On Propositional QBF Expansions and Q-Resolution

Mikoláš Janota and Joao Marques-Silva

[1] IST/INESC-ID, Lisbon, Portugal
[2] University College Dublin, Ireland

Abstract. Over the years, proof systems for propositional satisfiability (SAT) have been extensively studied. Recently, proof systems for quantified Boolean formulas (QBFs) have also been gaining attention. Q-resolution is a calculus enabling producing proofs from DPLL-based QBF solvers. While DPLL has become a dominating technique for SAT, QBF has been tackled by other complementary and competitive approaches. One of these approaches is based on expanding variables until the formula contains only one type of quantifier; upon which a SAT solver is invoked. This approach motivates the theoretical analysis carried out in this paper. We focus on a two phase proof system, which expands the formula in the first phase and applies propositional resolution in the second. Fragments of this proof system are defined and compared to Q-resolution.

This paper follows the line of research on proof systems for propositional and quantified Boolean formulas (QBFs). This research is motivated by complexity theory and more recently by the objective to develop and certify QBF solvers [11,18,8,14]. Proof systems for QBF come in different styles and flavors. Krajíček and Pudlák propose a Genzen-style calculus *KP* for QBF [18]. Büning et al. propose a refutation calculus *Q-resolution* [8], an extension of propositional resolution. Giunchiglia et al. extend the work of Büning et al. into *term resolution* for proofs of true formulas [14] . Certain separation results were shown between KP and Q-resolution recently by Egly [12].

While many QBF solvers are based on the DPLL procedure [21,9,23,20,13], other solvers tackle the given formula by *expanding* out quantifiers until a single quantifier type is left. At that point, this formula is handed to a SAT solver [1,4,19,15]. Experimental results show that expansion-based QBF solvers can outperform DPLL-based solvers on a number of families of practical instances. Also, expansion can be used in QBF preprocessing [6,5].

This practical importance of expansion motivates the study carried out in this paper. We define a proof system $\forall$Exp+Res, which eliminates universal quantification from the given *false* formula and then applies propositional resolution to refute the remainder.

We show that $\forall$Exp+Res can p-simulate *tree* Q-resolution refutations. Conversely, we show that Q-resolution can p-simulate $\forall$Exp+Res refutations under certain restrictions on the propositional resolution part of the proofs.

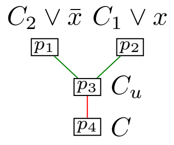
1 Preliminaries

A *literal* is a Boolean variable or its negation. The literal complementary to a literal l is denoted as $\bar{l}$, i.e. $\bar{x} = \neg x$, $\overline{\neg x} = x$. A *clause* is a disjunction of zero or more noncomplementary literals. A formula in *conjunctive normal form* (CNF) is a conjunction of clauses. Whenever convenient, a clause is treated as a set of literals and a CNF formula as a set of sets of literals. For a literal $l = x$ or $l = \bar{x}$, we write $\mathsf{var}(l)$ for x. For a clause C, we write $\mathsf{var}(C)$ to denote $\{\mathsf{var}(l) \mid l \in C\}$ and for a CNF ψ, $\mathsf{var}(C)$ denotes $\{l \mid l \in \mathsf{var}(C), C \in \psi\}$

Substitutions are denoted as $x_1/\psi_1, \ldots, x_n/\psi_n$, with $x_i \neq x_j$ for $i \neq j$. The set of variables $x_1, \ldots, x_n$ is called the *domain* of the substitution. An application of a substitution is denoted as $\phi[x_1/\psi_1, \ldots, x_n/\psi_n]$ meaning that variables x_i are simultaneously substituted with corresponding ψ_i in ϕ. A substitution is called an *assignment* iff each ψ_i is one of the constants 0, 1. An assignment is called *total*, or *complete*, for a set of variables $\mathcal{X}$ if each $x \in X$ is in the domain of the assignment. For substitutions $\tau_1 = x_1/\psi_1, \ldots, x_n/\psi_n$ and $\tau_2 = y_1/\xi_1, \ldots, y_m/\xi_m$ with distinct domains we write $\tau_1 \cup \tau_2$ for the substitution $x_1/\psi_1, \ldots, x_n/\psi_n, y_1/\xi_1, \ldots, y_m/\xi_m$.

Quantified Boolean Formulas (QBFs) [7] are an extension of propositional logic with quantifiers with the standard semantics that $\forall x.\, \Psi$ is satisfied by the same truth assignments as $\Psi[x/0] \wedge \Psi[x/1]$ and $\exists x.\, \Psi$ as $\Psi[x/0] \vee \Psi[x/1]$. Unless specified otherwise, we assume that QBFs are in *closed prenex* form with a CNF *matrix*, i.e. $\mathcal{Q}_1 X_1 \ldots \mathcal{Q}_k X_k.\ \phi$, where X_i are pairwise disjoint sets of variables; $\mathcal{Q}_i \in \{\exists, \forall\}$ and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$. The formula ϕ is in CNF and is defined only on variables $X_1 \cup \ldots \cup X_k$. The propositional part ϕ is called the *matrix* and the rest the *prefix*. If a variable x is in the set X_i, we say that x is at *level* i and write $\mathsf{lv}(x) = i$; we write $\mathsf{lv}(l)$ for $\mathsf{lv}(\mathsf{var}(l))$. A closed QBF is *false* (resp. *true*), iff it is semantically equivalent to the constant 0 (resp. 1).

For a clause C, a universal literal $l \in C$ is *blocked* by an existential literal $k \in C$ iff $\mathsf{lv}(l) < \mathsf{lv}(k)$. $\forall$-*reduction* is the operation of removing from a clause C all universal literals that are *not* blocked by some literal. For two $\forall$-reduced clauses $x \vee C_1$ and $\bar{x} \vee C_2$, where x is an existential variable, a *Q-resolvent* [8] is obtained in two steps. (1) Compute $C_u = C_1 \cup C_2 \smallsetminus \{x, \bar{x}\}$. If C_u contains complementary literals, the $Q-resolvent$ is undefined. (2) $\forall$-reduce C_u. For a QBF $\mathcal{P}\,.\phi$, a A *Q-resolution proof* of a clause C is a sequence of clauses $C_1, \ldots, C_n$ where $C_n = C$ and any C_i in the sequence is part of the given matrix ϕ or it is a Q-resolvent for some pair of the preceding clauses. A Q-resolution proof is called a *refutation* iff C is the empty clause, denoted $\bot$.



In this paper Q-resolution proofs treated as connected directed acyclic graphs so that the each clause in the proof corresponds to some node p_n labeled with that clause. We assume that the input clauses are already $\forall$-reduced. Q-resolution steps are depicted as on the right. Note that $\forall$-reduction corresponds to a separate node. A proof system P_1 *p-simulates* a proof system P_2 iff any proof in P_2 of a formula Φ can be translated into a proof in P_1 of Φ in polynomial time (c.f. [11,22]).

2 Expansions

Modern SAT solvers can be easily used in a black box setting which suggests a straightforward approach to solving QBF by expanding variables until only one type of quantifier is left; at that point a SAT solver can be invoked. Here we are assuming the mainstream type of a SAT solver that accepts formula in CNF and produces resolution proofs for unsatisfiable inputs.

Existential quantification can be expanded by the equivalence $\exists x.\, \Phi = \Phi[x/0] \vee \Phi[x/1]$ and universal quantification by the equivalence $\forall x.\, \Phi = \Phi[x/0] \wedge \Phi[x/1]$. These equivalences reveal two main obstacles to developing a calculus using both expansion and plain resolution (besides the exponential growth). The first obstacle is that the result of an expansion is not in prenex form; this can be overcome by prenexing the expansion. The second obstacle is that the result of expanding the existential quantifier does not yield CNF. Hence, in this paper we focus only on expansion of the universal quantifier. We show that this limitation still leads to a refutation complete calculus with many interesting properties.

Expansion of universal quantifiers enables decreasing the number of quantifiers and maintain prenex normal form at the cost of introducing fresh variables. For instance, expanding $\exists x \forall y \exists z.\, \phi$ yields $\exists x.\, (\exists z.\, \phi[y/0]) \wedge (\exists z.\, \phi[y/1])$. To get back to prenex form, we add two fresh copies of z, one for the sub-QBF where $y = 0$ and one for the sub-QBF where $y = 1$, thus obtaining $\exists x z^0 z^1.\, \phi[y/0, z/z^0] \wedge \phi[y/1, z/z^1]$.

A significant drawback of expansion is that the formula grows in size exponentially. This effect can be mitigated by observing that only *partial expansions* may be sufficient to show unsatisfiability. For instance, for the formula $\forall y \exists x.\, (y \vee x) \wedge (y \vee \bar{x})$ it is sufficient to consider an expansion with $y/0$ to show the formula false. Another source of rapid growth lies in the number of the formula's quantification levels. Expanding y in $\exists x \forall y \exists z \forall u \exists w.\, \phi$ yields $\exists x.\, (\exists z \forall u \exists w.\, \phi[y/0]) \wedge (\exists z \forall u \exists w.\, \phi[y/1])$. We could again prenex all variables but since we are aiming at eventually expanding *all* universal variables, we can expand more carefully by prenexing only z: $\exists x z^0 z^1.\, \forall u \exists w.\, \phi[y/0, z/z^0] \wedge \forall u \exists w.\, \phi[y/1, z/z^1]$. Such expansion gives us a finer control over the expansion process (see [15, Sec. 3.1] for more detailed discussion). If for instance now we wish to expand u as 1 in the first sub-formula and 0 in the second sub-formula we obtain the following:

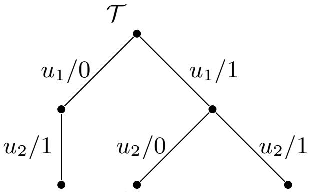
$$\exists x z^0 z^1 w^{01} w^{10}.\, \phi[y/0, z/z^0, u/1, w/w^{01}] \wedge \phi[y/1, z/z^1, u/0, w/w^{10}]$$

Consider a general QBF $\Phi = \forall \mathcal{U}_1 \exists \mathcal{E}_2 \ldots \forall \mathcal{U}_{2N-1} \exists \mathcal{E}_{2N} .\, \phi$ (WLOG we start with a universal quantifier to simplify notation). For succinctness reasons, from now on Φ refers to this formula. An expansion consists of expanding variables $\mathcal{U}_1$ with some values and introducing fresh variables for $\mathcal{E}_2$ variables yielding a sub-QBF for each considered assignment to the $\mathcal{U}_1$ variables. These sub-QBFs are recursively expanded in an analogous fashion. Note that if we expanded from the highest quantification level (innermost level), we would lose the structural information, which is enabling the abovementioned finer expansion steps. The following definitions formalize this process.

Definition 1 **($\forall$-expansion tree).** *A* $\forall$-expansion tree *is a rooted tree $\mathcal{T}$ such that each path $p_0 \xrightarrow{\tau_1} p_1 \ldots \xrightarrow{\tau_N} p_N$ in $\mathcal{T}$ from the root p_0 to some leaf p_N has exactly N edges and each edge $p_{i-1} \xrightarrow{\tau_i} p_i$ is labeled with a total assignment τ_i to the variables $\mathcal{U}_{2i-1}$, for $i \in 1..N$. Each path in $\mathcal{T}$ is uniquely determined by its labeling.*

Convention Since paths from the root in an $\forall$-expansion tree are uniquely determined by the labeling of the edges, i.e. assignments, we treat paths and the union of the appropriate assignments interchangeably.

Definition 2 **($\forall$-expansion).** *Let $\mathcal{T}$ be a $\forall$-expansion tree. For a root-to-leaf path P in $\mathcal{T}$ and a clause C, the following rules define $\forall$-expansion of C by P, $\forall$-expansion of ϕ*



(a) ∀-expansion tree

$\Psi = \forall u_1 \exists e_1 \forall u_2 \exists e_2 .\, \psi$	$\mathscr{E}(\mathcal{T}, \Psi)$
$u_1 \vee e_1 \vee \bar{u}_2 \vee e_2$	$e_1^{u_1/0} \vee e_2^{u_1/0, u_2/1}$
$u_1 \vee e_1 \vee \bar{u}_2 \vee \bar{e}_2$	$e_1^{u_1/0} \vee \bar{e}_2^{u_1/0, u_2/1}$
$u_1 \vee \bar{e}_1$	$\bar{e}_1^{u_1/0}$
$\bar{u}_1 \vee \bar{e}_1 \vee \bar{u}_2 \vee \bar{e}_2$	$\bar{e}_1^{u_1/1} \vee \bar{e}_2^{u_1/1, u_2/1}$
$\bar{u}_1 \vee \bar{e}_1 \vee \bar{u}_2 \vee e_2$	$\bar{e}_1^{u_1/1} \vee e_2^{u_1/1, u_2/1}$
$\bar{u}_1 \vee e_1 \vee u_2 \vee \bar{e}_2$	$e_1^{u_1/1} \vee \bar{e}_2^{u_1/1, u_2/0}$
$\bar{u}_1 \vee e_1 \vee u_2 \vee e_2$	$e_1^{u_1/1} \vee e_2^{u_1/1, u_2/0}$

(b) ∀-expansion

Fig. 1. Example expansion tree and its application

by P, and ∀-expansion of Φ by $\mathcal{T}$. These expansions are denoted as $\mathscr{E}(P, C)$, $\mathscr{E}(P, \psi)$, and $\mathscr{E}(\mathcal{T}, \Phi)$, respectively.

1. *For each path P_k in $\mathcal{T}$ from the root, labeled by assignments $\tau_1, \ldots, \tau_k$, and an existential variable x with $\mathsf{lv}(x) = 2k$ define a fresh variable $x^{\tau_1, \ldots, \tau_k}$.*
2. *For each path P in $\mathcal{T}$ from the root to some leaf labeled by $\tau_1, \ldots, \tau_N$, and a clause $C \in \phi$ define $\mathscr{E}(P, C)$ as $C[\tau_1 \cup \ldots \tau_N \cup \tau_R]$ where*

$$\tau_R = \{x/x^{\tau_1, \ldots, \tau_k} \mid 1 \leq k \leq N, x \textit{ an existential variable s.t. } \mathsf{lv}(x) = 2k\}$$

3. *For each path P in $\mathcal{T}$ from the root to some leaf define $\mathscr{E}(P, \phi)$ as a union of $\mathscr{E}(P, C)$ for $C \in \phi$.*
4. *Define $\mathscr{E}(\mathcal{T}, \Phi)$ as the union of all $\mathscr{E}(P, \phi)$ for each root-to-leaf path P in $\mathcal{T}$.*

Example 1. Figure 1(a) shows an example of a ∀-expansion tree and Figure 1(b) shows a ∀-expansion of some formula Ψ based on this tree. The expansion considers both values of u_1 but only the value 1 is considered for u_2 when $u_1 = 0$. The tree has 3 leafs so the formula could potentially grow 3 times. But because the formula is very simple, for each clause C there is only a single path P from the root to some leaf for which $\mathscr{E}(P, C) \neq 1$. Hence, the expansion has the same size as the original formula. Note that there are as many copies of e_2 as there are leafs in the expansion tree ($e_2^{u_1/0, u_2/1}$, $e_2^{u_1/1, u_2/0}$, $e_2^{u_1/1, u_2/1}$) but only two copies of e_1 ($e_1^{u_1/0}$,$e_1^{u_1/1}$).

Definition 3 **(∀Exp+Res).** ∀Exp+Res refutation *for Φ is a pair $(\mathcal{T}, \pi)$ where $\mathcal{T}$ is a ∀-expansion tree for Φ and π is a resolution refutation for $\mathscr{E}(\mathcal{T}, \Phi)$. A* size *of $(\mathcal{T}, \pi)$, denoted $|(\mathcal{T}, \pi)|$, is the sum of the numbers of nodes in $\mathcal{T}$ and π.*

Note that for a ∀-expansion $\mathcal{T}$ the size of $\mathscr{E}(\mathcal{T}, \Phi)$ is bounded by the number of leafs of $\mathcal{T}$ times the size of the matrix ϕ. Therefore a ∀Exp+Res refutation can be validated in polynomial time.

Theorem 1. *A formula Φ is false iff there exists a ∀Exp+Res refutation for Φ.*

Proof. If Φ is false, consider $\mathcal{T}_{\text{full}}$ capturing a full expansion of all of the quantifiers. More precisely, each node p_i of $\mathcal{T}_{\text{full}}$ at depth i (with the root being at depth 0) has $2^{|\mathcal{U}_{2i+1}|}$ children, each corresponding to a total assignment to variables $\mathcal{U}_{2i+1}$. Since this expansion mirrors semantics of QBF, $\mathscr{E}(\mathcal{T}_{\text{full}}, \Phi)$ is false iff Φ is false.

Throughout the $\forall$-expansion process, (sub-)QBFs $\forall\mathcal{U} \,.\, \Psi$ are replaced with the conjuncts $\Xi = \bigwedge_{\tau\in\omega} \Psi[\tau]$ for some ω, a set of total assignments to $\mathcal{U}$. Since Ξ is equivalent to $\forall\mathcal{U} \,.\, \Psi$ when ω is the set of *all* assignments, it is weaker if ω is a set of only some total assignments, i.e. $(\forall\mathcal{U} \,.\, \Psi) \to \Xi$. Consequently $\Phi \to \mathscr{E}(\mathcal{T}, \Phi)$ for any $\forall$-expansion tree $\mathcal{T}$. Therefore, if $\mathscr{E}(\mathcal{T}, \Phi)$ is false, then Φ is false. $\square$

3 Simulating Tree Q-resolution by ∀Exp+Res

Consider a tree Q-resolution refutation π of Φ. Our objective is to construct a ∀Exp+Res refutation $(\mathcal{T}, \pi')$ based on π. We should stress that DPLL-based solvers enable producing non-tree Q-resolution proofs due to learning [23]. Hence, this proof is *not* a proof of the fact ∀Exp+Res can simulate DPLL-based solving in general.

We will construct $\mathcal{T}$ and π' so that π' will share its basic structure with π but with universal variables removed and existential variables renamed (according to the definition of $\mathscr{E}$). We observe that if π consists of a single node $\bot$, $\mathcal{T}$ and π' are easily constructed by setting $\mathcal{T}$ to the empty tree and setting π'. Therefore, from now on, we assume that all leafs of π are labeled with nonempty clauses. For the sake of succinctness, within this section, π always refers to the given Q-resolution proof that we wish to translate to a ∀Exp+Res refutation.

We first observe that if two clauses $x \vee C_1$ and $\bar{x} \vee C_2$ are resolved in π, the $\forall$-expansion tree being constructed must ensure that x is substituted by the same fresh variable x' in both clauses so that the same resolution step can be carried out in π' on variable x'. The literals x and $\bar{x}$ can appear inside the Q-resolution tree π only if they were introduced by some of its leafs. Consequently, the corresponding leafs of the resolution tree π' must contain the same copy of x. This observation motivates the construction. In the first phase of the construction, we identify sets of leafs of π where a certain existential variable must be substituted by the same fresh copy. In the second phase we construct a $\forall$-expansion tree $\mathcal{T}$ that will respect the sets identified in the first phase. The $\forall$-expansion tree $\mathcal{T}$ will provide us with the leafs of π'.

Consider a resolution step in π on some variable x corresponding to nodes p_1 and p_2 with the resolvent (parent) node r. Let C_1, C_2, and C_r be the clauses labeling p_1, p_2, and r, respectively. Hence, $C_r = C_1 \cup C_2 \smallsetminus \{x, \bar{x}\}$ (recall that $\forall$-reduction is modeled as a separate step). Let D be the set of universal literals $l \in C_1 \cup C_2$ such that $\mathsf{lv}(l) < \mathsf{lv}(x)$. Let S be the set of leafs p of π such that there is a path from either p_1 or p_2 to p for which all clauses on the path contain the variable x (including the clause labeling p). Record the quadruple (r, x, D, S). In the following text we write $\mathcal{Q}_\pi$ to denote the set of quadruples generated for each resolution step in π.

Consider any two leafs p_1,p_2 of π s.t. $p_1, p_2 \in S$ for some $(r, x, D, S) \in \mathcal{Q}_\pi$. Once we ensure that x is replaced with the same fresh copy in the clauses labeling p_1 and p_2, the plain resolution refutation π' is easy to construct.

Proposition 1. *Let $\mathcal{T}$ be a $\forall$-expansion tree of Φ and let M be a total mapping from the leafs of π to paths of $\mathcal{T}$. If the following conditions $\mathscr{C}_1$–$\mathscr{C}_3$ hold for $\mathcal{T}$ and M, then there is a resolution refutation π' of $\mathscr{E}(\mathcal{T}, \Phi)$ linear in size of π.*

($\mathscr{C}_1$) *If p is a leaf of π, then $M(p)$ is a path from the root to some leaf in $\mathcal{T}$.*

($\mathscr{C}_2$) *If p is a leaf of π, labeled by a clause C, and $M(p) = P$, then P assigns to 0 all universal literals of C.*

($\mathscr{C}_3$) *If leafs p_1, p_2 of π appear in the same S for some quadruple $(r, x, D, S) \in \mathcal{Q}_\pi$, $M(p_1) = P_1$, and $M(p_2) = P_2$, then P_1 and P_2 assign the same value to all universal variables with level $l < \mathsf{lv}(x)$.*

Proof. We construct π' from π in the leaf-to-root direction; during this construction we mark each node of p' in π' as *corresponding* with some node p in π. The construction follows the following rules $\mathscr{R}_l$, $\mathscr{R}_r$, $\mathscr{R}_u$.

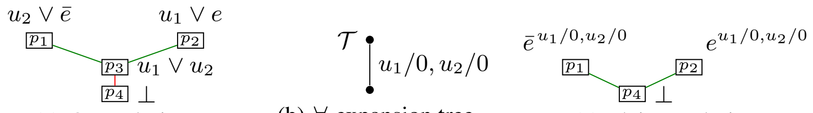
($\mathscr{R}_l$) For each leaf p in π labeled with C create a leaf $p' \in \pi'$ labeled with $\mathscr{E}(M(p), C)$; mark p and p' as corresponding.

($\mathscr{R}_r$) Let r be a node, with children p_1, p_2 labeled C, C_1, and C_2, respectively, where $C = C_1 \cup C_2 \smallsetminus \{x, \bar{x}\}$. Further, consider the nodes p'_1 and p'_2 corresponding to p_1 and p_2, respectively, and their respective labels C'_1 and C'_2. If there is a literal $x^P \in C'_1 \cup C'_2$ for some P, create a node r' in π' and label it with $C' = C'_1 \cup C'_2 \smallsetminus \{x^P, \bar{x}^P\}$. Mark r and r' as corresponding.

($\mathscr{R}_u$) Let p_u be node in π with a single child r labeled C_u and C_r, respectively, where C_u is a result of $\forall$-reduction of C_r. If p_r corresponds to p'_r mark p_u and p'_r also corresponding.

By induction on resolution depth, we show that the above construction results in a valid resolution tree π'. Additionally we prove, that if p' in π', labeled with a clause C', corresponds to some p in π, labeled with a clause C, then for any existential literal $l \in C$, with $\mathsf{var}(l) = x$ there is one and only one literal $l' \in C'$ s.t. $\mathsf{var}(l') = x^P$, for some P, and, the literals l, l' have the same polarity. Consequently, the root of π' must be labeled with the empty clause.

Rule $\mathscr{R}_l$ is well-defined due to conditions ($\mathscr{C}_1$) and ($\mathscr{C}_2$); it establishes the induction hypothesis due to definition of $\mathscr{E}$. For rule $\mathscr{R}_r$ we first observe that there must be a $x^{P_1} \in C'_1 \cup C'_2$, for some P_1, from the induction hypothesis because $x \in C_1 \cup C_2$. WLOG let $x^{P_1} \in C'_1$. From induction hypothesis we also have, $x \in C_1$, $\bar{x} \in C_2$, and $\bar{x}^{P_2} \in C'_2$ for some P_2. Since C'_1 and C'_2 were obtained by valid resolution steps, there must be a path in π' from some leaf p'_{l_1} to p'_1 where all clauses contain the literal x^{P_1}; analogously there a is path in π' from some leaf p'_{l_2} to p'_2 where all clauses contain the literal $\bar{x}^{P_2}$. Both paths correspond to some paths from p_{l_1} to p_1 and p_{l_2} to p_2 in π. Hence, $p_{l_1}, p_{l_2} \in S$ for some $(r, x, D, S) \in \mathcal{Q}_\pi$. Due to condition ($\mathscr{C}_3$), the variable x must be substituted with the same copy in the leafs and therefore also $P_1 = P_2$. Because $x^{P_1} \in C'_1$ and $\bar{x}^{P_1} \in C'_2$, the resolution step on C'_1 and C'_2 is possible. It remains to be shown that the resolution step does not introduce more than one copy of some literal. Assume that there are literals y^{R_1} and y^{R_2} in C'_1 and C'_2, respectively, where $y \neq x$. From induction hypothesis, $y \in C_1$ and $y \in C_2$. Consequently, there are some leafs p_{l_1}, p_{l_2} of π s.t. y appears in all clauses on the paths from p_{l_1} to p_1 and from p_{l_2} to p_2. Because π is a refutation proof, y gets eventually resolved away. Therefore there is some $(r_y, y, D_y, S_y) \in \mathcal{Q}_\pi$ for which $p_{l_1}, p_{l_2} \in S_y$ and therefore $R_1 = R_2$ from condi-



(a) Q-resolution (b) $\forall$-expansion tree (c) plain resolution

Fig. 2. Examples

tion ($\mathscr{C}_3$). Rule $\mathscr{R}_u$ preserves the induction hypothesis as universal reduction does not modify the set of existential literals. □

Example 2. Consider $\forall u_1 u_2 \exists e. (u_1 \vee e) \wedge (u_2 \vee \bar{e})$ with the Q-resolution refutation in Figure 2(a), which induces a single quadruple $(p_4, e, \{u_1, u_2\}, \{p_1, p_2\})$. To obtain a $\forall$Exp+Res refutation, generate the single-branch tree $\mathcal{T}$ in Figure 2(b) and mapping M with $M(p_1) = M(p_2) = \{u_1/0, u_2/0\}$ yielding the $\forall$-expansion $e^{u_1/0,u_2/0} \wedge \bar{e}^{u_1/0,u_2/0}$ with the corresponding resolution tree Figure 2(c). Observe that conditions $\mathscr{C}_1$–$\mathscr{C}_3$ from Proposition 1 are fulfilled. Clauses participating in the Q-resolution step are expanded so that e is replaced with the same copy. The universal literals u_1, u_2 are assigned to 0 by the expansion. Consequently, this Q-resolution step can be reproduced in a plain resolution refutation. Note that universal reduction steps are unnecessary in the resolution refutation since expansions remove all universal literals.

3.1 Construction of $\mathcal{T}$ and M

Proposition 1 gives us conditions $\mathscr{C}_1$–$\mathscr{C}_3$ on a $\forall$-expansion tree $\mathcal{T}$ and a mapping M so that any $\mathcal{T}$ and M satisfying these conditions enable us to construct the desired plain-resolution refutation π' for $\mathscr{E}(\mathcal{T}, \Phi)$. This subsection shows that such $\mathcal{T}$ and M can be constructed for any given Q-resolution refutation π.

For a quadruple $q = (r, x, D, S) \in \mathcal{Q}_\pi$ we say that *q is at level* $\mathsf{lv}(x)$ and we say that a leaf p of π *is in* q iff $p \in S$. Recall that the intuition behind a quadruple $(r, x, D, S) \in \mathcal{Q}_\pi$ is that the expanded counterparts of clauses labeling the leafs in S will contain the same fresh copy of x. Further, the assignment used for the expansion must assign to 0 the universal literals in those clauses. This poses the following question: *If some leaf p of π is in two different quadruples $q_1, q_2 \in \mathcal{Q}_\pi$, how do we ensure that the conditions are not conflicting?*

We say that $(r, x, D, S), (r', x', D', S') \in \mathcal{Q}_\pi$ are *connected* iff $S \cap S' \neq \emptyset$. We say that leafs p_1, p_2 of π *share level* k iff there exists a sequence (with possible repetitions) of quadruples $q_1, \ldots, q_n \subseteq \mathcal{Q}_\pi$, s.t. p_1 is in q_1; p_2 is in q_n; each q_i in the sequence has a level $l \geq k$; and each two adjacent quadruples are connected.

Observation 1 *The relation "*share level k*" is an equivalence relation on the leafs of π. All leafs of π share level 2 (recall that existential variables start at level 2). If two leafs share level k, then they share a level $l \leq k$.*

Let us look more closely at quadruples that share some level k. Recall that the given Φ formula has the prefix $\forall \mathcal{U}_1 \exists \mathcal{E}_2 \ldots \forall \mathcal{U}_{2N-1} \exists \mathcal{E}_{2N}$. Consider two connected

Algorithm 1: Expansion tree construction from $\mathcal{Q}_\pi$

```
Function Build ($k$, StopLev, $L$)
in  : StopLev..base-case level, $k \leq$ StopLev..current level, $L$..subset of leafs of $\pi$
out : a pair $(\mathcal{T}', M')$, where $\mathcal{T}'$ is an expansion tree for universal variables with
      level $\geq k$, $M'$ is a mapping from leafs in $L$ to root-to-leaf paths in $\mathcal{T}'$
begin
    if $k = \mathsf{StopLev}$ then
        $\mathcal{T}' \leftarrow$ create a tree with a single node, the root $r$
        $M' \leftarrow$ map all nodes in $L$ to the empty path starting in $r$
        return $(\mathcal{T}', M')$
    $T' \leftarrow$ a tree with the root node $r$
    $M' \leftarrow$ empty mapping
    $\Xi \leftarrow$ partition nodes $L$ by the "share level $k+1$" relation
    foreach $\rho \in \Xi$ do
        $Q_\rho \leftarrow \{q \in \mathcal{Q}_\pi \mid$ there exists $p \in \rho$ in $q,\ q$ is at level $> k\}$
        $D_\rho \leftarrow \{l \mid (p, e, D, S) \in Q_\rho, l \in D, \mathsf{lv}(l) = k\}$
        $\tau_\rho \leftarrow \{u/0 \mid u \in D_\rho\} \cup \{u/1 \mid \bar{u} \in D_\rho\} \cup \{u/0 \mid u, \bar{u} \notin D_\rho, \mathsf{lv}(u) = k\}$
        $(\mathcal{T}_\rho, M_\rho) \leftarrow \mathtt{Build}(k+2, \mathsf{StopLev}, \rho)$
        add $\mathcal{T}_\rho$ to $\mathcal{T}'$, connect $r$ to the root of $\mathcal{T}_\rho$ with an edge labeled with $\tau_\rho$
        if $M_\rho$ maps a leaf $p \in L$ to $\tau$, map $p$ to $\tau_\rho \cup \tau$ in $M'$
    return $(\mathcal{T}', M')$
```

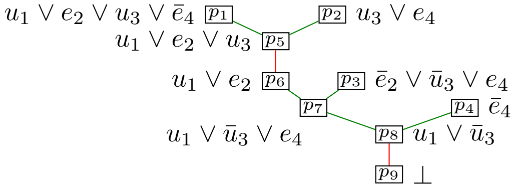
quadruples $(r, x, D, S), (r', x', D', S') \in \mathcal{Q}_\pi$, both at some level $\geq k$, i.e. $\mathsf{lv}(x) \geq k$ and $\mathsf{lv}(x') \geq k$. Our objective is to build such mapping M that for any two $p_1, p_2 \in S$, the paths $M(p_1)$ and $M(p_2)$ share the prefix of length $\mathsf{lv}(x)/2$ corresponding to assignments to variables $\mathcal{U}_1 \mathcal{U}_2 \dots \mathcal{U}_{\mathsf{lv}(x)-1}$; this ensures that x is renamed to the same fresh copy in clauses of the leafs. The same holds for leafs in S'. Since the quadruples are connected, there is some leaf p that belongs to $p \in S \cap S'$. Further, since both x and x' are at a level greater or equal to k, by transitivity, *all* leafs in $S \cup S'$ must be mapped to such paths of the $\forall$-expansion tree $\mathcal{T}$ that they share their prefixes of length $k/2$. This immediately generalizes to sequences of connected quadruples. If two leafs p_1, p_2 of π share level $k = 2l$, then $M(p_1)$ and $M(p_2)$ must have common prefix of length l, corresponding to assignments to variables $\mathcal{U}_1 \mathcal{U}_2 \dots \mathcal{U}_{k-1}$.

This observation motivates Algorithm 1, which is represented as a recursive function. The recursion is initiated by the call $\mathtt{Build}(1, 2N+1, L_{\text{all}})$ where L_{all} is the set of leafs of π. After this initial call terminates, any root-to-leaf paths with the same labeling in the returned tree are merged to obtain the required $\mathcal{T}$.

The function returns $\mathcal{T}'$, a subtree of the tree $\mathcal{T}$ being constructed, and a mapping M' that maps the given leafs L to paths of $\mathcal{T}'$. The labeling of root-to-leaf paths in $\mathcal{T}'$ are total assignments to variables $\mathcal{U}_k, \mathcal{U}_{k+2}, \dots, \mathcal{U}_{2N-1}$, where k is an odd natural number. Hence, for the base case of the recursion, i.e. $k = 2N+1$, the function creates a single-node tree $\mathcal{T}'$ and maps all given leafs L to an empty path starting and ending in the root of $\mathcal{T}'$.

For the non-base case, the function partitions the given leafs L of π by the "share level $k+1$" relation. From the conditions on $\mathcal{T}$, clauses labeling leafs that share level $k+1$ must be expanded such that existential variables with level $> k$ are replaced with the same copies. At the same time, the universal literals in these clauses with level $\leq k$ must be assigned to 0. The algorithm visits each partition ρ of the "share level $k+1$" partition and collects quadruples $q \in \mathcal{Q}_\pi$ for which there is some leaf $p \in \rho$ in q. Subsequently, it collects all universal literals at level k that appear in these quadruples and computes an assignment τ_ρ which assigns them to 0 and other literals assigns arbitrarily (line 14).

Example 3. Consider the following Q-resolution proof π with the prefix $\forall u_1 \exists e_2 \forall u_3 \exists e_4$.


$u_1 \vee e_2 \vee u_3 \vee \bar{e}_4$ p_1
p_2 $u_3 \vee e_4$
$u_1 \vee e_2 \vee u_3$ p_5
$u_1 \vee e_2$ p_6
p_3 $\bar{e}_2 \vee \bar{u}_3 \vee e_4$
p_7
p_4 $\bar{e}_4$
$u_1 \vee \bar{u}_3 \vee e_4$
p_8 $u_1 \vee \bar{u}_3$
p_9 $\bot$

$(p_5, e_4, \{u_1, u_3\}, \{p_1, p_2\})$

$(p_7, e_2, \{u_1\}, \{p_1, p_3\})$

$(p_8, e_4, \{u_1, \bar{u}_3\}, \{p_3, p_4\})$

This yields the quadruples depicted on the right hand side. All leafs share level $1+1$ and are put into a single partition $\rho = \{p_1, p_2, p_3, p_4\}$ labeled with $\{u_1/0\}$. Based on sharing of level $3+1$, ρ is split into $\{p_1, p_2\}$ and $\{p_3, p_4\}$, labeled $\{u_3/0\}$ and $\{u_3/1\}$, respectively. The resulting mapping is $M(p_1) = M(p_2) = \{u_1/0, u_3/0\}$ and $M(p_3) = M(p_4) = \{u_1/0, u_3/1\}$.

Let us now focus on the correctness of Algorithm 1. The algorithm is terminating because the set of quadruples $\mathcal{Q}_\pi$ is finite. That the algorithm constructs mapping M and the tree $\mathcal{T}$ satisfying the conditions $(\mathscr{C}_1)$–$(\mathscr{C}_3)$ of Proposition 1 hinges on proving that the set of literals D_ρ (line 12) does not contain complementary literals. Consequently, that the assignment τ_ρ (line 14) is indeed an assignment. For now we assume that this holds and show it later in order to first focus on the overall workings of the algorithm.

Since π has no empty clauses in leafs and all input clauses are $\forall$-reduced, every leaf p labeled with some clause C must be in some quadruple in $\mathcal{Q}_\pi$. At each level k, quadruples are partitioned so eventually there will be one and only one path P in $\mathcal{T}$ s.t. $M(p) = P$. Thus satisfying condition $(\mathscr{C}_1)$ of Proposition 1. If C contains some universal literal l with $\mathsf{lv}(l) = k$, l must be blocked by some existential literal $b \in C$ with $\mathsf{lv}(b) > k$. This literal b is eventually resolved away and therefore there must be a quadruple $q_b = (r, \mathsf{var}(b), D_b, S_b) \in \mathcal{Q}_\pi$ s.t. $p \in S_b$. Since b blocks l on a path from p to some child of r, it also holds that $l \in D_b$. Hence $q_b \in Q_\rho$, defined on line 11, and $l \in D_\rho$, defined on line 12. The algorithm places p into a subtree prepended by an edge labeled with τ_ρ, which sets l to 0. Thus satisfying condition $(\mathscr{C}_2)$. Consider two leafs p_1, p_2 of π such that they are in the same quadruple q at some level l. These leafs are connected at level $\leq l$. Hence they will be part of the same partition for levels $k < l$. Therefore, the algorithm puts the leafs in the same subtree while $k < l$ and therefore $M(p_1)$ and $M(p_2)$ assign the same value to all universal variables with level $k < l$ thus satisfying condition $(\mathscr{C}_3)$.

Now it remains to be shown that the set D_ρ constructed on line 12 is not contradictory. This will be shown in Lemma 5. However, before we reach this lemma, a series of auxiliary lemmas need to be derived. Since Q-resolution enables resolving two clauses $C_1 \vee x$ and $C_2 \vee \bar{x}$ only if $C_1 \cup C_2$ does not contain complementary literals, we can make the following observation.

Observation 2 *For any $(r, x, D, S) \in \mathcal{Q}_\pi$, the literals D are noncontradictory.*

Lemma 1. *If any two quadruples (r_1, x_1, D_1, S_1), $(r_2, x_2, D_2, S_2) \in \mathcal{Q}_\pi$ are connected, then r_1 dominates r_2, i.e. r_2 is in a subtree of r_1, or r_2 dominates r_1.*

Proof. Since the quadruples are connected, there is some leaf p_l of π s.t. $p_l \in S_1$ and $p_l \in S_2$. At the same time there is an undirected path from both r_1 and r_2 to p_l. If neither r_1 dominated r_2 nor r_2 dominated r_1 there would be a cycle from root to r_1, p_l, r_2, and back to the root. □

Lemma 2. *Consider any two quadruples $(r_1, x_1, D_1, S_1), (r_2, x_2, D_2, S_2) \in \mathcal{Q}_\pi$ such that r_1 dominates r_2 and r_2 dominates some $p_l \in S_1$. Then all the clauses on the path from r_1 to r_2 except for r_1 contain a literal $b \in \{x_1, \bar{x}_1\}$.*

Proof. Since the leaf p_l is dominated by both r_1 and r_2, there is a path from the root of π going through r_1, r_2, and ending in p_l. Since $p_l \in S_1$, from definition of the quadruples, there is a literal $b \in x_1, \bar{x}_1$ that appears everywhere on the path except for the node r_1. □

The following lemma shows that for any sequence of connected quadruples that are all at some level $l \geq k$, there is a quadruple pertaining to a resolution node r such that r dominates all the other resolution nodes in the sequence, and, all paths from this node to these resolution nodes contain some existential literal b with $\mathsf{lv}(b) \geq k$. Consequently, these literals block all universal literals with level $l < k$ on these paths.

Lemma 3. *Consider a sequence of quadruples $\gamma = q_1, \ldots, q_n$, such that each $q_i \in \mathcal{Q}_\pi$ in the sequence has a level $l \geq k$ and each two adjacent quadruples are connected. Then there is $(r, x, D, S) \in \gamma$ such that for any quadruple $(r_j, x_j, D_j, S_j) \in \gamma$ the node r dominates r_j and all the clauses on the path from r to r_j, except for r, contain some existential literal b with $\mathsf{lv}(b) \geq k$.*

Proof. Proof by induction on the length of prefix of γ. For the base case choose (r, x, D, S) as q_1. For the inductive case consider $i > 1$ and $q' = (r', x', D', S')$ from the induction hypothesis such that q' satisfies the condition for $q_1, \ldots, q_{i-1}$. Since adjacent quadruples are connected, for $q_i = (r_i, x_i, D_i, S_i)$ and $q_{i-1} = (r_{i-1}, x_{i-1}, D_{i-1}, S_{i-1})$ there is a leaf $p_c \in S_{i-1} \cap S_i$. Split on the following cases.

If q_i is equal to any of the q_j for $j < i$, choose (r, x, D, S) to be q'. If r_i dominates r' then invoke Lemma 2 whose preconditions are satisfied because r_i dominates r' and r' dominates p_c, from the induction hypothesis. Hence there is a path from one of the children of r_i to r containing the literal $b \in \{x_i, \bar{x}_i\}$. Note that b does not appear in r_i but does appear in r'. From induction hypothesis, for any r_j, $j < i$ there is a path from a child of r' to r_j where each clause is blocked by some literal with level $l \geq k$. Concatenating the path from r_i to r' with the path r' to r_j satisfies the condition for j. Choose (r, x, D, S) to be q_i.

From Lemma 1, either r_i is dominated by r_{i-1} or r_{i-1} is dominated by r_i. Hence we need to consider only these two remaining cases. If r_{i-1} dominates r_i, then from Lemma 2 there is a $b_{i-1} \in \{x_{i-1}, \bar{x}_{i-1}\}$ that appears on the path from one of the children of r_{i-1} to r_i (inclusively). From induction hypothesis, there is a path from r' to r_{i-1}, excluding r', that contains some existential literals b with $\mathsf{lv}(b) \geq k$. Concatenating this path with the path from r_{i-1} to r_i gives us a path satisfying the required condition for the node r_i. In particular, there is a path from a child of r' to r_i such that each clause on the path contains a some existential literals b with $\mathsf{lv}(b) \geq k$.

If r_{i-1} is dominated by r_i and r_i does not dominate r', then r' must dominate r_i otherwise there would be a cycle from the root to r', r_{i-1}, r_i, and back to root. From induction hypothesis, each clause on the path from r' to r_{i-1} contains some existential literal b with $\mathsf{lv}(b) \geq k$. Since r' dominates r_i, which in turn dominates r_{i-1}, the path from r' to r_i is a prefix of the path from r' to r_{i-1} and therefore also satisfies the required condition. Choose (r, x, D, S) to be q'. □

Lemma 4. *Consider ρ a subset of leafs of π that is an equivalence class of the* share level $k+1$ *relation for some odd number k. Define $Q^k_\rho \subseteq \mathcal{Q}_\pi$ as follows.*

$$Q^k_\rho = \{(r, x, D, S) \in \mathcal{Q}_\pi \mid p \in \rho, p \in S, \mathsf{lv}(x) > k\}$$

Then for any $q_a, q_b \in Q^k_\rho$ there is a sequence of quadruples $q_1, \ldots, q_m$ where $q_a = q_1$, $q_b = q_m$, each q_i is at a level $> k$ and $q_i \in Q^k_\rho$, and each two adjacent q_i,$q_{i+1} \in Q^k_\rho$ are connected.

Proof. From definition of Q^k_ρ there are leafs $p_a, p_b \in \rho$ s.t. $p_a \in q_a$, $p_b \in q_b$. Since ρ is an equivalence class of *share level* $k+1$ relation, there is a sequence of connected quadruples $s_1, \ldots, s_n$ such that p_a is in s_1 and p_b is in s_n, and each quadruple in the sequence is at a level $> k$. Since for any $s_i = (r_i, x_i, D_i, S_i)$, the set S_i is non-empty, all leafs $p \in S_i$ share level $k+1$ with p_a and $p \in \rho$. Hence, all the quadruples s_i in the sequence are in Q^k_ρ. Since q_a and s_1 are connected because of p_a and q_b and q_b are connected because of p_b, constructing the sequence $q_a, s_1, \ldots, s_n, q_b$ yields the required sequence. □

Lemma 5. *Let k, ρ, and Q^k_ρ be defined as in Lemma 4. Define a set of literals D^k_ρ as $D^k_\rho = \left\{l \mid (r, x, D, S) \in Q^k_\rho, \mathsf{lv}(l) = k, l \in D\right\}$. The set D^k_ρ does not contain complementary literals.*

Proof. Lemma 4 gives us that Q^k_ρ can be organized into a sequence γ where each two adjacent quadruples are connected and each $q_i \in \gamma$ is at a level $> k$. From Lemma 3 there is a quadruple $(r_d, x_d, D_d, S_d) \in \gamma$ s.t. for any quadruple $(r_j, x_j, D_j, S_j) \in \gamma$ the node r_d dominates r_j and all the clauses on the path from r_d to r_j, except for r_d, contain some existential literal b with $\mathsf{lv}(b) > k$. Hence, no universal literals with level $l \leq k$ can be $\forall$-reduced on a path from r_j to r_d in π. Therefore necessarily, D_d contains all literals D_j. Consequently, $D^k_\rho \subseteq D_d$. From Observation 2, the set D_d is noncontradictory and therefore D^k_ρ is also noncontradictory. □

This last lemma gives us what we needed to conclude the correctness of Algorithm 1, i.e. that the set of literals D_ρ, constructed on line 12 is not contradictory. Algorithm 1 operates in time polynomial to the size of π because the size of the set $\mathcal{Q}_\pi$

is linear to the size of π and partitioning by "share level $k+1$" relation can be done in polynomial time. This fact, together with Proposition 1 lets us derive the following.

Theorem 2. *For any tree Q-resolution refutation π there exists a $\forall$Exp+Res refutation $(\mathcal{T}, \pi_{\mathcal{T}})$ s.t. both $\mathcal{T}$ and $\pi_{\mathcal{T}}$ are polynomial in size of π. This $\forall$Exp+Res refutation can be constructed in time polynomial to π. Hence, $\forall$Exp+Res p-simulates tree Q-resolution.*

4 Simulating Restricted ∀Exp+Res by Q-Resolution

This section shows that a certain *fragment* of ∀Exp+Res refutations can be simulated by Q-resolution. This fragment allows expansions of universal quantifiers as before but puts a restriction on the resolution proof of the expansion. In particular, it allows only resolutions that follow the order of the quantifier prefix.

Definition 4 (level-ordered). *Consider a $\forall$Exp+Res refutation $(\mathcal{T}, \pi)$ of Φ. We say that $(\mathcal{T}, \pi)$ is* level-ordered *iff the following holds. Let $x^P \vee C_1$ and $\bar{x}^P \vee C_2$ be some clauses resolved in π, then $\mathsf{lv}(y) \leq \mathsf{lv}(x)$ for any $y^{P_1} \in \mathsf{var}(C_1 \vee C_2)$.*

Lemma 6. *Let $(\mathcal{T}, \pi)$ be a level-ordered $\forall$Exp+Res refutation of Φ. Let C be some clause in π and $x_1^{P_1}$,$x_2^{P_2} \in \mathsf{var}(C)$. If $\mathsf{lv}(x_1) \leq \mathsf{lv}(x_2)$, then the path P_1 is a prefix of the path P_2.*

Proof. By induction on the number of resolution steps that led to C. The condition is true for the leafs of π from the definition of $\mathscr{E}$. For the induction step consider clauses $C_1 \vee \bar{x}_r^P$ and $C_2 \vee x_r^P$ with the resolvent $C = C_1 \vee C_2$. If C is empty or unit, the condition is trivially satisfied. Let $x_1^{P_1}$,$x_2^{P_2} \in \mathsf{var}(C)$ with $\mathsf{lv}(x_1) \leq \mathsf{lv}(x_2)$. Because π is level-ordered, $\mathsf{lv}(x_1) \leq \mathsf{lv}(x_r)$ and $\mathsf{lv}(x_2) \leq \mathsf{lv}(x_r)$, from which the induction hypothesis gives that both paths P_1 and P_2 are prefixes of the path P. Since $\mathsf{lv}(x_1) \leq \mathsf{lv}(x_2)$, then $|P_1| \leq |P_2|$ from definition of $\mathscr{E}$. Hence the path P_1 is a prefix of the path P_2. □

Lemma 7. *Let $(\mathcal{T}, \pi)$ be a level-ordered $\forall$Exp+Res refutation of Φ. Let C be a clause in π and x^{P_1},$x^{P_2} \in \mathsf{var}(C)$, then $P_1 = P_2$.*

Proof. Immediate consequence of Lemma 6. □

Theorem 3. *Let $(\mathcal{T}, \pi)$ be a level-ordered $\forall$Exp+Res refutation of Φ. Then a Q-resolution refutation of Φ can be constructed in polynomial time with respect to $|(\mathcal{T}, \pi)|$. Hence, Q-resolution p-simulates level-ordered $\forall$Exp+Res.*

Proof (sketch). The proof is similar to the one of Proposition 1, i.e. we construct a Q-resolution refutation π' based on π and prove its correctness by induction on resolution depth. For each leaf p in π labeled with a clause C, there exists a path P from the root to some leaf in $\mathcal{T}$ and a clause $C' \in \phi$ such that $\mathscr{E}(P, C') = C$. Replace C with C'. Whenever there is a resolution on some variable x^P in π, perform resolution on x in π'. Add ∀-reduction steps after each resolution step. Effectively, the Q-resolution refutation will have the same shape as the plain resolution refutation but each variable x^P will be

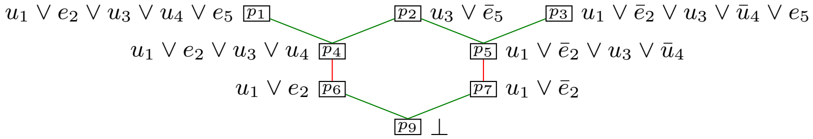

$u_1 \vee e_2 \vee u_3 \vee u_4 \vee e_5$ p_1
p_2 $u_3 \vee \bar{e}_5$
p_3 $u_1 \vee \bar{e}_2 \vee u_3 \vee \bar{u}_4 \vee e_5$
$u_1 \vee e_2 \vee u_3 \vee u_4$ p_4
p_5 $u_1 \vee \bar{e}_2 \vee u_3 \vee \bar{u}_4$
$u_1 \vee e_2$ p_6
p_7 $u_1 \vee \bar{e}_2$
p_9 $\perp$

Fig. 3. Nontree Q-resolution example

replaced with the variable x (“removed superscripts”), and, some universal literals will be inserted into the clauses.

The correctness of the resulting π' follows from Lemmas 6 and 7. Lemma 7 guarantees that in the plain resolution refutation there are no clauses containing variables x^{P_1} and x^{P_2} with $P_1 \neq P_2$. Consequently, removing the superscripts does not yield complementary existential literals in clauses of π'.

It remains to be shown that there are no complementary universal literals within clauses of π'. If there's a universal literal $k \in C'$ for some clause $C' \in \pi$, there most be some existential literal $x \in C'$ that blocks it. At the same time there's a corresponding literal $x^P \in C$ for the corresponding clause in π. We observe that P assigns k to 0. For leaf clauses this follows from the definition of $\mathscr{E}$. For resolution steps this follows from the level-orderndess which guarantees that the literal being resolved on blocks all universal literals in the clause. So if there's a resolution on a x^P in π, the clauses involved in the corresponding resolution in π' may contain only universal literals that are assigned to 0 by P and therefore complementary universal literals cannot meet.

5 Examples

This section illustrates some of the practical implications of the results derived so far. Section 3 shows that *tree* Q-resolution refutations can be simulated by ∀Exp+Res refutations. This result points in the direction of formulas where ∀Exp+Res will perform significantly worse than Q-resolution. In particular, this hints that *non-tree* Q-resolution refutations might prove nontrivial to simulate for ∀Exp+Res. The following example illustrates why that is the case.

For the quantifier prefix $\forall u_1 \exists e_2 \forall u_3 u_4 \exists e_5$, Figure 3 shows a simple non-tree Q-resolution proof that demonstrates a drawback of ∀-expansion-based proofs. Assume that clauses on p_1, p_3 are expanded to some clauses C'_1, C'_3, respectively. The clauses will contain some copies of e_5: $e_5^{P_1} \in C'_1$, $e_5^{P_3} \in C'_3$, let's say. It must be that $P_1(u_1) = P_1(u_3) = P_1(u_4) = 0$ and $P_3(u_1) = P_3(u_3) = P_3(\bar{u}_4) = 0$ Because of the different polarity of literal u_4 in the assignments, $P_1 \neq P_3$. This means that there must be 2 different expansions of clause on p_2. Hence, formulas leading to a high level of sharing in Q-resolution are likely to be easier for DPLL-based solvers than for expansion-based solvers.

Section 4 shows that Q-resolution can simulate ∀Exp+Res refutations where the plain resolution part follows a certain variable order. Again, this points us in the direction of formulas where ∀Exp+Res might perform better than Q-resolution, i.e. formulas with proofs not respecting this order. To support this hypothesis, we construct

$\mathbf{x_i \lor z \lor C_i^1}$	$\mathbf{\bar{x}_i \lor \bar{z} \lor C_i^2}$	$\mathbf{z/0}$	$\mathbf{z/1}$
$x_1 \lor z \lor \bar{y}_1$	$\bar{x}_1 \lor \bar{z} \lor \bar{y}_1$	$x_1 \lor \bar{y}_1^{z/0}$	$\bar{x}_1 \lor \bar{y}_1^{z/1}$
$x_2 \lor z \lor y_1$	$\bar{x}_2 \lor \bar{z} \lor \bar{y}_1$	$x_2 \lor y_1^{z/0}$	$\bar{x}_2 \lor \bar{y}_1^{z/1}$
$x_3 \lor z \lor \bar{y}_1$	$\bar{x}_3 \lor \bar{z} \lor y_1$	$x_3 \lor \bar{y}_1^{z/0}$	$\bar{x}_3 \lor y_1^{z/1}$
$x_4 \lor z \lor y_1$	$\bar{x}_4 \lor \bar{z} \lor y_1$	$x_4 \lor y_1^{z/0}$	$\bar{x}_4 \lor y_1^{z/1}$

Fig. 4. Example formula for $n = 1$

the following formula[3]. Let $n \in \mathbb{N}^+$ and $H = 2^{2n}$. Consider the set of variables $y_1, \ldots, y_n$,$x_1, \ldots, x_H$, z and the prefix $\exists x_1, \ldots, x_H \forall z \exists y_1, \ldots, y_n$. We construct the matrix as follows. For each $i \in 1 \ldots H$ construct two clauses of the form $x_{i+1} \lor z \lor C_i^1$, $\bar{x}_{i+1} \lor \bar{z} \lor C_i^2$, where $\mathsf{var}(C_i^1) = \mathsf{var}(C_i^2) = y_1 \ldots y_n$ and the pair C_i^1, C_i^2 goes over all the possible $2^{2n} = H$ pairs of sets of literals on the pertaining variables. More precisely, Let i_j be the jth bit of i, where $j \in 0..(2N-1)$. Add to C_i^1 the literal $\bar{y}_j$ if $i_j = 0$, where $j \in 0..(N-1)$. Add to C_i^1 the literal y_j if $i_j = 1$, where $j \in 0..(N-1)$. Add to C_i^2 the literal $\bar{y}_j$ if $i_j = 0$, where $j \in N..(2N-1)$. Add to C_i^2 the literal y_j if $i_j = 1$, where $j \in N..(2N-1)$. For the expansion we consider an expansion that includes both possible assignments: $z/0$ and $z/1$. Figure 4 shows the matrix and the expansion for $n = 1$.

While the expansion duplicates the y_i variables, it is easily shown unsatisfiable. Any total assignment to the copies of y_i variables gives a conflict and therefore a SAT solver that assigns these variables first, will need at most $2^{2n} = H$ conflicts to show unsatisfiability.

We show that this formula requires exponential computation by a conflict-driven DPLL QBF solver [23]. (However, this does not mean that there is no polynomial Q-resolution proof.) We first make the following observation.

Lemma 8. *If a CNF ψ is unsatisfiable and $|C| \geq k$ for all $C \in \psi$, then $|\psi| \geq 2^k$.*

Proof. Let $V = \mathsf{var}(\psi)$. Each clause $C \in \psi$ is 0 under $2^{|V|-|C|} \leq 2^{|V|-k}$ assignments to variables V. Since ψ is unsatisfiable, for *each* assignment τ to variables V there is a clause that is 0 under τ. By averaging $|\psi| \geq \frac{2^{|V|}}{2^{|V|-k}} = 2^k$.

A conflict-driven QBF solver first assigns the x_i variables, then z, and then y_i variables. Since long-distance resolution is not invoked in this example, clauses containing z do not give propagation while x_i variables are being assigned. Since the formula is false, after all x_i variables are assigned by some assignment τ_x, the solver eventually finds such value v_z for z that $\phi[\tau_x, z/v_z]$ is unsatisfiable. Once z is assigned a value, either all $x_i \lor z \lor C_i^1$ are satisfied or all $\bar{x}_i \lor \bar{z} \lor C_i^2$ are satisfied. For the solver to backtrack to the level of x_i variables, it must learn a clause containing only x_i variables. From Lemma 8, 2^n clauses must be used in learning this clause since this clause is a result of a resolution tree that forms a refutation proof once all z and x_i variables are removed from it. Consequently, the learned clause containing only x_i variables has at least 2^n variables. This is repeated until the set of learned clauses containing only

[3] The formula's generator is found at http://sat.inesc-id.pt/~mikolas/sat13.

x_i variables is unsatisfiable. Invoking again Lemma 8 gives that this must be repeated at least $2^{2^n} = 2^{\sqrt{H}}$ times (exponentially more than the expansion approach). We note that QuBE7.2 [13], DepQBF [20], and, non-CEGAR version of GhostQ [17] were able solve this formula only for $n \leq 3$. The expansion-based solver RAReQS [15] was able to solve the formula up to $n = 10$ (which has $1,048,587$ variables).

6 Conclusions and Future Work

This paper introduces and studies a proof system ∀Exp+Res aimed at refuting false QBFs based on expansion of universal variables and propositional resolution. Besides preprocessing [6,5] expansion of variables plays an important role in QBF solving. The solvers QUBOS [1], Nenofex [19], Quantor [4] expand universal variables from inner- to outermost levels. However, these expansions are possibly interleaved with operations for removal of existential quantifiers. In future work, we wish to investigate if these interleaved expansions give additional proving power to the solvers. The solver sKizzo [3] expands all universal quantifiers as is done in ∀Exp+Res (even though the process is called Skolemization). sKizzo expands the formula clause by clause, ignoring assignments to universal variables that satisfy the clause. So even though sKizzo does not explicitly avail of partial expansions, trivial parts of the expansion are not generated.

The solver RaReQS [16,15] constructs two types of expansions: one for universal variables and one for existential ones. For false QBFs, universal expansion eventually becomes false. Hence, the workings of RaReQS mimics the ∀Exp+Res in the case of false formulas. It should also be noted that out of the mentioned solvers, only RaReQS constructs *partial* expansions, i.e. both polarities of the expanded variable are considered in the other solvers.

It is the ability of ∀Exp+Res to expand partially that was crucial in showing that ∀Exp+Res can p-simulate *tree* Q-resolution refutations. In the opposite direction, we showed that Q-resolution can polynomially simulate ∀Exp+Res if the plain resolution part follows certain order of variables.

Hence, at this point it remains open how unrestricted ∀Exp+Res compares to unrestricted Q-resolution or possibly *long distance Q-resolution* [23,2,17]. However, Section 5 hints towards formulas that will be easy for one calculus and hard for the other. We conjecture that exponential separations can be shown in both directions. Such separation would be of high practical importance. Firstly, it would explain why expansion-based solvers are better for some classes of instances than DPLL solvers, and the other way around. Secondly, the separation would necessitate QBF certification formats supporting both types of solvers.

Acknowledgments. We would like to thank Uwe Egly and Will Klieber for various helpful conversations on QBFs. We would also like to thank the anonymous reviewers for their stimulating feedback. This work is partially supported by SFI PI grant BEACON (09/IN.1/I2618), FCT grants ATTEST (CMU-PT/ELE/0009/2009) and POLARIS (PTDC/EIA-CCO/123051/2010), and multiannual PIDDAC program funds (PEst-OE-/EEI/LA0021/2011).

References

1. Ayari, A., Basin, D.A.: QUBOS: Deciding quantified Boolean logic using propositional satisfiability solvers. In: Aagaard, M., O'Leary, J.W. (eds.) FMCAD. pp. 187–201 (2002)
2. Balabanov, V., Jiang, J.H.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012)
3. Benedetti, M.: Evaluating QBFs via symbolic Skolemization. In: LPAR (2004)
4. Biere, A.: Resolve and expand. In: SAT (2004)
5. Bubeck, U.: Model-based transformations for quantified Boolean formulas. Ph.D. thesis, University of Paderborn (2010)
6. Bubeck, U., Büning, H.K.: Bounded universal expansion for preprocessing QBF. In: SAT (2007)
7. Büning, H.K., Bubeck, U.: Theory of quantified boolean formulas. In: Handbook of Satisfiability. IOS Press (2009)
8. Büning, H.K., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. 117(1) (1995)
9. Cadoli, M., Schaerf, M., Giovanardi, A., Giovanardi, M.: An algorithm to evaluate quantified Boolean formulae and its experimental evaluation. J. Autom. Reasoning 28(2), 101–142 (2002)
10. Cimatti, A., Sebastiani, R. (eds.): Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7317. Springer (2012)
11. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. J. Symb. Log. 44(1), 36–50 (1979)
12. Egly, U.: On sequent systems and resolution for QBFs. In: Cimatti and Sebastiani [10], pp. 100–113
13. Giunchiglia, E., Marin, P., Narizzano, M.: QuBE 7.0 system description. Journal on Satisfiability, Boolean Modeling and Computation 7 (2010)
14. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause/term resolution and learning in the evaluation of quantified Boolean formulas. Journal of Artificial Intelligence Research 26(1), 371–416 (2006)
15. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. In: Cimatti and Sebastiani [10], pp. 114–128
16. Janota, M., Marques-Silva, J.: Abstraction-based algorithm for 2QBF. In: Sakallah, K.A., Simon, L. (eds.) SAT (2011)
17. Klieber, W., Sapra, S., Gao, S., Clarke, E.M.: A non-prenex, non-clausal QBF solver with game-state learning. In: SAT (2010)
18. Krajíček, J., Pudlák, P.: Quantified propositional calculi and fragments of bounded arithmetic. Mathematical Logic Quarterly 36(1), 29–46 (1990)
19. Lonsing, F., Biere, A.: Nenofex: Expanding NNF for QBF solving. In: SAT (2008)
20. Lonsing, F., Biere, A.: DepQBF: A dependency-aware QBF solver. JSAT (2010)
21. Rintanen, J.: Improvements to the evaluation of quantified Boolean formulae. In: Dean, T. (ed.) IJCAI. pp. 1192–1197. Morgan Kaufmann (1999)
22. Urquhart, A.: The complexity of propositional proofs. Bulletin of the EATCS 64 (1998)
23. Zhang, L., Malik, S.: Conflict driven learning in a quantified Boolean satisfiability solver. In: ICCAD (2002)