On the Complexity of the CQF Hierarchy

Anja Remshagen
University of West Georgia
Carrollton, GA 30118

1 Introduction

The following hierarchy of quantified Boolean formulas is introduced in Truemper [7].

$$F_1 = \exists Y\, S \tag{1}$$

$$F_i = \begin{cases} \forall Q_i\, (\exists X_i\, R_i \rightarrow F_{i-1}) & \text{if } i > 1, i \text{ even} \\ \exists Q_i\, (\exists X_i\, R_i \wedge F_{i-1}) & \text{if } i > 1, i \text{ odd} \end{cases} \tag{2}$$

where all R_i and S are formulas in conjunctive normal form (CNF), and where in general all CNF formulas occurring in F_i contain the variable set Q_i Each formula F_i is a *Constrained Quantified Formula (CQF).* For a given $i \geq 1$, the problem of evaluating formula F_i is called *CQF*(i). Note that CQF(i) is at the ith complexity level of the polynomial hierarchy. In particular, CQF(i) is Σ_i^{P}-complete if i is odd and Π_i^{P}-complete if i is even. More recent work—evidently not aware of Truemper [7]—defines similar hierarchies. See, for example, Sabharwal, Ansotegui, Gomes, Hart, and Selman [4], and Benedetti, Lallouet, and Vautard [2].

In this paper, we determine the complexity of two subclasses of CQF. A *Horn formula* is a CNF formula where each clause contains at most one nonnegated literal, and a *2CNF formula* is a CNF formula where each clause contains at most two literals. Then *Horn CQF*(i) is the problem to evaluate an instance of CQF(i) where all CNF formulas are Horn formulas. Analogously, *2CNF CQF*(i) is the problem to evaluate an instance of CQF(i) where all CNF formulas are 2CNF formulas. In Section 2, we show that the complexity of Horn CQF(i) is reduced by one level in the polynomial hierarchy. In Section 3, we show that the complexity of 2CNF CQF(i) is reduced by two levels.

2 The Complexity of Horn CQF

Kleine Büning, Karpinski, and Flögel [3] have shown that QBF for Horn formulas is in $\mathcal{P}$. In contrast, the complexity of Horn CQF(i) is only reduced by one level.

Theorem 1. For $k \geq 1$, Horn CQF(k) is $\Sigma^{\mathrm{P}}_{k-1}$-complete if k is odd and Π^{P}_{k-1}-complete if k is even.

In order to prove Theorem 1, we establish the equivalence between a CNF formula T and a related quantified formula T^*. The CNF formula T is given by a variable set V and set of clauses C. The formula T^* has as variable set the union of V and the following sets.

$$Z = \{z_c \,|\, c \in C\} \tag{3}$$

$$G = \{g_v \,|\, v \in V\} \tag{4}$$

$$V^* = \{v^* \,|\, v \in V\} \tag{5}$$

The formula T^* contains two Horn formulas R and S. The clauses of R are of the form $\neg v \vee \neg v^*$, for all $v \in V$. In the definition of formula S we use the notation $v \in c$ (resp. $\neg v \in c$) when v (resp. $\neg v$) occurrs as literal in clause c for some variable v and clause c of T. Then formula S is as follows.

$$\bigwedge_{v\in V} (\neg v \vee g_v) \wedge (\neg v^* \vee g_v) \wedge \tag{6}$$

$$\bigwedge_{c\in C} \left(\bigwedge_{v\in c} (\neg v \vee z_c) \right) \wedge \tag{7}$$

$$\bigwedge_{c\in C} \left(\bigwedge_{\neg v\in c} (\neg v^* \vee z_c) \right) \wedge \tag{8}$$

$$\left(\bigvee_{v\in V} \neg g_v \vee \bigvee_{c\in C} \neg z_c) \right) \tag{9}$$

Then T^* is the formula

$$\exists V^* \, R \wedge \forall G \cup Z \, \neg S \tag{10}$$

Note that the variables V of formula T^* are precisely the variables of the formula that are not quantified. The following lemma holds.

Lemma 1. CNF formula T is logically equivalent to the formula T^*.

Proof. Let α be a truth assignment to V under which T evaluates to *True*. We extend α to an assignment α^* for $V \cup V^*$ by setting v^* to *True* if $\alpha(v)$ is *False* and to *False* otherwise. That means, exactly one variable v or v^* is set to *True* while the other variable is set to *False*. In particular, R evaluates to *True* under α^*. Let β be an assignment to $G \cup Z$. If $\beta(g_v) = \mathit{False}$ for some $v \in V$, then one of the clauses of S defined in (6) is *False* under α^* and β. We assume that all g_v are set to *True* by β but at least one variable z_{c_0} is set to *False* for a clause c_0 of T. Since T evaluates to *True* under α, there is at least one literal v (resp. $\neg v$) in clause c_0 of T with $v \in V$ so that $\alpha(v) = \mathit{True}$ (resp. $\alpha(\neg v) = \mathit{True}$), and the clause $\neg v \vee z_{c_0}$ (resp. $\neg v^* \vee z_{c_0}$) is *False* under α^* and β. Finally, we assume that all variables in $G \cup Z$ are set to *True*. Then the clause (9) is *False*. Thus S evaluates to *False* under any assignment for $G \cup Z$ and under α^*, and hence T^* is *True* under α.

Now assume that T evaluates to *False* under an assignment α to V. We show that T^* evaluates to *False* as well. Assume we extend α to an assignment α^* for $V \cup V^*$. If there is at least one $v \in V$ so that $\alpha(v) = \mathit{True}$ and $\alpha(v^*) = \mathit{True}$, then R is *False*. Assume there is a $v_0 \in V$ so that $\alpha^*(v_0) = \mathit{False}$ and $\alpha^*(v_0^*) = \mathit{False}$. We can find an assignment for $G \cup Z$ that satisfies S as follows. Set g_{v_0} to *False* and all other variables in $G \cup Z$ to *True*. Then all clauses of S are satisfied. Now we assume that $\alpha^*(v) = \neg\alpha^*(v^*)$ for all variables $v \in V$ holds. Again we will determine an assignment for $G \cup Z$ under which S evaluates to *True*. Since T is *False* under α, there is a clause c_0 of T whose literals are all set to *False* by α. We set z_{c_0} to *False* and all other variables to *True*. Due to the construction of the clauses $\neg v \vee z_{c_0}$ and $\neg v^* \vee z_{c_0}$, formula S evaluates to *True*. Thus T^* is *False* under the assignment α for V.

Proof of Theorem 1. Consider the quantified Boolean formula

$$\Phi_k V_k \, \Phi_{k-1} V_{k-1} \ldots \Phi_1 V_1 \, T \tag{11}$$

where T is a CNF formula with variable set $V = V_1 \cup \cdots \cup V_k$ and where all V_i are disjoint. The Φ_i are existential quantifiers if i is odd and universal quantifiers if i is even. For example, for $k = 3$ we obtain

$$\exists V_3 \, \forall V_2 \, \exists V_1 \, T \tag{12}$$

Stockmeyer [6] and Warthall [8] show results for the polynomial hierarchy that prove that evaluating formula (11) is Σ^{P}_k-complete if k is odd and Π^{P}_k-complete if k is even. We define the formula $\exists V^* \, R \wedge \forall G \cup Z \, \neg S$ exactly as in formula (10) above. By Lemma 1, we know that the quantified formula

$$\Phi_k V_k \Phi_{k-1} V_{k-1} \ldots \Phi_1 V_1 \exists V^* \, R \wedge \forall G \cup Z \, \neg S \tag{13}$$

evaluates to *True* if and only if formula (11) evaluates to *True*. Formula (13) can be reduced to the complement of CQF$(k+1)$. We only give an example. All other reductions are carried out analogously. For $k = 2$, we obtain for formula (13)

$$\forall V_2 \, \exists V_1 \, \exists V^* \, R \wedge \forall G \cup Z \, \neg S \tag{14}$$

For $i = 1$, 2, we introduce a new variable set $X_i = \{x_i\}$, and let R_2 be the Horn formula $x_2 \vee \neg x_2$. Then equation (14) is logically equivalent to

$$\forall V_2 \, [\exists X_2 \, R_2 \rightarrow \exists V_1 \cup V^* \, (\exists X_1 \, R \wedge \forall G \cup Z \, \neg S)] \tag{15}$$

which is the complement problem of CQF(3).

In summary, we have shown that Horn CQF(k) is reduced at most by one level in the polynomial hierarchy compared to the general case of CQF(k). In particular, the problem Horn CQF(k) is $\Sigma^{\mathrm{P}}_{k-1}$-hard if k is odd and Π^{P}_{k-1}-hard if k is even.

Inductively we can argue that Horn CQF(k) is reduced at least by one level. It is well-known that Horn CQF(1), which is the satisfiability problem for Horn formulas, is in Σ^{P}_0, that is, in $\mathcal{P}$. Assume the problem Horn CQF(i) is $\Sigma^{\mathrm{P}}_{i-1}$-complete if i is odd and Π^{P}_{i-1}-complete if i is even for some $i \geq 1$. Say, i is odd. Then Horn CQF$(i+1)$ is the problem to evaluate

$$\forall Q_{i+1} \, (\exists X_{i+1} \, R_{i+1}) \rightarrow F_i \tag{16}$$

where R_{i+1} is a Horn formula and F_i is an instance of Horn CQF(i) if the variables Q_{i+1} are set to some truth values. We can verify whether an instance of CQF$(i+1)$ evaluates to *False* as follows. We guess an assignment for Q_{i+1}. Then we use a polynomial-time algorithm to check that the Horn formula R_{i+1} is satisfiable under the assignment for Q_{i+1}. We use a $\Sigma^{\mathrm{P}}_{i-1}$-oracle to determine whether F_i evaluates to *False* under the guessed assignment. If R_{i+1} is satisfiable and F_i is *False* under the guessed assignment for Q_{i+1}, then the instance of Horn CQF$(i+1)$ is *False*. Thus, there exists a Π_i^{P}-algorithm that solves CQF$(i+1)$. Analogously, we can construct a Σ_i^{P}-algorithm for CQF$(i+1)$ if i is odd.

3 The Complexity of 2CNF CQF

Aspvall, Plass, and Tarjan [1] prove that QBF for 2CNF formulas is in $\mathcal{P}$. Remshagen and Truemper [5] have shown that 2CNF CQF(2) is in $\mathcal{P}$ as well. For $k > 2$, the problem 2CNF CQF(k) is only reduced by two level in the polynomial hierarchy.

Theorem 2. For $k \geq 2$, 2CNF CQF(k) is $\Sigma_{k-2}^{\mathrm{P}}$-complete if k is odd and Π_{k-2}^{P}-complete if k is even.

Before we prove Theorem 2, we establish the equivalence between a CNF formula and a related quantified formula. Let T be a CNF formula with variable set V. Define the variable set $Z = \{z_c \,|\, c \text{ is a clause of } T\}$. We construct two 2CNF formulas R_2 and R_1 with variable set $V \cup Z$. The clauses of R_2 and R_1 are as follows. For each literal l contained in a clause c of T, formula R_2 contains the clause $\neg l \vee z_c$. Observe that the clause is equivalent to $l \rightarrow z_c$. For each clause c of T, formula R_1 contains the clause z_c. We show the equivalence of T and the quantified formula

$$T' = \forall Z \, R_2 \rightarrow R_1 \tag{17}$$

Lemma 2. CNF formula T is logically equivalent to T'.

Proof. Assume T evaluates to *True* under an assignment α to V. Let β be a truth assignment for the variable set Z so that R_2 evaluates to *True* under α and β. We show that R_1 evaluates to *True* as well under β. Let z_c be a clause of R_1 for some clause c of T. The clause c must contain a literal l that evaluates to *True* under α. By construction of formula R_2, the formula R_2 contains a clause $\neg l \vee z_c$. Since this clause is *True* under α and β and since $\neg l$ set to *False* by α, assignment β sets z_c to *True*. Then the clause z_c of R_1

evaluates to *True* as well under β. Thus if all variables V are set in R_1 and R_2 according to α, formula R_1 is *True* under every truth assignments under which formula R_2 is *True*. In other words, T' is *True* under α.

Now assume that T evaluates to *False* under a truth assignment α to V. Then all literals of some clause c_0 of T are set to *False* by α. Note that all clauses of R_2 that contain variable z_{c_0} are satisfied by α. We define a truth assignment β for Z that sets z_{c_0} to *False* and all other variables z_c to *True*. Then R_2 evaluates to *True* under α and β. However, the 2CNF formula R_1 is *False* since the clause z_{c_0} is *False* under the assignment β. Thus, the formula $\forall Z\, R_2 \to R_1$ evaluates to *False* under the assignment α.

Proof of Theorem 2. We know that 2CNF CQF(2) is in $\mathcal{P}$, and thus the complexity of CQF(2) is reduced by two levels. We show that 2CNF CQF(i) is also reduced by two levels in the polynomial hierarchy for $i \geq 3$. Consider the quantified Boolean formula

$$\Phi_k V_k\, \Phi_{k-1} V_{k-1} \ldots \Phi_1 V_1 T \tag{18}$$

where T is a CNF formula with variable set $V = V_1 \cup \cdots \cup V_k$ and where all V_i are disjoint. The Φ_i are existential quantifiers if i is odd and universal quantifiers if i is even. We define the formula T'

$$\forall Z\, R_2 \to R_1 \tag{19}$$

exactly as formula (17) above. By Lemma 2, the quantified formula

$$\Phi_k V_k\, \Phi_{k-1} V_{k-1} \ldots \Phi_1 V_1 \forall Z\, R_2 \to R_1 \tag{20}$$

evaluates to *True* if and only if formula (18) evaluates to *True*. Formula (20) can be easily reduced to 2CNF CQF($k+2$). For example, for the formula

$$\forall V_2\, \exists V_1\, \forall Z\, R_2 \to R_1 \tag{21}$$

we introduce, for $i = 1, 2, 3, 4$, new variable sets $X_i = \{x_i\}$ and, for $j = 3, 4$, a 2CNF formula R_j with the single clause $x_j \vee \neg x_j$. Then equation (21) is equivalent to

$$\forall V_2 \left(\exists X_4\, R_4 \to \left(\exists V_1\, \exists X_3\, R_3 \wedge \forall Z\, (\exists X_2\, R_2 \to \exists X_1\, R_1)\right)\right) \tag{22}$$

which is an instance of 2CNF CQF(4). Thus we have shown that 2CNF CQF(k) is reduced by at most two levels in the polynomial hierarchy. In particular, the problem 2CNF CQF(k) is $\Sigma^{\mathrm{P}}_{k-2}$-hard if k is odd and Π^{P}_{k-2}-hard if k is even.

Inductively we can argue that 2CNF CQF(k) is reduced by at least two levels for $k \geq 2$. We know that 2CNF CQF(2) is in Π^{P}_0. Assume the problem CQF(i) is $\Sigma^{\mathrm{P}}_{i-2}$-complete if i is odd and Π^{P}_{i-2}-complete if i is even for some $i \geq 2$. Say, i is odd. Then 2CNF CQF($i+1$) is the problem to evaluate

$$\forall Q_{i+1}\,(\exists X_{i+1}\,R_{i+1}) \rightarrow F_i \tag{23}$$

where R_{i+1} is a 2CNF formula and F_i is an instance of 2CNF CQF(i) if the variables Q_{i+1} are set to some truth values. We can verify whether an instance of 2CNF CQF($i+1$) evaluates to *False* as follows. We guess an assignment for Q_{i+1}. Then we use a polynomial-time algorithm to check that the 2CNF formula R_{i+1} is satisfiable under the assignment to Q_{i+1}. We use a Π^{P}_{i-2}-oracle to determine whether F_i evaluates to *False* under the guessed assignment. If R_{i+1} is satisfiable and F_i is *False* under the assignment for Q_{i+1}, then instance (23) is *False*. Thus there exists a Π^{P}_{i-1}-algorithm that solves 2CNF CQF($i+1$). Analogously, we can show that there is a $\Sigma^{\mathrm{P}}_{i-1}$-algorithm for 2CNF CQF($i+1$) if i is even.

References

[1] Aspvall, B., Plass, M. F., Tarjan, E.: A linear-time algorithm for testing the truth of certain quantified Boolean formulas. Information Processing Letters 8 (3), (1979) 121–123.

[2] Benedetti, M., Lallouet, A., Vautard, J.: QCSP made Practical by Virtue of Restricted Quantification. Proceedings of the 20th International Joint Conference on Artificial Intelligence (2007).

[3] Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Information and Computation 117(1), (1995) 12–18.

[4] Sabharwal, A., Ansotegui, C., Gomes, C. P., Hart, J. W., Selman, B.: QBF Modeling: Exploiting Player Symmetry for Simplicity and Efficiency. Proceedings of the Ninth International Conference on Theory and Applications of Satisfiability Testing (2006).

[5] Remshagen, A., Truemper, K.: The Complexity of Futile Questioning. Proceedings of the International Conference on Foundations in Computer Science (2007) 132–138.

[6] Stockmeyer, L. J.: The polynomial hierarchy. Theoretical Computer Science 3, (1976) 1–22.

[7] Truemper, K.: Design of Logic-based Intelligent Systems. Wiley (2004).

[8] Warthall, C.: Complete sets and the polynomial hierarchy. Theoretical Computer Science 3, (1976) 23–33.