[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

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

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

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

[14] 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.

Since for any $C \in \mathcal{F}^*$ we have $\#(\mathcal{F}(C)) \leq n-1$, it follows that

$$\#(\mathcal{F}) \leq \sum_{C \in \mathcal{F}^*} (\#\mathcal{F}(C)) \leq (n-1)(\#(f)). \ \square$$

The bound in Theorem 5.1 is not far from being sharp. It can be seen that the CNF (1) (in the introduction) is irredundant, and the CNF (2) is the minimum CNF of that function. The CNFs are over $2n$ variables, and we have $\#(1) = \frac{2n+1}{2}(\#(2))$.

It was observed in [4] that given any Horn CNF $\mathcal{F}$ of a function f, the algorithm HORN in [1] can be used to construct an equivalent CNF $\mathcal{F}'$ such that $\#(\mathcal{F}') \leq (n+1)(\#(f))$. Theorem 5.1 and Lemma 2.2 show that by transforming a given CNF to an equivalent irredundant one a slightly better result can be achieved faster.

Let $l(\mathcal{F})$ denote the length of a CNF $\mathcal{F}$, and $l(f)$ denote the length of a minimum CNF of a Boolean function f.

Corollary 5.2 *If $\mathcal{F}$ is an arbitrary irredundant Horn CNF of a Horn function f over n variables, then $l(\mathcal{F}) \leq \binom{n}{2} l(f)$.*

Proof Since the function f has no unit prime implicates, we have $l(f) \geq 2(\#(f))$. Therefore,

$$l(\mathcal{F}) \leq n(\#(\mathcal{F})) \leq n(n-1)(\#(f)) \leq \frac{n(n-1)}{2} l(f). \ \square$$

Corollary 5.2 proves that the algorithm in [9], which transforms any Horn CNF to an equivalent irredundant and prime one, can be used to create in time quadratic in the length of a given Horn production rule base an equivalent Horn base, the size of which does not exceed significantly the minimum possible one.

It was shown in [10] that the problem of Horn function minimization can be solved in quadratic time for the special class of *quasi-acyclic* Horn functions, which properly includes the practically important classes of acyclic and quasi-quadratic Horn functions.

References

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

[2] 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.

5 Length of Irredundant Horn CNFs

While the construction of the shortest CNF representation of a Horn function turns out to be computationally intractable, we shall prove below that the length of any irredundant Horn CNF of a Horn function cannot be "substantially larger" than the length of a minimum CNF of the function.

Let $\#(\mathcal{F})$ denote the number of clauses in a CNF $\mathcal{F}$, and $\#(f)$ denote the minimum of $\#(\mathcal{F})$ taken over all CNF of a Boolean function f.

Theorem 5.1 *If $\mathcal{F}$ is an arbitrary irredundant Horn CNF of a Horn function f over n variables, then $\#(\mathcal{F}) \leq (n-1)(\#(f))$.*

Proof Let $\mathcal{F}_h$ be the definite Horn part and $\mathcal{F}_n$ the negative part of $\mathcal{F}$ respectively, i.e. $\mathcal{F} = \mathcal{F}_h \bigwedge \mathcal{F}_n$. Let $\mathcal{F}^*$ be an irredundant prime CNF of f such that $\#(\mathcal{F}^*) = \#(f)$. For each clause $C \in \mathcal{F}^*$ let us denote by $\mathcal{F}(C)$ a conjunction of the minimum number of clauses in $\mathcal{F}$ such that C is an implicate of $\mathcal{F}(C)$. Since the CNF $\mathcal{F}$ is irredundant, we have $\bigcup_{C \in \mathcal{F}^*} \mathcal{F}(C) = \mathcal{F}$ (where the CNFs are interpreted as the sets of clauses).

Let C be a definite Horn clause in $\mathcal{F}^*$, and $C = x' \bigvee \bigvee_{x \in S} \neg x$. Obviously, $\mathcal{F}(C) \subseteq \mathcal{F}_h$, and by Lemma 3.1 the forward chaining procedure for $\mathcal{F}(C)$ starting with the set S includes eventually the variable x' into the set $\mathcal{R}$. The minimality of $\mathcal{F}(C)$ implies that all the clauses in $\mathcal{F}(C)$ have to be used in this procedure. Since for each variable in $\mathcal{R} \setminus S$ exactly one clause of $\mathcal{F}(C)$ is used in the procedure, we have

$$\#(\mathcal{F}(C)) \leq n - |S| \leq n - 1$$

(the set S is not empty, since the function f has no unit prime implicates).

Let C be a negative clause in $\mathcal{F}^*$, and $C = \bigvee_{x \in S_1} \neg x$. The structure and completeness of the *resolution (consensus)* procedure (see e.g. [12]) imply (similarly to [9]) that $\mathcal{F}(C)$ contains just one negative clause, i.e. $\mathcal{F}(C) = C' \bigwedge \mathcal{F}_h(C)$, where $C' = \bigvee_{x \in S_2} \neg x \in \mathcal{F}_n$ and $\mathcal{F}_h(C) \subseteq \mathcal{F}_h$. By Lemma 3.2 the forward chaining procedure for $\mathcal{F}_h(C)$ starting with the set S_1 includes eventually all the variables in S_2 into the set $\mathcal{R}$. The minimality of $\mathcal{F}(C)$ implies that all the clauses in $\mathcal{F}_h(C)$ have to be used in this procedure. Since for each variable in $\mathcal{R} \setminus S_1$ exactly one clause of $\mathcal{F}_h(C)$ is used in the procedure, it follows that

$$\#(\mathcal{F}(C)) \leq 1 + n - |S_1| \leq n - 1$$

(we have $|S_1| \geq 2$, since the function f has no unit prime implicates).

Let $\mathcal{F}_H^*$ denote the conjunction of all the prime implicates of $h(f)$. Obviously, a clause $C \in \mathcal{F}_H^*$ is essential iff C is not an implicate of $\mathcal{F}_H^* \setminus C$. Suppose that a prime implicate C is not essential, i.e. C is an implicate of $\mathcal{F}_H^* \setminus C$.

Let us first consider the case when C is of the form $y_{j'} \bigvee \bigvee_{i \in I(j')} \neg x_i$. Then Lemma 3.1 implies that the forward chaining procedure for $\mathcal{F}_H^* \setminus C$ starting with the set $\{x_i \mid i \in I(j') \setminus \{i'\}\}$ must eventually include $y_{j'}$ into the set $\mathcal{R}$. But the structure of $\mathcal{F}_H^*$ shows that this is impossible, since no additional variable will be included into $\mathcal{R}$.

Let us finally examine the case when C is of the form $x_{i'} \bigvee \neg y_{j'}$. Then Lemma 3.1 implies that the forward chaining procedure for $\mathcal{F}_H^* \setminus C$ starting with $y_{j'}$ must include eventually $x_{i'}$ into the set $\mathcal{R}$. But the structure of $\mathcal{F}_H^*$ shows that this is impossible, since the procedure will stop after having included into $\mathcal{R}$ all the variables x_i, $i \in I(j') \setminus \{i'\}$. □

Corollary 4.4 *The CNF $\mathcal{F}_H$ is the unique irredundant prime CNF of $h(f)$.*

Note that the number of literals in $\mathcal{F}_H$ is

$$l = m + 3 \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}.$$

Proof of Theorem 4.1 Suppose that the matrix A has a covering $S \subseteq \{1, \ldots, m\}$ such that $|S| \leq k'$. Let $C = \bigvee_{j \in S} \neg y_j$. It can be easily seen using Lemma 3.2 that C is an implicate of $\mathcal{F}$ and $\bigvee_{i=1}^{n} \neg x_i$ is an implicate of $\mathcal{F}_H \bigwedge C$. Therefore, the function f can be represented by the CNF $\mathcal{F}_H \bigwedge C$ of length $l + |S| \leq l + k'$.

Suppose that the function f can be represented by a CNF of length $l' \leq l + k'$. It follows that f can be represented by an irredundant prime CNF of length l'. Corollary 4.4 and Theorem 2.5 imply that $\mathcal{F}_H$ is the definite Horn part of any irredundant prime CNF of f. The negative part of any irredundant prime CNF of f consists of exactly one clause. Let an irredundant prime CNF representing f and having length l' be the CNF $\mathcal{F}_H \bigwedge C$, where $C = (\bigvee_{i \in I} \neg x_i) \bigvee (\bigvee_{j \in J} \neg y_j)$. Then $|I| + |J| \leq k'$. Since C is an implicate of $\mathcal{F}$, by Lemma 3.2 the forward chaining procedure for $\mathcal{F}_H$ starting with the set $\{x_i \mid i \in I\} \bigcup \{y_j \mid j \in J\}$ must eventually include into the set $\mathcal{R}$ every variable x_i, $i \in \{1, \ldots, n\} \setminus I$. The structure of $\mathcal{F}_H$ shows that this inclusion takes place iff for every $i \in \{1, \ldots, n\} \setminus I$ there exists $j \in J$ such that $i \in I(j)$. Therefore, the set J of columns covers the set $\{1, \ldots, n\} \setminus I$ of rows. Since every row $i \in I$ can be covered by any column j such that $a_{ij} = 1$, the matrix A has a covering of cardinality $|I| + |J| \leq k'$. □

Now we can show by contradiction that all the other clauses in $\mathcal{F}$ are also prime. Let $\mathcal{F}_H$ denote the conjunction of all the definite Horn clauses in $\mathcal{F}$. Suppose the clause $y_{j'} \bigvee \bigvee_{i \in I(j')} \neg x_i$ is not prime. Since $\bigvee_{i=1}^{n} \neg x_i$ is prime, there must exist i' such that the clause $y_{j'} \bigvee \bigvee_{i \in I(j') \setminus \{i'\}} \neg x_i$ is an implicate of $\mathcal{F}$ and $a_{i'j'} = 1$. Then Lemma 3.1 implies that the forward chaining procedure for $\mathcal{F}_H$ starting with the set $\{x_i \mid i \in I(j') \setminus \{i'\}\}$ has to include eventually $y_{j'}$ into the set $\mathcal{R}$. But the structure of $\mathcal{F}_H$ shows that this is impossible, since the procedure will never include into $\mathcal{R}$ any variables not in the original set. □

Lemma 4.2 and Theorem 2.5 imply that

$$\mathcal{F}_H = (\bigwedge_{j=1}^{m} (\bigwedge_{i \in I(j)} (x_i \bigvee \neg y_j)))(\bigwedge_{j=1}^{m} (y_j \bigvee \bigvee_{i \in I(j)} \neg x_i))$$

is a prime CNF of the definite Horn component $h(f)$ of the Horn function f represented by the CNF $\mathcal{F}$.

Lemma 4.3 *Each clause in $\mathcal{F}_H$ is an essential prime implicate of $h(f)$.*

Proof Let us first prove (by contradiction) that all those prime implicates of $h(f)$ which contain positive x literals are contained in $\mathcal{F}_H$. Suppose that the clause $x_{i'} \bigvee (\bigvee_{i \in I} \neg x_i) \bigvee (\bigvee_{j \in J} \neg y_j)$ is a prime implicate of $h(f)$ not belonging to $\mathcal{F}_H$. Whenever $a_{i'j} = 1$, $x_{i'} \bigvee \neg y_j$ is a clause of $\mathcal{F}_H$. Therefore, for every $j \in J$: $a_{i'j} = 0$. By Lemma 3.1 the forward chaining procedure for $\mathcal{F}_H$ starting with the set $\{x_i \mid i \in I\} \bigcup \{y_j \mid j \in J\}$ must eventually include $x_{i'}$ into the set $\mathcal{R}$. This is possible only if at an earlier stage a variable $y_{j'}$ such that $a_{i'j'} = 1$ is included into the set $\mathcal{R}$. But $y_{j'}$ can be included into $\mathcal{R}$ only if $\mathcal{R}$ already contains all the variables x_i such that $a_{ij'} = 1$. This implies that $\mathcal{R}$ must contain $x_{i'}$, which is impossible.

Let us now examine prime implicates of $h(f)$ containing a negative y literal. Suppose that the clause $y_{j'} \bigvee (\bigvee_{i \in I} \neg x_i) \bigvee (\bigvee_{j \in J} \neg y_j)$ is a prime implicate of $h(f)$ not belonging to $\mathcal{F}_H$. Let us first note that for every $i' \in I(j') \setminus I$ there exists a $j'' \in J$ such that $i' \in I(j'')$. If this were not so, then $x_{i'}$ would never be included into $\mathcal{R}$, and hence $y_{j'}$ also would never be included into $\mathcal{R}$, which is impossible. Clearly, $I(j') \not\subseteq I$. On the other hand, the relation $I \subset I(j')$ must hold. Suppose it does not. Let $I' = I \bigcap I(j')$. Since for every $i' \in I(j') \setminus I$ there exists $j'' \in J$ such that $i' \in I(j'')$, the clause $y_{j'} \bigvee (\bigvee_{i \in I'} \neg x_i) \bigvee (\bigvee_{j \in J} \neg y_j)$ is also an implicate of $h(f)$. Therefore, $I \subset I(j')$. Since the matrix A has no comparable columns, $I(j) \not\subseteq I$ for any $j = 1, \ldots, m$. Also, the relation $I \bigcap I(j) = \emptyset$ holds for every $j \in J$. If $I = \emptyset$, then J consists of at least two elements, since the matrix A has no comparable columns.

The set covering problem will remain **NP**-complete even if we assume in addition that the matrix A does not have comparable columns, i.e. for every two columns $j', j'' \in \{1, \ldots, m\}$ there exist two rows $i', i'' \in \{1, \ldots, n\}$ such that $a_{i'j'} = 1$, $a_{i'j''} = 0$, $a_{i''j'} = 0$, $a_{i''j''} = 1$. We may also assume that the matrix A does not have rows or columns consisting only of zeros.

We introduce a variable x_i for every row $i \in \{1, \ldots, n\}$, and a variable y_j for every column $j \in \{1, \ldots, m\}$. Let $I(j) = \{i \mid a_{ij} = 1\}$ for every $j = 1, \ldots, m$, and let us consider the following Horn CNF

$$\mathcal{F} = (\bigwedge_{j=1}^{m}(\bigwedge_{i \in I(j)}(x_i \bigvee \neg y_j)))(\bigwedge_{j=1}^{m}(y_j \bigvee \bigvee_{i \in I(j)} \neg x_i))(\bigvee_{i=1}^{n} \neg x_i).$$

In order to prove Theorem 4.1, we shall show at the end of this section that the solution of the Horn function minimization problem which has as its inputs the CNF $\mathcal{F}$ and the number

$$k = m + 3\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} + k'$$

is equivalent to the solution of the given set covering problem.

Lemma 4.2 *The CNF $\mathcal{F}$ is prime.*

Proof The CNF $\mathcal{F}$ is satisfiable since it evaluates to 1 when all the variables are 0. Also, no unit clause is an implicate of $\mathcal{F}$. Indeed,

(i) it is obvious that no positive literal can be an implicate of $\mathcal{F}$;

(ii) $\neg y_{j*}$ cannot be an implicate of $\mathcal{F}$, since if $y_{j*} = 1$, $x_i = a_{ij*}$ for all i, and all other $y_j = 0$, then $\mathcal{F} = 1$;

(iii) $\neg x_{i*}$ cannot be an implicate of $\mathcal{F}$, since if $x_{i*} = 1$ and all the other variables are 0, except the variable y_j (if any) for which $a_{i*j} = 1$ and $a_{ij} = 0$ (for all $i \neq i*$), then $\mathcal{F} = 1$.

It follows that all the clauses in $\mathcal{F}$ which consist of exactly two literals are prime.

The primality of the negative clause in $\mathcal{F}$ can be shown by contradiction. Suppose that a clause $\bigvee_{i \neq i'} \neg x_i$ is an implicate of $\mathcal{F}$. Then Lemma 3.2 implies that the forward chaining procedure starting with the set $\{x_i \mid i \neq i'\}$ has to include eventually $x_{i'}$ into the set $\mathcal{R}$. This can only happen if a variable y_j having $a_{i'j} = 1$ is included in $\mathcal{R}$. The structure of $\mathcal{F}$ shows that this cannot occur.

If $C_1 = \bigvee_{x \in S_1} \neg x$ is an implicate of f, then for every $x' \in S_2 \setminus S_1$ the clause $C' = x' \bigvee C_1$ is an implicate of f. Suppose that C' is not an implicate of $\mathcal{F}$. This implies that there is an assignment α such that $\mathcal{F}(\alpha) = 1$ and $C'(\alpha) = 0$. The last equality implies that $x' = 0$ in this assignment α, and hence $C_2(\alpha) = 1$, which implies that $f(\alpha) = 1$, in contradiction with the fact that C' is an implicate of f. It follows that C' is an implicate of $\mathcal{F}$. Therefore, by Lemma 3.1 the forward chaining procedure for $\mathcal{F}$ starting with the set S_1 includes eventually x' into the set $\mathcal{R}$. □

4 Intractability of Minimization

In this section we shall study 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 $\mathcal{F}$?

We observe first that this problem belongs to **NP**. Indeed, if the answer is *yes* and we are given a CNF $\mathcal{F}'$, we first count the number of literals in $\mathcal{F}'$ to check that this number is not larger than k. Then we check whether $\mathcal{F} \leq \mathcal{F}'$ by verifying in quadratic time that every clause of $\mathcal{F}'$ is an implicate of $\mathcal{F}$. If yes, then for each clause of $\mathcal{F}'$ we construct (in quadratic time, see [9]) a prime implicate of $\mathcal{F}$ subsuming it. Let $\mathcal{F}''$ be the conjunction of these prime implicates. Clearly, $\mathcal{F}''$ is Horn, and we can check whether $\mathcal{F}'' \leq \mathcal{F}$ by verifying in quadratic time that every clause of $\mathcal{F}$ is an implicate of $\mathcal{F}''$.

Theorem 4.1 *The Horn function minimization problem is* **NP***-complete.*

We shall prove this theorem by reducing the well known **NP**-complete set covering problem (see e.g. [6]) to a Horn function minimization problem.

Set Covering Problem

Instance: A matrix $A = \{a_{ij} \mid a_{ij} \in \{0,1\};\ i = 1, \ldots, n;\ j = 1, \ldots, m\}$ and a positive number k'.

Question: Is there a set $S \subseteq \{1, \ldots, m\}$ (called a *covering*) such that $|S| \leq k'$, and such that for every $i \in \{1, \ldots, n\}$ there exists a $j \in S$ such that $a_{ij} = 1$?

is found, the variable x' is included into $\mathcal{R}$, and the search is repeated as many times as possible.

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 $\mathcal{R}$.*

Proof We remind that a clause C is an implicate of a Boolean function h iff $h(x_1, \ldots, x_n) = 1$ implies $C(x_1, \ldots, x_n) = 1$.

To prove the **if** part of the lemma it is sufficient to show that $h(x_1, \ldots, x_n) = 1$ and $x = 1$ for every $x \in S$ imply $x' = 1$. We shall prove this by induction on the number of steps the forward chaining procedure takes to include x' in $\mathcal{R}$. The basis of induction follows immediately from the description of the procedure (i.e. if the variable x' is included in $\mathcal{R}$ in one step, then there is a clause in $\mathcal{F}$ subsuming C). Let $\mathcal{R}_r$ denote the set $\mathcal{R}$ after r steps of the procedure, and let the variable x' be included in $\mathcal{R}$ in the $(r+1)$-st step. By induction, $h(x_1, \ldots, x_n) = 1$ and $x = 1$ for every $x \in S$ imply $x = 1$ for every $x \in \mathcal{R}_r$. Since x' is included in $\mathcal{R}$ in the $(r+1)$-st step, the clause $C_r = x' \bigvee \bigvee_{x \in \mathcal{R}_r} \neg x$ is subsumed by some clause in $\mathcal{F}$, and hence C_r is an implicate of h. Therefore, $h(x_1, \ldots, x_n) = 1$ and $x = 1$ for every $x \in \mathcal{R}_r$ imply $x' = 1$, completing the proof of the **if** part by induction.

We shall prove the **only if** part by contradiction. Let us assume that C is an implicate of h, and that x' is not included into $\mathcal{R}$ by the procedure. Let us consider the following assignment α: $x(\alpha) = 1$ iff $x \in \mathcal{R}$; $x(\alpha) = 0$ iff $x \notin \mathcal{R}$. Since each clause of $\mathcal{F}$ evaluates to 1 at α (either the clause contains a negated variable which is not in $\mathcal{R}$, or the non-negated variable of the clause is in $\mathcal{R}$), the function h evaluates to 1 at α. Therefore, $x' = 0$ contradicts the fact that $x = 1$ for every $x \in S$ and $h = 1$ at α, thus completing the proof of the lemma. □

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 $\mathcal{R}$.*

Proof If the forward chaining procedure for $\mathcal{F}$ starting with the set S_1 includes eventually every variable $x \in S_2$ into the set $\mathcal{R}$, then by Lemma 3.1 for every $x' \in S_2 \setminus S_1$ the clause $x' \bigvee C_1$ is an implicate of $\mathcal{F}$. The fact that the clause C_1 is an implicate of $C_2 \bigwedge \bigwedge_{x' \in S_2 \setminus S_1} (x' \bigvee C_1)$ proves the **if** part of the lemma.

2.5 Decomposition of Horn Functions

The main result of [9] 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 the minimum number of clauses iff $\mathcal{F}(h(f))$ consists of the minimum number of clauses.

3 Forward Chaining Procedure

This section will provide some technical tools needed for the development of the results presented in this paper. We formulate here some properties of the definite Horn and the negative implicates (Lemmas 3.1 and 3.2) of a Horn function.

Let us first 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.**

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

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, 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 components with no incoming arcs are at the beginning of the list; let m be the number of such 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 $C_i \in \mathcal{N}_i(f)$.*

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 [9] 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. This gives an efficient way of constructing a negative restriction of a Horn function.

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 [9] 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 [5]. On this basis an algorithm was given in [9] 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. 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) $f = h(f) \wedge g$, *where g is a negative function;*

(ii) if $f = h' \wedge g'$, *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 [9] 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

2.1 Basic Definitions

A *Boolean function* f of n variables $x_1, \ldots, x_n$ is a mapping $\{0,1\}^n \rightarrow \{0,1\}$. 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 CNF of a function is called *minimum* iff there is no CNF representing the same function and containing fewer literals. Obviously, any minimum CNF is both irredundant and prime. The *problem of Boolean function minimization* consists in constructing a minimum CNF of the function.

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.

consider the Horn CNF

$$\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{i-1}\neg x_j \bigvee x_i \bigvee \bigvee_{j=n+1}^{2n}\neg x_j) \bigwedge_{i=n+1}^{2n}(\bigvee_{j=1}^{n}\neg x_j \bigvee \bigvee_{j=n+1}^{i-1}\neg x_j \bigvee x_i)(\bigvee_{i=1}^{2n}\neg x_i) \tag{1}$$

involving $2n+1$ clauses and $3n^2+3n$ literals. It can be seen that the Horn CNF

$$(\bigvee_{i=1}^{n}\neg x_i)(\bigvee_{i=n+1}^{2n}\neg x_i) \tag{2}$$

is equivalent to (1), but has only 2 clauses and $2n$ literals.

The goal of this paper is to study the problem of Horn function minimization. Since the minimized conjunctive normal forms of any Horn function are also Horn, the minimized knowledge bases maintain the Horn clause structure.

In the next section we introduce the main definitions and review the results of [9] 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).

In the third section we provide some technical tools needed for the development of the results presented in this paper. We formulate some properties of the negative and the definite Horn implicates (Lemmas 3.1 and 3.2) of a Horn function.

In Section 4 we prove that the problem of minimizing the number of literals in CNFs of Horn functions is **NP**-complete, by reducing to this problem the well-known **NP**-complete set covering problem.

In Section 5 we prove that the number of clauses in any irredundant Horn CNF involving n propositional variables is at most $n-1$ times the minimum possible number of clauses. Therefore, the quadratic time transformation of an arbitrary Horn CNF to an equivalent irredundant and prime one (presented in [9]) may be considered as a reasonable approximation algorithm.

2 Background Information

In this section we introduce some basic Boolean terminology and review those results of [9] 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).

1 Introduction

The problem of improving the computational performance of propositional expert systems is being actively investigated, and much attention is given to the analysis and simplification of production rule knowledge bases (see e.g. [2, 3, 7, 8, 11, 14]). Due to the dependence of the complexity of queries on the length of the knowledge base, the use of a "shortest" representation of a given knowledge base reduces the memory requirements and increases the computational efficiency of expert systems.

Obviously, 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:

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

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

Due to a large extent to the fact that for propositional Horn formulae the satisfiability problem is linearly solvable ([5]), propositional Horn clause knowledge bases play a prominent role in artificial intelligence. This is a motivation for active research in learning and identification of Horn formulae ([1, 4]) and in constructing Horn approximations of non-Horn formulae ([13]).

We concentrate here on the problem of optimal compression of propositional Horn clause knowledge bases. Therefore, we study the class of Horn functions, i.e. the class of those Boolean functions which are associated to Horn knowledge bases. The concept of a Horn function leads naturally to the use of various representations of the same function. In particular, the minimization (i.e. the choice of the most efficient representation) of a Horn function does not have to be limited to simple local reductions of a given Horn formula.

Minimization can drastically reduce the size of Horn knowledge bases. As an example,

ABSTRACT

Horn formulae play a prominent role in artificial intelligence and logic programming. In this paper we investigate the problem of optimal compression of propositional Horn production rule knowledge bases. The standard approach to this problem, consisting in the removal of redundant rules from a knowledge base, leads to an "irredundant" but not necessarily optimal knowledge base. We prove here that the number of rules in any irredundant Horn knowledge base involving n propositional variables is at most $n-1$ times the minimum possible number of rules. In order to formalize the optimal compression problem, we define a Boolean function of a knowledge base as being the function whose set of true points is the set of models of the knowledge base. In this way the optimal compression of production rule knowledge bases becomes a problem of Boolean function minimization. In this paper we prove that the minimization of Horn functions (i.e. Boolean functions associated to Horn knowledge bases) is **NP**-complete.

Optimal Compression of Propositional Horn Knowledge Bases: Complexity and Approximation[1]

Peter L. Hammer[2] and Alexander Kogan[2,3]

January 19, 1993

[1]The authors gratefully acknowledge the partial support of AFOSR (Grants 89-0512B and F49620-93-1-0041) and ONR (Grants N00014-92-J1375 and N00014-92-J4083).

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

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