CNF of a function is called *minimum* (*optimum*) iff there is no CNF representing the same function and containing fewer literals (clauses). Obviously, any minimum CNF is both irredundant and prime. The *problem of Boolean function minimization* consists in constructing a minimum CNF of the function.

As an example, consider the CNF

$$(x \bigvee \neg y \bigvee \neg z)(x \bigvee \neg y \bigvee z)(\neg x \bigvee y \bigvee \neg z)(\neg x \bigvee y \bigvee z)(x \bigvee \neg w \bigvee \neg z)(x \bigvee \neg w \bigvee z) \bigwedge$$
$$\bigwedge(\neg x \bigvee w \bigvee \neg z)(\neg x \bigvee w \bigvee z)(x \bigvee \neg z)(z \bigvee \neg v).$$

It is easy to see that an equivalent prime CNF is

$$(x \bigvee \neg z)(x \bigvee \neg y)(y \bigvee \neg z)(\neg x \bigvee y)(x \bigvee \neg z)(x \bigvee \neg w)(w \bigvee \neg z)(\neg x \bigvee w)(x \bigvee \neg z)(z \bigvee \neg v).$$

After dropping the clauses $y \vee \neg z$ (an implicate of $(x \vee \neg z)(\neg x \vee y)$) and $w \vee \neg z$ (an implicate of $(x \vee \neg z)(\neg x \vee w)$), and dropping the duplicates of the clause $x \vee \neg z$, we arrive at the equivalent prime CNF

$$(x \bigvee \neg z)(x \bigvee \neg y)(\neg x \bigvee y)(x \bigvee \neg w)(\neg x \bigvee w)(z \bigvee \neg v),$$

which can be seen to be irredundant. It can be shown that this last CNF is not minimum, since the following equivalent CNF (which is actually a minimum one)

$$(x \bigvee \neg z)(x \bigvee \neg y)(y \bigvee \neg w)(\neg x \bigvee w)(z \bigvee \neg v)$$

contains fewer literals. It can be proved that the clause $z \vee \neg v$ is an essential prime implicate of this Boolean function, the clauses $x \vee \neg v$, $y \vee \neg v$ and $w \vee \neg v$ are redundant prime implicates of this function, and no other prime implicate of this function is essential or redundant.

2.2 Horn Functions

A clause is called *negative* iff it contains only negative literals; a clause is called *definite Horn* iff it contains exactly one positive literal. A CNF is called *negative* iff it contains only negative clauses. A CNF is called *Horn* iff it contains only definite Horn and negative clauses. A CNF is called *definite Horn* iff it contains only definite Horn clauses.

An example of a negative CNF is $(\neg x \vee \neg y \vee \neg z)(\neg x \vee \neg y \vee \neg w)$, an example of a definite Horn CNF is $(\neg x \vee \neg y \vee z)(\neg w \vee \neg y \vee z)$, while their conjunction

$$(\neg x \bigvee \neg y \bigvee \neg z)(\neg x \bigvee \neg y \bigvee \neg w)(\neg x \bigvee \neg y \bigvee z)(\neg w \bigvee \neg y \bigvee z) \qquad (1)$$

is an example of a Horn CNF.

Definition 2.1 *A Boolean function is called* **negative** *iff it has at least one negative CNF; a Boolean function is called* **(definite) Horn** *iff it has at least one (definite) Horn CNF.*

It is known that each prime implicate of a negative function is negative, and that each negative function has a unique irredundant and prime CNF, consisting of all the prime implicates of the function.

It was shown in [12] that each prime implicate of a Horn function is either definite Horn or negative, and each prime implicate of a definite Horn function is definite Horn. Hence, any prime CNF representing a (definite) Horn function is (definite) Horn.

It is known that Horn satisfiability problems can be solved in linear time [8]. On this basis an algorithm was given in [12] showing that

Lemma 2.2 *Any Horn CNF can be transformed to an equivalent Horn irredundant prime CNF in time quadratic in the length of the given CNF.*

2.3 Definite Horn Components

Any Horn CNF consists of a negative part (the conjunction of its negative clauses) and of a definite Horn part (the conjunction of its definite Horn clauses). It can be seen that different Horn CNFs of the same Horn function may lead to different such splittings (e.g. the CNF $(\neg x \bigvee \neg y)(\neg w \bigvee \neg y \bigvee z)$ is equivalent to the CNF (1)). The following definition will be useful in constructing a canonical splitting of a Horn function.

Definition 2.3 *A definite Horn function is called the* **definite Horn component** $h(f)$ *of a Horn function* f *iff the following two conditions hold:*

(i) $f = h(f) \bigwedge g$*, where* g *is a negative function;*

(ii) if $f = h' \bigwedge g'$*, where* h' *is a definite Horn function and* g' *is a negative function, then* $h' \leq h(f)$*.*

The uniqueness of the definite Horn component of a Horn function follows directly from the definition. It was shown in [12] that the definite Horn component $h(f)$ of an arbitrary Horn function f exists, and that it can be represented as the conjunction of all the definite Horn clauses of an arbitrary prime CNF of f. This implies that the conjunction of all the definite Horn clauses of a prime irredundant CNF of a Horn function f is a prime irredundant CNF of $h(f)$. Therefore, if a Horn function is given by a Horn CNF, then a prime irredundant CNF of its definite Horn component can be constructed in quadratic time.

It should be noted that the definite Horn part of a non-prime Horn CNF does not necessarily represent its definite Horn component.

2.4 Negative Restrictions

Given a Horn function f, the set of its definite Horn prime implicates and the set of its negative prime implicates will be denoted by $\mathcal{H}(f)$ and $\mathcal{N}(f)$ respectively. We shall make use below of the CNF $H(f) = \bigwedge_{C \in \mathcal{H}(f)} C$.

In order to analyze the structure of the set $\mathcal{N}(f)$ of negative prime implicates of an arbitrary Horn function f, we shall construct a directed graph $G_N(f)$ in the following way. The vertices of $G_N(f)$ correspond to the negative prime implicates of f. The vertices corresponding to two prime implicates C_1 and C_2 are linked by an arc $C_1 \to C_2$ iff C_2 is an implicate of $C_1 \bigwedge H(f)$. It follows from the definition that $G_N(f)$ is transitively closed, and hence each strongly connected component of $G_N(f)$ is a complete directed subgraph.

Let $\mathcal{N}_i(f)$, $i \in \{1, \ldots, k\}$, be the subsets of negative prime implicates of f corresponding to the strongly connected components of $G_N(f)$, indexed in such a way that the "starting components", i.e. those with no incoming arcs are at the beginning of the list; let m be the number of starting components.

Definition 2.4 *A negative conjunctive normal form $\mathcal{F}_N$ is called a* **negative restriction** *of a Horn function f iff it has the form $\mathcal{F}_N = \bigwedge_{i=1}^{m} C_i$, where $C_i \in \mathcal{N}_i(f)$ and m is the number of starting components.*

Note that different negative restrictions represent different negative functions. Obviously, the number of different negative restrictions of a Horn function f is equal to $\prod_{i=1}^{m} |\mathcal{N}_i(f)|$.

It can be seen that if $\mathcal{F}_N$ is an arbitrary negative restriction of a Horn function f, and if $\mathcal{F}_H$ is the conjunction of all the definite Horn clauses of an arbitrary Horn CNF of f, then the function f admits the representation $f = \mathcal{F}_N \bigwedge \mathcal{F}_H$.

It was shown in [12] that the conjunction of all the negative clauses of an arbitrary irredundant and prime CNF of a Horn function f is a negative restriction of f. Therefore, using Lemma 2.2 a negative restriction of a Horn function given by an arbitrary Horn CNF can be constructed efficiently.

2.5 Decomposition of Horn Functions

The main result of [12] shows that the analysis of the irredundant and prime CNFs of Horn functions can be separated into that one of definite Horn functions and that one of negative restrictions. More precisely,

Theorem 2.5 *A conjunctive normal form $\mathcal{F}$ is an irredundant and prime representation of a Horn function f if and only if*

$$\mathcal{F} = \mathcal{F}(h(f)) \bigwedge \mathcal{F}_N,$$

where $\mathcal{F}(h(f))$ is an irredundant prime CNF of the definite Horn component $h(f)$, and where $\mathcal{F}_N$ is a negative restriction of f.

Since all the negative restrictions of a Horn function contain the same number of negative clauses, $\mathcal{F}$ consists of a minimum number of clauses iff $\mathcal{F}(h(f))$ consists of a minimum number of clauses.

We remind that a *unit* clause is a clause consisting of exactly one literal. It can be seen that any unit prime implicate of a Boolean function is essential. Moreover, if x or $\neg x$ is a unit prime implicate of a Boolean function f, then no other prime implicate of f depends on the variable x. Hence, any Horn function f (given by a Horn CNF) can be decomposed in quadratic time into a (possibly empty) conjunction of literals g, and a Horn function f' having no unit prime implicates, such that g and f' depend on disjoint sets of variables, and $f = g \bigwedge f'$. Therefore, **without loss of generality, from now on we shall restrict our attention to Horn functions which do not have unit prime implicates.**

2.6 Intractability of Minimization

In [14] we studied the computational complexity of the

Horn Function Minimization Problem

Instance: A Horn CNF $\mathcal{F}$ and a positive number k.

Question: Is there a CNF, containing at most k literals, equivalent to $\mathcal{F}$?

It was proved that

Theorem 2.6 *The Horn function minimization problem is* **NP***-complete.*

The intractability of the Horn function minimization problem clearly points out the importance of finding polynomially solvable subclasses of this problem.

3 Forward Chaining Procedure

Given a definite Horn CNF $\mathcal{F}$, we shall define a *forward chaining* procedure which associates to any subset S of variables a superset $\mathcal{R}$ of it in the following way. The procedure takes as input the subset S of variables, initializes the set $\mathcal{R} = S$, and at each step it looks for a definite Horn clause $x' \bigvee \bigvee_{x \in S'} \neg x$ in $\mathcal{F}$ such that $S' \subseteq \mathcal{R}$, and $x' \notin \mathcal{R}$. If such a clause is found, the variable x' is included into $\mathcal{R}$, and the search is repeated as many times as possible. The following statements proved in [14] characterize the implicates of an arbitrary Horn CNF in terms of the forward chaining procedure.

Lemma 3.1 *A definite Horn clause $C = x' \bigvee \bigvee_{x \in S} \neg x$ is an implicate of a definite Horn function h given by a Horn CNF $\mathcal{F}$ iff the forward chaining procedure starting with the set S includes eventually the variable x' into the set $\mathcal{R}$.*

Lemma 3.2 *A negative clause* $C_1 = \bigvee_{x \in S_1} \neg x$ *is an implicate of a Horn function* $f = \mathcal{F} \bigwedge C_2$*, where* $\mathcal{F}$ *is a definite Horn CNF and* $C_2 = \bigvee_{x \in S_2} \neg x$ *is a negative clause, iff the forward chaining procedure for* $\mathcal{F}$ *starting with the set* S_1 *includes eventually every variable* $x \in S_2$ *into the set* $\mathcal{R}$*.*

The following statement shows that the forward chaining procedure can be efficiently implemented. We assume that the starting subset of variables S contains only variables present in the given definite Horn CNF $\mathcal{F}$.

Lemma 3.3 *Given a definite Horn CNF* $\mathcal{F}$ *and a subset of variables* S*, the forward chaining procedure for* $\mathcal{F}$ *starting with* S *can be performed in* $O(L)$ *time, where* L *is the length of* $\mathcal{F}$*.*

Proof To implement the forward chaining procedure, the following data structures are used:

- The set of variables. The element for a variable x is a structure which includes a Boolean label $\alpha(x)$ initialized at 0, and the list of addresses of the structures for clauses in $\mathcal{F}$ containing $\neg x$.
- The set of clauses in $\mathcal{F}$. The element for a clause $C' = x' \bigvee \bigvee_{x \in S'} \neg x$ is a structure which includes an integer $d(C)$ initialized at $|S'|$, and the address of the structure for the variable x'.
- The first-in-first-out queue Q of variables which is initialized at $\emptyset$.
- The output set $\mathcal{R}$ initialized at $\emptyset$.

The initialization step consists in assigning $\alpha(x) := 1$ and including the address of the structure for x in the queue Q for all $x \in S$. Clearly, the data structure setup and the initialization step can be performed in $O(L)$ time.

If Q is empty, the procedure outputs the current set $\mathcal{R}$. If Q is not empty, the following steps are performed:

- Remove from Q the first variable x, and add it to $\mathcal{R}$, i.e. $\mathcal{R} := \mathcal{R} \bigcup \{x\}$.

For all clauses C in the list of x do the following:

- Reduce $d(C)$ by 1, i.e. $d(C) := d(C) - 1$.
- If $d(C) = 0$, then take the address of x' in the structure for C and check $\alpha(x')$.
- If $\alpha(x') = 0$, then assign $\alpha(x') := 1$ and add the address of x' to the queue Q.

The procedure repeats until Q is empty.

Each step in the above description requires a constant number of operations. It can be seen easily that the number of steps does not exceed the number of literals in $\mathcal{F}$. Therefore, the forward chaining procedure for $\mathcal{F}$ starting with S will stop and output the set $\mathcal{R}$ in time linear in the length of the given CNF. $\square$

4 CNF Graphs

This section will provide some technical tools needed for the development of the results presented in this paper.

To a definite Horn CNF $\mathcal{F}$ we shall associate a directed graph $G(\mathcal{F})$ in the following way. The vertices of $G(\mathcal{F})$ correspond to the variables of $\mathcal{F}$. Two vertices corresponding to the variables x and y are linked by an arc $x \rightarrow y$ iff there is a clause in $\mathcal{F}$ containing both $\neg x$ and y. It means that each clause $C = x' \bigvee (\bigvee_{x \in S} \neg x)$ in $\mathcal{F}$ defines in $G(\mathcal{F})$ a (not necessarily induced) subgraph which is a star having its center at x' and having its end-points at the vertices $x \in S$. Sometimes we will not distinguish between a variable and the corresponding vertex.

Lemma 4.1 *If a definite Horn clause $C = x' \bigvee (\bigvee_{x \in S} \neg x)$ is a prime implicate of a definite Horn function h given by a Horn CNF $\mathcal{F}$, then $G(\mathcal{F})$ contains a directed path from x to x' for all $x \in S$.*

Proof Since C is a prime implicate of h, the variable x' is included in $\mathcal{R}$ by the forward chaining procedure starting with the set S. Let us terminate the procedure as soon as it includes x' into $\mathcal{R}$, and use the procedure to construct a (not necessarily induced) subgraph of $G(\mathcal{F})$ in the following way (usually called *backward chaining*). First, the vertex x' is included in the subgraph. At the last step of the forward chaining procedure (when x' is included into $\mathcal{R}$) we use a clause $C' = x' \bigvee (\bigvee_{x \in \mathcal{R}'} \neg x)$ such that $C' \in \mathcal{F}$ and all the variables in $\mathcal{R}'$ have already been included in $\mathcal{R}$. We include in the subgraph the star corresponding to C' in $G(\mathcal{F})$ (having its center at x' and its end-points at the vertices $x \in \mathcal{R}'$). Similarly, we shall successively include in the subgraph every star corresponding to a clause of $G(\mathcal{F})$ used by the forward chaining procedure as soon as the center of the star has already been included in the subgraph. In the resulting subgraph all the vertices without incoming arcs belong to S. We remark that every vertex of S has been included in the subgraph, since otherwise, if a subset $S' \subseteq S$ were not included in the subgraph, then by Lemma 3.1 the clause $x' \bigvee (\bigvee_{x \in S \setminus S'} \neg x)$ would be an implicate of $\mathcal{F}$, contradicting the primality of C. Clearly, the constructed subgraph does not have oriented cycles, and the only vertex of the subgraph which does not have outcoming arcs is the vertex x'. Hence, there is a directed path in the subgraph from any other vertex to x'. $\Box$

Note that non-prime definite Horn implicates may contain variables for which the corresponding paths do not exist. For example, if $\mathcal{F} = (\neg x \bigvee y)(\neg z \bigvee w)$, then $\neg x \bigvee \neg z \bigvee w$ is a non-prime definite Horn implicate of the function, and $G(\mathcal{F})$ does not contain a path from x to w.

Lemma 4.2 *If a negative clause $C_1 = \bigvee_{x \in S_1} \neg x$ is an implicate of a Horn function $f = \mathcal{F} \bigwedge C_2$, where $\mathcal{F}$ is a definite Horn CNF, and $C_2 = \bigvee_{x \in S_2} \neg x$ is a prime implicate of f, then for every $x' \in S_2 \setminus S_1$ there exists $x \in S_1 \setminus S_2$ such that the graph $G(\mathcal{F})$ contains a directed path from x to x'.*

Proof If C_1 is an implicate of f, then, by Lemmas 3.1 and 3.2, for every $x' \in S_2 \setminus S_1$ the clause $x' \bigvee C_1$ is an implicate of $\mathcal{F}$. Therefore, since C_2 is a prime implicate of f, there must exist a subset $S' \subseteq S_1$ such that $S' \setminus S_2 \neq \emptyset$, and the clause $x' \bigvee (\bigvee_{x \in S'} \neg x)$ is a prime implicate of $\mathcal{F}$. Hence, by Lemma 4.1, there exists $x \in S_1 \setminus S_2$ such that the graph $G(\mathcal{F})$ contains a directed path from x to x'. □

To any definite Horn CNF $\mathcal{F}$ of a definite Horn function h we associate its graph $G(\mathcal{F})$. We shall use below the graph $G(H(h))$ associated to the CNF $H(h)$ consisting of all the prime implicates of h, and shall simply call this graph the *implicate graph* $G(h)$ associated to the definite Horn function h.

5 Acyclic Horn Functions

In this section we introduce the concept of acyclic Horn functions, generalizing the class of functions represented by acyclic Horn formulae (see [1, 10, 16]), and give an efficient minimization procedure for acyclic Horn functions.

Definition 5.1

- *A definite Horn CNF $\mathcal{F}$ is called* **acyclic** *iff the associated graph $G(\mathcal{F})$ does not contain oriented cycles.*
- *A definite Horn function h is called* **acyclic** *iff it has an acyclic Horn CNF.*
- *A Horn function f is called* **acyclic** *iff its definite Horn component $h(f)$ is acyclic.*

Lemma 5.2 *A definite Horn function h is acyclic iff its implicate graph $G(h)$ does not have oriented cycles.*

Proof Let $\mathcal{F}$ be a Horn CNF of h such that the graph $G(\mathcal{F})$ is acyclic. It follows from Lemma 4.1 that for every arc in $G(h)$ there is an oriented path in $\mathcal{F}$ connecting the tail of the arc to its head. Therefore, if there were a cycle in $G(h)$, then there would be a cycle in $G(\mathcal{F})$, contradicting the assumption of the lemma. □

In order to apply this last lemma to the recognition of the acyclicity of a Horn function given by a Horn CNF, it is necessary to generate first all the prime implicates of the function. This generation in general may be time consuming, since the number of prime implicates of an acyclic definite Horn function may be exponentially large in the length of a Horn CNF representing the function. As an example of such a situation, consider the acyclic definite Horn function h represented by

$$\mathcal{F} = (x_0 \bigvee \bigvee_{i=1}^{n} \neg x) \bigwedge_{i=1}^{n} ((x_i \bigvee \neg x_{n+i})(x_i \bigvee \neg x_{2n+i})).$$

The CNF $\mathcal{F}$ is prime and irredundant, and it contains only $2n+1$ clauses. The function h has $3^n - 1$ additional prime implicates, since every clause of the form $x_0 \bigvee \bigvee_{i=1}^{n} \neg x_{s(i)}$, with $s(i) \in \{i, n+i, 2n+i\}$, is a prime implicate of h.

The following corollary shows that the recognition of the acyclicity of a definite Horn function does not require the generation of all its prime implicates.

Corollary 5.3 *Every prime CNF of an acyclic definite Horn function is acyclic.*

In view of this result, in order to check whether a given Horn CNF represents an acyclic function, it is sufficient to transform this CNF to a prime irredundant one, and then check in linear time whether the graph of the resulting CNF has any oriented cycle. Therefore, this corollary and Lemma 2.2 imply

Corollary 5.4 *The acyclicity of a function represented by a Horn CNF can be checked in quadratic time.*

It should be remarked that a non-prime Horn CNF of an acyclic definite Horn function may not be acyclic even when it is irredundant. For example, the CNF

$$\mathcal{F} = (\neg x \bigvee z)(\neg y \bigvee z)(\neg x \bigvee \neg z \bigvee y)$$

is Horn, irredundant and not acyclic, since the graph $G(\mathcal{F})$ contains the cycle $y \to z \to y$. Nevertheless, $\mathcal{F}$ represents an acyclic definite Horn function, since the equivalent irredundant prime CNF is $(\neg x \vee y)(\neg y \vee z)$.

The following theorem presents a fundamental property of acyclic Horn functions.

Theorem 5.5 *Each prime implicate of an acyclic Horn function is either essential or redundant.*

Proof First we prove the theorem for definite Horn prime implicates. Let $\mathcal{F}$ be an arbitrary prime irredundant CNF of the definite Horn component $h(f)$ of a Horn function f. Theorem 2.5 implies that it is sufficient to prove that every clause of $\mathcal{F}$ is an essential prime implicate of $h(f)$. We prove this by contradiction.

Suppose there is a clause C' in $\mathcal{F}$ which is not an essential prime implicate of $h(f)$. This means that there is a subset $\mathcal{H}'$ of the set $\mathcal{H}(h(f))$ of prime implicates of $h(f)$ such that C' is not in $\mathcal{H}'$, C' is an implicate of $\mathcal{H}'$, and for any $C \in \mathcal{H}'$, the clause C' is not an implicate of $\mathcal{H}' \setminus C$. Then there must exist a clause $C'' \in \mathcal{H}'$ such that C'' is not an implicate of $\mathcal{F} \setminus C'$, since otherwise C' would be an implicate of $\mathcal{F} \setminus C'$, contradicting the irredundancy of $\mathcal{F}$.

Let $C' = x' \vee (\bigvee_{x \in S_1} \neg x) \vee (\bigvee_{x \in S_2} \neg x)$ and $C'' = x'' \vee (\bigvee_{x \in S_1} \neg x) \vee (\bigvee_{x \in S_3} \neg x)$, where $S_2 \bigcap S_3 = \emptyset$, and where x' and x'' may or may not be distinct variables. Since C' is an implicate of $\mathcal{H}'$ but it is not an implicate of $\mathcal{H}' \setminus C''$, it follows from Lemma 3.1 that there must be an oriented path in $G(h(f))$ from x'' to x'. On the other hand, C'' is an implicate of $\mathcal{F}$, but it is not an implicate of $\mathcal{F} \setminus C'$. Therefore, analogously, there must be an oriented path in $G(h(f))$ from x' to x''. Since $G(h(f))$ is acyclic, it follows that $x'' = x'$. Hence, $C'' = x' \vee (\bigvee_{x \in S_1} \neg x) \vee (\bigvee_{x \in S_3} \neg x)$, and $S_2 \neq \emptyset$, $S_3 \neq \emptyset$.

Since C' is an implicate of $\mathcal{H}'$ but it is not an implicate of $\mathcal{H}' \setminus C''$, it follows from Lemma 3.1 that the forward chaining procedure for the CNF $\mathcal{H}'$ starting with the set $S_1 \bigcup S_2$

will include x' into $\mathcal{R}$, using the clause C''. This implies that every variable of S_3 was already included into $\mathcal{R}$ in some previous steps. This fact and Lemma 3.1 imply that for every variable $x^* \in S_3$ the clause $x^* \bigvee (\bigvee_{x \in S_1} \neg x) \bigvee (\bigvee_{x \in S_2} \neg x)$ is an implicate of $\mathcal{H}'$, and hence an implicate of $h(f)$. Therefore, there are sets $S_1^* \subseteq S_1$ and $S_2^* \subseteq S_2$ such that the clause $x^* \bigvee (\bigvee_{x \in S_1^*} \neg x) \bigvee (\bigvee_{x \in S_2^*} \neg x)$ is a prime implicate of $h(f)$ and $S_2^* \neq \emptyset$, since otherwise C'' would not be a prime implicate of $h(f)$. Hence, there is a variable $x^{**} \in S_2$ such that the arc $x^{**} \to x^*$ is in the graph $G(h(f))$. Analogously, for every variable $\hat{x} \in S_2$ there is a variable $\check{x} \in S_3$ such that the arc $\check{x} \to \hat{x}$ is in the graph $G(h(f))$. Since the sets S_2 and S_3 are non-empty and finite, it follows that the graph $G(h(f))$ has an oriented cycle, contradicting the assumption of the theorem. This shows that the prime implicate C' is essential.

Now we shall prove the theorem for positive prime implicates. By Theorem 2.5 it is sufficient to show that each of the sets $\mathcal{N}_i(f)$, $i \in \{1, \ldots, m\}$, (defined in section 2) contains exactly one clause. Suppose that the set $\mathcal{N}_i(f)$, $i \in \{1, \ldots, m\}$, contains two distinct clauses $C_1 = (\bigvee_{x \in S} \neg x) \bigvee (\bigvee_{x \in S_1} \neg x)$ and $C_2 = (\bigvee_{x \in S} \neg x) \bigvee (\bigvee_{x \in S_2} \neg x)$, where $S_1 \bigcap S_2 = \emptyset$, $S_1 \neq \emptyset$, and $S_2 \neq \emptyset$. It follows that C_1 and C_2 are prime implicates of both $h(f) \bigwedge C_1$ and $h(f) \bigwedge C_2$. Therefore, by Lemma 4.2, for every $x' \in S_1$ there exists $x'' \in S_2$ such that the graph $G(h(f))$ contains an oriented path from x'' to x', and on the other hand, for every $x^{**} \in S_2$ there exists $x^* \in S_1$ such that the graph $G(h(f))$ contains an oriented path from x^* to x^{**}. Since the sets S_1 and S_2 are non-empty and finite, it follows that $G(h(f))$ has an oriented cycle, contradicting the assumption of the theorem. $\Box$

Corollary 5.6 *Any acyclic Horn function has a unique irredundant prime CNF, which is at the same time the minimum and the optimum CNF of the function.*

We have seen that an irredundant (but not prime) Horn CNF of an acyclic definite Horn function is not necessarily acyclic. Moreover, such CNF may be considerably longer than the irredundant prime CNFs of the function. For example, the CNF

$$\mathcal{F} = (\bigwedge_{i=1}^{n} (\neg x_i \bigvee z))(\neg y \bigvee z)(\bigwedge_{i=1}^{n} (\neg x_i \bigvee \neg z \bigvee y))$$

is Horn and irredundant. But it is neither prime nor acyclic, since the graph $G(\mathcal{F})$ contains the cycle $y \to z \to y$. The CNF $\mathcal{F}$ has $2n+1$ clauses and $5n+2$ literals, while the equivalent irredundant and prime acyclic Horn CNF $(\neg y \bigvee z)(\bigwedge_{i=1}^{n} (\neg x_i \bigvee y))$ has only $n+1$ clauses and $2n+2$ literals.

The last corollary shows that both the minimization and the optimization of acyclic Horn functions can be done by simply transforming a given Horn CNF to its unique equivalent irredundant prime one. By Lemma 2.2, this transformation can be done in quadratic time. Therefore

Corollary 5.7 *If an acyclic Horn function is given by a Horn CNF, then the function can be both minimized and optimized in quadratic time.*

6 2-Condensation of a Horn Function

In this section we shall introduce the *2-component* of an arbitrary Horn function as the conjunction of all its quadratic definite Horn prime implicates. The 2-component induces an equivalence relation between certain pairs of variables. The function obtained by "condensing" all the equivalent variables of the initial function will be called its *2-condensation*. This section is devoted to the investigation of relations between the Horn CNFs of the initial function and the Horn CNFs of its 2-condensation. The results obtained here lead to the development in the next section of efficient algorithms for optimizing and minimizing "quasi-acyclic" Horn functions – an important generalization of acyclic Horn functions.

A clause is called *quadratic* iff it contains exactly two literals. A CNF is called *quadratic* iff all its clauses are quadratic. A Boolean function is called *quadratic* iff among the CNFs representing it there is at least one which is quadratic. Quadratic functions and CNFs have been extensively investigated (for a survey see [15]). In particular, the knowledge bases of quadratic CNFs were recently studied in [11].

We shall denote by $\mathcal{H}^2(f)$ the set of quadratic definite Horn prime implicates of a Horn function f, and by $H^2(f)$ the conjunction of all the clauses in $\mathcal{H}^2(f)$. It can be easily seen that the graph $G(H^2(f))$ is transitively closed.

Lemma 6.1 *Given any Horn CNF $\mathcal{F}$ of a Horn function f, the set $\mathcal{H}^2(f)$ and the graph $G(H^2(f))$ can be constructed in $O(nL)$ time, where n is the number of variables and L is the length of the given CNF.*

Proof By Lemma 3.3, for every variable x the forward chaining procedure for the definite Horn part of $\mathcal{F}$ starting with x will output in linear time $O(L)$ the set $\mathcal{R}$. It follows from Lemma 3.1 that $\neg x \bigvee y$ is an implicate of f iff $y \in \mathcal{R} \setminus \{x\}$. Since f does not have unit prime implicates, and the total number of variables is n, all the quadratic definite Horn prime implicates of f will be constructed in $O(nL)$ time. □

Remark that the algorithm described in the above proof will take $O(nL)$ time even if the function does not have any quadratic definite Horn prime implicates. An adaptive algorithm for the construction of all quadratic definite Horn prime implicates, whose worst case computational complexity is also $O(nL)$, is developed in [3].

For any Horn function f we shall denote by $V_r(f)$, $r \in \{1, \dots, q\}$, the strongly connected components of $G(H^2(f))$, and define $S_r(f) = \{i \mid x_i \in V_r(f)\}$. Let us assume that $|V_r(f)| > 1$ for $r \leq p$, and $|V_r(f)| = 1$ for $r > p$. The graph $G(H^2(f))$ is acyclic iff $p = 0$ (i.e. every $V_r(f)$ consists of a single vertex).

If a clause in $\mathcal{H}^2(f)$ corresponds to an arc belonging to a strongly connected component $V_r(f)$ of $G(H^2(f))$, we shall say that the clause belongs to $V_r(f)$. Let $\mathcal{H}^2_*(f)$ denote the set of all those clauses in $\mathcal{H}^2(f)$, whose associated arcs in $G(H^2(f))$ belong to strongly connected components; let $H^2_*(f)$ denote the conjunction of all these clauses.

Definition 6.2 *For any Horn function f, the quadratic definite Horn function represented by the CNF $H^2_*(f)$ is called the* **2-component** *of f and is denoted by $q(f)$.*

In order to illustrate the above concepts let us consider the following modification of an example from [4]:

$$\mathcal{F}(h) = (\bigwedge_{i=1}^{n}((\neg x_i \bigvee x_{i+n})(x_i \bigvee \neg x_{i+n}))) \bigwedge (x_0 \bigvee \bigvee_{i=1}^{n} \neg x_i).$$

$\mathcal{F}(h)$ is an irredundant prime CNF of the definite Horn function h. In addition to $2n+1$ clauses in $\mathcal{F}(h)$ the function h has $2^n - 1$ additional prime implicates of the form

$$x_0 \bigvee \bigvee_{i=1}^{n} \neg x_{s(i)},$$

with $s(i) \in \{i, n+i\}$, thus the number of prime implicates of this definite Horn function h is $2^n + 2n$. For this function h we have:

$$H_*^2(h) = \bigwedge_{i=1}^{n}((\neg x_i \bigvee x_{i+n})(x_i \bigvee \neg x_{i+n})),$$

and the graph $G(H^2(h))$ consists of n strongly connected components of the form $\{i, n+i\}$.

The following lemma characterizes completely all the minimum CNFs of $q(f)$.

Lemma 6.3 *A CNF $\mathcal{F}$ is a minimum prime CNF of the 2-component $q(f)$ of an arbitrary Horn function f iff*

$$\mathcal{F} = \bigwedge_{r=1}^{p} \bigwedge_{i \in S_r(f)} (x_i \bigvee \neg x_{\pi_r(i)}),$$

where π_r is a cyclic permutation of $S_r(f)$, $r \in \{1, \ldots, p\}$.

Proof Clearly, all the prime implicates of $q(f)$ are quadratic and definite Horn, the graph $G(H_*^2(f))$ is transitively closed, and $\neg x \vee y$ is a prime implicate of $q(f)$ iff $G(H_*^2(f))$ contains the arc $x \to y$. It follows from Lemma 4.1 that a quadratic definite Horn CNF $\mathcal{F}$ represents $q(f)$ iff the transitive closure of $G(\mathcal{F})$ is the graph $G(H_*^2(f))$. In other words, the graph $G(\mathcal{F})$ must have the same strongly connected components as the graph $G(H_*^2(f))$. It is known that a directed graph is strongly connected iff it has an oriented cycle going through all its vertices. Such a cycle consists of a minimum number of arcs iff it is a simple cycle. Therefore, $\mathcal{F}$ is a minimum prime CNF iff every strongly connected component of $G(\mathcal{F})$ is a simple cycle. □

It can be easily checked that for any Horn function f no prime implicate in

$$(\mathcal{N}(f) \bigcup \mathcal{H}(f)) \setminus \mathcal{H}_*^2(f)$$

can contain two variables belonging to the same strongly connected component of $G(H^2(f))$.

Definition 6.4 *Let f be a Horn function, and let $\mathcal{F}(f)$ be a prime CNF of f; the* **2-condensation** *of $\mathcal{F}(f)$ is the CNF $\mathcal{F}^c(f)$, obtained by replacing all the variables x_i, $i = 1, \ldots, n$, belonging to the same strongly connected component $V_r(f)$ by the same variable v_r $(r = 1, \ldots, q)$; the function $c(f)$ represented by $\mathcal{F}^c(f)$ is called a* **2-condensation** *of f.*

For every variable x we shall denote by $v(x)$ the variable corresponding to the strongly connected component of $G(H^2(f))$ containing x.

Lemma 6.5 *If $\mathcal{F}_1(f)$ and $\mathcal{F}_2(f)$ are arbitrary prime CNFs of a Horn function f, then their 2-condensations $\mathcal{F}_1^c(f)$ and $\mathcal{F}_2^c(f)$ represent the same function, i.e. every Horn function has a unique 2-condensation.*

Proof It is sufficient to show that each clause of $\mathcal{F}_2^c(f)$ is an implicate of $\mathcal{F}_1^c(f)$, and each clause of $\mathcal{F}_1^c(f)$ is an implicate of $\mathcal{F}_2^c(f)$. Let $\mathcal{F}_1(h)$ and $\mathcal{F}_2(h)$ denote the definite Horn parts of $\mathcal{F}_1(f)$ and $\mathcal{F}_2(f)$ respectively, and $\mathcal{F}_1^c(h)$ and $\mathcal{F}_2^c(h)$ denote their 2-condensations, which by definition are the definite Horn parts of $\mathcal{F}_1^c(f)$ and $\mathcal{F}_2^c(f)$ respectively. We shall first show that each clause of $\mathcal{F}_2^c(h)$ is an implicate of $\mathcal{F}_1^c(h)$, and each clause of $\mathcal{F}_1^c(h)$ is an implicate of $\mathcal{F}_2^c(h)$.

If $C = x' \vee \bigvee_{x \in S} \neg x \in \mathcal{F}_2(h)$ and $C \notin \mathcal{H}_*^2(h)$, then by Lemma 3.1 x' will be included into the set $\mathcal{R}$ by the forward chaining procedure for $\mathcal{F}_1(h)$ starting with the set S. Let us consider the forward chaining procedure for $\mathcal{F}_1^c(h)$ starting with the set $v(S) = \{v(x) \mid x \in S\}$. In the latter procedure we shall skip every step of the former procedure in which a clause $\neg x'' \vee x''' \in \mathcal{H}_*^2(h)$ is used, since $v(x'') = v(x''')$ and by that step $v(x'')$ has already been included in $\mathcal{R}$. Obviously, for every other step of the former procedure there is a corresponding step of the latter procedure which can be reproduced. Therefore, the latter procedure will eventually include $v(x')$ into $\mathcal{R}$. Therefore, by Lemma 3.1, $v(x') \vee \bigvee_{x \in S} \neg v(x)$ is an implicate of $\mathcal{F}_1^c(h)$. Thus, we have proved that $\mathcal{F}_1^c(h)$ and $\mathcal{F}_2^c(h)$ represent the same definite Horn function.

Now we may assume that $\mathcal{F}_1(f)$ and $\mathcal{F}_2(f)$ contain the same definite Horn clauses which form the CNF $\mathcal{F}_1(h)$. Both $\mathcal{F}_1(f)$ and $\mathcal{F}_2(f)$ must contain at least one clause from every set $\mathcal{N}_i(f)$ (defined in section 2), $i \in \{1, \ldots, m\}$. Let C_1 and C_2 be in $\mathcal{N}_i(f)$, and $C_1 \in \mathcal{F}_1(f)$, $C_2 \in \mathcal{F}_2(f)$. Let C_1^c and C_2^c be the "images" of C_1 and C_2 respectively after the substitution. Repeating a similar argument to that in the previous paragraph and using Lemma 3.2 (instead of Lemma 3.1), it can now be easily seen that C_2^c is an implicate of $C_1^c \bigwedge \mathcal{F}_1^c(h)$, and C_1^c is an implicate of $C_2^c \bigwedge \mathcal{F}_1^c(h)$. □

The following statement relates the prime implicates of a Horn function to the prime implicates of its 2-condensation.

Lemma 6.6

- *A definite Horn clause $C = x' \vee \bigvee_{x \in S} \neg x$ is a prime implicate of a Horn function f iff either $C \in \mathcal{H}_*^2(f)$, or the definite Horn clause $v(x') \vee \bigvee_{x \in S} \neg v(x)$ is a prime implicate of $c(f)$.*

- *A negative clause $C = \bigvee_{x \in S} \neg x$ is a prime implicate of f iff the negative clause $\bigvee_{x \in S} \neg v(x)$ is a prime implicate of $c(f)$.*

Proof The proof of Lemma 6.5 shows that if $C = x' \vee \bigvee_{x \in S} \neg x$ is a prime implicate of f and $C \notin \mathcal{H}_*^2(f)$, then $v(x') \vee \bigvee_{x \in S} \neg v(x)$ is an implicate of $c(f)$. To complete the proof of the first statement it is sufficient to show that if $C^c = v' \vee \bigvee_{v\in S^c} \neg v$ is an implicate of $c(f)$, and if $x(v)$ is an arbitrary variable which belongs to the strongly connected component of $G(H^2(f))$ corresponding to v, then $x(v') \vee \bigvee_{v\in S^c} \neg x(v)$ is an implicate of f. This fact is obtained by using Lemma 3.1 and reversing the arguments in the proof of Lemma 6.5: if $\mathcal{F}(h)$ is a prime CNF of $h(f)$, and $\mathcal{F}^c(h)$ is its 2-condensation, then the forward chaining procedure for $\mathcal{F}^c(h)$ can be obviously reproduced as the forward chaining procedure for $\mathcal{F}(h) \bigwedge H^2_*(f)$. The same reasoning (using Lemma 3.2 instead of Lemma 3.1) proves the second statement of the lemma. □

Given a Horn CNF of a function f, according to Lemma 6.1 and using the linear time algorithm for the determination of the strongly connected components of a directed graph (see [19]), the CNF $H^2_*(f)$ representing the 2-component $q(f)$ can be constructed in $O(nL)$ time, where n is the number of variables and L is the length of the given CNF. Therefore,

Corollary 6.7 *Given a Horn CNF of an arbitrary Horn function, a prime CNF of its 2-condensation can be constructed in $O(nL + L^2)$ time, where n is the number of variables and L is the length of the given CNF.*

Lemmas 6.5 and 6.6 imply that the 2-condensation of a prime CNF is prime. However, the following example shows that the *irredundancy* of CNFs is not preserved by 2-condensation. The CNF

$$\mathcal{F}(f) = (\neg x_{11} \bigvee x_{12})(\neg x_{12} \bigvee x_2)(\neg x_{12} \bigvee x_3)(\neg x_2 \bigvee \neg x_3 \bigvee x_{11})(\neg x_3 \bigvee x_4)(\neg x_2 \bigvee \neg x_4 \bigvee x_{12})$$

is prime and irredundant. The strongly connected components of $G(H^2(f))$ are $\{x_{11}, x_{12}\}$, $\{x_2\}$, $\{x_3\}$, and $\{x_4\}$, and the 2-condensation of $\mathcal{F}(f)$,

$$\mathcal{F}^c(f) = (\neg v_1 \bigvee v_2)(\neg v_1 \bigvee v_3)(\neg v_3 \bigvee v_4)(\neg v_2 \bigvee \neg v_3 \bigvee v_1)(\neg v_2 \bigvee \neg v_4 \bigvee v_1),$$

is not irredundant, since the clause $\neg v_2 \vee \neg v_3 \vee v_1$ can be dropped. An equivalent irredundant prime CNF (which turns out to be unique) is

$$(\neg v_1 \bigvee v_2)(\neg v_1 \bigvee v_3)(\neg v_3 \bigvee v_4)(\neg v_2 \bigvee \neg v_4 \bigvee v_1).$$

For every variable v let $x(v)$ be an arbitrary variable belonging to the strongly connected component of $G(H^2(f))$ corresponding to v. A *selector* σ of f associates to every definite Horn clause $C = v' \vee \bigvee_{v\in S} \neg v$ a clause $\sigma(C) = x(v') \vee \bigvee_{v\in S} \neg x(v)$, and to every negative clause $C = \bigvee_{v\in S} \neg v$ a clause $\sigma(C) = \bigvee_{v\in S} \neg x(v)$. Note that when a variable v is present in two different clauses C' and C'', the variables $x(v)$ in the clauses $\sigma(C')$ and $\sigma(C'')$ may be different. If $\mathcal{F}$ is a Horn CNF of $c(f)$, then the CNF $\bigwedge_{C\in\mathcal{F}} \sigma(C)$ will be denoted by $\sigma(\mathcal{F})$.

Theorem 6.8 *If f is a Horn function, $\mathcal{F}(q(f))$ is an arbitrary prime CNF of the 2-component $q(f)$ of f, $\mathcal{F}(c(f))$ is an arbitrary prime CNF of the 2-condensation $c(f)$ of f, and σ is an arbitrary selector of f, then*

$$\mathcal{F} = \mathcal{F}(q(f)) \bigwedge \sigma(\mathcal{F}(c(f)))$$

is a prime CNF of f. Moreover, if $\mathcal{F}(c(f))$ is irredundant, then the CNF $\mathcal{F} \setminus \sigma(C)$ does not represent f for any $C \in \mathcal{F}(c(f))$.

Proof Obviously, $\mathcal{F}$ consists of prime implicates of f, and every clause in $\mathcal{H}^2_*(f)$ is a prime implicate of $\mathcal{F}$. In order to prove that the CNF $\mathcal{F}$ represents f, it is sufficient to show that every prime implicate of f is also an implicate of $\mathcal{F}$. This fact can be shown as in the proof of Lemma 6.6, by reproducing the forward chaining procedure for $\mathcal{F}(c(f))$ for the corresponding prime implicate of $c(f)$ as a forward chaining procedure for $\mathcal{H}^2_*(f) \bigvee \mathcal{F}$.

The irredundancy part of the theorem can be proved by contradiction. If there were a clause $C \in \mathcal{F}(c(f))$ such that the CNF $\mathcal{F} \setminus \sigma(C)$ represented f (i.e. if every prime implicate of f were an implicate of $\mathcal{F} \setminus \sigma(C^v)$), then the CNF $\mathcal{F}(c(f)) \setminus C$ would represent $c(f)$ (i.e. every prime implicate of $c(f)$ would be an implicate of $\mathcal{F}(c(f)) \setminus C$) by the same argument as in the proof of Lemma 6.5, contradicting the assumption of the theorem. □

Theorem 6.8 implies in particular that a CNF $\mathcal{F}_N$ is a negative restriction of a Horn function f iff $\mathcal{F}_N = \sigma(\mathcal{F}^c_N)$, where $\mathcal{F}^c_N$ is a negative restriction of $c(f)$ and σ is a selector of f.

Remark that it may happen that no minimum CNF of a Horn function is of the form described in Theorem 6.8. Consider for example a definite Horn function f represented by the CNF

$$\mathcal{F}(f) = (\neg x_{11} \bigvee x_{12})(\neg x_{12} \bigvee x_2)(\neg x_{12} \bigvee x_3)(\neg x_2 \bigvee \neg x_3 \bigvee x_{11}),$$

having 9 literals. The strongly connected components of $G(H^2(f))$ are $\{x_{11}, x_{12}\}$, $\{x_2\}$, and $\{x_3\}$, and the 2-component is

$$q(f) = (\neg x_{11} \bigvee x_{12})(\neg x_{12} \bigvee x_{11}).$$

The CNF $\mathcal{F}(f)$ can be seen to be minimum, and it does not contain a CNF of $q(f)$. Since the minimum CNF of the 2-condensation $c(f)$ of f is

$$\mathcal{F}^c(f) = (\neg v_1 \bigvee v_2)(\neg v_1 \bigvee v_3)(\neg v_2 \bigvee \neg v_3 \bigvee v_1)$$

having 7 literals, any CNF of f containing a CNF of $q(f)$ will have at least 11 literals, and therefore it cannot be minimum.

7 Quasi-Acyclic Horn Functions

In this section we introduce the concept of *quasi-acyclic* Horn functions which generalizes the practically important classes of acyclic and quadratic Horn functions considered above. We present below a polynomial solution of both the minimization and the optimization problems for the case of quasi-acyclic Horn functions.

Definition 7.1 *A Horn function is called* **quasi-acyclic** *iff its 2-condensation is acyclic.*

Note that, in contrast with the case of acyclic Horn functions, graphs of prime CNFs of definite Horn components of quasi-acyclic Horn functions may contain oriented cycles. Remark that a quasi-acyclic Horn function f is acyclic iff every strongly connected component of $G(H^2(f))$ consists of a single vertex ($p = 0$).

Lemma 7.2 *The recognition of the quasi-acyclicity of a function represented by a Horn CNF can be performed in time quadratic in the length of the given CNF.*

Proof The recognition of quasi-acyclicity of a function f represented by a Horn CNF $\mathcal{F}$ can be established performing the following steps.

Step 1: *transform the given Horn CNF $\mathcal{F}$ to an equivalent irredundant prime CNF $\mathcal{F}'$;*

Step 2: *use the forward chaining procedure (as in Lemma 6.1) to construct all the quadratic definite Horn prime implicates of f;*

Step 3: *determine the strongly connected components of the graph $G(H^2(f))$ associated to the quadratic Horn CNF $H^2(f)$ obtained in Step 2;*

Step 4: *construct the 2-condensation $\mathcal{F}^c$ of the CNF $\mathcal{F}'$;*

Step 5: *check whether the graph $G(\mathcal{F}^c)$ contains an oriented cycle.*

By Definition 7.1, the function f is quasi-acyclic iff the graph $G(\mathcal{F}^c)$ has no oriented cycles. It follows from Lemmas 2.2 and 6.5 and from the linear time complexity of the graph algorithms used here (see [19]), that the computational complexity of the given algorithm is $O(nL+L^2)$, where n is the number of variables and L is the length of the given CNF. $\Box$

The following lemma states an important property of quasi-acyclic Horn functions.

Lemma 7.3 *Given a prime CNF $\mathcal{F}$ of a quasi-acyclic Horn function f, there is a subset of clauses of $\mathcal{F}$ whose conjunction is a prime CNF of the 2-component $q(f)$.*

Proof It is sufficient to show that for any x' and x'' belonging to the same strongly connected component of $G(H^2(f))$ the graph $G(\mathcal{F}\bigcap\mathcal{H}^2_*(f))$ contains an oriented path from x' to x''. Assume no such path exists. Since $\neg x'\bigvee x''$ is a prime implicate of f, by Lemma 4.1 there must be an oriented path from x' to x'' in the graph $G(\mathcal{F})$. Since at least one arc in this path has to correspond to a clause in $\mathcal{F}\setminus\mathcal{H}^2_*(f)$, this path goes through some vertex which does not belong to the strongly connected component of $G(H^2(f))$ containing x' and x''. Let $\mathcal{F}^c$ be the 2-condensation of $\mathcal{F}$. Since x' and x'' correspond to the same variable v', the graph $G(\mathcal{F}^c)$ must have an oriented cycle containing the vertex v', in contradiction with the fact that $c(f)$ is quasi-acyclic. Therefore, $\mathcal{F}\bigcap H^2_*(f)$ must be a prime CNF of the 2-component $q(f)$ of f. $\Box$

The following theorem completely describes all the irredundant prime CNFs of a quasi-acyclic Horn function.

Theorem 7.4 *A conjunctive normal form $\mathcal{F}$ is an irredundant prime CNF of a quasi-acyclic Horn function f if and only if*

$$\mathcal{F}=\mathcal{F}(q(f))\bigwedge\sigma(\mathcal{F}_{\min}(c(f))),$$

where $\mathcal{F}(q(f))$ is an irredundant prime CNF of the 2-component $q(f)$ of f, $\mathcal{F}_{\min}(c(f))$ is the (unique) minimum prime CNF of the 2-condensation $c(f)$ of f, and σ is a selector of f.

Proof We shall prove first the **if** part of the theorem. By Theorem 6.8 and Lemma 6.6, the CNF $\mathcal{F}$ is a prime CNF of f. Theorem 6.8 states that no clause in $\sigma(\mathcal{F}_{\min}(c(f)))$ can be dropped from $\mathcal{F}$ without changing the function. No clause in $\mathcal{F}(q(f))$ can be dropped from $\mathcal{F}$ either, since by Lemma 7.3 the CNF $\mathcal{F}$ must contain a prime CNF of $q(f)$. Therefore, $\mathcal{F}$ is not only a prime but also an irredundant CNF of f.

We shall prove now the **only if** part of the theorem. Lemma 7.3 implies that $\mathcal{F} \bigcap H_*^2(f)$ is an irredundant prime CNF of $q(f)$. All what is left to be proved is that the 2-condensation of any irredundant prime CNF $\mathcal{F}$ of f is not only a prime, but also an irredundant CNF of $c(f)$. This can be shown by contradiction, using simulated forward chaining procedures and Lemma 7.3. □

The following corollary of Theorem 7.4 together with Lemma 6.3 completely describes all the minimum prime CNFs of a quasi-acyclic Horn function:

Theorem 7.5 (Main) *A conjunctive normal form $\mathcal{F}$ is a minimum prime CNF of a quasi-acyclic Horn function f if and only if*

$$\mathcal{F} = \mathcal{F}(q(f)) \bigwedge \sigma(\mathcal{F}_{\min}(c(f))),$$

where $\mathcal{F}(q(f))$ is a minimum prime CNF of the 2-component $q(f)$ of f, $\mathcal{F}_{\min}(c(f))$ is the (unique) minimum prime CNF of the 2-condensation $c(f)$ of f, and σ is a selector of f.

This result implies that all the minimum prime CNFs of a quasi-acyclic Horn function have exactly the same structure. In particular,

Corollary 7.6 *A prime CNF of a quasi-acyclic Horn function is minimum iff it is optimum.*

In order to illustrate the obtained results, let us consider the following Horn CNF

$$\mathcal{F} = (\neg x \bigvee y)(\neg y \bigvee x)(\neg x \bigvee z)(\neg y \bigvee w)(\neg z \bigvee \neg w \bigvee u)(\neg u \bigvee s)(\neg s \bigvee u)(\neg s \bigvee t)(\neg t \bigvee s).$$

It can be seen that $\mathcal{F}$ is irredundant and prime, but it is not acyclic, since the graph $G(\mathcal{F})$ contains oriented cycles, e.g. $x \to y \to x$ (see Figure 1).

The conjunction of all the quadratic definite Horn prime implicates of the function f represented by $\mathcal{F}$ is $H^2(f)$, and its graph $G(H^2(f))$ is presented in Figure 2. The strongly connected components of $G(H^2(f))$ are $\{x, y\}$, $\{u, s, t\}$, $\{z\}$, and $\{w\}$, and the 2-condensation $\mathcal{F}^c$ of the CNF $\mathcal{F}$,

$$\mathcal{F}^c = (\neg v_1 \bigvee v_4)(\neg v_1 \bigvee v_3)(\neg v_3 \bigvee \neg v_4 \bigvee v_2)$$

is acyclic. Therefore, the function f is quasi-acyclic, and one of its minimum CNFs is

$$\mathcal{F} = (\neg x \bigvee y)(\neg y \bigvee x)(\neg x \bigvee w)(\neg y \bigvee z)(\neg z \bigvee \neg w \bigvee s)(\neg u \bigvee s)(\neg s \bigvee t)(\neg t \bigvee u).$$

It should be noted that an irredundant prime CNF of a quasi-acyclic Horn function may be almost twice as long as an equivalent minimum prime CNF. More precisely,

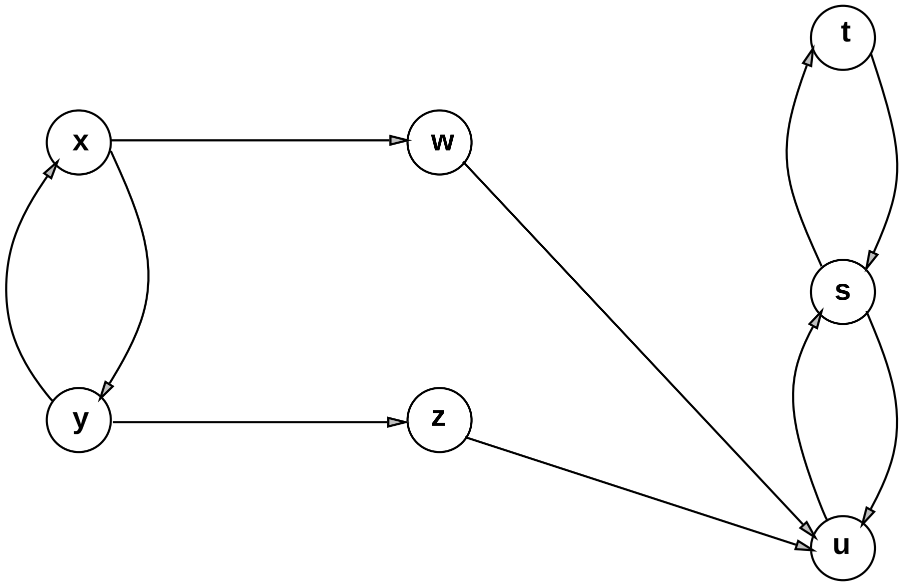

Figure 1: Graph $G(\mathcal{F})$

Lemma 7.7 *Given any quasi-acyclic Horn function f, the ratio of the number of clauses (the length) of any irredundant prime CNF of f to the number of clauses (the length) of a minimum prime CNF of f is at most $2 - \frac{2}{n}$, where n is the number of variables; this bound is sharp.*

Proof Theorem 7.4 implies that the number of those clauses of an irredundant prime CNF of a quasi-acyclic Horn function f which do not belong to $\mathcal{H}^2_*(f)$ is constant. Therefore, the ratio defined in the statement of the lemma will achieve its maximum at a function the 2-condensation of which is identically 0. Hence, it is sufficient to determine the maximum ratio for the quadratic function h, a minimum prime CNF of which is

$$\mathcal{F}^m = (\neg x_1 \bigvee x_n)(\bigwedge_{i=1}^{n-1}(x_i \bigvee \neg x_{i+1}))$$

(see Lemma 6.3); this minimum CNF has only n clauses. On the other hand, the function h can be represented by the following irredundant prime conjunctive normal form

$$\bigwedge_{i=1}^{n-1}((x_i \bigvee \neg x_n)(\neg x_i \bigvee x_n)),$$

which has $2n-2$ clauses. We shall prove now that any irredundant prime CNF $\mathcal{F}$ of the function h cannot have more than $2n-2$ clauses. Let us fix an arbitrary vertex x' in the graph $G(\mathcal{F})$. We can construct oriented paths from x' to all the other vertices of the graph using $n-1$ arcs, since each used arc adds a vertex to the set of vertices reachable from x'. On the other hand, it is sufficient to use $n-1$ more arcs to construct oriented paths from all

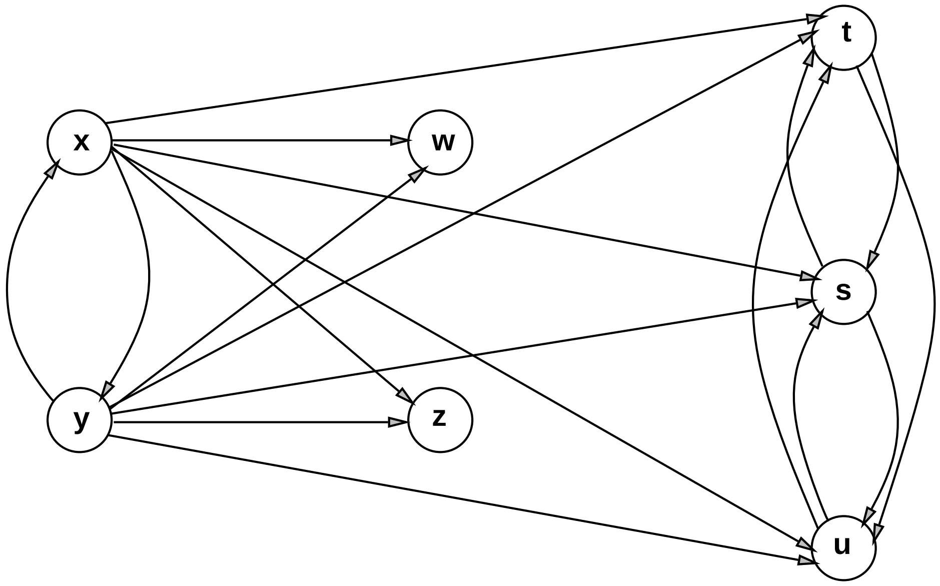

Figure 2: Graph $G(H^2(f))$

the other vertices to x', since each used arc adds a vertex to the set of vertices which reach x'. $\Box$

From a computational point of view, the main result of this paper is the following

Theorem 7.8 *The minimization and optimization of a quasi-acyclic Horn function represented by a Horn CNF can be performed in time quadratic in the length of the given CNF.*

Proof Theorem 7.5 implies that a quasi-acyclic Horn function represented by a Horn CNF can be simultaneously minimized and optimized performing the following quadratic time algorithm.

Step 1: *transform the given Horn CNF to an equivalent irredundant and prime one;*

Step 2: *extract the quadratic definite Horn clauses in the resulting CNF;*

Step 3: *determine the strongly connected components of the graph associated to the quadratic Horn CNF obtained in Step 2;*

Step 4: *replace in the resulting irredundant and prime CNF the conjunction of all the clauses of $\mathcal{H}^2_*(f)$ with a minimum prime CNF of $q(f)$ (obtained as in Lemma 6.3).* $\Box$

References

[1] D. Angluin. *Learning Propositional Horn Sentences With Hints.* Technical Report YALE/DCS/RR-590, Department of Computer Science, Yale University, 1987.

[2] D. Angluin, M. Frazier, L. Pitt. Learning conjunctions of Horn clauses. *Machine Learning*, **9**(1992), 147–164.

[3] E. Boros and O. Čepek. *Restricted Consensus Method and Quadratic Implicants of Pure Horn Formulas.* Unpublished manuscript, RUTCOR, Rutgers University, New Brunswick, NJ, 1993.

[4] E. Boros, Y. Crama, P.L. Hammer. Polynomial-time inference of all valid implications for Horn and related formulae. *Annals of Mathematics and Artificial Intelligence*, **1**(1990), 21–32.

[5] E. Charles and O. Dubois. MELODIA: Logical methods for checking knowledge bases. In: *Validation, Verification and Test of Knowledge-based Systems*, Eds.: M. Ayel and J.-P. Laurent, Wiley & Sons 1991, 95–105.

[6] R.M. Colomb and C.Y.C. Chung. Very fast decision table execution of propositional expert systems. *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90)*, 1990, 671–676.

[7] R. Dechter and J. Pearl. Structure identification in relational data. *Artificial Intelligence*, **58**(1992), 237–270.

[8] W.F. Dowling and J.H. Gallier. Linear time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, **3**(1984), 267–284.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of* **NP**-*Completeness.* W.H. Freeman and Company, San Francisco, 1979.

[10] A. Ginsberg. Knowledge-base reduction: A new approach to checking knowledge bases for inconsistency & redundancy. *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-88)*, 1988, 585–589.

[11] F. Glover and H.J. Greenberg. Logical testing for rule-base management. *Annals of Operations Research*, **12**(1988), 199–215.

[12] P.L. Hammer and A. Kogan. Horn functions and their DNFs. *Information Processing Letters*, **44**(1992), 23–29.

[13] P.L. Hammer and A. Kogan. *Horn Function Minimization and Knowledge Compression in Production Rule Bases.* RUTCOR Research Report RRR 8-92, Rutgers University, New Brunswick, NJ, March 1992.

[14] P.L. Hammer and A. Kogan. *Optimal Compression of Propositional Horn Knowledge Bases: Complexity and Approximation.* RUTCOR Research Report RRR 1-93, Rutgers University, New Brunswick, NJ, January 1993.

[15] P.L Hammer and B. Simeone. Quadratic functions of binary variables. In: *Combinatorial Optimization*, Ed.: B. Simeone, *Lecture Notes in Mathematics*, **1403**, Springer-Verlag, 1989, 1–56.

[16] T.A. Nguyen, W.A. Perkins, T.J. Laffey, D. Pecora. Knowledge Base Verification. *AI Magazine*, Summer 1987, Vol. 8, No. 2, 69–75.

[17] W. Quine. A way to simplify truth functions. *American Mathematical Monthly*, **62**(1955), 627–631.

[18] B. Selman and H. Kautz. Knowledge compilation using Horn approximations. *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-91)*, 1991, 904–909.

[19] R.E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, **1**(1972), # 2, 146–160.

[20] J. Tepandi. Comparison of expert system verification criteria: redundancy. In: *Validation, Verification and Test of Knowledge-based Systems*, Eds.: M. Ayel and J.-P. Laurent, Wiley & Sons 1991, 49–62.