A Resolution Method for Quantified Boolean Formulas

Marek Karpinski *
Department of Computer Science
University of Bonn
5300 Bonn 1
and
International Computer Science Institute
Berkeley, California

Hans Kleine-Büning
Department of Computer Science
University of Duisburg
4100 Duisburg 1

December, 1990

Abstract

A complete and sound resolution operation directly applicable to quantified boolean formulas is presented. If we restrict the resolution to unit resolution, then the completeness and soundness for extended quantified Horn formulas is shown. We prove that the truth of a quantified Horn formula can be decided in $O(rn)$ time, where n is the length of the Formula and r is the number of universal variables, whereas in contrast the evaluation problem for extended quantified Horn formulas is coNP-complete for formulas with prefix $\forall\exists$. Further we show that the resolution is exponential for extended quantified Horn formulas.

*Supported in part by the Leibniz Center for Research in Computer Science, by the DFG Grant KA 673/2-1 and by the SERC Grant GR-E 68297

1 Introduction

We are interested in the evaluation problem for quantified Boolean formulas , that means to determine whether a formula $Q_1 z_1 \ldots Q_n z_n(\alpha_1 \wedge \ldots \wedge \alpha_n)$ is true, where Q is either $\exists$ or $\forall$ and α_i is a propositional clause. It is well known that the evaluation problem is PSPACE–complete [StM 73] and that subclasses are decidable in polynomial time. A linear time algorithm for quantified Boolean formulas with clauses of at most 2 variables only can be found in [APT 78]. Motivated by the possibility of several natural query–like extensions of standard Prolog, in [KKBS 87] a cubic time algorithm for quantified Horn formulas has been established based on a generalized unit resolution for such formulas. We adept this approach and introduce a generalized resolution, called Q–resolution. A Q–resolution is an ordinary resolution, where only existential variables can be matched and additionally, each universal variable of a clause, for which no existential variable exists in that clause occuring after the universal variable in the prefix, is omitted. We prove that Q–resolution is *complete* and *sound* for quantified Boolean formulas.

If we restrict Q–resolution to Q–unit resolution, where one of the clauses contains one existential variable and arbitrarily many universal variables, analogously to Horn formulas, the completeness and soundness for quantified Horn formulas is shown.

A natural extension are formulas, where the universal literals of a clause can be arbitrary, but the existential literals are in Horn form. We will see that for this class, called extended quantified Horn formulas, the evaluation problem is coNP–complete, but stringly the Q–unit resolution remains complete and sound.

As mentioned above the evaluation problem for quantified Horn formulas is solvable in polynomial time. Based on simplification rules, which transform quantified Horn formulas in linear time to Horn formulas with prefix $\forall\exists$, we improve the upper bound of the evaluation problem.

In the last chapter we will investigate sets of Horn formulas denoted as multi Horn formulas. We will show that the satisfiability problem of multi Horn formulas, i.e. each of the Horn formulas is satisfiable, is closely related to the evaluation problem for quantified Horn formulas.

2 Generalized Resolution

We demand for technical reasons, that quantified formulas are of the form
$\forall x_1 \exists y_1 \ldots \forall x_{k-1} \exists y_{k-1} \forall x_k(\alpha_1 \wedge \ldots \wedge \alpha_n)$, where
for all i $(1 \leq i \leq k)$ $x_i = \mathrm{x}_{n_{i-1}+1}, ..., \mathrm{x}_{n_i}$ with $n_0 = 0$,
for all i $(1 \leq i < k)$ $y_i = \mathrm{y}_{m_{i-1}+1}, ..., \mathrm{y}_{m_i}$
and $\alpha_1, ..., \alpha_n$ clauses.

Further we assume that all variables occuring in $\alpha_1, ..., \alpha_n$ are bounded and no variable appears positive and negative in some α_i.

We say a literal L_1 is before a literal L_2, if the variable of L_1 occurs in the order of the prefix before the variable of L_2 and we write clauses in the form $(L_1 \vee ... \vee L_t)$, where L_{i-1} is before L_i for $1 < i \leq t$. An $X - literal$ (resp. $Y - literal$) is a literal of the form x_l or $\overline{\mathrm{x}_l}$ (resp. y_l or $\overline{\mathrm{y}_l}$). A *pure* $X - clause$ is a non–tautological clause consisting exclusively of X-literals. In particular the empty clause $\sqcup$ is a pure X-clause.

Looking for the desired resolution operation we can observe the following:

If a non–tautological clause with universal variables only appears in the quantified Boolean formula, then the formula is false. Thus, a pure X–clause can be replaced by the empty clause.

A formula $\exists \mathrm{y}_1 \ldots \mathrm{y}_n(\alpha_1 \wedge \ldots \wedge \alpha_m)$ is false iff ordinary resolution leads to the empty clause. That means existential variables will be matched. Therefore we willl restrict the resolution in the case of clauses with existential and universal variables to existential literals only.

If a clause contains an existential variable and a universal variable x not before an existential variable, then the universal variable x can be removed without effect to the truth of the formula. Hence, we will remove all universal variables of a resolvent which are not before an existential variable of the resolvent.

A more technical problem are the tautological clauses. The ordinary resolution may lead to tautological clauses, but in order to generate the empty clause they are not useful. For the sake of the simplicity we assume that quantified boolean formulas doesn't contain tautological clauses and we will forbid tautological resolvents.

Now we introduce our generalized resolution operation, called Q–resolution:

Definition (1)

a: *Let α be a non–tautological X–clause, then we replace α by the empty clause.*

b: *Let α_1 a clause with Y-literal y_l and α_2 a clause with Y-literal $\overline{\mathrm{y}_l}$, then the resolvent α of α_1 and α_2 is obtained as follows:*

1. *Remove all occurences of y_l and $\overline{\mathrm{y}_l}$ in $\alpha_1 \vee \alpha_2$.*
2. *Remove all occurences of X-literals, that are not before any Y-literal occuring in the disjunction.*
3. *If the resulting clause contains complementary literals, then no resolvent exists. Otherwise the resulting clause is the resolvent.*

Comparing ordinary resolution with Q–resolution, we see that only literals bounded by existential quantifiers can be matched and universal variables not before an Y-literal will be eliminated.

Example:

$$\forall \mathrm{x}_1\mathrm{x}_2\exists \mathrm{y}_1\mathrm{y}_2\forall \mathrm{x}_3((x_1 \vee \overline{\mathrm{x}_2} \vee \overline{\mathrm{y}_1} \vee \overline{\mathrm{y}_2} \vee \mathrm{x}_3) \wedge (\overline{\mathrm{x}_1} \vee \overline{\mathrm{y}_1}) \wedge (\overline{\mathrm{x}_2} \vee \mathrm{y}_2) \wedge (\overline{\mathrm{x}_1} \vee \overline{\mathrm{y}_2}))$$

The following Q–resolution steps can be performed:

$$(\mathrm{x}_1 \vee \overline{\mathrm{x}_2} \vee \overline{\mathrm{y}_1} \vee \overline{\mathrm{y}_2} \vee \mathrm{x}_3),\ (\overline{\mathrm{x}_2} \vee \mathrm{y}_2) \ \vdash_{\overline{Q-Res}} \ (\mathrm{x}_1 \vee \overline{\mathrm{x}_2} \vee \overline{\mathrm{y}_1})$$

$$(\overline{\mathrm{x}_1} \vee \overline{\mathrm{y}_2}), (\overline{\mathrm{x}_2} \vee \mathrm{y}_2) \ \vdash_{\overline{Q-Res}} \sqcup$$

where $\vdash_{\overline{Q-Res}}$ denotes the application of Q–resolution steps.

It is well known that resolution is complete and sound for propositional formulas in CNF, i.e. a formula $\exists \mathrm{y}_1 \ldots \mathrm{y}_n$ $(\alpha_1 \wedge \ldots \wedge \alpha_n)$ is false iff the empty clause can be derived by resolution applied to $\alpha_1 \wedge ... \wedge \alpha_n$. A similar result will be shown for Q–resolution and quantified Boolean formulas.

Theorem 2.1 *A quantified Boolean formula $\Phi = \forall x_1 \exists y_2 \ldots \exists y_{k-1} \forall x_k(\phi_1 \wedge \ldots \wedge \phi_m)$ is false iff $\Phi \vdash_{\overline{Q-Res}} \sqcup$.*

Proof: At first we assume Φ is false and show by induction that the empty clause can be generated by Q–resolution.

If $k = 2$ then Φ is of the form $\exists y_1 \ldots y_{m_1} \forall x_1 \ldots x_{n_1} (\phi_1 \wedge \ldots \wedge \phi_m)$.
We can omitt all occurences of x_i-variables and obtain an equivalent formula

$\exists y_1 \ldots y_1 (\phi'_1 \wedge \ldots \wedge \phi'_m)$. Since in that case Q–resolution equals ordinary resolution and resolution is complete the empty clause can be derived.

Let be $k = 3$ and the formula of the form
$\Phi = \forall x_1 \ldots x_{n_1} \exists y_1 \ldots y_{m_1} \forall x_{n_1+1} \ldots x_{n_2} (\phi_1 \wedge \ldots \wedge \phi_m)$.
Now let Π_0 (resp. Π_1) be the formula obtained by removing all clauses with literals x_1 (resp.$\overline{x_1}$) and omitting all literals $\overline{x_1}$ (resp. x_1). That means Π_0 is the formula with assignment $x_1 = 1$ and Π_1 the formula with assignment $x_1 = 0$.

Since Φ is false at least one of Π_0 and Π_1 is false. Without loss of generality let Π_0 be false. Then by induction on (n_1) and the induction hypothesis (k=2) the empty clause is derivable by Q–resolution starting with Π_0. If we add the removed $\overline{x}_1$-occurences in clauses of Π_0, then no literal x_1 occurs. Thus we can generated the empty clause.

Let be $k > 3$ then the formula is of the form $\exists\ y_1 \ldots y_{m_1}\ \forall x_2 \ldots \exists y_{k-1} \forall x_k (\phi_1 \wedge \ldots \wedge \phi_m)$.

If $m_1 = 1$ then let Π_1 (resp. Π_0) be the formula which we obtain by an assignment $y_1 = 1$ (resp. $y_1 = 0$). By the induction hypothesis (k) it yields $\Pi_1 \vdash_{\overline{Q-Res}} \sqcup$ and $\Pi_0 \vdash_{\overline{Q-Res}} \sqcup$.

Hence we see that $\Phi \vdash_{\overline{Q-Res}} (\sqcup \text{ or } y_1)$ and $\Phi \vdash_{\overline{Q-Res}} (\sqcup \text{ or } \overline{y_1})$. Therefore we can conclude $\Phi \vdash_{\overline{Q-Res}} \sqcup$.

For $m_1 > 1$ again we assign $y_1 = 1$ (resp. $y_1 = 0$) and obtain by means of the induction hypothesis (k, m_1) that the empty clause is derivable.

Altogether we have shown that the empty clause is derivable, if the formula Φ is false.

To prove the converse direction it suffices to show the following :
Let $\Phi = \forall x_1 \exists y_2 \ldots \exists y_{k-1} \forall x_k (\phi_1 \wedge ... \wedge \phi_m)$ a quantified formula and ϕ a clause derivable by an Q–resolution of ϕ_i and ϕ_j, then $\forall x_1 \exists y_2 \ldots \forall x_k (\phi_1 \wedge \ldots \wedge \phi_m \wedge \phi)$ is true, if Φ is true. That is obvious, which can be seen by considering the definition of Q–resolution.

q.e.d.

3 Q–Unit–Resolution

Analogously to the case of unit–resolution for ordinary resolution we can define a Q–unit resolution. A clause ϕ is called a $Y-$ *unit clause* if ϕ contains exactly one Y–literal and arbitrarily many X–literals. A *positive* Y–unit clause is a unit clause with a positive Y–literal.

Definition (2) *The Q–unit–resolution (Q–U–Res) is a Q–resolution, where one of the clauses is a positive Y–unit clause.*

The Q–unit–resolution is useful not only for quantified Horn formulas, but for an extension of quantified Horn formulas too. As we will see the X–part of the clauses can be arbitrarily given, whereas the Y–part must be in Horn form.

Definition (3) *A formula $\Phi = \forall x_1 \exists y_1 \ldots \forall x_k (\phi_1 \wedge \ldots \wedge \phi_m)$ is an extended quantified Horn formula, if for each ϕ_i the Y–part is a Horn clause, i.e. the clause without all X–literals is a*

Horn clause.

Theorem 3.1 *The Q–unit–resolution is complete and sound for extended quantified Horn formulas.*

Proof: We show, that a formula Φ is false iff the empty clause can be derived from Φ by Q–unit–resolution. Since Q–unit resolution is a restriction of Q–resolution it suffices to prove that the empty clause is derivable, if Φ is false.

Let $\mathrm{U} = \{ x\mathrm{w} : \mathrm{w}$ is a Y–literal, x disjunction of X–literals, $\Phi \vdash_{Q-U-Res} x\mathrm{w} \}$ be the set of derivable Y–unit clauses, i.e. clauses containing exactly one Y–literal.

Let $a = \mathrm{a}_1, \ldots, \mathrm{a}_k$ be an arbitrary assignment of the X–variables $\mathrm{x}_1, \ldots, \mathrm{x}_{n_k}$. Then we define the assignment b_i for the variable y_i as follows: If there is some Y–unit clause $x\mathrm{y}_i$ in U that is not true under the partial assignment of x, then let b_i equal 1. In all other cases we define b_i equal 0.

Now we will show, that Φ is true under the above assignment, if the empty clause isn't derivable. For each clause ϕ in Φ, let t be the number of Y–literals occuring in ϕ. If t=1 and the empty clause isn't derivable, then the clause ϕ is in U and true under the assignment.

Now we want to treat clauses with $t > 1$. Because of the Horn property ϕ must contain a negative Y–literal $\overline{\mathrm{y}_i}$. Let x be the X–part of the clause ϕ. We asssume ϕ is false under the assignment, that means $\mathrm{b}_i = 1$. Hence there is some Y–unit clause $x_1\mathrm{y}_i$ in U, such that x_1 is false for a. Since the X–part x of ϕ and x_1 are false, no X–literal occurs positive and negative in $(x \vee x_1)$. Therefore Q–unit resolution can be applied to ϕ and $x_1\mathrm{y}_i$ obtaining a clause $\phi^* = (\ldots[x, x_1]\ldots)$. $[x, x_1]$ denotes the resulting disjunction of the X–part x and x_1.

Since ϕ^* contains t–1 Y–literals we can apply the induction hypothesis with ϕ^* and obtain our desired result ϕ is true.

q.e.d.

Theorem 3.2 *There exist quantified extended Horn formulas Φ_t $(t \geq 1)$ of length $18t + 1$, which are false and a refutation to the empty clause requires at least 2^t Q–resolution steps.*

Proof:
The formula Φ_t is defined as follows:
$\exists \mathrm{y}_0\mathrm{y}_1\mathrm{y}_1' \, \forall \mathrm{x}_1 \exists \mathrm{y}_2\mathrm{y}_2' \, \forall \mathrm{x}_2 \exists \mathrm{y}_3\mathrm{y}_3' \ldots \forall \mathrm{x}_{t-1} \exists \mathrm{y}_t\mathrm{y}_t' \forall \mathrm{x}_t \exists \mathrm{y}_{t+1} \ldots \mathrm{y}_{t+t}$

$$
\begin{array}{l}
(\bar{\mathrm{y}}_0) \wedge \\
(\mathrm{y}_0 \leftarrow \mathrm{y}_1\mathrm{y}_1') \wedge \\
(\mathrm{y}_1 \leftarrow \mathrm{x}_1\mathrm{y}_2\mathrm{y}_2') \wedge (\mathrm{y}_1' \leftarrow \bar{\mathrm{x}}_1\mathrm{y}_2\mathrm{y}_2') \wedge \\
(\mathrm{y}_2 \leftarrow \mathrm{x}_2\mathrm{y}_3\mathrm{y}_3') \wedge (\mathrm{y}_2' \leftarrow \bar{\mathrm{x}}_2\mathrm{y}_3\mathrm{y}_3') \wedge \\
\vdots \qquad\qquad\qquad \vdots \\
\vdots \qquad\qquad\qquad \vdots \\
(\mathrm{y}_{t-1} \leftarrow \mathrm{x}_{t-1}\mathrm{y}_t\mathrm{y}_t') \wedge (\mathrm{y}_{t-1}' \leftarrow \bar{\mathrm{x}}_{t-1}\mathrm{y}_t\mathrm{y}_t') \wedge \\
(\mathrm{y}_t \leftarrow \mathrm{x}_t\mathrm{y}_{t+1} \ldots \mathrm{y}_{t+t}) \wedge (\mathrm{y}_t' \leftarrow \bar{\mathrm{x}}_t\mathrm{y}_{t+1} \ldots \mathrm{y}_{t+t}) \wedge
\end{array}
$$

$$(x_1 y_{t+1}) \wedge \ldots \wedge (x_t y_{t+t}) \wedge$$
$$(\bar{x}_1 y_{t+1}) \wedge \ldots \wedge (\bar{x}_t y_{t+t})$$

As easily can be seen, each positive y–unit clause with y–literal $y_i(y_i^{'})$ $(i = 1 \ldots t)$ is of the form $\bar{x}_1\bar{x}_2 \ldots \bar{x}_{i-1}y^{'}$. and is obtained by Q–unit–resolution in at least 2^{t-i} steps. So each of y_1 od $y_1^{'}$ is obtained in at least 2^{t-1} steps leading to the empty clause in at least 2^t Q–unit–resolution steps. (This also shows the falsity of Φ_t)

A gain in complexity by using unrestricted Q–resolution would result from the merging of literals form the two parents of a resolution step. But each clause with head $y_i(y_i^{'})$, that is no y–unit, contains $x_i(\bar{x}_i)$ and at least one y_j or $y_j^{'}$, that is no y–unit, contains $x_i(\bar{x}_i)$ and at least one y_j or $y_j^{'}$ with $j > i$ in the body.

So, if one has resolved a clause containing $y_i y_i^{'}$ in the body with a clause with head $y_i(y_i^{'})$, that is no y–unit the resolvent contains $y_i^{'} x_i(y_i\bar{x}_i)$ and all $y_j^{'}$ of the clause with head $y_i(y_i^{'})$ with $j > i$.

There is no possibility to use the resolvent in a resolution step with a clause with head $y_i^{'}(y_i)$ before all $y_j^{'}$ with $j > i$ are "resolved away", so that the blocking $x_i(\bar{x}_i)$ is deleted. That means, that even if the clause with head $y_i^{'}$ has the same $y_j^{'}(j > i)$ in the body as the clause with head y_i no merging is possible and the same $y_j^{'}$ have to be "resolved away" a second time.

Since $(\bar{y}_0)$ is the only negative clause in Φ_t, one has to resolve with a clause containing $y_1 y_1^{'}$ in the body. That means one has to resolve upon y_2 and $y_2^{'}$ when $x_1 y_2 y_2^{'}$ is introduced in the resolution tree and again when $\bar{x}_1 y_2 y_2^{'}$ is introduced, one has to resolve upon $y_3 y_3^{'}$ four lines (when $x_1 x_2$, $x_1\bar{x}_2$, $\bar{x}_1 x_2$ and $\bar{x}_1\bar{x}_2$ is introduced) and so on.

This results in at least 2^t Q–resolution steps.

Theorem 3.3 *The evaluation problem for extended quantified Horn formulas with prefix* $\forall\exists$ *is coNP–complete.*

Proof: That the falsity of an extended quantified Horn formula can be decided nondeterministically in polynomial time is obvious. Thus the evaluation problem belongs to coNP and the completeness remains to show.

We will associate to each formula $\alpha = \alpha_1 \wedge \ldots \wedge \alpha_n$ in 3–CNF an extended quantified Horn formula, such that the propositional formula α is satisfiable iff the extended quantified Horn formula is false.

To each clause $\alpha_i = (L_{i_1} \vee L_{i_2} \vee L_{i_3})$ with literals L_{i_j} we associate three clauses $\phi_{i_1} = (L_{i_1} \vee y_i)$, $\phi_{i_2} = (L_{i_2} \vee y_i)$ and $\phi_{i_3} = (L_{i_3} \vee y_i)$.

Then we define $\Phi := \forall x_1 \ldots x_s \exists y_1 \ldots y_n((\overline{y_1} \vee \ldots \vee \overline{y_n}) \wedge \phi_{1_1} \wedge \phi_{1_2} \wedge \ldots \wedge \phi_{n_3})$, where $x_1, \ldots x_s$ are the variables of the propositional formula α.

If Φ is false, then it yields $\Phi \vdash_{Q-Res} \sqcup$. Since for each i one of the clauses ϕ_{i_1} or ϕ_{i_2} or ϕ_{i_3} must be matched, there is some $L_{i_{j_i}}$ in α_i, such that $\{L_{1_{j_1}}, \ldots L_{n_{jn}}\}$ doesn't contain complementary literals. Thus, we can define a truth assignment I for α choosing $I(L_{i_{j_i}})$=1.

In the converse direction let α be satisfiable. Then there is some truth assignment I, such that $I(\alpha) = 1$.Hence, for each i there is some literal $L_{i_{j_i}}$ with $I(L_{i_{j_i}}) = 1$. Now we can apply the the Q–resolution to Φ obtaining the empty clause as follows:

$(\bar{y}_1 \vee \ldots \vee \bar{y}_n), (L_{1_{j_1}} \vee y_1) \vdash_{\overline{Q-Res}} (L_{1_{j_1}} \vee \bar{y}_2 \vee \ldots \vee \bar{y}_n), (L_{2_{j_2}} \vee \bar{y}_2) \vdash_{\overline{Q-Res}} \cdots \vdash_{\overline{Q-Res}}$ $(L_{1_{j_1}} \vee \ldots \vee L_{n_{j_{n-1}}} \vee \bar{y}_n), (L_{n_{jn}} \vee y_n) \vdash_{\overline{Q-Res}} \sqcup.$

Since the satisfiability problem for 3–CNF is NP–complete and the transformation can be performend in polynomial time, we have proved our desired result.

q.e.d.

In the remainder of this chapter we will give some simplification rules, which transform in linear time quantified Horn formulas in equivalent Horn formulas with prefix $\forall x \exists y$.

Let Φ be the formula $\forall x_1 \exists y_1 \ldots \exists y_{k-1} \forall x_k (\phi_1 \wedge \ldots \wedge \phi_m)$.

At first we associate to each clause ϕ_j the clause $\phi_j(Y)$ omitting all X–literals and define $\Phi(Y) = \phi_1(Y) \wedge \ldots \wedge \phi_m(Y)$. The formula $\Phi(Y)$ can be divided into NEG(Φ(Y)), the set of clauses with negated literals only, and REST(Φ(Y)) = Φ(Y) - NEG(Φ(Y)).

Furthermore Unit(Φ(Y)) := { y : REST(Φ(Y)) $\models$ y } is the set of literals, which are consequences of clauses with at least one positive literal.

Unit(Φ(Y)) can be computed in O(n) time adepting the linear time algorithm for the satisfiability problem for Horn formulas cf. [DoGa 84].

The **simplification rules** are defined as follows: Let Unit(Φ(Y)) = { $y_{i_1}, \ldots, y_{i_r}$ } be given.

For each clause ϕ_i we consider the following two cases:
Case 1 : $\phi_i = (\overline{x_{i_1} y_{i_1}} \ldots \overline{x_{i_r} y_{i_r} x_p} y_j \overline{y_k x_{j_1} y_{j_1}} \ldots \overline{x_{j_t} y_{j_t}})$.

If $A_i := \{y_{i_1}, \ldots y_{i_r}, y_k, y_{j_1}, \ldots y_{j_t}\} \subseteq$ Unit(Φ(Y))
  then replace ϕ_i by $\delta_i :=(\overline{x_{i_1}} \ldots \overline{x_{i_r} x_p y_{i_1}} \ldots \overline{y_{i_r}} y_j \overline{y_k y_{j_1}} \ldots \overline{y_{j_t}})$
  else delete ϕ_i.

If $A_i \not\subseteq$ Unit(Φ(Y)) then some of the Y–Literals $\overline{y_t}$ $(t \neq j)$ can't be eliminated by the Q–unit resolution. Hence Φ is false iff $\Phi - \phi_i$ is false. If $A_i \subseteq$ Unit(Φ(Y)), we can apply each Y–unit clause $x y_p$, $y_p \in A_i$ to ϕ_i, because x consists of negative X–literals only. Since X–literals not before any Y–literals will be deleted, ϕ_i can be replaced by δ_i.

Case 2 : $\phi_i = \{\overline{x_{i_1} y_{i_1}} \ldots \overline{x_{i_s} y_{i_s} x_{t_1}} x_t \overline{x_{t_2} y_{j_1} x_{j_1}} \ldots \overline{y_{j_r} x_{j_r}}\}$
if $A_i = \{y_{i_1}, \ldots, y_{i_s}, y_{j_1}, \ldots, y_{j_r}\} \subseteq$ Unit(Φ(Y))
  then replace ϕ_i by $\delta_i := (x_t \overline{y_{j_1}} \ldots \overline{y_{j_r}})$
  else delete ϕ_i

It yields, the Y–part of ϕ_i is in NEG(Φ(Y)). If $A_i \not\subseteq$ Unit(Φ(Y)), then the empty clause can't be obtained by means of ϕ_i, because some of the Y–literals of ϕ_i remain. Hence we can delete ϕ_i. If $A_i \subseteq$ Unit(Φ(Y)), then the only problem arises with the X–literals x_t. Since all X–literals of clauses with Y–part in RESTΦ(Y)) are negative, we can omitt all negative X–literals different from x_t in ϕ_i. Furthermore, if ϕ_i is used to generate the empty clause, at first we can match the Y–literals $y_{j_1}, \ldots, y_{j_r}$. If a Y–unit with positive Y–literals $y_{i_1}, \ldots, y_{i_s}$ is applied to ϕ_i, then no x_t–literal occurs in the unit–clause, because X–literals not before a Y–literal are removed. Therefore we can move the literal x_t before each Y–literal.

Altogether we have clauses, where the remaining X–literals are before the Y–literals. Therefore we can rearrange the prefix and obtain a formula

$$\Phi_{SR} := \forall x \exists y(\delta_1 \wedge \ldots \wedge \delta_m).$$

for which the following theorem yields.

Theorem 3.4 *Let Φ be a quantified Horn formula, then Φ is true iff Φ_{SR} is true.*

The computation of Unit(Φ(Y)) can be done in linear time and the required time for the simplification depends linearly on the length of the clauses. We can store Unit(Φ(Y)) in an array $[a_1, \ldots, a_t]$, where t is the number of Y–variables and $a_j = 1$, if $y_j \in$ Unit(Φ(Y)) and $a_j = 0$ otherwise.

Lemma 3.5 *Each quantified Horn formula Φ can be simplified in linear time. The resulting quantified Horn formula Φ_{SR} is of the form $\forall x \exists y(\phi_1 \wedge \ldots \wedge \phi_m)$ with clauses $x_i\overline{y}$ and $\overline{x}\overline{y}y_j$.*

On the basis of the simplification we can improve the upper bound for the evaluation problem for quantified Horn formulas.

Theorem 3.6 *The evaluation problem for quantified Horn formulas can be decided in O(rn) time, where n is length of the formula and r is the number of X–variables occuring positive in the formula.*

Proof: We simplify Φ and obtain in linear time a formula Φ_{SR} of the form $\forall x \exists y(\alpha_1 \wedge \ldots \wedge \alpha_m)$ with clauses $x_i\overline{y}$ and $\overline{x}\overline{y}y_j$. We can assume that no pure X–clause exists, otherwise Φ is false. Without loss of generality $x_1, \ldots, x_r$ are the X–variables with positive occurence in Φ_{SR}. Now let $\Phi_{SR}(i)$ be defined as follows
$\Phi_{SR}(i) = \{\overline{y} | \exists j : \alpha_j = x_i\overline{y}\} \cup \{\overline{y} | \exists j : \alpha_j = \overline{y}\} \cup \{\overline{y}y | \exists j : \alpha_j = \overline{x}\overline{y}y$ and $\overline{x_i}$ not in $\overline{x}\}$.
Then it yields: Φ is false iff Φ_{SR} is false iff there is some i, such that $\Phi_{SR}(i)$ isn't satisfiable. The satisfiability of $\Phi_{SR}(i)$ can be decided in linear time applying the linear algorithm for Horn formulas [DoGa 84]. Altogether we require O(rn) time, where n is the length of the formula.

q.e.d.

4 Multi Horn Formulas

In this chapter we present a problem which is in a certain sense equivalent to the evaluation problem for quantified Horn formulas. This problem is concerned with sets of formulas. When we are dealing with a set of formulas sometimes some formulas have clauses in common and therefore we can store those clauses once and add an index to the clause indicating to which formula the clause belongs.

An *index–clause* is a tuple $([i_1, \ldots, i_m], \alpha)$, where α is a clause and $[i_1, \ldots i_m]$ is a list of natural numbers $i_1 < i_2 < \ldots < i_m$. A *multi Horn formula* Φ with *Index* $Ind(\Phi) = \{1, \ldots, m\}$ is a conjunction of index–Horn clauses $(I_1, \alpha_1) \wedge \ldots \wedge (I_n, \alpha_n)$ and $Ind(\Phi) = \{i \mid \exists j : i \in I_j\}$.

The Horn formula $\Phi(i)$ of a multi Horn formula Φ is defined as follows: $\Phi(i) := \{\alpha_j \mid (I_j, \alpha_j)$ index-clause of $\Phi, i \in I_j\}$

Example:
$\Phi = \{([1], \overline{A_1}), ([2], \overline{A_2}), ([2,3], A_1 \vee \overline{A_2}), ([1,2], A_1 \vee \overline{A_3}), ([3], A_2 \vee \overline{A_4}), ([2,3], A_3)\}$

is a multi Horn formula with $Ind(\Phi) = \{1,2,3\}$ and with formulas

$\Phi(1) = \{\overline{A_1}, A_1 \vee \overline{A_3}\}$
$\Phi(2) = \{\overline{A_2}, A_1 \vee \overline{A_2}, A_1 \vee \overline{A_3}, A_3\}$
$\Phi(3) = \{A_1 \vee \overline{A_2}, A_2 \vee \overline{A_4}, A_3\}$.

One typical problem is the satisfiability problem for multi Horn formulas.

$SAT(Multi\ Horn) := \{\Phi\ multi\ Horn \mid \forall i \in Ind(\Phi) : \Phi_i \in SAT\}$

As immediatly can be seen the formula given in the above example belongs to $SAT(Multi\ Horn)$. Since the satifiability problem for Horn formulas can be solved in linear time obviously the satisfiability of a multi Horn formula Φ can be decided in $O(mn)$ time, where n is the length of Φ and $m = |Ind(\Phi)|$.

Since an quantified Horn formula can be transformed in linear time into a formula with 3–clauses only, we restrict ourselves in the following to multi Horn formulas with 3–clauses only. Then we can improve the above mentioned upper bound for $SAT(Multi\ Horn)$.

Lemma 4.1 *$3 - SAT(Multi\ Horn)$ is solvable in linear time for multi Horn formulas with index–clauses.*

Proof: Let be given a multi Horn formula Φ. We replace each index–clause $([i_1, \ldots i_r], \alpha_i)$ by $([i_1], \alpha_i) \wedge \ldots \wedge ([i_r], \alpha_i)$. Then the resulting formula Φ^* has a length $O(length(\Phi))$ and each clause belongs to exactly one $\Phi^*(i)$. Thus we have separated Φ^* obtaining $|Ind(\Phi)|$ Horn formulas, each of them decidable in linear time.

q.e.d.

Instead of index–lists containing the indices to which a clause belongs, we can list the complement, that means for $Ind(\Phi) = \{1, \ldots, m\}$ and a index–clause (I, α) we can write (I^c, α), where $I^c := [1, \ldots, m] - I$. Such clauses are called *index − complement − clauses* and in order to obtain short expressions, in the following we allow both representations.

Example:
$\Phi = \{([1,2], \alpha_1), ([2]^c, \alpha_2), ([1,3], \alpha_3), ([1], \alpha_4), ([2]^c, \alpha_5)\}$ and $Ind(\Phi) = \{1,2,3,4\}$ is a multi Horn formula with
$\Phi(1) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$
$\Phi(2) = \{\alpha_1\}$
$\Phi(3) = \{\alpha_2, \alpha_3, \alpha_5\}$
$\Phi(4) = \{\alpha_2, \alpha_5\}$

In chapter 2 we have investigated simplification rules for quantified Horn formulas obtaining in linear time a formula of the form $\Phi = \forall x \exists y(\phi_1 \wedge \ldots \wedge \phi_r)$ with clauses $(\mathrm{x}_i \vee \overline{y})$ or $(\overline{\mathrm{x}_{i_1}} \vee \ldots \vee \overline{\mathrm{x}_{i_q}} \vee \overline{y} \vee \mathrm{y}_{j_t})$ Furthermore we can assume that no X–literal occurs positive or negative only in Φ.

Transformation 1:

Let be given $\forall x_1 \ldots x_s \exists y_1 \ldots y_t(\phi_1 \wedge \ldots \wedge \phi_r)$, then the associated multi Horn formula Φ is constructed a follows:

If a pure Y–clause ϕ_i exists **then** $Ind(\Phi) := \{0, \ldots, s\}$
else $Ind(\Phi) := \{1, \ldots, s\}$

and
if $\phi_i = (x_i \vee \overline{y})$ **then** $m(\phi_i) := ([i], \overline{y})$
if $\phi_i = (\overline{x_{i_1}} \vee \ldots \vee \overline{x_{i_q}} \vee \overline{y} \vee y_s)$ **then** $m(\phi_i) := ([i_1, \ldots, i_q]^c, \overline{y} \vee y_s)$
if ϕ_i contains no X–literals **then** $m(\phi_i) := ([]^c, \phi_i)$
and $m(\Phi) := (m(\phi_1) \wedge \ldots \wedge m(\phi_m))$

Theorem 4.2 *Each quantified Horn formula $\Phi = \forall x \exists y(\phi_1 \wedge \ldots \phi_m)$ is true iff the associated multi Horn formula $m(\Phi)$ is in $SAT(Multi\ Horn)$.*

Proof: Let Φ be the formula $\forall x_1 \ldots x_s \exists y_1 \ldots y_t(\phi_1 \wedge \ldots \phi_m)$. We abbreviate
$Pos(0) := \{\phi | \phi$ clause in Φ without X–literals and negative Y–literals only $\}$,
$Neg(0) := \{\phi | \phi$ clause in Φ with one positive Y–literal and without X–literals $\}$ and for each $i > 0$:
$Neg(i) := \{\overline{x}\alpha$ clause in $\Phi \mid \overline{x}$ contains $\overline{x_i}\ \}$
$Pos(i) := \{x_i\overline{y} \mid x_i\overline{y}$ clause in $\Phi\}$

Then Φ is false iff $\Phi \vdash_{Q-U-Res} \sqcup$ iff there is some i
$K(i) := Pos(i) \cup\ Pos(0) \cup\ Neg(0) \cup_{j \neq i}\ Neg(j) \vdash_{Q-U-Res} \sqcup$.

Thus we have sets of clauses K(i) for which the X-part does‘nt play any role.

The associated index clauses to K(i) are exactly the Horn formulas $m(\Phi(i))$:

$$(x_i\overline{y}) \in Pos(i) : ([i], \overline{y})$$
$$(\overline{y}) \in Pos(0) : ([]^c, \overline{y})$$
$$(\overline{y}y) \in Neg(0) : ([]^c, \overline{y}y)$$
$$(\overline{x_{j_1}} \ldots \overline{x_{j_r}}\overline{y}y) \in Neg(j)(j \neq i) : ([j_1, \ldots j_r]^c, \overline{y}y).$$

Obviously $m(\Phi)(i)$ isn't satisfiable iff K(i) $\vdash_{Q-U-Res} \sqcup$.

Hence we have proved Φ is true iff $m(\Phi)$ is in SAT(Multi Horn).

q.e.d.

Each quantified Horn formula can be transformed in linear time in a formula with clauses of at most 3 literals. By lemma 3.5 the formula can be simplified in linear time obtaining a prefix $\forall x \exists y$ and 3–clauses. Then we can associate a multi Horn formula with 3–clauses too. By theorem 4.2 the resulting Horn formula is in 3–SAT(Multi Horn) iff the quantified Horn formula is true.

Corollary 4.3 *We can compute in linear time to each quantified Horn formula Φ a multi Horn formula $m(\Phi)$, such that Φ is true iff $m(\Phi)$ is in 3–SAT(Multi Horn).*

By theorem 4.1 the problem 3–SAT(Multi Horn) is solvable in linear time, if the formula contains index–clauses only. In the case of index–complement–clauses used in tranformation 1, it is not known, whether a linear time algorithm exists. Since this problem is directly related to the evaluation problem for quantified Horn formulas, a linear algorithm for multi Horn formulas with index and index–complement clauses would imply a linear algorithm for the evaluation problem.

In the other direction each multi Horn formula can be described in terms of quantified Horn formulas with prefix $\forall x \exists y$.

Transformation 2:
Let Π be a multi Horn formula $(I_1{}^c, \alpha_1) \wedge \ldots \wedge (I_s{}^c, \alpha_s) \wedge (I_{s+1}, \alpha_{s+1}) \wedge \ldots \wedge (I_r, \alpha_r)$ and index set $Ind(\Pi) = \{1, \ldots, m\}$, then we associate with Π the quantified Horn formula Φ as follws: Without loss of generality we assume that $\alpha_1, \ldots, \alpha_r$ contain the variables $\mathrm{y}_1, \ldots, \mathrm{y}_p$.

For each clause α_i with negative literals only and representation $(I_i{}^c, \alpha_i)$ we replace $I_i{}^c$ by $I_i := I - I_i{}^c$ obtaining (I_i, α_i)
and for each clause α_i with some positive literal and representation (I_i, α_i) we replace I_i by $I_i{}^c = I - I_i$ obtaining $(I_i{}^c, \alpha_i)$.

Then we define for each (I_i, α_i) (note: α_i contains negative Y–literals only)
if $I_i = [i_1, \ldots, i_p]$ **then** $\phi_{i_1} = \mathrm{x}_{i_1}\alpha_i, \ldots, \phi_{i_p} = \mathrm{x}_{i_p}\alpha_i$
and for each $(I_i{}^c, \alpha_i)$ (α_i contains some positive Y–literal):

if $I_i{}^c = [i_1, \ldots, i_p]^c$ **then** $\phi_{i_1} = \overline{\mathrm{x}_{i_1}} \ldots \overline{\mathrm{x}_{i_p}}\alpha_i$
and $\Phi := \forall \mathrm{x}_1 \ldots \mathrm{x}_m \exists \mathrm{y}_1 \ldots \exists \mathrm{y}_p(\phi_{1_1} \wedge \ldots \wedge \phi_{r_p})$.

Obviously it yields: Φ is true iff Π is in SAT(Multi Horn), because the previous transformation from quantified Horn formulas to multi Horn formulas applied to Φ leads to Π.

q.e.d.

5 Conclusion

We have introduced a complete and sound resolution operation directly applicable to quantified Boolean formulas. For propositional formulas in conjunctive normal form Haken[Ha 85] has shown that ordinary resolution requires exponential time. Since Q – resolution is a generalization of ordinary resolution, obviously for a false formula at least exponential many clauses must be generated. Because of the PSPACE – completeness of the evaluation problem it would be interesting to prove essentially larger lower bound for our resolution operation.

References

[APT 78] B. Aspvall, M. F. Plass and R.E. Tarjan: *A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas*, Information Processing Letters, Volume 8, number 3, 1978

[DoGa 84] W. F. Dowling and J. H. Gallier: *Linear-Time Algorithms For Testing The Satisfiability Of Propositional Horn Formulae*, J. of Logic Programming, 1984

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

[Ha 85] A. Haken: *The intractability of resolution*, Theoret. Comput. Sci. 39, 1985, 297–308

[KKBS 87] M. Karpinski, H. Kleine Büning and P.H. Schmitt: *On the computational complexity of quantified Horn clauses*, Lecture Notes in Computer Science 329, Springer-Verlag, 1987

[StM 73] L. J. Stockmeyer and A. R. Meyer: *Word problems requiring exponential time*, Proc. 5th Ann. ACM Symp. Theory Computer, 1973