QBF Resolution Systems and their Proof Complexities *

Valeriy Balabanov[1], Magdalena Widl[2], and Jie-Hong R. Jiang[1]

[1]National Taiwan University, [2]Vienna University of Technology
{balabasik@gmail.com, widl@kr.tuwien.ac.at, jhjiang@ntu.edu.tw}

Abstract. Quantified Boolean formula (QBF) evaluation has a broad range of applications in computer science and is gaining increasing attention. Recent progress has shown that for a certain family of formulas, Q-resolution, which forms the foundation of learning in modern search-based QBF solvers, is exponentially inferior in proof size to two of its extensions: Q-resolution with resolution over universal literals (QU-resolution) and long-distance Q-resolution (LQ-resolution). The relative proof power between LQ-resolution and QU-resolution, however, remains unknown. In this paper, we show their incomparability by exponential separations on two families of QBFs, and further propose a combination of the two resolution methods to achieve an even more powerful proof system. These results may shed light on solver development with enhanced learning mechanisms. In addition, we show how QBF Skolem/Herbrand certificate extraction can benefit from polynomial LQ-resolution proofs in contrast to their exponential Q-resolution counterparts.

1 Introduction

Quantified Boolean formulas (QBFs) can naturally express many decision problems encountered in verification [4,16], planning [15], two-player games [8], electronic design automation [10,12], and other fields in computer science. QBFs extend formulas of propositional logic by adding quantifiers over the (Boolean) variables, which makes them more expressive and allows a more compact representation of logical constraints. Their efficient evaluation has significant practical impacts and is gaining more and more research attention. State of the art evaluation methods for QBF have been considerably influenced by the advancement of satisfiability (SAT) solving of propositional logic [14] and contain methods based on SAT techniques like conflict-driven clause learning (CDCL) [14]. However, possibly to its higher complexity, QBF evaluation remains premature for robust industrial applications and awaits new insights for a breakthrough.

Resolution is a fundamental technique in automated reasoning, in particular for SAT [20]. CDCL, the key technique for efficiency in modern SAT solvers,

* This work was supported in part by the National Science Council (NSC) of Taiwan under grants 101-2923-E-002-015-MY2 and 102-2221-E-002-232, by the Vienna Science and Technology Fund (WWTF) through project ICT10-018, and by the Austrian Science Fund (FWF) under grant S11409-N23.

can be considered as a guided resolution process. Not surprisingly, resolution also plays an essential role in the learning mechanism (QCDCL) of modern QBF solvers [2,7,9,13] In QBF, the existence of more than one sound resolution rule enables different proof systems. In particular, Q-resolution [11], which allows resolution only over existential variables and uses universal reduction to remove universal variables, and its extensions by allowing resolution over universal variables (QU-resolution) [18] and by allowing tautological long-distance derivations (LQ-resolution) [1,19], have been proposed.

Recent studies have shown that members of a certain family of QBFs [11] have proofs in QU-resolution or LQ-resolution of polynomial size in the formula size but any Q-resolution proof is claimed to be of exponential size [6,11,18]. On the practical side, an embedding of LQ-resolution in the QCDCL-based solver DepQBF [13] has resulted in significant performance gains [6]. This gives rise to the question whether other resolution systems can have similar impacts. Also, the relative proof complexity between QU-resolution and LQ-resolution remains unknown.

In addition to its contribution to learning, resolution can produce a syntactic proof of the truth or falsity of a QBF. However, besides a validation of the decision result, many applications require a semantic certificate to represent a concrete solution. Such certificates are typically represented in terms of Skolem functions for true QBFs and Herbrand functions for false QBFs. They can be extracted from Q-resolution proofs in time linear in the proof size [1] and their size is usually related to the proof size. Thus, the study of certificate extraction from the potentially smaller QU-resolution and LQ-resolution proofs is very important.

The quests for efficient QBF evaluation and for the extraction of compact QBF (counter)models motivate the investigation of more powerful resolution systems. In this work, we present the following related results. First, we show the incomparability of QU-resolution and LQ-resolution with respect to their proof complexities. To this end, we construct two families of QBFs for which either of the two calculi has only proofs of exponential size, but the other can produce proofs of polynomial size. Second, we define two stronger proof systems and show an exponential separation to QU- and LQ-resolution for one of them. Third, we propose a new procedure for (counter)model extraction from resolution proofs in all the discussed proof systems. Finally, we present an experimental evaluation of the new certificate extraction method.

2 Preliminaries

A Boolean variable over the domain $\{\top \text{ (true)}, \bot \text{ (false)}\}$ appears in a propositional formula ϕ as a *positive literal* or a *negative literal.* We refer to the opposite polarity of a (positive or negative) literal l by $\bar{l}$ and to the variable of a literal l by $\mathsf{var}(l)$. We use $\mathsf{lit}(v) \in \{v, \overline{v}\}$ to refer to either literal of a variable v. A propositional conjunctive normal form (CNF) formula is a conjunction of *clauses*, each of which is a disjunction of literals. We denote a CNF formula by a set of clauses, a clause by a set of literals, and the empty clause by $\Box$. We use the Boolean connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ with their standard interpretation.

Given a set V of Boolean variables, a set $L = V \cup \{\overline{v} \mid v \in V\}$ of positive and negative literals over V, and the existential ($\exists$) and universal ($\forall$) quantifiers, a *quantified Boolean formula* (QBF) $\mathcal{P}.\phi$ in *prenex conjunctive normal form* (PCNF) consists of the prefix $\mathcal{P} = Q_1 v_1 \dots Q_k v_k$ with $Q_i \in \{\exists, \forall\}$, $v_i \in V$, and $v_i \neq v_j$ if $i \neq j$, and the CNF matrix $\phi \subset 2^L$. All QBFs in this work are assumed to be in PCNF, to be *closed*, i.e., all literals in the matrix are quantified in the prefix, and to be free of tautological clauses. For each variable $v_i \in V$, its *quantifier level* $\mathsf{lev}(v_i)$ is the number of alternations between $\exists$ and $\forall$ quantifiers from Q_1 to Q_i. We apply this definition also to literals, i.e., $\mathsf{lev}(l) = \mathsf{lev}(\mathsf{var}(l))$. The *quantifier index* of v_i is $\mathsf{idx}(v_i) = i$. Similarly, for literal l, $\mathsf{idx}(l) = \mathsf{idx}(\mathsf{var}(l))$. The set V of variables is partitioned into the set $V_\exists = \{v_i \in V \mid Q_i = \exists\}$ of *existential variables* and the set $V_\forall = \{v_i \in V \mid Q_i = \forall\}$ of *universal variables*. We use letters from the beginning of the Latin alphabet for existential variables/literals, letters from the end for universal variables/literals, and v for either.

A (partial) *assignment* to a QBF $\Phi = \mathcal{P}.\phi$ is a set $\sigma \subset L$ where it holds that if $l \in \sigma$ then $\overline{l} \notin \sigma$. The *assignment condition* $\mathsf{cond}(\sigma)$ is the conjunction $(\bigwedge_{l \in \sigma} l)$ of literals in σ. A clause C is evaluated under an assignment σ to $C_{\lceil \sigma}$ such that $C_{\lceil \sigma} = \top$ if $C \cap \sigma \neq \emptyset$, $C_{\lceil \sigma} = \bot$ if $C \setminus \{v \mid \overline{v} \in \sigma\} = \emptyset$, and $C_{\lceil \sigma} = C \setminus \{v \mid \overline{v} \in \sigma\}$ otherwise. A QBF Φ is evaluated under an assignment σ to $\Phi_{\lceil \sigma}$ by replacing each $C \in \phi$ by $C_{\lceil \sigma}$. The QBF $\forall x \mathcal{P}.\phi$ is true if and only if $\mathcal{P}.\phi_{\lceil \{x\}}$ and $\mathcal{P}.\phi_{\lceil \{\overline{x}\}}$ are true. The QBF $\exists e \mathcal{P}.\phi$ is true if and only if $\mathcal{P}.\phi_{\lceil \{e\}}$ or $\mathcal{P}.\phi_{\lceil \{\overline{e}\}}$ is true.

A clause containing a variable in both polarities is tautological. In QBF reasoning the derivation of such clauses can be useful under certain conditions. A universal variable x contained in a clause C as both x and $\overline{x}$ is called a *merged variable*. A *merged literal* l^* is used to replace both literals l and $\overline{l}$ in C. We define $\mathsf{var}(l^*) = \mathsf{var}(l)$, $\mathsf{lev}(l^*) = \mathsf{lev}(l)$, and $\mathsf{idx}(l^*) = \mathsf{idx}(l)$.

The QBF proof systems considered in this work are based on the two derivation rules *resolution* and *universal reduction*. Given two clauses C_1 and C_2, and a *pivot variable* p with $p \in C_1, \overline{p} \in C_2$, resolution produces the clause $\mathsf{resolve}(C_1, p, C_2) = C_1 \setminus \{p\} \cup C_2 \setminus \{\overline{p}\}$. We call this rule an *ordinary* resolution if the following condition holds: For all (merged or regular) literals $l_1 \in C_1 \setminus \{p\}$ and $l_2 \in C_2 \setminus \{\overline{p}\}$ it holds that if $\mathsf{var}(l_1) = \mathsf{var}(l_2)$ then $l_1 = l_2$ and l_1 is not merged. Otherwise we refer to it as *long-distance* resolution. We further distinguish ordinary resolution into $\mathsf{resolve}_\exists$ if $p \in V_\exists$ and $\mathsf{resolve}_\forall$ if $p \in V_\forall$. We call long-distance resolution over pivot $p \in V_\exists$ *proper* and denote it by $\mathsf{resolve}_{\exists\mathsf{L}}$ if the following *index restriction* holds: For all (merged or regular) literals $l_1 \in C_1 \setminus \{p\}$ and $l_2 \in C_2 \setminus \{\overline{p}\}$ it holds that if $\mathsf{var}(l_1) = \mathsf{var}(l_2)$ and either $l_1 \neq l_2$ or l_1 is merged, then $\mathsf{var}(l_1) \in V_\forall$ and $\mathsf{idx}(l_1) = \mathsf{idx}(l_2) > \mathsf{idx}(p)$. Note that since $p \in V_\exists$ and $l_1, l_2 \in V_\forall$, lev can be used instead of idx. Given a clause C, universal reduction produces the clause $\mathsf{reduce}(C) = C \setminus \{l \mid \mathsf{var}(l) \in V_\forall \text{ and } \mathsf{lev}(l) > \mathsf{lev}(l') \text{ for all } l' \in C \text{ with } \mathsf{var}(l') \in V_\exists\}$, i.e., it removes from C all universal variables whose quantifier levels are greater than the largest level of any existential variable in C. Note that reduce applies to merged literals from C in the same way as it applies to regular literals.

The following three QBF resolution proof systems are sound and complete: Q-resolution [11] contains the derivation rules reduce and $\mathsf{resolve}_{\exists}$. QU-resolution [17] and LQ-resolution [1] extend Q-resolution by the rules $\mathsf{resolve}_{\forall}$ and $\mathsf{resolve}_{\exists\mathsf{L}}$, respectively. A $\{\mathsf{Q},\mathsf{QU},\mathsf{LQ}\}$-resolution proof Π of the falsity of a QBF $\Phi = \mathcal{P}.\phi$ is a directed acyclic graph (DAG) representing clauses derived from ϕ by repeated applications of the respective rules in process of deriving $\Box$. The operation reduce is applied to any clause in Π from which it can remove a literal. (Note that the definition of a QU-resolution proof in [17] does not include the mandatory application of reduce. We discuss the influence of arbitrarily *postponing* the reduce operation in Section 3.1.) We call application of a derivation rule a *step*. The *size* of Π is the number of clauses in Π that are derived by resolution (not by reduction). By *topological order* we refer to any order following the derivation steps in Π from the clauses in ϕ to $\Box$.

To witness the falsity (truth) of a QBF, a countermodel (model) can be built in terms of *Herbrand* (*Skolem*) functions. A false (true) QBF $\Phi = \mathcal{P}.\phi$ warrants the existence of a Herbrand (Skolem) function h_v (s_v) for each $v \in V_{\forall}$ ($v \in V_{\exists}$) referring only to the variables $\{e \in V_{\exists} \mid \mathsf{lev}(e) < \mathsf{lev}(v)\}$ ($\{x \in V_{\forall} \mid \mathsf{lev}(x) < \mathsf{lev}(v)\}$) such that substituting each appearance of a variable v in ϕ by its function h_v (s_v) makes the resultant formula, denoted $\Phi[\mathcal{H}]$ for $\mathcal{H} = \{\mathsf{h}_v \mid v \in V_{\forall}\}$ ($\Phi[\mathcal{S}]$ for $\mathcal{S} = \{\mathsf{s}_v \mid v \in V_{\exists}\}$), unsatisfiable (tautological).

3 Resolution Proof Systems and their Complexities

In this section, we first show an exponential gap between the proof complexities of LQ-resolution and QU-resolution with respect to two families of QBFs obtained by modifications of a family of QBFs introduced in [11] (in the sequel called "KBKF family"). Then we introduce two new resolution proof systems, both of which are extensions of Q-resolution, and show an exponential separation between QU-resolution, LQ-resolution, and one of the new resolution systems.

3.1 Incomparability of LQ- and QU-resolutions

We first give an intuition of how to engineer a false QBF that inhibits $\mathsf{resolve}_{\forall}$ and $\mathsf{resolve}_{\exists\mathsf{L}}$ steps in any of its resolution proofs. Ex. 1 shows a false QBF for which any resolution proof cannot contain $\mathsf{resolve}_{\forall}$ or $\mathsf{resolve}_{\exists\mathsf{L}}$ steps.

Ex. 1. Consider the false QBF $\Phi = \exists a \forall x \forall y \exists b.(a, x, y, b)(\overline{a}, \overline{x}, \overline{y}, b)(x, y, \overline{b})(\overline{x}, \overline{y}, \overline{b})$. The falsity of Φ is shown by the Herbrand functions $\mathsf{h}_y = \mathsf{h}_x = a$. Let Π be a QU-resolution proof of Φ. Since $\mathsf{lev}(x) = \mathsf{lev}(y)$, the universal reduction reduce always removes both x and y at once. Thus, any clause in Π either contains both x and y in the same polarity, or neither x nor y in any polarity. It follows that Π cannot contain any clause derived by $\mathsf{resolve}_{\forall}$. Alternatively, let Π be an LQ-resolution proof of Φ. Due to the level restriction, any $\mathsf{resolve}_{\exists\mathsf{L}}$ step must have a as pivot variable, so $\mathsf{resolve}_{\exists\mathsf{L}}((a, x, y, b), a, (\overline{a}, \overline{x}, \overline{y}, b)) = (x^{*}, y^{*}, b)$ is the only possible such step. However, this resolvent can never be used in a derivation of $\square$, because the necessary pivot literal $\overline{b}$ always occurs in clauses together with literals of x and y, which forbids any further resolution.

Definition 1 reproduces the definition of the KBKF family [11]. Theorem 3.2 in [11] claims that any Q-resolution proof for members of this family is of size exponential in t [11], but its proof is not completely given. It has further been shown that there exist a QU-resolution proof [17] and an LQ-resolution proof [6] of size polynomial in t. For the remainder of this section it is important to keep in mind that for all $i \in [1..t]$, $\mathsf{lev}(e_i) = \mathsf{lev}(d_i) < \mathsf{lev}(x_i)$ and $\mathsf{lev}(x_t) < \mathsf{lev}(f_i)$.

Definition 1 (KBKF family[11]). *For $t > 1$, the t^{th} member KBKF[t] of the KBKF family consists of the following prefix and clauses:*

$$\begin{array}{llll}
\multicolumn{4}{l}{\exists d_1 e_1\ \forall x_1\ \exists d_2 e_2\ \forall x_2\ ..\ \exists d_t e_t\ \forall x_t\ \exists f_1..f_t} \\
B\ = (\overline{d}_1, \overline{e}_1) & & & \\
D_i = (d_i, x_i, \overline{d}_{i+1}, \overline{e}_{i+1}) & E_i = (e_i, \overline{x}_i, \overline{d}_{i+1}, \overline{e}_{i+1}) & \text{for } i \in [1..t-1] \\
D_t = (d_t, x_t, \overline{f}_1, .., \overline{f}_t) & E_t = (e_t, \overline{x}_t, \overline{f}_1, .., \overline{f}_t) & \\
F_i\ = (x_i, f_i) & F'_i = (\overline{x}_i, f_i) & \text{for } i \in [1..t]
\end{array}$$

We now apply ideas from Ex. 1 to transform the KBKF family into the family KBKF-qu, for which, based on Theorem 3.2 in [11], the smallest QU-refutations are of exponential size but there exist LQ-refutations of size polynomial in t. It follows from the existence of these proofs that the members of this family are false. For $t > 1$, KBKF-qu[t] is obtained from KBKF[t] by adding fresh universal variables y_i to some clauses.

Definition 2 (KBKF-qu family). *For $t > 1$, the t^{th} member KBKF-qu[t] of the KBKF-qu family consists of the following prefix and clauses:*

$$\begin{array}{llll}
\multicolumn{4}{l}{\exists d_1 e_1\ \forall x_1 y_1\ \exists d_2 e_2\ \forall x_2 y_2\ ..\ \exists d_t e_t\ \forall x_t y_t\ \exists f_1..f_t} \\
B\ = (\overline{d}_1, \overline{e}_1) & & & \\
D_i = (d_i, x_i, y_i, \overline{d}_{i+1}, \overline{e}_{i+1}) & E_i = (e_i, \overline{x}_i, \overline{y}_i, \overline{d}_{i+1}, \overline{e}_{i+1}) & \text{for } i \in [1..t-1] \\
D_t = (d_t, x_t, y_t, \overline{f}_1, .., \overline{f}_t) & E_t = (e_t, \overline{x}_t, \overline{y}_t, \overline{f}_1, .., \overline{f}_t) & \\
F_1\ = (x_i, y_i, f_i) & F'_i = (\overline{x}_i, \overline{y}_i, f_i) & \text{for } i \in [1..t]
\end{array}$$

The following proposition shows that the shortest Q-refutation for KBKF-qu[t] is at least as long as the shortest Q-refutation for KBKF[t].

Proposition 1. *Given a false QBF $\Phi = Q_1 v_1\ ..\ Q_k v_k.\ C_1 \wedge C_2 \wedge .. \wedge C_n$ over the set V of variables, it holds that for any variable $v \in V$, if $\Phi^* = Q_1 v_1\ ..\ Q_k v_k.\ C_1 \wedge .. \wedge (C_j \cup \{\mathsf{lit}(v)\}) \wedge .. \wedge C_n$ is false, then the smallest $\{$Q,QU,LQ$\}$-resolution proof for Φ^* is at least as large as that for Φ.*

The validity of Proposition 1 can be understood by the fact that removing the literal $\mathsf{lit}(v)$ from the clause $(C_j \cup \{\mathsf{lit}(v)\})$ can only decrease the proof size of Φ^*. Note that adding a fresh variable $v \notin V$ to $\mathcal{P}$ influences neither the satisfiability of Φ, nor the validity of any of its Q-resolution proofs. Thus Proposition 1 can be extended for addition of fresh variables to $\mathcal{P}$ and their literals to ϕ.

Theorem 1. *For $t > 1$ there exists an LQ-refutation of polynomial size for KBKF-qu[t], but any QU-refutation for KBKF-qu[t] is of exponential size in t (based on Theorem 3.2 in [11]).*

Proof. Except for the clause B, each clause of KBKF-qu[t] contains two universal variables x, y with the same level and the same polarity. For any QU-refutation, in order to have a resolve$_\forall$ step over two clauses C_1 and C_2 with x (respectively y) as pivot, y (x) must be removed from one of the clauses, which can only be done by reduce. Whenever y (x) is reduced, so is x (y). Therefore, any QU-refutation will be a Q-refutation , and by Theorem 3.2 in [11] and Proposition 1, the shortest Q-refutation for KBKF-qu is exponential. On the other hand, by following the method proposed in Proposition 1 of [6], a polynomial LQ-refutation can be obtained. □

We continue with the following modification of the KBKF family that inhibits resolve$_{\exists\mathsf{L}}$ steps but allows polynomial QU-refutations. For $t > 1$, KBKF-lq[t] is retrieved from KBKF[t] by adding literals $\overline{f}_1, .., \overline{f}_t$ to clauses B, D_i and E_i, and literals $\overline{f}_{i+1}, .., \overline{f}_t$ to clauses F_i and F'_i, for all $i \in [1..t-1]$.

Definition 3 (KBKF-lq family). *For $t > 1$, the t^{th} member KBKF-lq[t] of the KBKF-lq family consists of the following prefix and clauses:*

$$\begin{array}{llll}
\multicolumn{4}{l}{\exists d_1 e_1\ \forall x_1\ \exists d_2 e_2\ \forall x_2\ ..\ \exists d_t e_t\ \forall x_t\ \exists f_1..f_t} \\
B\ = (\overline{d}_1, \overline{e}_1, \overline{f}_1, .., \overline{f}_t) & & & \\
D_i = (d_i, x_i, \overline{d}_{i+1}, \overline{e}_{i+1}, \overline{f}_1, .., \overline{f}_t) & E_i = (e_i, \overline{x}_i, \overline{d}_{i+1}, \overline{e}_{i+1}, \overline{f}_1, .., \overline{f}_t) & \text{for } i \in [1..t-1] \\
D_t = (d_t, x_t, \overline{f}_1, .., \overline{f}_t) & E_t = (e_t, \overline{x}_t, \overline{f}_1, .., \overline{f}_t) & \\
F_i\ = (x_i, f_i, \overline{f}_{i+1}, .., \overline{f}_t) & F'_i = (\overline{x}_i, f_i, \overline{f}_{i+1}, .., \overline{f}_t) & \text{for } i \in [1..t-1] \\
F_t\ = (x_t, f_t) & F'_t = (\overline{x}_t, f_t) &
\end{array}$$

Observation 1. For $t > 1$ any member KBKF-lq[t] of the KBKF-lq family is an extended quantified Horn (QE-Horn) formula [11] and QE-Horn formulas are closed under LQ-resolution.

The closure of QE-Horn formulas under LQ-resolution directly follows from their closure under Q-resolution (observe that the resolve$_{\exists\mathsf{L}}$ rule does not influence existential literals in the clauses). On the other hand, note that QE-Horn formulas are not closed under QU-resolution. Further, the following three invariants hold for any member of KBKF-lq family.

Lemma 1 (Invariant 1). *Given any LQ-resolution proof Π of a formula KBKF-lq[t], the following holds for any clause $C \in \Pi$: For all $i \in [1..t]$, if $f_i \in C$ then $\mathsf{lit}(x_i) \in C$, and if $\overline{f}_i \in C$ then for any $j \in [i..t]$ either $\overline{f}_j \in C$ or $\mathsf{lit}(x_j) \in C$.*

Proof. First, observe that the invariant holds for any clause in the original clause set of KBKF-lq[t]. Let C be a clause derived from C' by exactly one derivation step, such that $f_i \in C$ and $f_i \in C'$. If $\mathsf{lit}(x_i) \in C'$ then it must hold that $\mathsf{lit}(x_i) \in C$, because resolution on universal variables is forbidden and the presence of f_i disallows the universal reduction of $\mathsf{lit}(x_i)$ in both C' and C. Thus by induction it holds for any clause C that if $f_i \in C$ then $\mathsf{lit}(x_i) \in C$.

Now let C be a clause derived from C' by exactly one derivation step, such that $\overline{f}_i \in C$ and $\overline{f}_i \in C'$. If $\mathsf{lit}(x_j) \in C'$ for some $j \in [i..t]$, then $\mathsf{lit}(x_j) \in C$ for the same reasons as above. If $\overline{f}_j \in C'$ for some $j \in [i..t]$, then either $\mathsf{lit}(x_j) \in C$ (in the case where f_j is the pivot variable, i.e., $C = \mathsf{resolve}(C', f_j, C'')$ with

$f_j, \mathsf{lit}(x_j) \in C''$ by the above discussion), or $\overline{f}_j \in C$ (in any other case). Thus by induction it holds for any clause C that if $\overline{f}_i \in C$ then for any $j \in [i..t]$ either $\overline{f}_j \in C$ or $\mathsf{lit}(x_j) \in C$. □

Lemma 2 (Invariant 2). *Given any* LQ*-resolution proof* Π *of a formula* KBKF-lq[t] *the following holds for any clause* $C \in \Pi$*: For all* $i \in [1..t]$*, if* $\mathsf{lit}(d_i) \in C$ *or* $\mathsf{lit}(e_i) \in C$ *then* $f_j \notin C$ *for any* $j \in [1..t]$*.*

Proof. First, the invariant holds for any clause in the original clause set of KBKF-lq[t]. Now let $C = \mathsf{resolve}(C_1, p, C_2)$, where $\mathsf{lit}(e_i) \in C$ or $\mathsf{lit}(d_i) \in C$, and $\mathsf{lit}(e_i) \in C_1$ or $\mathsf{lit}(d_i) \in C_1$ for some $i \in [1..t]$.

If $\mathsf{lit}(e_k) \in C_2$ or $\mathsf{lit}(d_k) \in C_2$ for some $k \in [1..t]$, then by inductive hypothesis it holds that $f_j \notin C_1$ and $f_j \notin C_2$ for all $j \in [1..t]$. Therefore, by the definition of resolve, it holds that $f_j \notin C$ for all $j \in [1..t]$.

Else, $\mathsf{lit}(e_i) \notin C_2$ and $\mathsf{lit}(d_i) \notin C_2$, thus we are left with $p = f_k$ for some $k \in [1..t]$. By inductive hypothesis, $f_j \notin C_1$ for all $j \in [1..t]$, therefore $\overline{f}_k \in C_1$ and $f_k \in C_2$. By Observation 1 it holds that $f_j \notin C_2$ for all $j \in [1..t]$ with $j \neq k$. Thus for all $j \in [1..t]$ it holds that $f_j \notin C$.

Therefore, by induction it holds for any clause C and for all $i \in [1..t]$ that if $\mathsf{lit}(d_i) \in C$ or $\mathsf{lit}(e_i) \in C$ then $f_j \notin C$ for any $j \in [1..t]$. □

Lemma 3 (Invariant 3). *Given any* LQ*-resolution proof* Π *of a formula* KBKF-lq[t] *the following holds for any clause* $C \in \Pi$*: For all* $i \in [1..t]$ *it holds that if* $\mathsf{lit}(d_i) \in C$ *or* $\mathsf{lit}(e_i) \in C$ *then for any* $j \in [1..i-1]$ *either* $\overline{f}_j \in C$ *or* $\mathsf{lit}(x_j) \in C$*.*

Proof. First, note that the invariant holds for any clause of the original clause set of KBKF-lq[t]. Now, let C be a clause retrieved from C' by one derivation step, such that $\mathsf{lit}(e_i) \in C'$ or $\mathsf{lit}(d_i) \in C'$, and $\mathsf{lit}(e_i) \in C$ or $\mathsf{lit}(d_i) \in C$. If for some $j \in [1..i-1]$ it holds that $\mathsf{lit}(x_j) \in C'$, then $\mathsf{lit}(x_j) \in C$ for the same reasons as in the proof of Invariant 1 (recall that $\mathsf{lev}(e_i) = \mathsf{lev}(d_i) > \mathsf{lev}(x_j)$ for $j \in [1..i-1]$, therefore disallowing universal reduction of $\mathsf{lit}(x_j)$ in the presence of either $\mathsf{lit}(e_i)$ or $\mathsf{lit}(d_i)$). If $\overline{f}_j \in C'$ for some $j \in [1..i-1]$, then either $\mathsf{lit}(x_j) \in C$ (in the case where f_j is the pivot variable, i.e., $C = \mathsf{resolve}(C', f_j, C'')$ with $\{f_j, \mathsf{lit}(x_j)\} \in C''$ by Invariant 1), or $\overline{f}_j \in C$ (in any other case).

Therefore by induction it holds for any clause C and for all $i \in [1..t]$ that if $\mathsf{lit}(d_i) \in C$ or $\mathsf{lit}(e_i) \in C$ then for any $j \in [1..i-1]$ either $\overline{f}_j \in C$ or $\mathsf{lit}(x_j) \in C$. □

Theorem 2. *For* $t > 1$ *there exists a* QU*-resolution proof of polynomial size for* KBKF-lq[t]*, but any* LQ*-resolution proof for* KBKF-lq[t] *is of exponential size in* t *(based on Theorem 3.2 in [11]).*

Proof. For $t > 1$, a QU-refutation of polynomial size in t for KBKF-lq[t] can be constructed as follows: The unit clause (f_t) is obtained by the resolution step $\mathsf{resolve}_\forall(F_t, x_t, F'_t)$. Then, for each $i \in [1..t-1]$, the unit clause (f_i) is obtained by recursively resolving all previous units $(f_{i+1})..(f_t)$ with the resolvent $\mathsf{resolve}_\forall(F_i, x_i, F'_i)$. For $i \in [1..t]$ these unit clauses are used to remove all $\overline{f_i}$ from the clauses D_i, E_i, and B, and the existential literals e_i and d_i are removed one after another by $\mathsf{resolve}_{\exists}$ over the remaining clauses.

For the remainder of this proof, let Π be an LQ-resolution proof for KBKF-lq[t]. Let the three clauses $C_1 = (A_1, p, X, R_1)$, $C_2 = (A_2, \overline{p}, \overline{X}, R_2)$, and $C = (A, X^*, R)$ be parts of a $\mathsf{resolve}_{\exists\mathsf{L}}$ step in Π, where X is a set of universal literals, $\overline{X} = \{\overline{x} \mid x \in X\}$, $X^* = \{x^* \mid x \in X\}$, $C = \mathsf{resolve}_{\exists\mathsf{L}}(C_1, p, C_2)$ is the resolvent of C_1 and C_2, $A = A_1 \cup A_2$, and $R = R_1 \cup R_2$. Let x_m and x_n be the variables with the lowest, respectively the highest, level among the variables in X. By definition of $\mathsf{resolve}_{\exists\mathsf{L}}$ it holds that $\mathsf{lev}(p) < \mathsf{lev}(x_m)$. Without loss of generality, for $i \in \{1, 2\}$ let $R_i = \{v \in C_i \mid v \notin X \land \mathsf{lev}(v) > \mathsf{lev}(x_m)\}$ and $A_i = \{v \in C_i \mid v \notin (X \cup R_i \cup \{p\})\}$. Therefore, $R = \{v \in C \mid v \notin X^* \land \mathsf{lev}(v) > \mathsf{lev}(x_m)\}$ and $A = \{v \in C \mid v \notin (X^* \cup R)\}$. It is important to notice that the existential literals in R have to be removed from successors of C before X^* can be reduced. Further, $f_i \notin R$ for all $i \in [1..t]$ by Invariant 2, and $R_1, R_2 \neq \emptyset$ because otherwise x_m would be reduced before deriving C. Hence $R \subset \{\mathsf{lit}(e_i), \mathsf{lit}(d_i), \mathsf{lit}(x_i) \mid m < i \leq t\} \cup \{\overline{f}_i \mid 1 \leq i \leq t\}$.

We now show by case distinction on the existential variables in R that the clause C can either not contribute to the derivation of $\Box$ in Π because at least one of the merged variables can never be reduced, or that the subclause A can be retrieved from C_1, C_2, and the input clauses in a polynomial number of derivation steps in the Q-resolution calculus. Under the assumption that Π is of polynomial size, its polynomial transformation into a Q-resolution proof contradicts with Proposition 1 and Theorem 3.2 in [11], stating that any Q-resolution for any member of KBKF-lq is exponential. Therefore, Π must be exponential.

Case 1. $\overline{f}_n \in R$. To remove $\overline{f}_n$, C has to be resolved with a clause C' containing f_n. By Invariant 1, C' contains $\mathsf{lit}(x_n)$. Thus $\overline{f}_n$ cannot be removed from R due to the level restriction on $\mathsf{resolve}_{\exists\mathsf{L}}$ steps. Therefore, $C \notin \Pi$.

Case 2. $\mathsf{lit}(d_i) \in R$ or $\mathsf{lit}(e_i) \in R$. Recall that $i > m$, and without loss of generality, let $d_i \in R$. To remove d_i, C has to be resolved with a clause C' containing $\overline{d}_i$. By Invariant 3, C' either contains $\mathsf{lit}(x_m)$ or $\overline{f}_m$. In the first case, the level restriction on $\mathsf{resolve}_{\exists\mathsf{L}}$ steps forbids the resolution, and in the latter case, Invariant 1 applies to the resolvent similarly as in Case 1. Therefore, $C \notin \Pi$.

Case 3. $\overline{f}_i \in R$ and $i < n$. Similarly to Case 2, to remove $\overline{f}_i$, C has to be resolved with a clause C' containing f_i. By Invariant 1, C' either contains $\mathsf{lit}(x_n)$, which blocks the resolution as in Case 1 and Case 2, or it contains $\overline{f}_n$ and therefore to its resolvent, Case 1 applies. Therefore, $C \notin \Pi$.

Case 4. $\overline{f}_i \in R$ and $i > n$. Assume without loss of generality that $\overline{f}_i \in R_1$. For $j \in [i..t]$ let the set X' contain x_j if $x_j \in R_1$ and contain $\overline{x}_j$ if $x_j \notin R_1$. By applying $\mathsf{resolve}_{\exists}$ over an adequate subset of the clauses $\{F_j, F'_j \mid i \leq j \leq t\}$, the clause (f_i, X') can be obtained in a polynomial number of steps and be resolved with C to eliminate $\overline{f_i}$. This procedure can be applied to eliminate all $\overline{f}_i$ literals from R_1 and thus enable reduce on the variables in X. By applying the same rewriting to C_2 eventually $\mathsf{resolve}_{\exists\mathsf{L}}(C_1, p, C_2)$ transforms into $\mathsf{resolve}_{\exists}(C_1 \setminus \{X, F_1\}, p, C_2 \setminus \{\overline{X}, F_2\})$, where $F_1 = \{\overline{f}_i \mid i > n \land \overline{f}_i \in C_1\}$ and $F_2 = \{\overline{f}_i \mid i > n \land \overline{f}_i \in C_2\}$. □

For {Q, QU, LQ}-resolution, we follow the assumption that universal reduction is performed whenever possible. If one allows postponing the reduction arbitrarily (as in the definition of QU-resolution in [17]), it will generalize the aforementioned proof systems and allow a larger number of sound refutations. In the sequel we call a refutation where the reduction of at least one universal variable has been postponed a *postponed refutation* and a clause that contains a universal variable which could be universally reduced, but is still present in at least one of its child clauses, a *postponed clause.* The following corollary from Proposition 1 shows that postponing cannot lead to shorter refutations in terms of the number of resolutions for any of the {Q, QU, LQ}-resolution proof systems.

Corollary 1. *Given a false QBF Φ, let Π be its shortest {Q, QU, LQ}-refutation, and let Π^* be its shortest postponed {Q, QU, LQ}-refutation. Then $|\Pi^*| \geq |\Pi|$.*

Proposition 1 can be applied to all topologically first postponed clauses in a postponed refutation and therefore, the corollary follows. By Corollary 1, Theorems 1 and 2 hold for postponed QU-refutations as well.

3.2 New Resolution Proof Systems

We propose two additional resolution systems for QBF. The first, LQU-resolution, is defined as an extension of Q-resolution by adding both the $\mathsf{resolve}_\forall$ and the $\mathsf{resolve}_{\exists\mathsf{L}}$ derivation rules. The second, LQU+-resolution, extends LQU-resolution by the new derivation rule $\mathsf{resolve}_{\forall\mathsf{L}}$ that allows proper long-distance resolutions under universally quantified pivots. The proof for soundness of $\mathsf{resolve}_{\forall\mathsf{L}}$ is similar to that of $\mathsf{resolve}_{\exists\mathsf{L}}$ rule in [1]. Note that the index restriction imposed on $\mathsf{resolve}_{\forall\mathsf{L}}$ cannot be simplified to level restriction as for $\mathsf{resolve}_{\exists\mathsf{L}}$, since the universal pivot may have the same level as a merged literal in the same proof step. The following example shows that relaxing the index restriction to the level restriction is unsound for $\mathsf{resolve}_{\forall\mathsf{L}}$.

Ex. 2. Consider the true QBF $\Phi = \forall x\, \forall y\, \exists a.\ (x, y, a)_1\ (\overline{x}, \overline{y}, a)_2\ (x, \overline{y}, \overline{a})_3\ (\overline{x}, y, \overline{a})_4$. The Skolem function $\mathsf{s}_a = (x \leftrightarrow y)$ shows that Φ is true. Note that $\mathsf{lev}(x) = \mathsf{lev}(y)$, but $\mathsf{idx}(x) < \mathsf{idx}(y)$. If the index restriction is neglected, then the following unsound proof Π can be built.

$$\Pi = \left\{ \begin{array}{l} 1.\ clause_5 = \mathsf{resolve}_{\forall\mathsf{L}}(clause_1, x, clause_2) = (y^*, a) \\ 2.\ clause_6 = \mathsf{resolve}_{\forall\mathsf{L}}(clause_3, y, clause_4) = (x^*, \overline{a}) \\ 3.\ clause_7 = \mathsf{resolve}_{\exists}(clause_5, a, clause_6) = (x^*, y^*) \\ 4.\ clause_\emptyset = \mathsf{reduce}(clause_7) = \Box \end{array} \right\}$$

Note that the index restriction on x and y would disallow $\mathsf{resolve}_{\forall\mathsf{L}}$ step 2.

Table 1 compares the five proof systems discussed in this section by listing their derivation rules. In each line, the derivation rules for each proof system are marked by “x”. All proof systems are sound and refutationally complete for QBF. The completeness of LQU-resolution and LQU+-resolution follows from the completeness of Q-resolution. The soundness of LQU-resolution and LQU+-resolution is an extension of Theorem 4 in [1] and can be proved similarly. We extend the definition of a {Q,QU,LQ}-resolution proof Π to {LQU,LQU+}-resolution by adding the corresponding derivation rules.

Table 1. Summary of Proof System Rules.

	reduce	resolve$_\exists$	resolve$_\forall$	resolve$_{\exists\mathsf{L}}$	resolve$_{\forall\mathsf{L}}$
Q-resolution [11]	x	x			
QU-resolution[17]	x	x	x		
LQ-resolution[1]	x	x		x	
LQU-resolution	x	x	x	x	
LQU+-resolution	x	x	x	x	x

3.3 Superiority and Limitation of LQU- and LQU+-resolutions

KBKF-qu and KBKF-lq families can be combined into a family KBKF-lqu that has exponential smallest proofs in both QU-resolution and LQ-resolution, but polynomial proofs in LQU-resolution. Proposition 2 and Theorem 3 below demonstrate bounds on the shortest proofs for combinations of QBF formulas.

Proposition 2 (cf. [6], Proposition 5). *Given a false QBF $\Phi = \mathcal{P}.\phi$, a literal $e \in V_\exists$, an LQU-resolution proof Π for Φ, both $\Phi_{\lceil\{e\}}$ and $\Phi_{\lceil\{\overline{e}\}}$ are false QBFs and Π can be modified in polynomial time with respect to its size to obtain a new proof $\Pi_{\lceil e}$ (respectively $\Pi_{\lceil\overline{e}}$) deriving $\Box$ from $\Phi_{\lceil\{e\}}$ (respectively $\Phi_{\lceil\{\overline{e}\}}$).*

This proposition extends Proposition 5 in [6] by allowing e to have an arbitrary quantifier level and allowing the proof to contain resolve$_\forall$ steps. The extension is sound, since the proof in [6] is independent of the quantifier levels of existential variables and can also be incorporated with resolve$_\forall$ and resolve$_{\exists\mathsf{L}}$ rules. [1] The same result for Q-resolution has been proposed in [8]. By Proposition 2 also $\Phi_{\lceil\sigma}$ and $\Pi_{\lceil\sigma}$ for any assignment σ to existential variables of Φ can be constructed.

Theorem 3. *Given two disjoint sets V_1 and V_2 of variables, let $\Phi_1 = \mathcal{P}_1.\phi_1$ and $\Phi_2 = \mathcal{P}_2.\phi_2$ be two false QBFs over V_1 and V_2, respectively. Let Π_1 and Π_2 be their respective shortest LQU-resolution proofs. Let $\Phi = \exists a\mathcal{P}_1\mathcal{P}_2.(\phi_1 \vee a) \wedge (\phi_2 \vee \overline{a})$, where $a \notin V_1 \cup V_2$, and for $i \in \{1,2\}$, $(\phi_i \vee a)$ stands for $\{C \cup \{a\} \mid C \in \phi_i\}$. Then Φ is false and the size of its shortest LQU-refutation is $|\Pi_1| + |\Pi_2| + 1$.*

Proof. By following the resolution steps of Π_1 on the clauses of $(\phi_1 \vee a)$ we retrieve the clause (a), by following the resolution steps of Π_2, on $(\phi_2 \vee \overline{a})$ we retrieve $(\overline{a})$, and resolving the two unit clauses results in $\Box$. Thus an LQU-resolution proof Π with $|\Pi| = |\Pi| = |\Pi_1| + |\Pi_2| + 1$ is constructed. Let Π be any LQU-resolution proof for Φ and let $\Pi_{\lceil\{a\}}$ be the proof generated for $\Phi_{\lceil\{a\}}$ as by Proposition 2 and let n_1 (resp. n_2) be the number of resolution steps in Π under any pivot $p \in V_1$ (resp. any pivot $p \in V_2$). By construction, $\Phi_{\lceil\{a\}} = \Phi_2$ and therefore $n_2 \geq |\Pi_{\lceil\{a\}}| \geq |\Pi_2|$. The dual case holds for $\Pi_{\lceil\{\overline{a}\}}$, resulting in $n_1 \geq |\Pi_{\lceil\{\overline{a}\}}| \geq |\Pi_1|$. Finally, in a derivation of $\Box$ from Φ, there must be at least one resolve$_\exists$ with pivot variable a. As $V_1 \cap V_2 = \emptyset$ and $a \notin V_1 \cup V_2$ we conclude $|\Pi| \geq |\Pi_1| + |\Pi_2| + 1$. Note that if V_1 and V_2 are not disjoint, then in similar way we can only prove a weaker bound $|\Pi| \geq max(|\Pi_1|, |\Pi_2|) + 1$. $\Box$

[1] The proof is found in the Appendix of [6], available in the online version of the paper at `http://www.kr.tuwien.ac.at/staff/widl/publications/2013/lpar13.pdf`

Definition 4 (KBKF-lqu family). *For $t > 1$, let $\mathcal{P}^q.\phi^q$ be the t^{th} member of the KBKF-qu family over variable set V^q, and let $\mathcal{P}^l.\phi^l$ be the t^{th} member of the KBKF-lq family over variable set V^l, where $V^q \cap V^l = \emptyset$. Let a be a fresh variable with $a \notin V^q \cup V^l$. The t^{th} member KBKF-lqu[t] in the KBKF-lqu family is defined as $\exists a \mathcal{P}^q \mathcal{P}^l.(\phi^q \vee a) \wedge (\phi^l \vee \overline{a})$.*

Corollary 2. *For $t > 1$, the smallest proofs for KBKF-lqu[t] are polynomial for LQU-resolution, but are exponential for LQ-resolution and exponential for QU-resolution (based on Theorem 3.2 in [11]).*

Whether the LQU+-resolution calculus has an exponential separation with respect to LQU remains an open problem. The following example, however, shows how LQU+-resolution can be more beneficial than LQU-resolution in some cases.

Ex. 3. Consider the false QBF $\Phi = \exists a \forall x \forall y \exists b.(a, x, b)_1(\overline{a}, \overline{x}, b)_2(x, y, \overline{b})_3(\overline{x}, \overline{y}, \overline{b})_4$.

Notice that Φ is similar as in Ex. 1, that it has Herbrand functions $\mathsf{h}_y = \mathsf{h}_x = a$, and that an LQU-resolution proof of the falsity of Φ cannot contain any of the steps $\mathsf{resolve}_{\forall}$ and $\mathsf{resolve}_{\exists \mathsf{L}}$, relevant to the derivation of an empty clause. There exists, however, an LQU+-resolution proof Π, which contains a $\mathsf{resolve}_{\forall \mathsf{L}}$ step.

$$\Pi = \left\{ \begin{array}{l} 1.\ clause_5 = \mathsf{resolve}_{\exists\mathsf{L}}(clause_1, a, clause_2) = (x^*, b) \\ 2.\ clause_6 = \mathsf{resolve}_{\forall\mathsf{L}}(clause_3, x, clause_4) = (y^*, \overline{b}) \\ 3.\ clause_7 = \mathsf{resolve}_{\exists}(clause_5, b, clause_6) = (x^*, y^*) \\ 4.\ clause_{\emptyset} = \mathsf{reduce}(clause_7) = \Box \end{array} \right\}$$

4 Certificate Extraction

In this section we examine existing methods for countermodel construction from {Q,LQ}-resolution proofs and extend them for {QU,LQU}-resolution proofs. All the discussions can be dually extended to cube resolution proofs for true QBFs as proposed in [1]. The Algorithm `Countermodel_construct` [1] was proposed to extract Herbrand functions from Q-resolution proofs. We show in the following proposition that this algorithm is also sound for QU-resolution proofs.

Proposition 3. *For a false QBF Φ and a corresponding QU-resolution proof Π, algorithm `Countermodel_construct` of [1] returns a correct countermodel for Φ.*

Proof. Theorem 3 of [1] shows the correctness of `Countermodel_construct` for Q-resolution proofs. Since the way it is proved is not affected by the presence of $\mathsf{resolve}_{\forall}$ steps, it is also sound for QU-resolution proofs. □

Note that the algorithm `Countermodel_construct` applied to QU-refutations of KBKF[t] proposed in [18] returns countermodel $\mathcal{H} = \bigcup_{i \in [1..t]} \mathsf{h}_{xi}$ with $\mathsf{h}_{xi} = d_i \wedge \overline{e_i}$, since for each $i \in [1..t]$ the literal $\mathsf{lit}(x_i)$ is universally reduced only twice in the whole proof, namely in clauses (d_i, x_i) and $(e_i, \overline{x}_i)$. It is also worth noticing that KBKF[t] has even simpler Herbrand functions than those constructed by

```
*LQU_countermodel_construct*
  **input**: a false QBF $\Phi$ and its LQ-resolution proof $\Pi$
  **output**: Herbrand model $\mathcal{H}$ for $\Phi$
  **begin**
      **let** $\Sigma$ the set of all assignments to variables $V_{P_\Pi}$
      **foreach** assignment $\sigma \in \Sigma$
          $(\Phi^\sigma, \Pi^\sigma) :=$ unmerge$(\Phi, \Pi, \sigma)$;
          $\mathcal{H}^\sigma :=$ Countermodel_construct$(\Phi^\sigma, \Pi^\sigma)$;
      $\mathcal{H} := \{\mathsf{h}_x \mid \mathsf{h}_x = \big(\bigvee_{\sigma\in\Sigma}(\mathsf{h}_x^\sigma \wedge \mathsf{cond}(\sigma))\big)$ for $\mathsf{h}_x^\sigma \in \mathcal{H}^\sigma\}$;
      **return** $\mathcal{H}$;
  **end**
```

Fig. 1. Algorithm: LQU Countermodel Construction.

Countermodel_construct, namely $\mathsf{h}_{xi} = d_i$ for all $i \in [1..t]$. The existence of these simple functions motivates to further investigate the (counter)model extraction from proofs of different resolution systems.

In contrast to QU-resolution proofs, the algorithm Countermodel_construct is unsound for LQ-resolution proofs due to the possible presence of resolve$_{\exists\mathsf{L}}$ steps. A conversion of an LQ-resolution proof into a Q-resolution proof in order to apply Countermodel_construct has been proposed [1], but it can result in an exponential blow-up. We propose an algorithm to extract Herbrand functions for a false QBF directly from its LQ-resolution proofs. The algorithm is outlined in Fig. 1. By Proposition 3 it applies LQU-resolution proofs as well. The procedure unmerge(Φ, Π, σ) is central to the algorithm. It transforms an LQU-resolution proof Π into a QU-refutation as follows. Let $V_{P_\Pi} \subseteq V_\exists$ be the exact set of the pivot variables in the resolve$_{\exists\mathsf{L}}$ steps of Π. Given an LQU-resolution proof Π for a false QBF $\Phi = \mathcal{P}.\phi$ and an assignment σ to a set V_σ of variables of Φ with $V_{P_\Pi} \subseteq V_\sigma \subseteq V_\exists$, unmerge$(\Phi, \Pi, \sigma)$ traverses Π in a topological order. Whenever it encounters two clauses $C_a = C_1 \cup \{l, p\}$ and $C_b = C_2 \cup \{\bar{l}, \bar{p}\}$ resolving into $C = \text{resolve}_{\exists\mathsf{L}}(C_a, p, C_b) = C_1 \cup C_2 \cup \{l^*\}$, it applies the following rewriting rule. Two cases are distinguished by the polarity of the pivot's literal in σ.

$$\frac{(C_1 \cup \{l,p\}) \quad (C_2 \cup \{\bar{l},\bar{p}\})}{(C_1 \cup C_2 \cup \{l^*\})} \xrightarrow{p\in\sigma} \frac{\frac{(C_1\cup\{l,p\}) \quad (C_1\cup\{\bar{l},p\})}{(C_1\cup\{p\})} \quad (C_2 \cup \{\bar{l},\bar{p}\})}{(C_1 \cup C_2 \cup \{\bar{l}\})}$$

$$\frac{(C_1 \cup \{l,p\}) \quad (C_2 \cup \{\bar{l},\bar{p}\})}{(C_1 \cup C_2 \cup \{l^*\})} \xrightarrow{\bar{p}\in\sigma} \frac{(C_1 \cup \{l,p\}) \quad \frac{(C_2\cup\{\bar{l},\bar{p}\}) \quad (C_2\cup\{l,\bar{p}\})}{(C_2\cup\{\bar{p}\})}}{(C_1 \cup C_2 \cup \{l\})}$$

If there are more than one merged literals in C, unmerge is applied several times to eliminate all of them. Intuitively, this procedure adds clauses to ϕ in order to substitute all resolve$_{\exists\mathsf{L}}$ steps. It preserves the order of reduce and does not create any new resolve$_{\exists\mathsf{L}}$ steps. It never encounters resolve$_{\exists\mathsf{L}}$ on two clauses containing merged literals because these literals are removed by the rewriting rule in an earlier iteration. We denote the QBF resulting from unmerge(Φ, Π, σ) by Φ^σ, and the resulting (QU-resolution) proof by Π^σ.

Given a Herbrand model $\mathcal{H}$ and an assignment σ, the Herbrand model $\mathcal{H}\lceil_\sigma$ results from replacing each variable v in $\mathcal{H}$ by $\top$ if $v \in \sigma$ and by $\bot$ if $\overline{v} \in \sigma$. The following two observations establish the connection between Φ^σ and $\Phi\lceil_\sigma$.

Observation 2. Let $\mathcal{H}$ be a set of Herbrand functions for a false QBF Φ, and σ be an assignment to some existential variables of Φ. Then $\mathcal{H}\lceil_\sigma$ is a set of Herbrand functions for the false QBF $\Phi\lceil_\sigma$.

Observation 3. For any assignment σ to variables in V_{P_Π}, it holds that $\Pi\lceil_\sigma = (\Pi^\sigma)\lceil_\sigma$. By Observation 2, if $\mathcal{H}$ is a set of Herbrand functions for Φ^σ, then $\mathcal{H}\lceil_\sigma$ is a set of Herbrand functions for $\Phi\lceil_\sigma$.

The algorithm LQU_countermodel_construct takes a false QBF Φ and an LQU-resolution proof Π of Φ as input. It then collects the pivots of all resolve$_{\exists L}$ steps in Π into the set V_{P_Π} and iteratively picks an assignment σ to the variables in V_{P_Π}. For each assignment, a QU-resolution proof is constructed by unmerge in Line 02. Note that unmerge was defined for any set of existential variables containing V_{P_Π}. It however suffices to consider the assignments to V_{P_Π} only. In Line 03, Countermodel_construct is applied to extract parts of the countermodel for Φ, which are then put together in Line 04. Note that the Herbrand function F_x returned by the algorithm LQU_countermodel_construct for a universal variable x permits its dependency on the universal variables x' with $lvl(x') < lvl(x)$. All occurrences of such x' in F_x should be substituted by the corresponding Herbrand functions $F_{x'}$, resulting into the function that depends only on existential variables.

Theorem 4 below states the soundness of LQU_countermodel_construct. Note that from this theorem also follows the soundness of LQU-resolution.

Theorem 4. *Given a false QBF $\Phi = \mathcal{P}.\phi$ and an LQU-resolution proof Π for Φ, LQU_countermodel_construct returns correct Herbrand functions for Φ.*

Proof. Consider any assignment σ to V_{P_Π} variables. By construction, $\mathcal{H}\lceil_\sigma = \mathcal{H}^\sigma$. Taking in account Observation 3, the Herbrand functions $\mathcal{H}\lceil_\sigma$ falsify the formula $\Phi\lceil_\sigma$. Thus $\mathcal{H}$ falsifies ϕ under any assignment to existential variables.

It remains to show that for each $x \in V_\forall$, its Herbrand function $\mathsf{h}_x \in \mathcal{H}$ respects the variable ordering of $\mathcal{P}$. (As constructed, h_x includes all variables in σ due to the assignment condition cond(σ).) Notice that under a given σ, the constructed h_x^σ is uniquely defined by the ordered set of clauses in Π^σ from which x is removed by universal reduction. By construction, Π^σ has exactly the same universal reduction steps as Π, with the only difference that every literal l^* is replaced by l or $\overline{l}$, depending on σ. For two assignments σ_1 and σ_2, compare clauses $C^{\sigma_1} \in \Pi^{\sigma_1}$ and $C^{\sigma_2} \in \Pi^{\sigma_2}$ that correspond to clause $C \in \Pi$ and result from universal reduction on x. If $(l \in \sigma_1) \wedge (\mathsf{lev}(l) < \mathsf{lev}(x))$ implies $l \in \sigma_2$ for any literal l, then by the definition of unmerge we conclude that $C^{\sigma_1} = C^{\sigma_2}$ and that x was universally reduced as the same literal to get both C^{σ_1} and C^{σ_2}. Thus h_x is independent of any variable in $\{v \in V_{P_\Pi} \mid \mathsf{lev}(v) > \mathsf{lev}(x)\}$. □

Table 2. Time and Memory Statistics for KBKF Family of QBF Instances.

t	DepQBF		ResQu			DepQBF-lq		ResQu-lqu		
	time	$\|\Pi\|$	time	memory	verify	time	$\|\Pi\|$	time	memory	verify
2	0	24	0	1	0	0	27	0	1	0
3	0	50	0	1	0	0	43	0	1	0
4	0	106	0	1	0.1	0	59	0	1	0.1
5	0	230	0	1	0.1	0	75	0	1	0.1
6	0	506	0	1	0.1	0	91	0	1	0.1
7	0	1.1k	0	1	0.1	0	107	0	1	0.1
8	0	2.5k	0	2	0.1	0	123	0	2	0.1
9	0	5.4k	0	3	0.1	0	139	0	2	0.1
10	0.1	11.8k	0.1	7	0.1	0	155	0	4	0.1
11	0.2	25.6k	0.1	14	0.3	0	171	0.1	8	0.1
12	0.5	55.3k	0.3	58	0.7	0	187	0.1	18	0.1
13	1.2	118.8k	0.6	123	2.3	0	203	0.3	37	0.1
14	2.8	254.0k	1.4	261	7.6	0	219	0.7	79	0.1
15	6.8	540.7k	3.0	550	30.5	0	235	1.8	169	0.1
16	16.6	1.15M	6.7	1.2G	-1	0	251	3.9	360	0.8
17	41.0	2.42M	15.1	2.4G	-1	0	267	9.4	767	5.4
18	102.8	5.11M	33.6	5.1G	-1	0	283	20.5	1.6G	40.4
19	261.5	10.75M	74.1	10.7G	-1	0	299	48.8	3.4G	-1
20	674.2	22.54M	175.7	22.5G	-1	0	315	95.1	7.2G	-1

The time complexity of LQU_countermodel_construct is in the worst case exponential in the proof size. In practice, however, it can be more efficient than converting LQ-resolution proofs into Q-resolution proofs [1], as will be evident in Section 5. Note that the algorithm LQU_countermodel_construct is unsound for LQU+-resolution proofs due to the presence of universal variables in V_{P_Π}.

5 Experiments

In this section we evaluate the proposed algorithm LQU_countermodel_construct on members of the KBKF family. To the best of our knowledge, there is currently no tool available to construct QU-resolution proofs (and consequently LQU-resolution proofs). Hence we test LQU_countermodel_construct on LQ-resolution proofs, and compare the results to those obtained by Countermodel_construct [1] from the corresponding Q-resolution proofs. The experiments were conducted on a Linux machine with a Xeon 2.3 GHz CPU and 32 GB RAM.

Table 2 summarizes time and memory statistics for solving, extracting, and verifying Herbrand functions for members of the KBKF family up to $t = 20$. ResQu implements the algorithm Countermodel_construct, ResQu-lqu implements LQU_countermodel_construct, DepQBF stands for the solver proposed in [13], and DepQBF-lq for its extension by LQ-resolution [6]. The column “time” refers to the runtime in seconds, “$|\Pi|$” to the size of the resulting proof, “memory” to the maximal memory consumption (in MB for unit unspecified entries), and “verify” to the time needed by the SAT-solver MiniSAT [5] embedded in ABC [3] to verify the certificate where “-1” stands for a timeout with a limit of 1,000s.

The superiority of LQ-resolution compared to Q-resolution is evident in all aspects. Since Q-resolution proofs produced by DepQBF are exponential in t. ResQu also requires resources exponential in t. On the other hand, LQ-resolution proofs produced by DepQBF-lq are linear in t. Despite its exponential worst-case behavior, ResQu-lqu considerably outperforms ResQu in both time and memory consumption, although it still requires exponential resources due to the exponential size of the constructed Herbrand functions.

6 Conclusions and Future Work

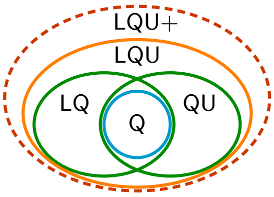

Fig. 2. Relations among the Proof Systems.

We have presented results related to both theoretical and practical aspects of QBF evaluation. On the theoretical side, we have shown the incomparability between two proof systems, QU-resolution and LQ-resolution, from literature. Additionally, we have proposed two new extended proof systems, LQU-resolution and LQU+-resolution, and have shown the two new systems to be exponentially stronger than both of the above. It remains open whether an exponential gap exists between the proof complexities of LQU-resolution and LQU+-resolution. Fig. 2 summarizes our results on the relations between the discussed proof systems. Since modern QBF solvers heavily rely on resolution techniques, we expect our theoretical results to inspire future work in the area of QBF solving.

On the practical side, we have designed a new algorithm to extract Herbrand certificates from LQU-resolution proofs. An implementation and experimental evaluation underline its practical applicability and advantage over the certificates from Q-resolution. For future work, a polynomial time algorithm for certificate extraction from LQU-resolution proofs would be very desirable.

References

1. V. Balabanov and J.-H. R. Jiang. Unified QBF Certification and Its Applications. *Formal Methods in System Design*, 41:45–65, 2012.
2. M. Benedetti. sKizzo: A suite to evaluate and certify QBFs. In *International Conference on Automated Deduction (CADE)*, pages 369–376. Springer, 2005.
3. Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification. `http://http://www.eecs.berkeley.edu/~alanmi/abc/`.
4. N. Dershowitz, Z. Hanna, and J. Katz. Bounded Model Checking with QBF. In *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 3569 of *LNCS*, pages 408–414. Springer, 2005.
5. N. Eén and N. Sörensson. An Extensible SAT-Solver. In *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 2919 of *LNCS*, pages 502–518. Springer, 2003.
6. U. Egly, F. Lonsing, and M. Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *International Conference on Logic Programming and Automated Reasoning (LPAR)*, pages 291–308. Springer, 2013.
7. E. Giunchiglia, M. Narizzano, and A. Tacchella. QuBE++: An Efficient QBF Solver. In *Formal Methods in Computer-Aided Design (FMCAD)*, pages 201–213, 2004.

8. A. Goultiaeva, A. Van Gelder, and F. Bacchus. A Uniform Approach for Generating Proofs and Strategies for Both True and False QBF Formulas. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 546–553. AAAI Press, 2011.
9. M. Janota, W. Klieber, J. Marques-Silva, and E. Clarke. Solving QBF with Counterexample Guided Refinement. In *International conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 7317 of *LNCS*, pages 114–128. Springer, 2012.
10. J.-H. R. Jiang, H.-P. Lin, and W.-L. Hung. Interpolating Functions from Large Boolean Relations. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pages 779–784. IEEE/ACM, 2009.
11. H. Kleine Büning, M. Karpinski, and A. Flögel. Resolution for Quantified Boolean Formulas. *Information and Computation*, 117(1):12–18, Feb. 1995.
12. C.-F. Lai, J.-H. R. Jiang, and K.-H. Wang. BooM: A Decision Procedure for Boolean Matching with Abstraction and Dynamic Learning. In *Design Automation Conference (DAC)*, pages 499–504. ACM/IEEE, 2010.
13. F. Lonsing and A. Biere. DepQBF: A Dependency-Aware QBF Solver (System Description). *Journal on Satisfiability, Boolean Modeling and Computation*, 7:71–76, 2010.
14. J. P. Marques Silva, I. Lynce, and S. Malik. Conflict-Driven Clause Learning SAT Solvers. In *Handbook of Satisfiability*, pages 131–153. IOS Press, 2009.
15. J. Rintanen. Asymptotically Optimal Encodings of Conformant Planning in QBF. In *National Conference on Artificial Intelligence (AAAI)*, pages 1045–1050. AAAI Press, 2007.
16. S. Staber and R. Bloem. Fault localization and correction with QBF. In *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 355–368, 2007.
17. A. Van Gelder. Input Distance and Lower Bounds for Propositional Resolution Proof Length. In *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 3569 of *LNCS*, pages 282–293. Springer, 2005.
18. A. Van Gelder. Contributions to the Theory of Practical Quantified Boolean Formula Solving. In *International Conference on Principles and Practice of Constraint Programming (CP)*, volume 7514 of *LNCS*, pages 647–663. Springer, 2012.
19. L. Zhang and S. Malik. Conflict Driven Learning in a Quantified Boolean Satisfiability Solver. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pages 442–449. ACM, 2002.
20. L. Zhang and S. Malik. The Quest for Efficient Boolean Satisfiability Solvers. In *International Conference on Computer Aided Verification (CAV)*, volume 2404 of *LNCS*, pages 17–36. Springer, 2002.