On the Computational Complexity of Quantified Horn Clauses

Marek Karpinski
Department of Computer Science
University of Bonn
5300 Bonn 1

Hans Kleine-Büning
Institut für angewandte Informatik und formale Beschreibungsverfahren
University of Karlsruhe
4100 Duisburg 1

Peter H. Schmitt
Scientific center of IBM Germany
Heidelberg

Abstract

A polynomial time algorithm is presented for the evaluation problem for quantified propositional Horn clauses. This answers an open problem of [IM 87].

1 Introduction

The basic idea behind the programming language PROLOG is, that a proof or refutation of Horn formulas can be viewed as an efficient computation from which one extracts an output. Horn formulas (or a program) are conjunctions of Horn clauses, i.e. clauses of the form: $A_1 \wedge A_2 \wedge \ldots A_m \rightarrow B$, where A_i and B are atomic formulas. The computation of the program consists in finding an assignment of values to the variables which satisfies all clauses. Two basic methods used for such a computation are **unifiction** (to produce assignments) and **resolution** of clauses (as a method of logical inference). The problem of testing a set of Horn formulas for satisfiability, e.g. using unit resolution is known to have linear time solution algorithms, see e.g. [DG 84], [IM 87].

The new development of Prolog query languages, cf. [GR 87], strongly motivated the search for general efficient solutions for quantified Horn clauses. In this paper we present such an efficient solution for the evaluation problem of quantified propositional Horn clauses. The algorithm works in $O(n^3)$ time. We stress in this paper polynomial time solutions for this problem, rather than the design of new data structures to make it work faster. We do hope though to present a linear time or $O(n \log n)$ time algorithm in a subseguent paper. Schäfer in [Sch 78] claimed a polynomial time algorithm for the above problem, but he did not give a proof.

It is interesting to note, that the evaluation problem for quantified Boolean formulas, even if restricted to formulas containing at most three literals per clause, is PSPACE-complete, [GJ 79]. In [APT 79] a linear time algorithm has been design for the case of quantified Boolean formulas in conjunctive normal form with at most two literals per clause.

Our result in this context entails suprising algorithmic efficiency of the evaluation problem for quantified Horn clauses and opens the possibility of several natural query-like extensions of standard PROLOG.

2 Terminology

We will mainly deal with propositional and quantified propositional formulas. We denote by **PV** the infinite set of propositional variables. The propositional connectives “and”, “or”, and “not” are designated by the symbols $''\wedge''$, “$\vee$”, and $''\neg''$. We shall have occasion to deal with the following special sets of propositional formulas: literals, all propositional formulas, clauses, conjuntive normal forms (conjunctions of clauses), Horn clauses (clauses containing at most one positive literal), totally negative clauses (clauses containing only negative literals), negative conjunctive normal forms (conjunctions of totally negative clauses). These will be denoted respectively by **LI, FML, CL, CNF, HC, TNC, NCNF**.

In case we allow the propositional constants 0 and 1 to occur, the corresponding sets are indexed by C, e.g. CNF_C, HC_C, etc.

The universal (existential) quantifier will be denoted by $\forall$ ($\exists$) as usual. For any set $\sum$ of propositional formulas $Q^* \sum$ is the set of all formulas of the form $Q_1 X_1 \ldots Q_n X_{n^\circ}$ where n is an arbitrary natural number, each Q_i is either $\forall$ or $\exists$, $o \in \sum$ and$\{X_1, \ldots, X_n\}$ is a set of propsitional variables. If $\{X_1, \ldots, X_n\}$ contains all variables occuring in o, we call $Q_1 X_1 \ldots Q_n X_{n^o}$ closed. Likewise $\exists^* \sum$ is the set of all formulas of the form $\exists X_1, \ldots, \exists X_{n^o}$ with $o \in \sum$.

As usual we distinquish free and bound occurences of variables in quantified propositional formulas. For $X \in \mathbf{PV}$ and t a constant symbol the formula $o(t/X)$ arises from o by replacing every free occurence of X in o by t.

We say that a variable X occurs in a clause ϕ if either X or $\neg X$ is a disjunctive component of ϕ. In contrast we say that X is a literal of ϕ if X is a disjunctive component of ϕ. Thus X occurs in $\neg X \vee Y \vee \neg Z$, but X is not a literal of this clause.

3 Gerneralized Unit Resolution

It is well known, that a formula of the form

$$\exists X_1 \ldots \exists X_{k\phi}$$

where ϕ is a cojunction of Horn clauses, is true if and only if the empty clause is not derivable from ϕ by unit resolution. We first generalize the operation of unit resolution to the case of arbitrary quantifier prefixes.
Let Φ be a formula of the form

$$\forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y}_{k-1} \forall \vec{X}_{k\phi}$$

where

for all $i, 1 \leq i \leq k \quad \vec{X}_i = X_{n_{i-1}+1}, \ldots, X_{n_i}$ with $n_0 = 0$

Since we want to treat formulas with arbitrary quantifier prefix, we allow n_1 and n_k to be 0, while for all $i, 1 < i < k$ we require $n_i \neq 0$.

for all $i, 1 \leq i < k \quad \vec{Y}_i = Y_{m_{i-1}+1}, \ldots, Y_{m_i}$, where all $m_i \neq 0$.

$\phi = \phi_1 \wedge \ldots \wedge \phi_r$
where all ϕ_i are Horn clauses:

we may furthermore assume without loss of generality:

for every $i, 1 \leq i \leq r$, there is no variable X_j, such that both literals X_j and $\neg X_j$ occur in ϕ_i

In the given representation of Φ we assume implicitely, that no propositional variable occurs both existentially and universally bound. This is clearly no restriction.

An **X-literal (Y-literal)** is a literal of the form X_i or $\neg X_i$ (resp. Y_i or $\neg Y_i$). A **pure X-clause** is a clause consisting exclusively of X-literals. In particular the empty clause is a pure X-clause.

A clause ϕ_j is called a Y_i**-unit clause** if Y_i occurs positively in ϕ_j and Y_i is the only Y-variable occuring in ϕ_j.

A clause ϕ_j is called a Y**-unit clause** if Y_i-unit clause for some i.

When we say, that the variable X_i is before Y_j, we refer to the order of occurences in the prefix of Φ, i.e. X_i is before Y_j if $n_{p-1} < i \leq n_p, m_{q-1} < j \leq m_q$ and $p \leq q$. Analogously we use the phrase X_i is after Y_j.

Let ϕ_p be a Y_i-unit clause and ϕ_q a clause containing the literal $\neg Y_i$. The **resolvent** ψ of ϕ_p and ϕ_q is obtained by

forming the disjunction $\phi_p \vee \phi_q$,

if for some variable X_j both literals X_j and $\neg X_j$ occur in $\phi_p \vee \phi_q$, then stop. No resolvent exists in this case.

omitting all occurences both negated and unnegated of the variable Y_i,

omitting all occurences of X-variables, that are not before any Y-variable occuring in the modified disjunction,

A **unit resolution step** on the formula ϕ

$\forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y} k - 1 \forall \vec{X}_k (\phi_1 \wedge \ldots \wedge \phi_k)$

is performed by adding the resolvent ψ of a Y-unit clause ϕ_p and a clause ϕ_q containing the literal $\neg Y$ to the matrix of ϕ, thus obtaining

$\forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y}_{k-1} \forall \vec{X}_k (\phi_1 \wedge \ldots \wedge \phi_k \wedge \psi)$

Lemma 1: Let $\sum$ be obtained from $\Phi = \forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y}_{k-1} \forall \vec{X}_k \phi$ by omitting in any clause of ϕ all X-literals for those X, that are not before any Y-variable occuring in this clause, then $\sum$ is true if and only if ϕ is true.

Proof: If $\sum$ is true, then ϕ is, of course, also true. So let us assume that Φ is true. There are functions f_i for all i, $1 \leq i \leq m_{k-1}$, the number of arguments of f_i equals the number of X-variables that are before Y_i, such that for any sequence $a_1, \ldots, a_{n_k}$ of 0 and 1 the formula

$\phi(a_1, \ldots, a_{n_k}, f_1(\vec{a}^1), \ldots, f_{m_{k-1}}(\vec{a}^{m_{k-1}}))$

is true, where $b_i = f_i(a_1, \ldots, a_n)$ for the appropriate number n of arguments.
Let ϕ_i be a clause in the matrix of Φ. Let $\phi_i = \phi_{i,1} \vee \phi_{i,2}$, where $\phi_{i,1}$ contains all Y-literals from ϕ_i and those X-literals of ϕ_i, such that X occurs before Y for some Y-literal in ϕ_i and $\phi_{i,2}$ contains all literals $\neg X_j$ or X_j such that X_j is not before any Y-variable occuring in ϕ_i.

Let us fix an assignment $a_1, \ldots, a_{n_k}$ of 0 and 1. We will use b_i as an abbrevation for $f_i(\vec{a}^1)$.

We will show, that

$\phi_{i,1}(a_1, \ldots, a_{n_k}, f_1(\vec{a}^1), \ldots, f_{m_{k-1}}(\vec{a}^{m_{k-1}}))$

is true.

Let $a'_1, \ldots, a'_{n_k}$ be another assignment of values 0 and 1 to the X-variables which agrees with the fixed assignment except possibly for variables X_j that are not before any Y-variable in ϕ_i. The assignment $\vec{a}'$ is chosen to have the property, that $\phi_{i,2}(\vec{a}', \vec{b}')$ is false. This is possible since by assumption on Φ the pure X-clause $\phi_{i,2}$ contains no complementary pair X_j, $\neg X_j$.

Since ϕ is true under the assignment $a'_1, \ldots, a'_{n_k}, b'_1, \ldots, b'_{m_{k-1}} \phi_{i,1}$ has to be true. Since $\phi_{i,1}$ contains only X-variables X_i for which $a_i = a'_i$ and since for all variables Y_j in $\phi_{i,1}$ the function f_j does not have any of the changed X-values among it arguments, $\phi_{p,1}$ is also true under the original assignment $\vec{a}$.

Lemma 2: Let $\sum$ be obtained from Φ by a resolution step, then $\sum$ is true if only if Φ is true.

Proof: If $\sum$ is true, then Φ is, of course, also true. To prove the converse direction we first observe, that adding to ϕ the ordinary resolvent of two clauses, without omitting any variables leads to a logically equivalent formula ϕ'. Now Lemma 1 is used to pass from $\forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y}_{k-1} \forall \vec{X}_k \phi'$ to $\sum$.

Theorem 3: Let Φ be a formula of the form $\forall \vec{X}_1 \exists \vec{Y}_1 \ldots \exists \vec{Y}_{k-1} \forall \vec{X}_k \phi$
Φ is true if and only of no pure X-clause can be derived from ϕ by Y-unit resolution.

Proof: Let Φ' be obtained from Φ by Y-unit resolution, such that the matrix ϕ' of Φ' contains a pure X-clause ψ. By assumption on Φ and the definition of unit resolution ψ cannot contain a complementary pair. Thus Φ' is obviously false. By Lemma 1 also Φ has to be false.

Now let us assume, that no pure X-clause can be derived from ϕ. Let o be the conjuntion of ϕ together with all resolvents, that can be derived by Y-unit resolution and let $\sum$ be the formula with the same prefix as Φ and the matrix o. We will show that $\sum$ is true, which immediately yields also the truth of Φ.

Let an arbitrary assignment $\vec{a} = a_1, \ldots, a_{n_k}$ of 0 and1 be given. We define the assignments b_i for the variables Y_i as follows:

- If there is a Y_i-unit clause in o that is not already made true under the partial assignment $\vec{a}$, then let b_i equal 1.
- Let b_i equal 0 otherwise.

We prove by induction on the number s of Y-variables in χ, that $\chi(\vec{a}, \vec{b})$ is true, for any clause χ of o.

If $s = 0$, then χ would consist entirely of X-variables. By assumption this is not possible.

Let $s = 1$. If χ is a Y_i-unit clause, then χ is either already true on the basis of the assignment $\vec{a}$ or b_i has been defined to be equal to 1. So let us assume that the only Y-literal in χ is $\neg Y_i$. If no Y_i-unit clause occurs in o, that is not true on the basis of $\vec{a}$ alone, then b_i is aquel to 0 and χ is true. Finally it remains to consider the case that o contains a Y_i-unit clause χ', that is not already true under the partial assignment $\vec{a}$. By the Horn property all X-literals in χ' are negative. We may therefore draw the conclusion that for all variables X_j in χ' $a_j = 1$. By assumption χ and χ' cannot have a resolvent, i.e. for some j X_j occurs in χ and $\neg X_j$ occurs in χ', which implies that χ is true, since $a_j = 1$.

Induction step. Let χ contains $s + 1$ Y-literals, where we may now assume $s > 1$. By the Horn property χ has to contain a negative Y-literal $\neg Y_i$. If there is no Y_i-unit clause in o, that is not already true under the assignment $\vec{a}$, then $b_i = 0$ and χ is true. Otherwise let χ' be such a clause. The resolvent ψ of χ and χ' contains s Y-literals and is thus true by induction hypothesis. Since the disjunctive part of ψ stemming from χ' is not true, the part stemming from χ has to be. Thus also χ is true.

4 Examples

Example 1 Let

$\Phi = \forall X \exists Y ((X \vee \neg Y) \wedge (\neg X \vee Y))$

The second clause is a Y-unit clause. Its resolvent with the first clause would contain the complementary pair x and $\neg X$ and is thus not performed. No pure X-clause is derivable, Φ is true.

Example 2 Let

$\Phi = \forall Y \exists X ((X \vee \neg Y) \wedge (\neg X \vee Y))$

Again the second clause is a Y-unit clause. Its resolvent with the first clause is the empty clause since all occurences of X-variables are dropped. Thus the formula is false.

Example 3 Let

$\Phi = \exists Y_1 \forall X \exists Y_2 ((\neg Y_1 \vee X \vee \neg Y_2) \wedge (\neg Y_1 \vee \neg X \vee Y_2) \wedge Y_1)$

The only Y-unit clause is Y. The resolvents with the first and second clause are $X \vee \neg Y_2$ and $\neg X \vee Y_2$ respectively.
The second clause is again a Y-unit clause. No resolution with the first clause is possible, because a complementary pair would arise. This Φ is true.

Example 4 Let

$\Phi = \forall X_1 \forall X_2 \exists Y_1 \exists Y_2 ((X_2 \vee \neg Y_2) \wedge (Y_2 \vee \neg Y_1) \wedge (\neg X_2 \vee Y_1) \wedge (\neg X_1 \vee Y_1))$

Using the Y-unit clauses $\neg X_2 \vee Y_1$ and $\neg X_1 \vee Y_1$ we obtain by resolution with the second clause two new Y-unit clauses $Y_2 \vee \neg X_2$ and $Y_2 \vee \neg X_1$. Only the second of these can be used to continue resolution with the first clause to obtain $X_3 \vee \neg X_2$. Thus is false.

Lemma 4: Let Φ be a formula of the form $\forall\vec{X}\exists\vec{Y}\phi$. Then Φ is false if and only if for some assignment $\vec{a} = a_1, \ldots, a_n$ of values 0 and 1 with at most one occurence of 0, the formula $\exists\vec{Y}\phi(a_1, \ldots, a_n)$, is false.

Proof: One implication of the lemma is trivial. So let us assume that $\forall\vec{X}\exists\vec{Y}\phi$ is false. Let ϕ_0 be the conjunction of ϕ together with all clauses, that can be derived from ϕ by Y-unit resolution. By Lemma 1 $\forall\vec{X}\exists\vec{Y}\phi$, is equivalent to $\forall\vec{X}\exists\vec{Y}\phi_0$. By our assumption the latter formula is false and thus contains by virtue of Theorem 3 a pure X-clause χ. Let $\vec{a} = a_1, \ldots, a_n$ be an assignment of values 0 and 1 to the X-variables, such that $\chi(a_1, \ldots, a_n)$ is false. Since χ is a Horn clause, we may choose $\vec{a}$, such that at most one 0 occurs. Obviously $\exists\vec{Y}\phi_0(a_1, \ldots, a_n)$ is false and therefore by the equivalence stated above also $\exists\vec{Y}\phi(a_1, \ldots, a_n)$ is false.

Lemma 5: The truth of an $\forall\exists$-quantified conjunction of Horn formulas can be decided in polynomial time.

Proof: Let $\Phi = \forall\vec{X}\exists\vec{Y}\phi$. The algorithm consists in all assignments $\vec{a}$ with at most once 0 the truth of $\exists\vec{Y}\phi(\vec{a})$. There are (number of X-variables) + 1 many assignments $\vec{a}$ with at most one 0. The reduction of $\phi(\vec{a})$ to a conjuction ϕ_1 of Horn clauses not containing the constants 0 and 1 can be affected in linear time. Finally the satisfiability of ϕ_1 can be decided in linear time. Thus the overall running time of the algorithm may be bounded by (length of input)^2.

5 An Algorithm

Let Φ be a formula of the form

$$\forall\vec{X}_1\exists\vec{Y}_1 \ldots \exists\vec{Y}_{k-1}\forall\vec{X}_k\phi$$

be given.

Let N_ϕ be the set of clauses in ϕ, that are not pure X-clauses and contain only negative Y-literals. Let P_ϕ be the set of clauses in ϕ, that contain at least one positive Y-literal.

for all clauses C in N_ϕ do

let S_C be the set of positive X-literals in C. {Thus SC may be a singleton set or the empty set}

if S_C is empty then do

remove all occurences of all X-literals in P_ϕ and C obtaining P'_ϕ and C'
apply standard unit-resolution to P'_ϕ and C'.
if the empty clause can be derived terminate with "Φ is false"
otherwise terminate with "ϕ is true".

end if

if $S_C = \{X_r\}$ then do

for all variables X different from X_r remove all X-literals from P_ϕ obtaining P'_ϕ.
begin 1

let L be the set of Y-unit clauses that may be derived from P'_ϕ by standard unit resolution without talking the obstacle X_r into consideration and such that Y occurs before X_r in the prefix of Φ.
for all $Y \in L$ remove all occurences of the literal $\neg Y$ in all clauses in P'_ϕ
remove all clauses containing a Y-literal, with Y occuring before X_r in the prefix of Φ obtaining P''_ϕ.
let U be the set of Y-unit clauses in P''_ϕ not containing the literal $\neg X_r$.
let R be the empty set
while U is not empty do

for $Y \in U$do
remove all occurences of the literal $\neg Y$ in all clauses in P''_ϕ.
remove all clauses in P''_ϕ containing the literal Y.
add new Y-unit clauses not containing $\neg X_r$ to U
remove Y from U
add Y to R
end for

end while

end 1

if all Y-variables in C occur among the variables in $L \cup R$, then is "Φ is false" otherwise "Φ is true".

end if
end for

The complexity of the above algorithm is $O(n^3)$ observing that unit-resolution can be performed in linear time.

References

[AvE 82] K.R. Apt and M.H. van Emden, Contributions to the Theory of Logic Programming, J. ACM vol. 29, 1982, pp. 841-862.

[APT 79] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, Inf. Processing Letters, Vol. 8, 1979, pp. 121-123.

[DG 84] W.F. Dowling and J.H. Gallier, Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae, J Logic Programming, vol 3, 1984, pp. 267-284.

[GR 87] J.H. Gallier and S.Raatz, HORNLOG: A Graph- Based Interpreter for General Horn Clauses, J. of Logic Programming, vol 4, 1987, pp. 119-155.

[GJ 79] M.R. Garey and D.S. Johnston, Computers and Intractability, A Guide to the Theory of Np-Completeness, W.H. Freeman and Co., San Fransicao, 1979.

[IM 87] A. Itai and J.A. Makowsky, Unification as a complexity measure for logic programming, J. Logic Programming, vol. 4, 1987, pp. 105-117.

[Sch 78] Th.J. Schäfer, The Complexity of Satisfiability Problems. Proc. 10 th. Ann. ACM STOCS 1978, pp. 216-226.