

Resolution on Quantified Generalized Clause-sets

Jiwei Jin *

Xishun Zhao

*Institute of Logic and Cognition,
Sun Yat-sen University,
People's Republic of China*

jinjiwei2004@gmail.com

hsszxs@mail.sysu.edu.cn

Abstract

This paper is devoted to investigate resolution for *quantified generalized clause-sets* (QCLS). The soundness and refutation completeness are proved. Then *quantified generalized Horn clause-sets* are introduced for which a restricted resolution, called *quantified positive unit resolution*, is proved to be sound and refutationally complete. Moreover, it is shown that the satisfiability for quantified generalized Horn clause-sets is solvable in polynomial time. On the one hand, the work of this paper can be considered as a generalization of resolution for generalized clause-sets (CLS). On the other hand, it also can be considered as a generalization of Q-resolution for quantified boolean CNF formulae (QCNF).

KEYWORDS: *quantified generalized clause-sets, satisfiability, Q-resolution, Horn*

Submitted July 2009; revised November 2009; published March 2010

1. Introduction

Boolean formulae and quantified boolean formulae (QBF) are widely used in computer science and AI. Many problems (such as reasoning, planing, verification, etc) can be expressed as satisfiability problems of boolean formulae or quantified boolean formulae. These formulae use boolean variables which can be assigned two values: true, false (or 1, 0). However, satisfiability problems with constraint variables which have more than two values occur at many places, for example in coloring problems. A popular method to deal with these problems is to translate them into boolean satisfiability problems, but this method hides to a certain degree the structure of the original problem. Hence it is not suitable for theoretical studies on the structure of the original problem. So, formulae with non-boolean variables (such as signed formulae, generalized clause-sets, etc.) have been introduced recently and some interesting results have been obtained (see e.g. [2, 13, 14, 18]). Among these non-boolean variables, negative monosigned literals can make formulae be processed efficiently [14]. Kullmann has generalized boolean formulae to generalized clause-sets (CLS) by using negative monosigned literals instead of boolean literals and obtained many interesting results [15, 16, 17] et al.

There have been some important algorithms which are proved to be powerful to solve some hard instances of satisfiability problem. Among these algorithms, the resolution ap-

* Research was partially supported by the NSFC projects under Grant No.60970040 and a MOE project under Grant No. 05JJJD72040122

proach plays an important role. The resolution calculus was first presented in a more general form by Robinson in [20], while the propositional version of the method was already introduced by A. Blake [3] in 1937, and later used by W. V. Quine et al under the name of “consensus” (e.g., [19]). The generalization of resolution to quantified boolean CNF formulae, called Q-resolution, was first introduced in [9]. Resolution and Q-resolution have not only been applied in some SAT or QSAT solvers (e.g., [5, 21]), but also adopted to prove some theoretical results (e.g., [10, 11, 12]). The generalization of resolution to generalized clause-sets was studied in [13].

For some problems such as gaming, reasoning, planning and so on, quantifiers can make formulae, which are encodings of these problems, more compact. Usually these formulae (with quantifiers) are quantified boolean formulae (QBF). In case when non-boolean variables are used, quantifiers on non-boolean variables should be adopted. Parallel to quantified boolean CNF formulae, it is natural to introduce quantified generalized clause-sets (QCLS), which can be considered as the generalization of quantified boolean CNF formulae. Because of non-boolean variables and quantifiers, quantified generalized clause-sets might express some problems more naturally. In Section 3, after the formal definition of quantified generalized clause-sets, we will give an example which uses quantified generalized clause-sets to express two-person coloring game problems.

One can define generalized k -clause-sets (k -CLS) naturally as the set of clause-sets in which each clause has at most k literals. It is well known that the satisfiability problem for boolean 2-CNF and Q2-CNF is tractable ([1, 6, 11]). However, this is not true for generalized 2-clause-sets and quantified generalized 2-clause-sets. From [14] it is easy to see that 3-coloring problem can be reduced to 2-CLS, hence the satisfiability problem for 2-CLS is NP-complete. In section 3, we show that the satisfiability problem for quantified generalized 2-clause-sets is PSPACE-complete.

In [14, 15, 16, 17], Kullmann defined several transformations from CLS to CNF. Besides preserving the satisfiability, these transformations preserve structural properties. Therefore, some results on CNF formulae (e.g., CNF formulae with fixed deficiency) can be adapted directly to CLSs. However, due to the quantifiers (in particular, the universal quantifiers for non-boolean variables), these transformations can not be extended to quantified generalized clause-sets with structural properties (e.g, the deficiency) preserved. This motivates us to investigate approaches on quantified generalized clause-sets. This paper is to introduce and investigate resolution for quantified generalized clause-sets. The soundness and refutation completeness of this kind of resolution will be proved.

The satisfiability problem for (boolean) Horn formulae can be solved in polynomial time, and this is also true for quantified Horn formulae. Also for some signed formulae, especially regular signed formulae, Horn structure has also been developed (e.g., [2]). These signed formulae are build over variables whose domains have special structure, for example, totally ordered. A literal is of the form $\uparrow i:v$ or $\downarrow i:v$, where v is a variable and i is a value in the domain of v . The literal $\uparrow i:v$ (resp. $\downarrow i:v$) means that v must take a value larger (resp. smaller) than i . By considering literals $\uparrow i:v$ (resp. $\downarrow i:v$) as positive (resp. negative), one can define signed Horn formulas in the usual way. However, domains of variables in CLS do not have such special structure. For each variable v we select exactly one literal on v as negative and all other literals as positive. Then we introduce generalized Horn clause-sets (HCLS), and prove that the satisfiability problem for generalized Horn clause-sets can be

solved in polynomial time. Further, we investigate quantified generalized Horn clause-sets (QHCLS), that is, quantified generalized clause-sets with Horn structure as matrices. It will be shown that a restricted resolution, called quantified positive unit resolution, is sufficient to refute false quantified generalized Horn clause-sets. Further, the satisfiability problem for quantified generalized Horn clause-sets is shown to be tractable.

The paper is organized as follows. In Section 2 we recall the definition of generalized clause-sets and some terminologies which will be used in this paper. Section 3 introduces quantified generalized clause-sets (QCLS). Besides an example which shows that it is quite natural to use QCLS to encode game problem, some properties of QCLS are also presented. Section 4 develops resolution for quantified generalized clause-sets, the soundness and refutation completeness will be proved. Section 5 is mainly devoted to the study of generalized Horn clause-sets and quantified generalized Horn clause-sets for which the satisfiability problem is shown to be tractable. Finally, Section 6 concludes this paper.

2. Preliminaries

In this section we recall the definition of generalized clause-sets [14, 15, 16, 17] and some terminology which will be used later on.

The universe of *variables* is a countably infinite set denoted by \mathcal{VA} , and the universe of *domains* is denoted by \mathcal{DOM} . Each variable $v \in \mathcal{VA}$ has a non-empty (value-)domain $D_v \subseteq \mathcal{DOM}$. For each $v \in \mathcal{VA}$ and each $\varepsilon \in D_v$, the pair (v, ε) is called a *literal*. Usually, we use x, y, z (also with subscripts) to denote literals. If x is the literal (v, ε) , then we call v the variable of x , denoted by $\text{var}(x)$, while ε the value of x , denoted by $\text{val}(x)$. That is, $\text{var}(v, \varepsilon) = v$, $\text{val}(v, \varepsilon) = \varepsilon$. Two literals x, y are called *clashing literals* if $\text{var}(x) = \text{var}(y)$ but $\text{val}(x) \neq \text{val}(y)$.

A *clause* C is a finite set of literals. Let \mathcal{CL} be the set of all clauses. We write \perp to represent the empty clause. For a clause C let $\text{var}(C) := \{\text{var}(x) : x \in C\}$. A *generalized clause-set* is a finite set $F \subseteq \mathcal{CL}$. Let $\text{var}(F) := \bigcup\{\text{var}(C) : C \in F\}$. For a variable $v \in \mathcal{VA}$ we write $\text{val}_v(F) := \{\varepsilon \in D_v \mid \exists C \in F : (v, \varepsilon) \in C\}$. The empty generalized clause-set is denoted by \top .

A *partial assignment* is a map φ such that $\text{dom}(\varphi) \subseteq \mathcal{VA}$ and $\varphi(v) \in D_v$ for all $v \in \text{dom}(\varphi)$. Let $n(\varphi) := |\text{dom}(\varphi)|$ be the number of elements in the domain of φ . \emptyset is the empty partial assignment. For convenience the partial assignment φ with $\text{dom}(\varphi) = \{v_1, \dots, v_m\}$ and $\varphi(v_i) = \varepsilon_i$ is written as $\langle v_1 \rightarrow \varepsilon_1, \dots, v_m \rightarrow \varepsilon_m \rangle$. We shorten partial assignment to *assignment* in this paper.

For a literal (v, ε) and a assignment φ with $v \in \text{dom}(\varphi)$, we say (v, ε) is true (resp. false) under φ if $\varphi(v) \neq \varepsilon$ (resp. $\varphi(v) = \varepsilon$). For a clause C and a assignment φ , if there is a literal x such that $\text{var}(x) \in \text{var}(C) \cap \text{dom}(\varphi)$ and x is true under φ (i.e., $\varphi(\text{var}(x)) \neq \text{val}(x)$), we say φ satisfies C . If for all $x \in C$, either $\text{var}(x) \notin \text{dom}(\varphi)$ or $\varphi(\text{var}(x)) = \text{val}(x)$, then φ do not satisfy C . In [14, 15, 16, 17], it is required that clauses should not contain clashing literals. The reason is that if C contains clashing literals x, y , then C is satisfied by any assignment φ such that $\text{var}(x) \in \text{dom}(\varphi)$. That is, a clause with clashing literals can be considered as a ‘‘tautology’’. This paper also requires that clauses should not contain clashing literals.

For a assignment φ we write the clause $\{(v, \varphi(v)) : v \in \text{dom}(\varphi)\}$ as C_φ , that is, C_φ denote the literals of C which are false under φ . For a clause C we can define an associated

assignment φ_C by $\text{dom}(\varphi_C) := \text{var}(C)$, $\varphi_C(v) := \varepsilon$ if $(v, \varepsilon) \in C$. For a generalized clause-set F and a assignment φ the result of application of φ on F is defined as

$$\varphi * F = \{C' : \exists C \in F, C \text{ is not satisfied by } \varphi \text{ and } C' = C \setminus C_\varphi\},$$

Obviously $\varphi * F$ is obtained from F by first deleting all clauses which are satisfied by φ , then deleting from the remaining clauses all literals which are false under φ . We say F is satisfied by φ if $\varphi * F$ is empty. A generalized clause-set is said to be satisfiable if it can be satisfied by some assignments.

For two assignments φ and ϕ , we define $\phi \circ \varphi$ as follows. $\text{dom}(\phi \circ \varphi) := \text{dom}(\phi) \cup \text{dom}(\varphi)$. For a variable $v \in \text{dom}(\phi \circ \varphi)$, if $v \in \text{dom}(\varphi)$, then $\phi \circ \varphi(v) = \varphi(v)$, else $\phi \circ \varphi(v) = \phi(v)$. So for a generalized clause-set F , $(\phi \circ \varphi) * F$ is the generalized clause-set obtained by first applying φ on F then applying ϕ on the resulting clause-set. Obviously if $\text{var}(\varphi) \cap \text{var}(\phi) = \emptyset$, then $(\varphi \circ \phi) * F = (\phi \circ \varphi) * F$.

3. Quantified Generalized Clause-sets

In this section, quantified generalized clause-sets will be introduced, and some properties of them which will be used in later sections will be shown. Roughly speaking, quantified generalized clause-sets are formulae obtained by adding a quantifier prefix to generalized clause-sets.

Definition 1. *The class of quantified generalized clause-sets (QCLS) is the least set satisfying the following conditions:*

1. *Every generalized clause-set is a quantified generalized clause-set.*
2. *For a variable $v \in \mathcal{VA}$ and a quantified generalized clause-set Φ , both $\forall v\Phi$ and $\exists v\Phi$ are quantified generalized clause-sets.*

It is easy to see that quantified generalized clause-sets have structures like $Q_1v_1 \dots Q_nv_n\alpha$, where $Q_i \in \{\exists, \forall\}$ and α is a generalized clause-set. The part $Q_1v_1 \dots Q_nv_n$ is called the prefix, α is the matrix. We use upper case Greek letters such as Φ , Ψ (or with subscripts) to denote quantified generalized clause-sets. Sometimes we use an abbreviation and write $\Phi = Q\alpha$. The variables following \exists (resp. \forall) directly are called existential (resp. universal) variables, literals on existential (resp. universal) variables are called existential (resp. universal) literals. For a quantified generalized clause-set $\Phi = Q_1v_1 \dots Q_nv_n\alpha$, we use $\text{var}(\Phi)$ to represent the set of variables occurring in Φ , that is, $\text{var}(\Phi) = \{v_1, \dots, v_n\} \cup \text{var}(\alpha)$.

For a quantified generalized clause-set $\Phi = Q_1v_1 \dots Q_nv_n\alpha$, if $\text{var}(\alpha) \subseteq \{v_1, \dots, v_n\}$ we say that Φ is closed. If Φ is not closed, there must be some $v \in \text{var}(\alpha)$ such that $v \notin \{v_1, \dots, v_n\}$, and these variables are called free variables. The set of all free variables in Φ is denoted by $\text{free}(\Phi)$, whereas variables occurring in the prefix are called bounded variables. For our later purpose, we define an (partial) order “ $<$ ” on variables of Φ . For two bounded variables v_i, v_j , we say v_i is smaller than v_j if $i < j$, that is, v_i is on the left of v_j in the prefix; free variables are always smaller than bounded variables (by convention). This order can be extended to literals as follows: a literal x is smaller than a literal y if $\text{var}(x)$ is smaller than $\text{var}(y)$.

For a quantified generalized clause-set $\Phi = Q\alpha$ and an assignment φ , we define the value of Φ as follows.

Definition 2. Consider $\Phi \in \text{QCLS}$ and an assignment φ such that $\text{free}(\Phi) \subseteq \text{dom}(\varphi)$. The value of Φ under φ , denoted by $V_\varphi(\Phi)$, is defined inductively as follows:

1. If the prefix of Φ is empty, i.e., Φ is a generalized clause-set α , then

$$V_\varphi(\Phi) = \begin{cases} 1, & \text{if } \varphi * \alpha = \top \\ 0, & \text{if } \perp \in \varphi * \alpha \end{cases}$$

2. If $\Phi = \exists v\Psi$, then $V_\varphi(\Phi) = \max\{V_{\varphi \circ (v \rightarrow \varepsilon)}(\Psi) : \varepsilon \in D_v\}$.
3. If $\Phi = \forall v\Psi$, then $V_\varphi(\Phi) = \min\{V_{\varphi \circ (v \rightarrow \varepsilon)}(\Psi) : \varepsilon \in D_v\}$.

Where, in item 2. and 3. \max (resp. \min) is the operation which selects the maximal (resp. minimal) element of a set.

We say Φ is satisfiable if there is some assignment φ such that $V_\varphi(\Phi) = 1$.

The next example shows that quantified generalized clause-sets can be used to encode the two-person (\forall and \exists) coloring game problem, and player \exists has a winning strategy if and only if the quantified generalized clause-set coded for the game problem is satisfiable.

Example 3. Given a hypergraph $G = (V, H)$ and a set Col of colors, where we equip the set of vertices $V = \{v_1, \dots, v_{2k}\}$ with a total order $<$ such that $v_i < v_j$ when $i < j$. Two players, called \forall and \exists , color vertices of G with colors in Col by the order of V in turn, and player \forall plays first. Player \exists wins the game if and only if G is Col -colored, that is, for each hyperedge $h_i \in H = \{h_1, \dots, h_n\}$, there must be at least two vertices in h_i with different colors. We use V as the set of variables and Col as the domain of each variable. Then this problem can be encoded naturally as a quantified generalized clause-set $Q\alpha$. The prefix Q is $\forall v_1 \exists v_2, \dots, \forall v_{2k-1} \exists v_{2k}$, and $\alpha = \alpha_1 \cup \dots \cup \alpha_n$, where α_i is the clause set representing that the hyperedge h_i should be “monochromatic”, that is, α_i is the set of all clauses $C = \{(v, c) \mid v \in h_i\}$ for each $c \in \text{Col}$.

Consider a hypergraph (V, H) with $V = \{v_1, v_2, v_3, v_4\}$, $H = \{\{v_1, v_2\}, \{v_1, v_3, v_4\}\}$, and $\text{Col} = \{0, 1, 2\}$. Then the encoded formula $Q\alpha$ is $\forall v_1 \exists v_2 \forall v_3 \exists v_4 (\alpha_1 \cup \alpha_2)$, where

$$\alpha_1 = \{\{(v_1, 0), (v_2, 0)\}, \{(v_1, 1), (v_2, 1)\}, \{(v_1, 2), (v_2, 2)\}\}, \text{ and}$$

$$\alpha_2 = \{\{(v_1, 0), (v_3, 0), (v_4, 0)\}, \{(v_1, 1), (v_3, 1), (v_4, 1)\}, \{(v_1, 2), (v_3, 2), (v_4, 2)\}\}.$$

It is not hard to see that $Q\alpha$ is satisfiable. Hence player \exists has a winning strategy. More precisely, whatever color is assigned to v_1 by player \forall , player \exists can assigned a different color to v_2 . After player \forall assigns a color to v_3 , player \exists colors v_4 differently.

In Example 3, if G is restricted to be a graph (i.e., each edge has two vertices), and Col has k colors, the game is termed as GRAPH k -COLORING GAME problem in [4], which has been proved to be PSPACE-complete for $k \geq 3$ (see [4]). It is easy to see that the encoding of GRAPH k -COLORING GAME problem is a quantified generalized clause-set in which each clause consists of two literals.

From Definition 2 we can see that the values of bounded variables under an assignment have no effect on the value of the quantified generalized clause-set. Especially, if Φ is a

closed quantified generalized clause-set, then Φ has the same value under all assignments, either 1 (true) or 0 (false). Thus, for closed quantified generalized clause-sets we call them true or false instead of satisfiable or unsatisfiable.

For a quantified generalized clause-set $\Phi = Q_1v_1 \dots Q_nv_n\alpha$ and an assignment φ such that $\text{free}(\Phi) \subseteq \text{dom}(\varphi)$ and $\{v_1, \dots, v_n\} \cap \text{dom}(\varphi) = \emptyset$, it is easy to see that

$$V_\varphi(\Phi) = 1 \text{ if and only if } Q_1v_1 \dots Q_nv_n(\varphi * \alpha) \text{ is true.}$$

For a variable $v \in \text{var}(\Phi)$, if D_v contains exactly one value ε , then any assignment with v in its domain must assign ε to v , and hence any occurrence of (v, ε) can be removed without changing the satisfiability. So, in this paper, we demand $|D_v| \geq 2$ for any variable v .

Suppose Φ is a quantified generalized clause-set with free variables among $\{v_1, \dots, v_k\}$. From Definition 2 it is easy to see that Φ is satisfiable if and only if $\exists v_1 \dots \exists v_k \Phi$ is true.

Since each variable can only take a fixed number of values, it is easy to see that the problem of determining the satisfiability of quantified generalized clause-sets is in PSPACE. On the other hand, quantified boolean formulae are also quantified generalized clause-sets, so we get the following proposition.

Proposition 4. *The satisfiability problem for quantified generalized clause-sets is PSPACE-complete.*

It is well known that the satisfiability problem for boolean 2-CNF formulae (resp. quantified boolean 2-CNF formulae) can be solved in polynomial time (e.g., [1, 6, 11]). However these results can not be extended to generalized clause-set and quantified generalized clause-set. We denote by 2-CLS (resp. Q2-CLS) the subclass of generalized clause-sets (resp. quantified generalized clause-sets) in which each clause has at most two literals. The satisfiability problem becomes NP-complete for 2-CLS (in fact, 3-coloring problem can be reduced to the satisfiability problem of 2-CLS). Next proposition states that the satisfiability problem for Q2-CLS is PSPACE-complete.

Proposition 5. *The satisfiability problem for Q2-CLS is PSPACE-complete.*

Proof. We have mentioned in Example 3 that GRAPH k -COLOURING GAME problem, which has been proved to be PSPACE-complete in [4] when $k \geq 3$, can be encoded into a quantified 2-clauses-set, in which each clause consists of two literals. The assertion follows. \square

Definition 6. *Two quantified generalized clause-sets Φ and Ψ are equivalent, in symbols $\Phi \approx \Psi$, if and only if for every assignment φ such that $\text{free}(\Phi) \cup \text{free}(\Psi) \subseteq \text{dom}(\varphi)$ we have $V_\varphi(\Phi) = V_\varphi(\Psi)$.*

The next proposition shows that equivalent transformations of the matrix do not affect the satisfiability of the whole quantified generalized clause-set.

Proposition 7. *Let Q be a prefix, Φ and Ψ quantified generalized clause-sets. Then $\Phi \approx \Psi \implies Q\Phi \approx Q\Psi$*

Proof. We proceed by induction on the length of the prefix. If the prefix Q is empty then the proposition holds trivially. Assume that for any prefix whose length is smaller than Q , the proposition holds. Next we will prove that the proposition holds for Q , there are two cases.

Suppose $Q = \exists vQ'$. For every assignment φ such that $\text{free}(\Phi) \cup \text{free}(\Psi) \subseteq \text{dom}(\varphi)$ we have:

$$V_\varphi(Q\Phi) = V_\varphi(\exists vQ'\Phi) = \max\{V_{\varphi \circ (v \rightarrow \varepsilon)}(Q'\Phi) : \varepsilon \in D_v\}.$$

$$V_\varphi(Q\Psi) = V_\varphi(\exists vQ'\Psi) = \max\{V_{\varphi \circ (v \rightarrow \varepsilon)}(Q'\Psi) : \varepsilon \in D_v\}.$$

By the induction hypothesis we have $V_{\varphi \circ (v \rightarrow \varepsilon)}(Q'\Phi) = V_{\varphi \circ (v \rightarrow \varepsilon)}(Q'\Psi)$ for every $\varepsilon \in D_v$. The theorem follows.

For the case $Q = \forall vQ'$ we can show $Q\Phi \approx Q\Psi$ in the same way. \square

Given a quantified generalized clause-set $\Phi = Q\alpha$, let $C = \{x_1, \dots, x_k\}$ be a clause in α such that $x_1 < \dots < x_k$ (for the order between literals please see the paragraph following Definition 1). Please note that if x_k is a universal literal then deleting x_k from C will not change the value of Φ . So the following proposition is obvious.

Proposition 8. *Consider $\Phi = Q\alpha \in \text{QCLS}$, and C_1, \dots, C_n are some non-tautological clauses in α . For each $i = 1, \dots, n$, we remove all universal literals in C_i which are not smaller than any existential literal in C_i , and the result clauses are denoted by C'_i . We have $\Phi \approx Q((\alpha \setminus \{C_1, \dots, C_n\}) \cup \{C'_1, \dots, C'_n\})$.*

From the above proposition, one might think that a clause should not contain largest universal literals because they can be removed without changing the satisfiability, but through the application of partial assignment such largest universal literals can be created, thus one had to alter the definition of application of partial assignments, which perhaps is too cumbersome. Thus, we do not forbid occurrences of largest universal literals, but we will removed them during Q-resolution (see Definition 10 in next section) .

4. Resolution for Quantified Generalized Clause-sets

Resolution is a powerful tool for investigating the satisfiability problem of boolean CNF formulae, the idea is based on uniting two clauses $(X \vee z)$ and $(Y \vee \neg z)$ into one clause $(X \vee Y)$, where X and Y are disjunctions of boolean literals. In this section we introduce the resolution for quantified generalized clause-sets and prove the soundness and refutational completeness.

For a clause C and a variable v , we write $\{v\} * C$ to denote the clause obtained from C by deleting literals on v .

Definition 9. ([13, 14]) *For a variable v and a group of non-tautological clauses $C_1, \dots, C_{|D_v|}$, if $\text{val}_v(\{C_1, \dots, C_{|D_v|}\}) = D_v$ and the set $R = \bigcup_{i=1}^{|D_v|} \{v\} * C_i$ is a non-tautological clause, we say this group of clauses can be resolved on v , and $C_1, \dots, C_{|D_v|}$ are called parent clauses, while R is called the the resolvent.*

From [13] we learn that the resolution defined above is sound and refutationally complete. More precisely, for a generalized clause-set F , we have

- F is equivalent to any clause-set obtained by adding a resolvent.

- F is unsatisfiable if and only if the empty set can be derived by a sequence of resolution steps.

Now we extend the above definition to quantified generalized clause-sets. For simplicity, we only consider closed quantified generalized clause-sets. This is not a restriction, because we are concerned with satisfiability, and $\exists v_1 \cdots \exists v_k \Phi$ has the same satisfiability with Φ , and it is closed if $\text{free}(\Phi) = \{v_1, \dots, v_k\}$.

Definition 10. For $\Phi = Q\alpha \in \text{QCLS}$, if there exists an existential variable v and a group of non-tautological clauses $C_1, \dots, C_{|D_v|} \in \alpha$ such that $\text{val}_v(\{C_1, \dots, C_{|D_v|}\}) = D_v$, then we define a set R of literals as follows.

1. From each C_i ($i \in \{1, \dots, |D_v|\}$) delete all universal literals which are bigger than all existential literal in C_i . These new clauses are denoted by $C'_1, \dots, C'_{|D_v|}$.
2. Construct a new set $R := \bigcup_{i=1}^{|D_v|} \{v\} * C'_i$.

If the new set R is a non-tautological clause, the operation above is called a Q -resolution step, R is called the resolvent, and $C_1, \dots, C_{|D_v|}$ are called the parent clauses. In symbols, $\Phi \stackrel{1}{\underset{Q\text{-res}}{\vdash}} R$.

Example 11. Let $\Phi = \forall v_1 \exists v_2 \forall v_3 \exists v_4 \{C_1, C_2, C_3, C_4, C_5\}$ be a quantified generalized clause-set, where $C_1 = \{(v_1, 0), (v_2, 0), (v_4, 0)\}$, $C_2 = \{(v_3, 1), (v_4, 1)\}$, $C_3 = \{(v_4, 2)\}$, $C_4 = \{(v_2, 1)\}$, $C_5 = \{(v_2, 2)\}$, and $D_{v_1} = D_{v_2} = D_{v_3} = D_{v_4} = \{0, 1, 2\}$. First we can resolve clauses C_1, C_2, C_3 on v_4 and obtain the resolvent $C_6 := \{(v_1, 0), (v_2, 0), (v_3, 1)\}$, then resolve C_4, C_5, C_6 on v_2 and get the resolvent $C_7 := \{(v_1, 0)\}$ which consists of only one universal literal. The resolution steps are illustrated by Fig. 1 below.

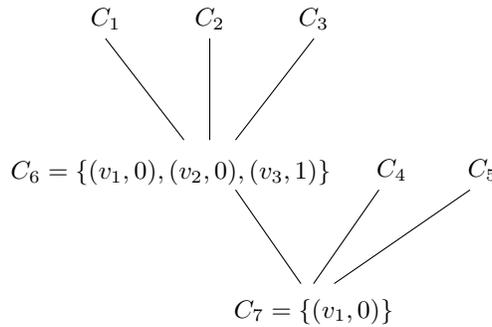


Fig. 1

Next we show that adding the resolvent to a quantified generalized clause-set does not affect the truth value.

Theorem 12. For $\Phi = Q\alpha \in \text{QCLS}$ and a clause R , if $\Phi \stackrel{1}{\underset{Q\text{-res}}{\vdash}} R$, then $Q\alpha \approx Q(\alpha \cup \{R\})$.

Proof. Let $C_1, \dots, C_{|D_v|}$ be parent clauses, and $R = \bigcup_{i=1}^{|D_v|} \{v\} * C'_i$, where $C'_1, \dots, C'_{|D_v|}$ are clauses obtained as in Definition 10. Then by Proposition 8 we have $\Phi \approx Q\alpha'$, where $\alpha' = (\alpha \setminus \{C_1, \dots, C_{|D_v|}\}) \cup \{C'_1, \dots, C'_{|D_v|}\}$.

Now we can apply a resolution step to derive R from $C'_1, \dots, C'_{|D_v|}$ which is on the same variable as the Q-resolution step, then $\alpha' \stackrel{1}{\underset{Res}{\vdash}} R$ (where $\stackrel{1}{\underset{Res}{\vdash}}$ represents the derivability by one resolution step for generalized clause-sets).

As mentioned above the resolution for generalized clause-sets is sound, so $\alpha' \approx \alpha' \cup \{R\}$. By Proposition 7, we have $Q\alpha' \approx Q(\alpha' \cup \{R\})$. Again by Proposition 8 we have $Q(\alpha \cup \{R\}) \approx Q(\alpha' \cup \{R\})$, so $\Phi \approx Q(\alpha \cup \{R\})$. \square

Definition 13. Consider $\Phi = Q\alpha \in \text{QCLS}$ and a sequence of clauses C_1, \dots, C_n . If for each $i = 1, \dots, n$, C_i is either a clause in α or the resolvent of some clauses before it in the sequence, then we say C_1, \dots, C_n is a Q-resolution sequence from Φ . We say C_i is derivable from Φ by Q-resolution for $i = 1, \dots, n$, in symbols, $\Phi \stackrel{1}{\underset{Q-res}{\vdash}} C_i$.

From Definition 13 it is easy to see that if C is a clause of Φ , then C is derivable from Φ by Q-resolution. A clause without existential literals is called a universal clause. In particular, the empty clause \perp is a universal clause. Obviously, a quantified generalized clause-set with a non-tautological universal clause is always false.

Next we explain concepts of refutational completeness and soundness. A resolution approach is said to be refutational complete for a subclass of quantified generalized clause-sets if some universal clause can be derived by a sequence of resolution steps from any unsatisfiable quantified clause-set in the class. Whereas, the soundness means that: for any Φ in the class, if some universal clause can be derived by the resolution from Φ , then Φ is unsatisfiable.

Remark 14. Consider $\Phi = Q'vQ\alpha$, $\varepsilon \in D_v$, $Q' \in \{\exists, \forall\}$ and C_1, \dots, C_n is a Q-resolution sequence from the quantified generalized clause-set $Q((v \rightarrow \varepsilon) * \alpha)$, denoted by Φ_ε . Recall that Φ_ε is obtained from $Q\alpha$ by (1) deleting all clauses containing a literal (v, ε') with $\varepsilon' \neq \varepsilon$, and (2) deleting all occurrences of the literals (v, ε) from the remaining clauses. Then we can obtain a Q-resolution sequence C'_1, \dots, C'_n from Φ by inductively recovering the occurrences of (v, ε) as follows. For each C_i , $i \leq n$,

- if C_i is a clause in Φ then $C'_i = C_i$,
- if $C_i \in \Phi_\varepsilon$, and $\{(v, \varepsilon)\} \cup C_i \in \Phi$, then $C'_i = \{(v, \varepsilon)\} \cup C_i$,
- if C_i is obtained from C_{i_1}, \dots, C_{i_k} ($i_1, \dots, i_k < i$) by a Q-resolution step, then C'_i is the resolvent of $C'_{i_1}, \dots, C'_{i_k}$

The above “lifting” process is a quit common approach in proofs about resolution, e.g. in first order logic.

We conclude this section by proving the refutational completeness and soundness of Q-resolution for quantified generalized clause-sets. For simplicity we suppose that tautological clauses do not occur in the (original) quantified generalized clause-set.

Theorem 15. *For any quantified generalized clause-set Φ ,
 Φ is false if and only if $\Phi \vdash_{Q\text{-res}} R$ for some universal clause R .*

Proof. The direction from right to left follows from Theorem 12. Now we prove the inverse direction by induction on the length of the prefix.

If $\Phi = \perp$, then the theorem holds trivially. Suppose $\Phi = \exists v\alpha$ where α is a generalized clause-set (please recall that Φ is closed). Since Φ is false, α is unsatisfiable. So, α must contain clauses $\{(v, \varepsilon_1)\}, \dots, \{(v, \varepsilon_n)\}$, where $D_v = \{\varepsilon_1, \dots, \varepsilon_n\}$. It is easy to see that $\Phi \vdash_{Q\text{-res}} \perp$.

Suppose $\Phi = \forall v\alpha$ where α is a generalized clause-set. Then all clauses in α are universal clauses. Since Φ is false, α is non-empty. So, there is a clause $C \in \alpha$, and we have $\Phi \vdash_{Q\text{-res}} C$.

Suppose $\Phi = \exists vQ\alpha$, and assume that the theorem holds for quantified clause-sets with prefix Q . Because Φ is false, for each $\varepsilon_i \in D_v = \{\varepsilon_1, \dots, \varepsilon_n\}$, the quantified generalized clause-set $Q((v \rightarrow \varepsilon_i) * \alpha)$, denoted by Φ_{ε_i} , is false. By the induction hypothesis we have $\Phi_{\varepsilon_i} \vdash_{Q\text{-res}} R_{\varepsilon_i}$, R_{ε_i} is a universal clause. That is, there is a Q-resolution sequence Π_{ε_i} of R_{ε_i} from Φ_{ε_i} . Then by Remark 14 we can obtain a Q-resolution sequence from Φ by inductively recovering the occurrences of the literal (v, ε_i) to clauses in Π_{ε_i} . Thus, we have either $\Phi \vdash_{Q\text{-res}} R_{\varepsilon_i}$ or $\Phi \vdash_{Q\text{-res}} \{(v, \varepsilon_i)\} \cup R_{\varepsilon_i}$. If $\Phi \vdash_{Q\text{-res}} R_{\varepsilon_i}$ for some $\varepsilon_i \in D_v$, then the theorem follows. So, we assume that $\Phi \vdash_{Q\text{-res}} \{(v, \varepsilon_i)\} \cup R_{\varepsilon_i}$ for each $\varepsilon_i \in D_v$. Then we can obtain the empty clause \perp by resolving clauses $\{(v, \varepsilon_1)\} \cup R_{\varepsilon_1}, \dots, \{(v, \varepsilon_n)\} \cup R_{\varepsilon_n}$. Therefore, $\Phi \vdash_{Q\text{-res}} \perp$. The theorem follows.

Suppose $\Phi = \forall vQ\alpha$, and assume that the theorem holds for quantified clause-sets with prefix Q . Because Φ is false, there must be a $\varepsilon_i \in D_v$ such that the quantified generalized clause-set Φ_{ε_i} is false, where Φ_{ε_i} is obtained by the same way as in the above paragraph. By the induction hypothesis we have $\Phi_{\varepsilon_i} \vdash_{Q\text{-res}} R_{\varepsilon_i}$ for some universal clause R_{ε_i} . By the same argument as in the above paragraph, we have either $\Phi \vdash_{Q\text{-res}} R_{\varepsilon_i}$ or $\Phi \vdash_{Q\text{-res}} \{(v, \varepsilon_i)\} \cup R_{\varepsilon_i}$. Both resolvents are universal clauses, and we complete the proof. \square

5. Quantified Generalized Horn Clause-sets

The class of boolean Horn formulae (in which each clause contains at most one positive literal) is a well-known tractable class in propositional logic. Please note that a boolean variable v is a variable with domain $D_v = \{0, 1\}$. The literal v can be considered as $(v, 0)$ while the literal $\neg v$ can be considered as $(v, 1)$. For a boolean variable v , we always take $\neg v$ (i.e. $(v, 1)$) as a negative literal while v (i.e. $(v, 0)$) as a positive one (In fact, if we would take $(v, 0)$ as negative then we get the notion of dual Horn formulae). For each non-boolean variable v we can specify exactly one value $\varepsilon \in D_v$ and consider (v, ε) as negative and others as positive. Then we can define generalized Horn formulae as those in which each clause contains at most one positive literal.

Before the definition of generalized Horn clause-sets, we explain why we do not adopt the alternative definition of h -Horn clause-sets by selecting $(v, h(v))$ as positive and all others as negative. Considering a boolean Horn clause-sets F without the empty clause, the two basic facts are:

1. If F has no unit-clause, then the assignment ϕ which sets all variables to 0 satisfies F .
2. If ϕ does not satisfy F , then there exists a positive unit-clause v in F , and after deleting clauses containing v , removing all $\neg v$, the formula is still staying in the class of (boolean) Horn clause-sets.

Suppose we declare (v, e) as positive if $e = h(v)$ and negative otherwise, then the partial assignment h plays the same role as ϕ above, i.e., they both set positive literals to be false. If F is a Horn clause-sets with no unit-clause then h satisfies F . Unfortunately, property 2 above fails. In fact we are able to show that, using this more liberal definition of “generalized Horn clause-sets”, one actually obtains an NP-complete class. One can reduce the satisfiability problem of boolean CNF formulae to the satisfiability problem of generalized Horn clause-sets as follows: For each boolean CNF formula F , first we expand the domain of boolean variables $\{0, 1\}$ to $\{0, 1, 2\}$, and select $(v, 2)$ as positive for each variable v ; then for each $v \in \text{var}(F)$ we change v to $(v, 0)$, $\neg v$ to $(v, 1)$, and add positive unit clauses $\{(v, 2)\}$. Obviously the resulting formula F' is clearly “Horn” and F' is satisfiable if and only if F is satisfiable.

Definition 16. Let $h : \mathcal{VA} \rightarrow \bigcup_{v \in \mathcal{VA}} D_v$ be a map such that $h(v) \in D_v$ for every $v \in \mathcal{VA}$. Then $(v, h(v))$ is called an h -negative literal, whereas every literal (v, ε) with $\varepsilon \neq h(v)$ is called an h -positive literal.

A clause is called an h -Horn clause if it contains at most one h -positive literal or it is \perp .

A generalized clause-set where all clauses are h -Horn clauses is called a generalized h -Horn clause-set.

Definition 17. A generalized clause-set is called a generalized Horn clause-set if it is a generalized h -Horn clause-set for some map $h : \mathcal{VA} \rightarrow \bigcup_{v \in \mathcal{VA}} D_v$

Lemma 18. The class of generalized Horn clause-sets is decidable in polynomial time.

Proof. Our approach to recognize generalized Horn clause-sets is similar to the algorithm deciding whether a boolean CNF formula is renamable Horn. That is, we shall construct a polynomial-time transformation T from generalized clause-sets F to boolean 2-CNF formulae such that F is generalized Horn if and only if $T(F)$ is satisfiable.

For each variable $v \in \text{var}(F)$ and each $\varepsilon \in D_v$, we introduce a boolean variable $\text{pos}(v, \varepsilon)$ which means that (v, ε) is positive.

Consider an arbitrary clause $C = \{(v_1, \varepsilon_1), \dots, (v_k, \varepsilon_k)\}$ in F . Since any Horn clause contains at most one positive literal, we need the following boolean formula

$$S(C) = \{\neg \text{pos}(v_i, \varepsilon_i) \vee \neg \text{pos}(v_j, \varepsilon_j) \mid 1 \leq i \neq j \leq k\}.$$

Please note that for each variable v , we can take only one literal with variable v as negative. This can be expressed as

$$N(v) = \{\text{pos}(v, \varepsilon_1) \vee \text{pos}(v, \varepsilon_2) \mid \varepsilon_1 \neq \varepsilon_2 \in D_v\}, \text{ and} \\ N'(v) := \{\bigvee_{\varepsilon \in D_v} \neg \text{pos}(v, \varepsilon)\}.$$

Please note that $N'(v)$ consists of only one clause which may contain more than two literals. Fortunately, $N'(v)$ can be omitted for our purpose, because when $N'(v)$ can't be satisfied we can select arbitrary value as negative. Define

$$T(F) := \left(\bigwedge_{C \in F} S(C) \right) \wedge \left(\bigwedge_{v \in \text{var}(F)} N(v) \right).$$

Our next task is to prove that F is a generalized Horn clause-set if and only if $T(F)$ is satisfiable.

For the direction from left to right we suppose F is generalized Horn. Then for some mapping $h : VA \rightarrow \bigcup_{v \in VA} D_v$, F is h -Horn. Now we define a truth assignment φ on variables like $\text{pos}(v, \varepsilon)$ as follows.

$\varphi(\text{pos}(v, \varepsilon)) = 1$ if and only if (v, ε) is h -positive, i.e., $h(v) \neq \varepsilon$.

Since each clause C contains at most one h -positive literal, it follows that $S(C)$ is satisfied by φ . Similarly, $N(v)$ is also satisfied by φ because for each v only the literal $(v, h(v))$ is h -negative, that is, $\varphi(\text{pos}(v, h(v))) = 0$.

For the inverse direction we suppose $T(F)$ is satisfiable. Let φ be a satisfying truth assignment on variables like $\text{pos}(v, \varepsilon)$. For each variable $v \in \text{var}(F)$, due to the formula $N(v)$, there is at most one $\varepsilon \in D_v$ such that $\varphi(\text{pos}(v, \varepsilon)) = 0$. Now we can define a mapping $h : VA \rightarrow \bigcup_{v \in VA} D_v$ as follows. For each v , if $\varphi(\text{pos}(v, \varepsilon)) = 0$ for some $\varepsilon \in D_v$, then let $h(v) := \varepsilon$; otherwise pick any one $\varepsilon \in D_v$ and let $h(v) = \varepsilon$.

It is easy to see for each variable $v \in \text{var}(F)$ and $\varepsilon \in D_v$ that (v, ε) is h -positive implies $\varphi(\text{pos}(v, \varepsilon)) = 1$. Thus each clause C of F contains at most one h -positive literal since φ satisfies $S(C)$.

Obviously, the transformation S costs polynomial time. The theorem follows because the satisfiability problem for boolean 2-CNF formulae is solvable in linear time. \square

From the proof of Lemma 18, we can see that if a generalized clause-set F is generalized Horn, then a map h witnessing that F is h -Horn can be computed in polynomial time. So, we will fix a map $h : \mathcal{VA} \rightarrow \bigcup_{v \in \mathcal{VA}} D_v$, and only consider generalized h -Horn clause-sets.

Let us recall a standard translation in [14, 15, 16, 17] from generalized clause-set to boolean CNF formulae.

- For every literal (v, ε) we consider a boolean variable $\tau((v, \varepsilon))$ expressing that v shall not get value ε .
- Clauses C are translated to positive boolean clauses $\tau(C)$ by replacing each literals $x \in C$ with the positive boolean literals $\tau(x)$.
- Adding ALO clauses to require that each variable gets at least one value. For a variable v with $D_v = \{\varepsilon_1, \dots, \varepsilon_n\}$ the ALO_v clause is $\{\neg\tau((v, \varepsilon_1)) \vee \dots \vee \neg\tau((v, \varepsilon_n))\}$.

That is, a generalized clause-set F is translated to $\Theta(F) = \{\tau(C) \mid C \in F\} \cup \{\text{ALO}_v \mid v \in \text{var}(F)\}$ which has the same satisfiability as F , where “the same satisfiability” means F is satisfiable if and only if $\Theta(F)$ is satisfiable. Suppose F is h -Horn, for every negative literal x , we rename $\tau(x)$ to $\neg\tau(x)$. Clearly, for every $C \in F$, $\tau(C)$ becomes a Horn clause

after renaming. Further, since there is exactly one literal on v is h -negative, after renaming only one negative literal in $\text{ALO}_v = \{\neg\tau((v, \varepsilon_1)) \vee \dots \vee \neg\tau((v, \varepsilon_n))\}$ becomes positive. Consequently, $\Theta(F)$ is renameable Horn.

As an example, consider

$$F = \{\{(v, 0), (w, 0)\}, \{(v, 1), (w, 0)\}, \{(v, 0), (w, 2)\}, \{(w, 1)\}, \{(v, 2)\}\}$$

with $D_v = D_w = \{0, 1, 2\}$. For F we choose $(v, 0), (w, 0)$ as negative and $(v, 1), (v, 2), (w, 1), (w, 2)$ as positive. For convenience we use x_i (resp. y_i) to represent $\tau((v, i))$ (resp. $\tau((w, i))$), then $\tau(F) = \alpha \wedge \beta$, where

$$\begin{aligned} \alpha &= (x_0 \vee y_0) \wedge (x_1 \vee y_0) \wedge (x_0 \vee y_2) \wedge y_1 \wedge x_2, \\ \beta &= \text{ALO}_v \wedge \text{ALO}_w = (\neg x_0 \vee \neg x_1 \vee \neg x_2) \wedge (\neg y_0 \vee \neg y_1 \vee \neg y_2). \end{aligned}$$

Clearly, if replace x_0, y_0 by $\neg x_0, \neg y_0$, respectively, then $\Theta(F)$ becomes Horn.

The translation and the renaming cost polynomial time. Since the satisfiability for boolean Horn formulae is decidable in linear time, we obtain the following lemma.

Lemma 19. *The satisfiability problem for generalized h -Horn clause-sets can be solved in polynomial time.*

Now, we can test the satisfiability of an h -Horn clause-set F by at first translating it to $\Theta(F)$, then renaming $\Theta(F)$ to a Horn formula, and then checking the satisfiability of the Horn formula. However, we shall show that the satisfiability of Horn clause-sets can also be decided by directly applying a restricted resolution.

A clause is called an *h -positive unit clause* if it contains only one literal, and the literal is h -positive. Positive unit resolution (P-U-resolution) is the resolution defined in Definition 9 with all but one of the parent clauses are h -positive unit clauses.

Next we will prove that P-U-resolution is sound and refutationally complete for generalized h -Horn clause-sets.

Lemma 20. *For a generalized h -Horn clause-set F , F is unsatisfiable if and only if $F \vdash_{\text{P-U-res}} \perp$, where $\vdash_{\text{P-U-res}}$ represents the derivability by P-U-resolution.*

Proof. The direction from right to left follows obviously, because P-U-resolution is also the resolution defined in Definition 9. Now we prove the inverse direction by induction on the number of the variables in F . For convenience we suppose that there are no tautological clauses in F .

If $F = \perp$, then the lemma holds trivially. Suppose there is only one variable in F . Since F is unsatisfiable, it must contain a variable v with $D_v = \varepsilon_1, \dots, \varepsilon_n$ and clauses $\{(v, \varepsilon_1)\}, \dots, \{(v, \varepsilon_n)\}$. So we have $F \vdash_{\text{P-U-res}} \perp$.

Suppose the number of variables in F is bigger than one, and assume that the theorem holds for generalized h -Horn clause-sets with fewer variables than F . Because F is unsatisfiable, there must be a variable v such that $\text{val}_v(F) = D_v$, and for each $\varepsilon_i \in D_v = \{\varepsilon_1, \dots, \varepsilon_n\}$, the clause-set $\langle v \rightarrow \varepsilon_i \rangle * F$, denoted by F_{ε_i} , is unsatisfiable. Because F_{ε_i} is still a generalized h -Horn clause-set, by the induction hypothesis $F_{\varepsilon_i} \vdash_{\text{P-U-res}} \perp$. That is, there is a P-U-resolution refutation Π_{ε_i} from F_{ε_i} . Then we can obtain a resolution sequence (maybe not a P-U-resolution sequence) from F by recovering the occurrences of the literal (v, ε_i) to clauses in Π_{ε_i} (see Remark 14). Since F is generalized h -Horn, if (v, ε_i) is h -positive,

there is no chance to add (v, ε_i) to a clause with an h -positive literal, hence, the new resolution sequence remains a P-U-resolution sequence. Without loss of generality we suppose $(v, \varepsilon_1), \dots, (v, \varepsilon_{n-1})$ are h -positive and (v, ε_n) is h -negative. Therefore, for each $i < n$, we have either $F \vdash_{P-U-res} \perp$ or $F \vdash_{P-U-res} \{(v, \varepsilon_i)\}$. If the former case is true for some $i < n$, then the lemma holds. So we assume that $F \vdash_{P-U-res} \{(v, \varepsilon_i)\}$ for each $i < n$.

Now we consider Π_{ε_n} which is a P-U-resolution refutation from F_{ε_n} . It is easy to see that for each input clause C (i.e., clause in F_{ε_n}) occurring in Π_{ε_n} , either C itself is a clause in F or $\{(v, \varepsilon_n)\} \cup C$ is in F . If $\{(v, \varepsilon_n)\} \cup C \in F$, we can obtain C by a P-U-resolution step resolving clauses $\{(v, \varepsilon_1)\}, \dots, \{(v, \varepsilon_{n-1})\}$ and $\{(v, \varepsilon_n)\} \cup C$. Then applying resolution steps in Π_{ε_n} , we can obtain \perp . The lemma holds. \square

Next we extend the above results to quantified generalized Horn clause-sets.

Definition 21. *A quantified generalized clause-set $\Phi = Q\alpha$ is called a quantified generalized Horn clause-set if the matrix α is a generalized Horn clause-set. The set of all quantified generalized Horn clause-sets is denoted by QHCLS.*

Due to the quantifiers, Kullmanns's translation can not be extended to one from quantified Horn clause-sets to quantified boolean Horn formulae. In order to show the poly-time solvability of the satisfiability of QHCLS, we introduce quantified positive unit resolution (Q-P-U-resolution), and prove the soundness and refutational completeness for quantified generalized Horn clause-sets. Before the definition we introduce some notions. A clause (of a quantified generalized clause-set) is called a *unit clause* if it contains exactly one existential literal; if furthermore this literal is h -positive then the clause is called an *h -positive unit clause*.

Definition 22. *A Q-resolution step is called a Q-P-U-resolution step if all but one of parent clauses are h -positive unit clauses, in symbols $\vdash_{Q-P-U-res}^1$.*

A sequence C_1, \dots, C_n is called a Q-P-U-resolution sequence from $\Phi = Q\alpha \in QCLS$ if for every C_i in the sequence we have either $C_i \in \alpha$ or C_i is obtained from some clauses before it in the sequence by a Q-P-U-resolution step.

If a clause C occurs in a Q-P-U-resolution sequence from Φ , then we say C is derivable from Φ , in symbols $\Phi \vdash_{Q-P-U-res} C$.

The soundness of Q-P-U-resolution for quantified generalized Horn clause-sets is trivial, because it is a special case of Q-resolution. Next we will prove that Q-P-U-resolution is refutationally complete for quantified generalized Horn clause-sets. The proof is similar to Lemma 20, and we use the same proof structure.

Theorem 23. *For a quantified generalized Horn clause-set Φ , we have:*

$$\Phi \text{ is false} \implies \Phi \vdash_{Q-P-U-res} R \text{ for some universal clause } R$$

Proof. We shall proceed by induction on the length of the prefix. By Proposition 8, we assume w.l.o.g. that in each clause the largest literals is existential.

If $\Phi = \perp$, then the theorem holds trivially. When Φ contains only one quantifier, i.e., $\Phi = \exists v\alpha$ or $\Phi = \forall v\alpha$, the proofs are the same as the corresponding parts in the proof of Theorem 15.

Suppose $\Phi = \exists v Q\alpha$, and assume that the theorem holds for formulae with prefix Q . Because Φ is false, for each $\varepsilon_i \in D_v = \{\varepsilon_1, \dots, \varepsilon_n\}$, the quantified generalized clause-set $Q(\langle v \rightarrow \varepsilon_i \rangle * \alpha)$, denoted by Φ_{ε_i} , is false. Please note that Φ_{ε_i} is still a quantified generalized Horn clause-set, so by the induction hypothesis $\Phi_{\varepsilon_i} \vdash_{Q-P-U-res} R_{\varepsilon_i}$, R_{ε_i} is a universal clause. That is, there is a Q-P-U-resolution sequence Π_{ε_i} of R_{ε_i} from Φ_{ε_i} . Then we can obtain a Q-resolution sequence (maybe not a Q-P-U-resolution sequence) from Φ by recovering the occurrences of the literal (v, ε_i) to clauses in Π_{ε_i} . Since Φ is a quantified generalized Horn clause-set, if (v, ε_i) is h -positive, there is no chance to add (v, ε_i) to a clause with an h -positive literal, hence, the new resolution sequence remains a Q-P-U-resolution sequence. Without loss of generality we suppose $(v, \varepsilon_1), \dots, (v, \varepsilon_{n-1})$ are h -positive and (v, ε_n) is h -negative. For each $i < n$, we have either $\Phi \vdash_{Q-P-U-res} R_{\varepsilon_i}$ or $\Phi \vdash_{Q-P-U-res} \{(v, \varepsilon_i)\} \cup R_{\varepsilon_i}$. If the former case is true for some $i < n$, then the theorem follows. So we assume that $\Phi \vdash_{Q-P-U-res} \{(v, \varepsilon_i)\} \cup R_{\varepsilon_i}$ for each $i < n$.

Now we consider Π_{ε_n} which is a Q-P-U-resolution sequence of a universal clause R_{ε_n} from Φ_{ε_n} . Our next task is to prove $\Phi \vdash_{Q-P-U-res} R_{\varepsilon_n}$. It is easy to see that for each input clause C (i.e., clause in Φ_{ε_n}) occurring in Π_{ε_n} , either C itself is a clause in α or $\{(v, \varepsilon_n)\} \cup C$ is in α . If $\{(v, \varepsilon_n)\} \cup C \in \alpha$, we can obtain C by resolving clauses $\{(v, \varepsilon_1)\} \cup R_{\varepsilon_1}, \dots, \{(v, \varepsilon_{n-1})\} \cup R_{\varepsilon_{n-1}}$ and $\{(v, \varepsilon_n)\} \cup C$ (please note that by the assumption at the beginning of the proof, the biggest literal in C is existential, thus no universal literal in C will be deleted during this resolution step). Then applying resolution steps in Π_{ε_n} , we can obtain R_{ε_n} . The theorem follows.

Suppose $\Phi = \forall v Q\alpha$. The proof is similar to the corresponding part in the proof of Theorem 15, because after recovering a universal literal to a Q-P-U-resolution sequence the result remains a Q-P-U-resolution sequence.

Altogether, we complete the proof of the theorem. \square

Next we prove that the satisfiability problem for quantified generalized Horn clause-sets can be decided in polynomial time. Our proof is based on the above theorem (Theorem 23) and we use the same ideas as the proof in [8].

Theorem 24. *The satisfiability problem for quantified generalized Horn clause-sets can be decided in polynomial time.*

Proof. In a Q-P-U-resolution step if each of parent clauses contains an h -positive existential literal the resolvent must contain an h -positive existential literal. Then, a Q-P-U-resolution refutation for a false quantified generalized Horn clause-set Φ is indeed a process which reduces h -negative existential literals from a clause without h -positive literals until a universal clause is obtained.

For a quantified generalized Horn clause-set $\Phi = Q\alpha$, let P_Φ be the clause-set of all clauses in Φ which contain h -positive existential literals, N_Φ be the clause-set of all remaining clauses, i.e., clauses without h -positive existential literals. As analyzed above, Φ is false if and only if $Q(P_\Phi \cup \{C\})$ is false for some clause C in N_Φ . Thus, it is sufficient to consider the satisfiability of $Q(P_\Phi \cup \{C\})$ for all $C \in N_\Phi$.

For convenience we introduce a notation: for a quantified generalized clause-set $Q'\beta$ we use $\beta_{|\exists}$ to denote the clause-set obtained from β by removing all occurrences of universal literals in β .

Consider an arbitrary clause C in N_Φ . Suppose there is no h -positive universal literal in C . Then any universal literal occurring in $Q(P_\Phi \cup \{C\})$ is h -negative. Because universal literals can not block the resolution, $Q(P_\Phi \cup \{C\})$ is false if and only if $(P_\Phi \cup \{C\})_{|\exists}$ is unsatisfiable. The satisfiability of $(P_\Phi \cup \{C\})_{|\exists}$ can be decided in polynomial time.

So, we assume that C contains an h -positive universal literal, say (v, ε) . Since in $Q(P_\Phi \cup \{C\})$ all universal literals on variables different from v are h -negative, we will disregard them. Define the clauses-set $U_{\Phi, v}$ as follows.

$$U_{\Phi, v} = \{(v', \varepsilon') \mid (v', \varepsilon') \text{ is an } h\text{-positive existential literal, } v < v', \\ \text{and } (P_\Phi)_{|\exists} \xrightarrow{P\text{-}U\text{-}res} \{(v', \varepsilon')\}\}.$$

(Where, $\xrightarrow{P\text{-}U\text{-}res}$ represents the derivability by P-U-resolution for generalized Horn clause-sets).

Because $(P_\Phi)_{|\exists}$ is a generalized Horn clause-set, $U_{\Phi, v}$ can be constructed in polynomial time.

Please note that if a clause in P_Φ contains the h -negative universal literal (v, ε') with $\varepsilon' \neq \varepsilon$ such that (v, ε') is smaller than the h -positive existential literal in the clause, then the clause has no contribution to derive a universal clause. Then we can delete all such clauses without change the truth. Let $Rest$ be the clause-set of the remaining clauses.

If $Q(P_\Phi \cup \{C\})$ is false, to obtain a resolution refutation, we can at first obtain positive unit clauses with their positive existential literals in $U_{\Phi, v}$, then resolve these unit clauses with C until the positive universal literal (v, ε) becomes biggest, then we can delete (v, ε) , and resolve further without worrying about the universal literals. Now it is not hard to see that $Q(P_\Phi \cup \{C\})$ is false if and only if $(U_{\Phi, v} \cup Rest \cup \{C\})_{|\exists}$ is unsatisfiable. The satisfiability of $(U_{\Phi, v} \cup Rest \cup \{C\})_{|\exists}$ can be decided in polynomial time because it is a generalized Horn clause-set. The proof completes. \square

6. Conclusion and Future Work

In this paper, resolutions for quantified generalized clause-sets and quantified generalized Horn clause-sets are introduced and studied. Soundness and refutational completeness have been proved. Further, the satisfiability problem for generalized Horn clause-sets and quantified generalized Horn clause-sets is shown to be decidable in polynomial time. On the one hand, the work of this paper can be considered as a generalization of resolution for generalized clause-sets [14] to quantified generalized clause-sets. on the other hand, it also can be considered as a generalization of Q-resolution [7, 11] for quantified boolean CNF formulae to quantified generalized clause-sets. Unfortunately, the results on boolean 2-CNF formulae and quantified boolean 2-CNF formulae can not be extended accordingly because the satisfiability problem is NP-complete for generalized 2-clause-sets, and PSPACE-complete for quantified generalized 2-clause-sets. The investigation of QHCLS is the first step to search for tractable subclass of quantified generalized clause-sets. Yet QHCLS is a very restricted subclass of quantified generalized clause-sets. Our future work is to identify some other larger tractable subclass of quantified generalized clause-sets.

References

- [1] 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*, **8**:121–123, 1979.
- [2] B. Beckert, R. Hähnle, and F. Manyà. The SAT problem of signed CNF formulas. *Applied Logic Series*, **17**:61–82, 2000.
- [3] A. Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.
- [4] F. Börner, A. Bulatov, H. Chen, P. Jeavons, and A. Krokhin. The complexity of constraint satisfaction games and QCSP. In *Proceedings of computer science logic*, **2803** of *LNCS*, pages 58–70. springer, 2003.
- [5] K. Chandrasekar and M. S. Hsiao. Q-PREZ: QBF evaluation using partition, resolution and elimination with ZBDDs. In *Proceedings of the IEEE VLSI Design Conference*, pages 189–194, January 2005.
- [6] S. Even, A. Itai, and A. Shamir. On the Complexity of Timetable and Multi-Commodity Flow Problems. *SIAM Journal on Computing*, **5**:691–703, 1976.
- [7] A. Flögel. *Resolution für quantifizierte Boole'sche Formeln*. PhD thesis, Universität Paderborn, 1993.
- [8] A. Flögel, M. Karpinski, and H. Kleine Büning. Resolution for Quantified Boolean Formulas. *Information and Computation*, **117**:12–18, 1995.
- [9] M. Karpinski, H. Kleine Büning, and P. H. Schmitt. On the Computational Complexity of Quantified Horn Clauses. In *Proceedings of 1st Workshop Computer Science Logic (1987)*, **329** of *LNCS*, pages 129–137. springer, 1988.
- [10] H. Kleine Büning. On subclasses of minimal unsatisfiable formulas. *Discrete Applied Mathematics*, **107**(1-3):83–98, 2000.
- [11] H. Kleine Büning and T. Lettman. *Propositional Logic: Deduction and Algorithms*. Cambridge University Press, 1999.
- [12] H. Kleine Büning, K. Subramani, and X. Zhao. Boolean Functions as Models for Quantified Boolean Formulas. *Journal of the Automated Reasoning*, **39**:49–75, 2007.
- [13] O. Kullmann. Upper and lower bounds on the complexity of generalised resolution and generalised constraint satisfaction problems. *Annals of Mathematics and Artificial Intelligence*, **40**(3-4):303–352, 2004.
- [14] O. Kullmann. Constraint satisfaction problem in clausal form: Autarkies, minimal unsatisfiability, and applications to hypergraph inequalities. Technical Report Computer Science Report Series, CSR 13, University of Wales Swansea, 2006.

- [15] O. Kullmann. Constraint satisfaction problem in clausal form: Autarkies and minimal unsatisfiability. Technical Report Technical Report TR 07-055, Electronic Colloquium on Computational Complexity (ECCC), June 2007.
- [16] O. Kullmann. Constraint satisfaction problem in clausal form I: Autarkies and deficiency. *Fundamenta Informaticae*, 2010. To appear.
- [17] O. Kullmann. Constraint satisfaction problem in clausal form II: Minimal unsatisfiability and conflict structure. *Fundamenta Informaticae*, 2010. to appear.
- [18] F. Manyà, R. Béjar, and G. Escalada-Imaz. The satisfiability problem in regular CNF-formulas. *Soft Computing: A fusion of Foundations, Methodologies and Applications*, **2**(3):116–123, 1998.
- [19] W. V. Quine. A Way to Simplify Truth Functions. *American Mathematical Monthly*, **66**:627–630, 1955.
- [20] J. A. Robinson. A Machine Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, **12**:23–41, 1965.
- [21] L. Zhang and S. Malik. Validating SAT solvers using an independent resolution-based checker: practical implementations and other applications. In *Proceedings of Design Automation and Test in Europe (DATE)*, pages 880–885, March 2003.