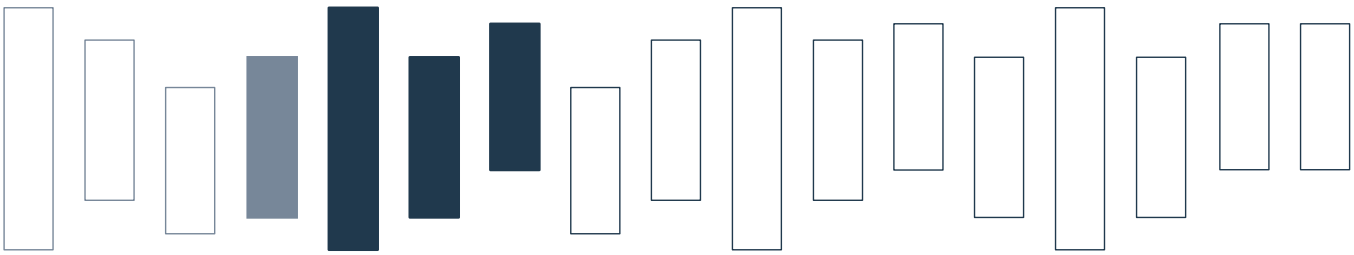
Technical Report AC-TR-20-007

June 2020

Short Q-Resolution Proofs with Homomorphisms

Ankit Shukla, Friedrich Slivovsky,
and Stefan Szeider



This is the authors' copy of a paper that will appear in the proceedings of SAT'20, the 23rd International Conference on Theory and Applications of Satisfiability Testing.

www.ac.tuwien.ac.at/tr

Short Q-Resolution Proofs with Homomorphisms*

Ankit Shukla[1], Friedrich Slivovsky[2], and Stefan Szeider[2]

[1] JKU, Linz, Austria ankit.shukla@jku.at
[2] Algorithms and Complexity Group, TU Wien, Vienna, Austria
[fs,sz]@ac.tuwien.ac.at

Abstract. We introduce new proof systems for quantified Boolean formulas (QBFs) by enhancing Q-resolution systems with rules which exploit local and global symmetries. The rules are based on homomorphisms that admit non-injective mappings between literals. This results in systems that are stronger than Q-resolution with (injective) symmetry rules. We further strengthen the systems by utilizing a dependency system D in a way that surpasses Q(D)-resolution in relative strength.

Keywords: Symmetries · Homomorphisms · QBF · Q(D)-Resolution · Dependency Schemes · Proof Complexity.

1 Introduction

In a 1985 paper, Krishnamurthy [12] introduced *symmetry rules* which strengthen the propositional resolution system to admit exponentially shorter proofs, for instance, linearly sized proofs for the Pigeon Hole Principle. The *global symmetry rule* exploits the automorphisms of the entire input formula. The even stronger *local symmetry rule* exploits the existence of isomorphic images of subsets of clauses within the formula. Szeider [19] further strengthened Krishnamurthy's proof systems, generalizing the symmetry rules to *homomorphism rules* by considering clause-preserving mappings that are not necessarily injective.

Recently, Kauers and Seidl [11] lifted Krishnamurthy's most basic symmetry rule, the global symmetry rule, to *Q-resolution* (Q-Res), the standard resolution-based proof system for quantified Boolean formulas (QBFs) in prenex conjunctive normal form (PCNF). They showed that several families of formulas that require exponentially-sized Q-resolution proofs admit polynomially-sized proofs if the generalized symmetry rule is added.

Our main contribution is the introduction and study of proof systems based on Q-resolution that are even stronger than the one studied by Kauers and Seidl [11]: we lift the local symmetry rule to the quantified setting, as well as the local and global homomorphism rules. A straightforward lifting of the rules from

* The authors acknowledge the support by the Austrian Science Funds (FWF), projects W1255 and P32441, and by the Vienna Science and Technology Fund (WWTF), projects ICT19-060 and ICT19-065.

the propositional case to the quantified case insists that the mapping between literals on which the symmetries (or more generally, homomorphisms) operate does not jump between quantifier blocks. A more general version allows a jump between quantifier blocks, as long as the relative position of the variables in the quantifier prefix is preserved. We go even a step further, and parameterize our systems by a *dependency scheme* D [15], and only require the mapping to preserve dependencies according to the chosen dependency scheme. Thus, our systems strengthen Kauers and Seidl's system along three dimensions:

1. from global symmetries to local symmetries,
2. from symmetries to homomorphisms, and
3. from quantifier-block preserving mappings to dependencies preserving mappings with respect to a dependency system.

Each of the three dimensions alone provides an exponential speedup.

Figure 1 gives an overview of the proof systems considered in this paper. In the figure, D stands for the reflexive resolution-path dependency scheme [18], a variant of the resolution-path dependency scheme [17,20], or any stronger dependency scheme. The separations between LH and LS, LH and GH, LS and GS, GH and GS, and GS and Q-Res, as well as the corresponding separations between the systems using a dependency scheme D, follows from the propositional case [19].

We show an exponential separation between LH(D) and LH for the reflexive resolution-path dependency scheme (Theorem 3). This result also provides separations between LS(D) and LS, GH(D) and GH, and GS(D) and GS (see the legend in Figure 1 for definitions).

2 Preliminaries

Formulas and Assignments. A *literal* is a negated or unnegated variable. If x is a variable, we write $\overline{x} = \neg x$ and $\overline{\neg x} = x$, and let $var(x) = var(\neg x) = x$. We sometimes call literals x and $\neg x$ the positive and negative *polarity* of variable x. If X is a set of literals, we write $\overline{X}$ for the set $\{\, \overline{x} : x \in X \,\}$. A *clause* is a finite disjunction of literals, and a *term* is a finite conjunction of literals. We call a clause *tautological* if it contains the same variable negated as well as unnegated. A *CNF formula* is a finite conjunction of non-tautological clauses. Whenever convenient, we treat clauses and terms as sets of literals, and CNF formulas as sets of sets of literals. We write $var(S)$ for the set of variables occurring (negated or unnegated) in a clause or term S, that is, $var(S) = \{\, var(\ell) : \ell \in S \,\}$. Moreover, we let $var(\phi) = \bigcup_{C \in \phi} var(C)$ denote the set of variables occurring in a CNF formula ϕ.

A *truth assignment* (or simply *assignment*) to a set X of variables is a mapping $\tau : X \to \{0,1\}$. We write $[X]$ for the set of truth assignments to X, and extend $\tau : X \to \{0,1\}$ to literals by letting $\tau(\neg x) = 1 - \tau(x)$ for $x \in X$. Let $\tau : X \to \{0,1\}$ be a truth assignment. The restriction $C[\tau]$ of a clause (term) S by τ is defined as follows: if there is a literal $\ell \in S \cap (X \cup \overline{X})$ such that $\tau(\ell) = 1$ ($\tau(\ell) = 0$) then $S[\tau] = 1$ ($S[\tau] = 0$). Otherwise, $S[\tau] = S \setminus (X \cup \overline{X})$.

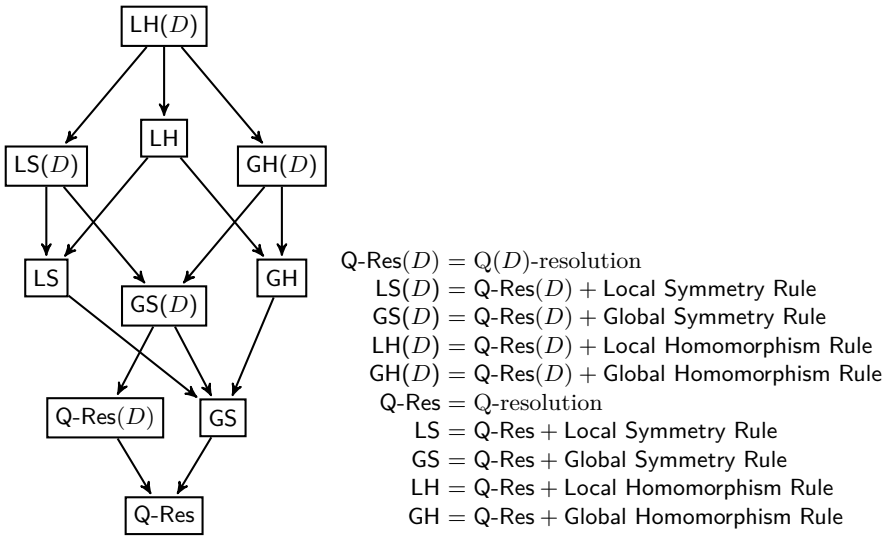


Fig. 1: Proof system map. $A \to B$ indicates that system A p-simulates system B, but B cannot p-simulate A.

The restriction $\phi[\tau]$ of a CNF formula ϕ by the assignment τ is defined $\phi[\tau] = \{\, C[\tau] : C \in \phi, C[\tau] \neq 1 \,\}$.

PCNF Formulas. A *PCNF formula* is denoted by $\Phi = \mathcal{Q}.\phi$, where ϕ is a CNF formula and $\mathcal{Q} = Q_1 X_1 \dots Q_n X_n$ is a sequence such that $Q_i \in \{\forall, \exists\}$, $Q_i \neq Q_{i+1}$ for $1 \leq i < n$, and the X_i are pairwise disjoint sets of variables. We call ϕ the *matrix* of Φ and $\mathcal{Q}$ the *(quantifier) prefix* of Φ, and refer to the X_i as *quantifier blocks*. We require that $var(\phi) = X_1 \cup \dots \cup X_n$ and write $var(\Phi) = var(\phi)$. We define a partial order $<_\Phi$ on $var(\phi)$ as $x <_\Phi y \Leftrightarrow x \in X_i, y \in X_j, i < j$. We extend $<_\Phi$ to a relation on literals in the obvious way and drop the subscript whenever Φ is understood. For $x \in var(\Phi)$ we let $R_\Phi(x) = \{\, y \in var(\Phi) : x <_\Phi y \,\}$ and $L_\Phi(x) = \{\, y \in var(\Phi) : y <_\Phi x \,\}$ denote the sets of variables *to the right* and *to the left* of x in Φ, respectively. Relative to the PCNF formula Φ, variable x is called *existential* (*universal*) if $x \in X_i$ and $Q_i = \exists$ ($Q_i = \forall$). The set of existential (universal) variables occurring in Φ is denoted $var_\exists(\Phi)$ ($var_\forall(\Phi)$). We define the set $\mathsf{lit}_\exists(\Phi)$ ($\mathsf{lit}_\forall(\Phi)$) as a set of all existential (universal) literals corresponding to $var_\exists(\Phi)$ ($var_\forall(\Phi)$), i.e., if $x \in var_\exists(\Phi)$ then both $x, \neg x \in \mathsf{lit}_\exists(\Phi)$ (resp. for the universal variable). The *length* of a PCNF formula $\Phi = \mathcal{Q}.\phi$ is given by its cardinality $|\Phi|$; the number of clauses in the matrix. The *size* of a PCNF formula $\Phi = \mathcal{Q}.\phi$ is defined as $\|\Phi\| = \sum_{C \in \phi} |C|$. If τ is an assignment, then $\Phi[\tau]$ denotes the PCNF formula $\mathcal{Q}'.\phi[\tau]$, where $\mathcal{Q}'$ is the quantifier prefix obtained from $\mathcal{Q}$ by deleting variables that do not occur in $\phi[\tau]$. *True* and *false* PCNF formulas are defined in the usual way.

Proof Systems. A *proof* of a formula F is a finite object x which certifies falsity of F in the sense that, if x is given, then falsity of F can be verified in polynomial time (proofs of falsity are also called *refutations*). A *proof system* Π is a set of proofs such that (i) elements of Π can be recognized in polynomial time, and (ii) a formula F is false if and only if Π contains a proof of F.

Let Π, Π' be proof systems. We say that Π' *p-simulates* Π if every proof $x \in \Pi$ can be transformed into a proof $x' \in \Pi'$ in polynomial time such that x and x' prove the same formula. If Π and Π' p-simulate each other, then we say that they are *p-equivalent*.

Q-Resolution. Q-resolution is a generalization of propositional resolution to PCNF formulas [6]. Q-resolution is of practical interest due to its relation to search-based QBF solvers that implement Quantified Conflict Driven Constraint Learning (QCDCL) [7, 21]: the traces of QCDCL solvers correspond to Q-resolution proofs [9, 10].

Q-resolution proof system consists of propositional resolution and the *universal reduction* rule for dealing with universally quantified variables. This system (Figure 2) was shown to be sound and complete for false PCNF formulas [6].

$$\frac{}{C}\ \text{(input clause)} \qquad \frac{C_1 \vee e \qquad \neg e \vee C_2}{C_1 \vee C_2}\ \text{(resolution)}$$

An *input clause* $C \in \varphi$ can be used as an axiom. From two clauses $C_1 \vee e$ and $\neg e \vee C_2$, where e is an existential variable, the *resolution* rule can derive the clause $C_1 \vee C_2$, provided that it is non-tautological. Here, e is an existential variable called the *pivot*.

$$\frac{C}{C \setminus \{u, \neg u\}}\ (\forall\text{-reduction})$$

The $\forall$*-reduction* rule derives the clause C from $C \vee l$ if $var(l)$ is universal and there is no existential variable $e \in var(C)$ with $l <_\Phi e$.

Fig. 2: Derivation rules of Q-resolution for a PCNF formula $\Phi = \mathcal{Q}.\phi$.

3 Dependency Schemes and Q(D)-Resolution

QCDCL generalizes the well-known DPLL procedure [8] from SAT to QSAT. In essence, DPLL is a recursive algorithm that picks a variable of its input formula and calls itself for both possible instantiations of that variable. Modern SAT solvers derived from the DPLL algorithm, delegate the choice of which variable to branch on to clever heuristics [16].

In QCDCL, the quantifier prefix imposes constraints on the order of variable assignments: a variable may be assigned only if it occurs in the leftmost quantifier block with unassigned variables. Often, this is more restrictive than necessary.

For instance, variables from disjoint subformulas may be assigned in any order. Intuitively, a variable can be assigned as long as it *does not depend* on any unassigned variable. This is the intuition underlying a generalization of QCDCL implemented in the solver DepQBF [13, 14]. Dependency schemes are mappings that associate every PCNF formula with a binary relation on its variables that refines the order of variables in the quantifier prefix.[3]

Definition 1 (Dependency Scheme). A *dependency scheme* is a mapping D that associates each PCNF formula Φ with a relation $D_\Phi \subseteq \{ (x, y) : x <_\Phi y \}$ called the *dependency relation* of Φ with respect to D.

The mapping which simply returns the prefix ordering of an input formula can be thought of as a baseline dependency scheme:

Definition 2 (Trivial Dependency Scheme). The *trivial dependency scheme* $\mathrm{D}^{\mathrm{trv}}$ associates each PCNF formula Φ with the relation $\mathrm{D}^{\mathrm{trv}}_\Phi = \{ (x, y) : x <_\Phi y \}$.

DepQBF uses a dependency relation to determine the order in which variables can be assigned: if y is a variable and there is no unassigned variable x such that (x, y) is in the dependency relation, then y is considered ready for assignment. DepQBF also uses the dependency relation to generalize the $\forall$-reduction rule used in clause learning [14]. As a result of its use of dependency schemes, DepQBF generates proofs in a generalization of Q-resolution called Q(D)-resolution [18], a proof system that takes a dependency scheme D as a parameter.

Dependency schemes can be partially ordered based on their dependency relations: if the dependency relation computed by a dependency scheme D_1 is a subset of the dependency relation computed by a dependency scheme D_2 for each PCNF formula, then D_1 is *more general* than D_2. The more general a dependency scheme, the more freedom a solver has in choosing decision variables. Currently, (aside from the trivial dependency scheme) DepQBF supports (a refined version [13, p.49] of) the *standard dependency scheme* [15]. We will work with the more general *reflexive resolution-path dependency scheme* [18], a variant of the resolution-path dependency scheme [17, 20]. This dependency scheme computes an overapproximation of variable dependencies based on whether two variables are connected by a (pair of) resolution path(s).

Definition 3 (Resolution Path). Let $\Phi = \mathcal{Q}.\phi$ be a PCNF formula and let X be a set of variables. A *resolution path* (from ℓ_1 to ℓ_{2k}) via X (in Φ) is a sequence $\ell_1, \ldots, \ell_{2k}$ of literals satisfying the following properties:

1. For all $i \in [k]$, there is a $C_i \in \phi$ such that $\ell_{2i-1}, \ell_{2i} \in C_i$.
2. For all $i \in [k]$, $var(\ell_{2i-1}) \neq var(\ell_{2i})$.
3. For all $i \in [k-1]$, $\{\ell_{2i}, \ell_{2i+1}\} \subseteq X \cup \overline{X}$.
4. For all $i \in [k-1]$, $\overline{\ell_{2i}} = \ell_{2i+1}$.

[3] The original definition of dependency schemes [15] is more restrictive than the one given here, but the additional requirements are irrelevant for the purposes of this paper.

If $\pi = \ell_1, \ldots, \ell_{2k}$ is a resolution path in Φ via X, then we say that ℓ_1 and ℓ_{2k} are *connected in* Φ (with respect to X). For every $i \in \{1, \ldots, k-1\}$, we say that π *goes through* $var(\ell_{2i})$.

One can think of a resolution path as a potential chain of implications: if each clause C_i contains exactly two literals, then assigning ℓ_1 to 0 requires setting ℓ_{2k} to 1. If, in addition, there is such a path from $\overline{\ell_1}$ to $\overline{\ell_{2k}}$, then ℓ_1 and ℓ_{2k} have to be assigned opposite values. Accordingly, the resolution path dependency scheme identifies variables connected by a pair of resolution paths as potentially dependent on each other.

Definition 4 (Dependency Pair). Let Φ be a PCNF formula and $x, y \in var(\Phi)$. We say $\{x, y\}$ is a *resolution-path dependency pair* of Φ with respect to $X \subseteq var_\exists(\Phi)$ if at least one of the following conditions holds:

- x and y, as well as $\neg x$ and $\neg y$, are connected in Φ with respect to X.
- x and $\neg y$, as well as $\neg x$ and y, are connected in Φ with respect to X.

Definition 5. The *reflexive resolution-path dependency scheme* is the mapping $\mathrm{D}^{\mathrm{rrs}}$ that assigns to each PCNF formula $\Phi = \mathcal{Q}.\phi$ the relation $\mathrm{D}^{\mathrm{rrs}}{}_\Phi = \{\, x <_\Phi y : \{x, y\}$ is a resolution-path dependency pair in Φ with respect to $R_\Phi(x) \setminus var_\forall(\Phi) \,\}$.

The derivation rules of Q(D)-resolution are shown in Figure 3. Here, as in the rest of the paper, D denotes an arbitrary dependency scheme.

$$\frac{}{C}\ \text{(input clause)} \qquad \frac{C_1 \vee e \qquad \neg e \vee C_2}{C_1 \vee C_2}\ \text{(resolution)}$$

An *input clause* $C \in \varphi$ can be used as an axiom. From two clauses $C_1 \vee e$ and $\neg e \vee C_2$, where e is an existential variable, the *resolution* rule can derive the clause $C_1 \vee C_2$, provided that $C_1 \vee C_2$ does not contain a universal variable in both polarities.

$$\frac{C}{C \setminus \{u, \neg u\}}\ (\forall\text{-reduction})$$

The $\forall$*-reduction* rule derives the clause $C \setminus \{u, \neg u\}$ from C, where $u \in var(C)$ is a universal variable such that $(u, e) \notin D_\Phi$ for every existential variable $e \in var(C)$.

Fig. 3: Derivation rules of Q(D)-resolution for a PCNF formula $\Phi = \mathcal{Q}.\phi$.

A derivation in a proof system consists of repeated applications of the derivation rules to derive a clause from the clauses of an input formula. A sequence $S = C_1, \ldots, C_k$ of clauses is a Q(D)-resolution derivation of C_k from a PCNF formula $\Phi = \mathcal{Q}.\phi$ if for each $i \in \{1, \ldots, k\}$ at least one of the following holds

1. $C_i \in \phi$ (C_i is an axiom).
2. C can be derived from C_1 and C_2 by the resolution rule.

3. C can be derived from C' by $\forall$-reduction with respect to the dependency scheme D.

The size $|S|$ of a derivation S is the number k of clauses in the sequence. A *refutation* is a derivation of the empty clause.

Proposition 1 (Slivovsky and Szeider [18]). $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$*-resolution is a complete proof system for false formulas; i.e., a PCNF formula is false if, and only if, there exists a* $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$*-resolution refutation of it.*

Definition 6 (Equality formulas [3]). For every $n \in \mathbb{N}$, the n^{th} equality formula is

$$\mathrm{EQ}(n) := \exists x_1 \dots x_n \forall u_1 \dots u_n \exists t_1 \dots t_n . \bigwedge_{i=1}^{n} \big((x_i \vee u_i \vee \overline{t_i}) \wedge (\overline{x_i} \vee \overline{u_i} \vee \overline{t_i})\big) \wedge \bigvee_{i=1}^{n} t_i .$$

For every $n \in \mathbb{N}$, the formula $\mathrm{EQ}(n)$ is *false*, and any Q-resolution refutation of $\mathrm{EQ}(n)$ has size exponential in n [3].

4 Homomorphisms

For a finite set $L \subseteq \mathsf{lit}$ of literals, a mapping $\rho : L \to \mathsf{lit}$ is a *renaming* if $\overline{\rho(\ell)} = \rho(\overline{\ell})$ for every pair $\ell, \overline{\ell} \in L$ of clashing literals in the domain of ρ. We generalize renamings to clauses and formulas in the obvious way. For a clause C, the image $\rho(C)$ under renaming may be tautological. We define $\rho_{cls}(\phi)$ as the set of all non-tautological $\rho(C)$ with $C \in \phi$. In the propositional case, the image of a resolution derivation under a renaming contains a resolution derivation. For QBFs, we have to take variable dependencies induced by the quantifier prefix into account to make sure universal reduction steps are applicable in the image. We define a notion of renaming that imposes additional restrictions to ensure that the image of a $\mathrm{Q}(D)$-resolution derivation is again a $\mathrm{Q}(D)$-resolution derivation.

Definition 7. Let $\Phi_1 = \mathcal{Q}_1.\phi_1$ and $\Phi_2 = \mathcal{Q}_2.\phi_2$ be PCNF formulas and let D be a dependency scheme for which $\mathrm{Q}(D)$-resolution is sound. For any $L \subseteq \mathsf{lit}(\Phi_1)$, a renaming $\rho : L \to \mathsf{lit}(\Phi_2)$ is a *D-renaming* from Φ_1 to Φ_2 if it satisfies the following conditions:

1. For every $\ell \in L$, $\mathit{qtype}_{\Phi_1}(\ell) = \mathit{qtype}_{\Phi_2}(\rho(\ell))$.
2. If $\rho(\ell) = \rho(\ell')$ and $\mathit{qtype}_{\Phi_1}(\ell) = \forall$, then $\ell = \ell'$.
3. If $\ell, \ell' \in L$ and $(\mathit{var}(\rho(\ell)), \mathit{var}(\rho(\ell'))) \in D_{\Phi_2}$ then $(\mathit{var}(\ell), \mathit{var}(\ell')) \in D_{\Phi_1}$.

This definition satisfies several desiderata. First, in the absence of universal variables, it boils down to a previously defined notion of homomorphisms of propositional formulas in CNF. Second, it enables us to transfer $\mathrm{Q}(D)$-resolution derivations, as stated in the following lemma.

Lemma 1. *Let $\Phi_1 = \mathcal{Q}_1.\phi_1$ and $\Phi_2 = \mathcal{Q}_2.\phi_2$ be PCNF formulas and let D be a dependency scheme for which $\mathrm{Q}(D)$-resolution is sound. If $C_1, \ldots, C_k$ is a $\mathrm{Q}(D)$-resolution derivation from $\psi \subseteq \phi_1$ in Φ_1 and $\rho : \mathsf{lit}(\psi) \to \mathsf{lit}(\Phi_2)$ is a D-renaming such that $\rho(C_k)$ is non-tautological and $\rho_{cls}(\psi) \subseteq \phi_2$, then $\rho(C_1), \ldots, \rho(C_k)$ contains a $\mathrm{Q}(D)$-resolution derivation of $C'_k \subseteq \rho(C_k)$ from clauses $\rho_{cls}(\psi)$ in Φ_2.*

Proof. We proceed by induction on the length k of the derivation and distinguish three cases. First, if $C_k \in \phi_1$ is an initial clause and $\rho(C_k)$ is non-tautological then $\rho(C_k) \in \phi_2$ by assumption. Second, if C_k is derived from clause C_i with $1 \leq i < k$ by universal reduction, then $C_k = C_i \setminus \{\ell_u\}$ for a universal literal ℓ_u with $\mathit{var}(\ell_u) = u$, and $(u, e) \notin D_{\Phi_1}$ for every existential variable e occurring in C_i. We argue that $\rho(C_i)$ is non-tautological. Towards a contradiction assume that $\rho(C_i)$ is tautological. Since $\rho(C_k)$ is assumed to be non-tautological the only way for $\rho(C_i) = \rho(C_k) \cup \{\rho(\ell_u)\}$ to be tautological is that $\overline{\rho(\ell_u)} \in \rho(C_i)$. Because ρ preserves quantifier types and C_i is non-tautological, there must be a universal literal $\ell' \in C_i$ with $\mathit{var}(\ell') \neq u$ such that $\rho(\ell') = \overline{\rho(\ell_u)}$. That means $\rho(\overline{\ell'}) = \rho(\ell_u)$ and thus $\ell' = \ell_u$ by Property 2 and in particular $\mathit{var}(\ell') = \mathit{var}(\ell_u) = u$, a contradiction. Thus $\rho(C_i)$ is non-tautological and we can apply the induction hypothesis to conclude that $\rho(C_1), \ldots, \rho(C_i)$ contains a $\mathrm{Q}(D)$-resolution derivation of $C'_i \subseteq \rho(C_i)$ from $\rho_{cls}(\psi)$ in Φ_2. By Property 1, every existential literal in $C'_i \subseteq \rho(C_i)$ is the image of an existential literal in C_i. Property 3 ensures that $(\rho(u), \rho(e)) \notin D_{\Phi_2}$ for every existential variable $\rho(e)$ occurring in $\rho(C_i)$, so a clause $C'_k \subseteq \rho(C_k)$ can be obtained from $C'_i \subseteq \rho(C_i)$ by universal reduction.

Finally, let C_k be derived by resolution on pivot variable e from C_i and C_j with $1 \leq i < j < k$. Assume without loss of generality that $e \in C_i$ and $\neg e \in C_j$, so that $C_i \subseteq C_k \cup \{e\}$ and $C_j \subseteq C_k \cup \{\neg e\}$. If $\rho(C_i)$ and $\rho(C_j)$ are both non-tautological we can apply the induction hypothesis to obtain $\mathrm{Q}(D)$-resolution derivations of clauses $C'_i \subseteq \rho(C_i)$ and $C'_j \subseteq \rho(C_j)$ from Φ_2. If the pivot variable is contained in both clauses we obtain $C'_k \subseteq \rho(C_k)$ by resolution, otherwise we choose as C'_k one among the clauses C'_i and C'_j that does not contain the pivot. Otherwise, since $\rho(C_k)$ is non-tautological, the clause $\rho(C_i) \subseteq \rho(C_k) \cup \{\rho(e)\}$ can be tautological only if there is a literal $\ell \in C_i$ such that $\rho(\ell) = \rho(\neg e)$. Symmetrically, the clause $\rho(C_j) \subseteq \rho(C_k) \cup \{\rho(\neg e)\}$ can be tautological only if there is a literal $\ell' \in C_j$ such that $\rho(\ell') = \rho(e)$. It follows that at most one of $\rho(C_i)$ and $\rho(C_j)$ can be tautological. Assume without loss of generality that $\rho(C_i)$ is tautological and let $\ell \in C_i$ such that $\rho(\ell) = \rho(\neg e)$. Then $\rho(C_j) \subseteq \rho(C_k) \cup \rho(\neg e) = \rho(C_k)$ and there is a $\mathrm{Q}(D)$-resolution derivation of $C'_j \subseteq \rho(C_k)$ from Φ_2 by induction hypothesis. □

This result states that if we apply a D-renaming to each clause in a $\mathrm{Q}(D)$-resolution derivation, a subsequence of the resulting sequence of clauses is a $\mathrm{Q}(D)$-resolution derivation of a clause subsuming the image of the final clause in the original derivation. In particular, the length of the derivation can only decrease.

If any of the conditions of Definition 7 is dropped, then Lemma 1 no longer holds, in the sense that we might obtain derivations that are not syntactically

correct. Mapping existential to universal literals may introduce tautologies that are removed by universal reduction, which is unsound in general and forbidden in Q(D)-resolution. The same problem can occur if universal literals are not mapped in an injective way. If universal literals can be mapped to existential literals, or independence according to the dependency scheme D is not preserved, universal reduction may not be applicable in the image.

A D-renaming ρ from Φ to itself is a *D-homomorphism* from clause set ϕ to clause set ψ with respect to Φ if $\rho(\phi) \subseteq \psi$. The set of all D-homomorphisms from ϕ to ψ with respect to Φ is denoted $\mathsf{Hom}_{\Phi}^{D}(\phi, \psi)$. D-homomorphisms generalize *symmetries* of PCNF formulas, which are renamings that may only change the order of variables within quantifier blocks [11]: any such mapping is bijective and preserves the type of a variable, as well as dependencies indicated by the trivial dependency scheme.

5 The Homomorphism Rule

Let $\Phi = \mathcal{Q}.\phi$ be PCNF formula and let D be a tractable dependency scheme for which Q(D)-resolution is sound. Consider a Q(D)-resolution derivation of a clause C from clauses $\psi \subseteq \phi$ in Φ. If there is a homomorphism $\varphi \in \mathsf{Hom}_{\Phi}^{D}(\psi, \phi)$ then the *local homomorphism rule* can derive the clause $\varphi(C)$. We call the restricted form of this rule, which can only be applied if $\psi = \phi$ the *global homomorphism rule*. The proof systems $\mathsf{GH}(D)$ and $\mathsf{LH}(D)$ arise from Q(D)-resolution by addition of the global and local homomorphism rule, respectively.

We present an example to illustrate the local homomorphism rule. Consider the PCNF formula $\Phi = \mathcal{Q}.\phi$ where $\mathcal{Q} = \forall a\, b\, \exists x\, \forall c\, \exists y\, z\, w$ and $\phi = \{C_1, C_2, C_3, C_4, C_5\}$ with $C_1 = \{\neg a, \neg y, z\}, C_2 = \{c, y, w\}, C_3 = \{c, \neg z\}, C_4 = \{b, \neg x\}, C_5 = \{\neg a, x\}$. We use trivial dependency scheme $\mathrm{D}^{\mathrm{trv}}_{\Phi}$ for this illustration. Consider the following Q(D)-resolution derivation S from the formula Φ:

C_1	axiom;
C_2	axiom;
$\{\neg a, c, w, z\}$	resolution from C_1 and C_2;
C_3	axiom;
$\{\neg a, c, w\}$	resolution from $\{\neg a, c, w, z\}$ and C_3.

Using the above resolution derivation S, we derive the clause $\{\neg a, c, w\}$ from Φ. We define a non-injective mapping ρ over the subset of variables of Φ as follows; $\rho(a) = a$, $\rho(c) = b$, $\rho(y) = \rho(w) = \neg x$ and $\rho(z) = x$. By the definition of renaming the complement of the literal takes the negation of the value defined by ρ, for example, $\rho(\neg z) = \neg x$. Note that the renaming jumps between the quantifier blocks by allowing the mapping of literals from one quantifier block to another. Let $\psi = \{C_1, C_2, C_3\} \subseteq \phi$, the image of ψ under the renaming ρ is $\rho(\psi) = \{C_4, C_5\} \subseteq \phi$. All the three restrictions of Definition 7 are satisfied, hence the renaming $\rho \in \mathsf{Hom}_{\Phi}^{D}(\psi, \phi)$. Thus, by using the local homomorphism rule, we can obtain the clause $\rho(\{\neg a, c, w\}) = \{\neg a, b, \neg x\}$ and add it to the matrix ϕ.

Proposition 2. *The systems* $\mathsf{GH}(D)$ *and* $\mathsf{LH}(D)$ *are sound for any dependency scheme* D *such that* $\mathrm{Q}(D)$*-resolution is sound. That is, a PCNF formula that has a refutation in* $\mathsf{GH}(D)$ *or* $\mathsf{LH}(D)$ *is false.*

Proof. Let $\Phi = \mathcal{Q}.\phi$ be a PCNF formula and let $S = C_1, \ldots, C_k$ be an $\mathsf{LH}(D)$-refutation of Φ. If S does not use the local homomorphism rule, then Φ is false by the soundness of $\mathrm{Q}(D)$-resolution. Otherwise, let C_j be derived from C_i by application of the local homomorphism rule, where $1 \leq i < j \leq k$. That is, there is a subset of clauses $\psi \subseteq \phi$ such that C_i can be derived from ψ in Φ and $\varphi \in \mathsf{Hom}^D_\Phi(\psi, \phi)$ is a homomorphism with $\varphi(C_i) = C_j$. By Lemma 1, the sequence $\varphi(C_1), \ldots, \varphi(C_i)$ contains a $\mathrm{Q}(D)$-resolution derivation of $C'_i \subseteq \varphi(C_i)$ from clauses $\varphi(\psi) \subseteq \phi$ in Φ. We can replace C_j with the corresponding derivation and (possibly) simplify the proof to obtain an $\mathsf{LH}(D)$-refutation of Φ with one less application of the local homomorphism rule. In this way, we can get rid of all uses of the local homomorphism rule one by one and obtain a $\mathrm{Q}(D)$-resolution refutation of Φ. □

6 Lifting Lower Bounds from $\mathbf{Q}(\mathrm{D}^{\mathrm{rrs}})$-Resolution to $\mathbf{LH}(\mathrm{D}^{\mathrm{rrs}})$

Let D be an arbitrary but fixed tractable and sound dependency scheme. Let $\Phi = \mathcal{Q}.\phi$ be a PCNF formula with a 3CNF matrix such that each clause contains at least two literals and cannot be simplified by universal reduction. Moreover, we assume that each clause contains at most one universal literal. Observe that any formula not solved by unit propagation can be transformed into this format by applying unit propagation and splitting clauses.

From Φ we construct a formula $\Phi^\circ = \mathcal{Q}^\circ.\phi^\circ$ as follows. Let $\ell_1, \ldots, \ell_s$ be the sequence of literals appearing in ϕ. For each existential literal ℓ_j we introduce new existential variables $y_{j,1}, \ldots, y_{j,j+9}$ and z_j at the same quantifier depth and create a chain of binary clauses

$$L'_j = \{\{\neg y_{j,1}, y_{j,2}\}, \{\neg y_{j,2}, y_{j,3}\}, \ldots, \{\neg y_{j,j+8}, y_{j,j+9}\}, \{\neg y_{j,j+9}, \ell_j\}\}.$$

We add the variable z_j to all clauses of L'_j except the fourth and (j+7)th one to obtain a formula L_j, called the *link* of ℓ_j. The clause widths of a link yield a sequence

$$3\ 3\ 3\ 2\ \underbrace{3\ \ldots\ 3}_{j+1 \text{ times}}\ 2\ 3\ 3$$

that uniquely identifies an existential literal ℓ_j.

Next, we replace each existential literal of Φ by the first literal in its link. More specifically, if $E_i = \{\ell_j, \ell_{j+1}, \ell_{j+2}\}$ is a clause of ϕ, we let

$$E^\circ_i := \{\, y_{k,1} : \ell_k \text{ is existential, } j \leq k \leq j+2 \,\} \cup \{\, \ell \in E_i : \ell \text{ is universal} \,\}.$$

We combine the above definitions to obtain the formula

$$\phi^\circ := \{E_1^\circ, \ldots, E_m^\circ\} \cup \bigcup_{j=1}^{s} (L_j \cup \{\{\neg z_j\}\}).$$

We refer to clauses E_i° as *main clauses*, clauses in L_j as *link clauses*, and to unit clauses $\{z_j\}$ as *auxiliary clauses*.

Since link clauses only contain existential variables, homomorphisms from L_j into ϕ° coincide with homomorphisms of propositional formulas defined by Szeider [19], and so the following result carries over to our setting.

Lemma 2 (Szeider [19]). $\mathsf{Hom}^D_{\Phi^\circ}(L_j, \phi^\circ) = \{id_{L_j}\}$ *for any* $1 \le j \le s$.

The formulas Φ and Φ° have the same dependencies according to the resolution-path dependency scheme.

Lemma 3. *For every existential literal* $\ell_j \in \mathsf{lit}(\Phi)$ *and* $y_{j,i}$ *with* $1 \le i \le j+9$, *we have* $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(\ell_j) = \mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(y_{j,i}) = \mathrm{D}^{\mathrm{rrs}}{}_{\Phi}(\ell_j)$, *as well as* $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(z_j) = \emptyset$.

Proof. There is a natural correspondence between resolution paths of Φ and Φ°. Each resolution path in Φ can be extended to a resolution path in Φ° by using links. Formally, if $\ell_{j_1}, \ell_{j_2}, \ldots, \ell_{j_{2k}}$ is a resolution path of Φ we obtain a resolution path of Φ' by replacing each literal $\ell_{j_{2i-1}}$ for $1 \le i \le k$ by the sequence

$$\ell_{j_{2i-1}}, \neg y_{j_{2i-1}, j_{2i-1}+9}, y_{j_{2i-1}, j_{2i-1}+9}, \ldots, y_{j_{2i-1},2}, \neg y_{j_{2i-1},1}, y_{j_{2i-1}}$$

of literals from $L'_{j_{2i-1}}$ in reverse order, and each adjacent literal $\ell_{j_{2i}}$ for $1 \le i \le k$ by the sequence

$$y_{j_{2i},1}, \neg y_{j_{2i},1}, y_{j_{2i},2}, \ldots, y_{j_{2i}, j_{2i}+9}, \neg y_{j_{2i}, j_{2i}+9}, \ell_{2i}$$

of literals from $L'_{j_{2i}}$ in order. Conversely, any resolution path of Φ° with original literals of Φ as endpoints can be transformed into a resolution path of Φ by removing sequences of link literals. Since link variables $y_{j,i}$ are introduced at the same quantifier depth as $var(\ell_j)$ and $var_\forall(\Phi) = var_\forall(\Phi^\circ)$, it follows that $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(\ell_j) = \mathrm{D}^{\mathrm{rrs}}{}_{\Phi}(\ell_j)$. Further, a dependency-inducing resolution path of Φ° from a universal variable u to its negation $\neg u$ goes through a literal ℓ_j if, and only if, it goes through all the link variables $y_{j,i}$, so $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(\ell_j) = \mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(y_{j,i})$ for $1 \le i \le j+9$. Finally, the variables z_j occur negatively exclusively in the unit clauses $(\neg z_j)$, so $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(z_j) = \emptyset$. □

For a QBF proof system Π and a false formula Φ, let $\mathsf{PSize}_\Pi(\Phi)$ denote the size of a shortest Π-refutation of Φ.

Corollary 1. $\mathsf{PSize}_{\mathsf{Q}(\mathrm{D}^{\mathrm{rrs}})\text{-}\mathsf{Res}}(\Phi^\circ) \le \mathsf{PSize}_{\mathsf{Q}(\mathrm{D}^{\mathrm{rrs}})\text{-}\mathsf{Res}}(\Phi) + O(\|\Phi\|^2)$.

Proof. The original matrix ϕ can be obtained from ϕ° by resolving each existential literal ℓ_j in a main clause E_i with the link clauses in L_j and the auxiliary clause $\{\neg z\}$. This requires $O(j)$ steps for each literal ℓ_j with $1 \le j \le s$, and $s \in O(\|\Phi\|)$. Let S denote the corresponding $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution derivation. As resolution-path dependencies are preserved by Lemma 3, a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation S' of Φ can be appended to S so as to obtain a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of Φ°. □

We want to show that any $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$-refutation of the "rigid" version Φ° of a PCNF formula Φ can be mapped back to a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of the original formula Φ. To do this, we introduce a new existential variable z and define a renaming $\rho : \mathsf{lit}(\Phi^\circ) \to \mathsf{lit}(\Phi) \cup \{z\}$ as follows:

$$\begin{aligned}
&\rho(u) := u, && \text{for each universal variable } u;\\
&\rho(y_{j,i}) := \ell_j, && \text{for } 1 \leq j \leq s, 1 \leq i \leq j+9;\\
&\rho(\ell_j) := \ell_j, && \text{for } 1 \leq j \leq s;\\
&\rho(z_j) := z, && \text{for } 1 \leq j \leq s.
\end{aligned}$$

With this renaming, every link clause becomes tautological, every auxiliary clause $\{\neg z_j\}$ becomes $\rho(\{\neg z_j\}) = \{\neg z\}$, and main clauses E_i° are mapped back to original clauses $\rho(E_i^\circ) = E_i$. Hence $\rho_{cls}(\phi^\circ)$ as defined in Section 4 is nothing but $\phi \cup \{\{\neg z\}\}$, and $\neg z$ is a pure literal of $\rho_{cls}(\phi^\circ)$.

Lemma 4. *The mapping $\rho : \mathsf{lit}(\Phi^\circ) \to \mathsf{lit}(\Phi) \cup \{z\}$ is a $\mathrm{D}^{\mathrm{rrs}}$-renaming from Φ° to $\Phi_z = \exists z \mathcal{Q}.\phi \cup \{\{\neg z\}\}$.*

Proof. By construction, the mapping preserves quantifier types and is injective with respect to universal variables. Moreover, the new variable z and clause $\{\neg z\}$ do not affect resolution-path dependencies in Φ_z and z has no dependencies itself, so $\mathrm{D}^{\mathrm{rrs}}{}_{\Phi^\circ}(v) = \mathrm{D}^{\mathrm{rrs}}{}_{\Phi_z}(\rho(v))$ holds for every variable $v \in \mathit{var}(\Phi^\circ)$ by Lemma 3. □

The following result establishes that "interesting" $\mathsf{LH}(D)$-derivations using at least two main clauses from Φ° cannot use the homomorphism rule in a non-trivial way.

Lemma 5. *Let $S = C_1, \ldots, C_k$ be a $\mathrm{Q}(D)$-resolution derivation from $\phi' \subseteq \phi^\circ$ in Φ° such that no subsequence of S is a $\mathrm{Q}(D)$-resolution derivation of C_k in Φ. If S contains at least two main clauses as input clauses then $\rho(C_k) = \rho(\varphi(C_k))$ for any homomorphism $\varphi \in \mathsf{Hom}^D_{\Phi^\circ}(\phi', \phi^\circ)$.*

A formal proof of Lemma 5 is rather tedious and has to be omitted due to space constraints, but the underlying intuition is fairly simple. Since main clauses are only connected through links, two main clauses can take part in a resolution proof only if the two links corresponding to the pivot literal are present, which by Lemma 2 leaves the identity as the only homomorphism that can be applied to the main clauses or the clauses in the links. Having identified such a "rigid" part of a proof, one can then show that any other clause C that participates in the proof has the same image under ρ as its homomorphic image $\varphi(C)$, in symbols $\rho(C) = \rho(\varphi(C))$.

The next result states that $\mathsf{LH}(D)$-derivations that use at most one single main clause cannot be too long.

Lemma 6. *For any $\mathrm{Q}(D)$-resolution derivation S of a clause C in Φ° with at most one main clause among its input clauses, there is a $\mathrm{Q}(D)$-resolution derivation S' of $C' \subseteq C$ in Φ° of length $O(|\Phi^\circ|^3)$.*

Proof. Let $S = C_1, \ldots, C_k$ be a $\mathrm{Q}(D)$-resolution derivation of $C = C_k$ in Φ° that contains at most one main clause among its input clauses. Any remaining input clauses are auxiliary or link clauses. We construct the derivation S' by first resolving each link clause containing a literal z_j with the auxiliary clause $\{\neg z_j\}$. This requires at most $|\Phi^\circ|$ resolution steps. We then proceed as in S while (possibly) omitting resolution steps on variables z_j. The length of S' can be crudely bounded as follows. After resolving out z_j we are left with binary link clauses L'_j and a single main clause of size at most three. Any $\mathrm{Q}(D)$-resolution derivation starting from these clauses can derive clauses of size at most three, and there are $O(|\Phi^\circ|^3)$ such clauses. □

Lemma 7. $\mathsf{PSize}_{\mathsf{Q}(\mathrm{D}^{\mathrm{rrs}})\text{-}\mathsf{Res}}(\Phi) \leq \mathsf{PSize}_{\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})}(\Phi^\circ) \cdot O(|\Phi^\circ|^3)$.

Proof. Let $C_1, \ldots, C_k$ be an $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$-refutation of Φ°. By Lemma 4, the mapping ρ is a $\mathrm{D}^{\mathrm{rrs}}$-renaming from Φ° to Φ_z, so if no clause of $C_1, \ldots, C_k$ is derived by the local homomorphism rule, we can apply Lemma 1 and conclude that $\rho(C_1), \ldots, \rho(C_k)$ is a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of Φ. Otherwise, we are going to turn $\rho(C_1), \ldots, \rho(C_k)$ into a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of Φ that is not too much larger. Suppose clause C_j is derived from C_i using the local homomorphism rule for some $1 \leq i < j \leq k$. That is, $C_1, \ldots, C_i$ contains a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution derivation S of C_i from clause set $\phi' \subseteq \phi^\circ$ in Φ°, and there is a homomorphism $\varphi \in \mathsf{Hom}^D_\Phi(\phi', \phi^\circ)$ such that $\varphi(C_i) = C_j$. If the derivation of C_i involves at most one main clause then its size is in $O(|\Phi^\circ|^3)$ by Lemma 6. By Lemma 1, the sequence $\varphi(C_1), \ldots, \varphi(C_i)$ contains a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution derivation of the clause $\varphi(C_i) = C_j$. We simply replace $\rho(C_j)$ by the image $\rho(\varphi(C_1)), \ldots, \rho(\varphi(C_i))$ of this entire derivation, increasing the proof size by $O(|\Phi^\circ|^3)$. Otherwise, the derivation S uses at least two main clauses. In this case, Lemma 5 tells us that $\rho(C_i) = \rho(\varphi(C_i)) = \rho(C_j)$, so we can simply use $\rho(C_i)$ instead of $\rho(C_j)$. In this manner, we obtain a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of Φ_z of size $k \cdot O(|\Phi^\circ|^3)$. Since $\neg z$ is pure in Φ_z, the refutation cannot contain the clause $\{\neg z\}$, and is in fact a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of the original formula Φ. □

7 Separating $\mathbf{LH}(\mathrm{D}^{\mathrm{rrs}})$ from LH

In this section, we will use Corollary 1 and Lemma 7 to lift known separations of Q-resolution systems *without* the homomorphism rule to systems *with* the homomorphism rule. First, we show that our assumption from the previous section that the formula Φ has a matrix in 3CNF does not affect certain semantic lower bound techniques. More specifically, we show that long clauses occurring in the equality formulas can be split without affecting the *cost* of these formulas [3].

Definition 8 (Universal Winning Strategy). For any set V of variables, let $[V]$ denote the set of assignments of V. Let $\Phi = \forall U_1 \exists E_1 \ldots \forall U_n \exists E_n . \phi$ be a false QBF. A *universal strategy* for Φ is a sequence $S = (S_i)_{1 \leq i \leq n}$ of functions $S_i : [E_1 \cup \cdots \cup E_{i-1}] \to [U_i]$. The *response* of S to an existential assignment τ :

$var_\exists(\Phi) \to \{0,1\}$ is the assignment $S(\tau) = \bigcup_{i=1}^{n} S_i(\tau|_{E_1 \cup \cdots \cup E_{i-1}})$. The universal strategy S is a *universal winning strategy* if the assignment $\tau \cup S(\tau)$ satisfies the matrix ϕ for every existential assignment $\tau : var_\exists(\Phi) \to \{0,1\}$.

Definition 9 (Cost). Let $\Phi = \forall U_1 \exists E_1 \ldots \forall U_n \exists E_n . \phi$ be a false QBF and let $S = (S_i)_{1 \leq i \leq n}$ be a universal winning strategy for Φ. The *cost* of S is defined as $cost(S) = \max\{\, |\mathrm{rng}(S_i)| : 1 \leq i \leq n \,\}$, where $\mathrm{rng}(f)$ denotes the range of function f. The *cost* of the QBF Φ is the minimum cost of any universal winning strategy for Φ.

The cost of a false QBF Φ is a lower bound on the size of any Q-resolution refutation of Φ.

Theorem 1 (Beyersdorff, Blinkhorn, and Hinde [3]). *Let $C_1, \ldots, C_k$ be a Q-resolution refutation of a QBF Φ. Then $k \geq cost(\Phi)$.*

Lemma 8. *Let $\Phi = \mathcal{Q}.\phi \cup \{C\}$ be a PCNF formula with clause $C = C_1 \cup C_2$ and let y be a fresh variable. Further, let $\Phi' = \mathcal{Q}\, \exists y.\phi \cup \{C_1 \cup \{y\}, C_2 \cup \{\neg y\}\}$ be the formula obtained from Φ by splitting C. Then Φ and Φ' have the same universal winning strategies.*

Corollary 2. *If Φ^* is obtained from Φ by splitting clauses, then Φ^* and Φ have the same cost.*

Proposition 3 (Beyersdorff, Blinkhorn, and Hinde [3]). *For each $n \in \mathbb{N}$, $\mathrm{EQ}(n)$ has cost 2^n.*

This implies an exponential proof size lower bound by Theorem 1. At the same time, it is known that these formulas have short $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutations.

Theorem 2 (Blinkhorn and Beyersdorff [2]). *For each $n \in \mathbb{N}$, $\mathrm{EQ}(n)$ has a $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation of size $O(n)$.*

We are now ready to prove an exponential separation of $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$ from $\mathsf{LH}(\mathrm{D}^{\mathrm{trv}})$.

Theorem 3. *There is an infinite sequence $(\Phi_n)_{n \in \mathbb{N}}$ of false formulas such that the shortest $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$-refutation of Φ_n is polynomial in n but any $\mathsf{LH}(\mathrm{D}^{\mathrm{trv}})$-refutation of Φ_n has length $2^{\Omega(n)}$.*

Proof. For each $n \in \mathbb{N}$, let $\mathrm{EQ}^*(n)$ denote a QBF obtained from $\mathrm{EQ}(n)$ by splitting clauses until each clause contains at most three literals in total and at most one universal literal. By Proposition 3 and Corollary 2, $\mathrm{EQ}^*(n)$ has cost 2^n and thus requires Q-resolution refutations of size at least 2^n by Theorem 1. At the same time, since the original formula can be obtained by resolving on existential variables introduced by splitting, and splitting does not introduce new resolution-path dependencies among the original variables, Theorem 2 implies that $\mathrm{EQ}^*(n)$ has a linear-size $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutation. Now, consider the "rigid" versions $\mathrm{EQ}^\circ(n)$ of $\mathrm{EQ}^*(n)$. Clearly, the size of $\mathrm{EQ}^\circ(n)$ is polynomially bounded in the size of $\mathrm{EQ}(n)$. By Theorem 2 and Corollary 1, the formulas $\mathrm{EQ}^\circ(n)$ have polynomial-size $\mathrm{Q}(\mathrm{D}^{\mathrm{rrs}})$-resolution refutations, and thus also

polynomial-size $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$-refutations. On the other hand, Lemma 7 tells us that any $\mathsf{LH}(\mathrm{D}^{\mathrm{trv}})$-refutation of $\mathrm{EQ}^{\circ}(n)$ can be shorter than a Q-resolution refutation of $\mathrm{EQ}^{*}(n)$ by at most a polynomial factor. □

Since the short $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$-refutations in the above theorem do not use the local homomorphism rule, analogous separations hold for weaker systems.

Corollary 3. *There is an infinite sequence $(\Phi_n)_{n\in\mathbb{N}}$ of false formulas such that the shortest $\Pi(\mathrm{D}^{\mathrm{rrs}})$-refutation of Φ_n is polynomial in n but any $\Pi(\mathrm{D}^{\mathrm{trv}})$-refutation of Φ_n has length $2^{\Omega(n)}$, for $\Pi \in \{\mathsf{GH}, \mathsf{LS}, \mathsf{GS}\}$.*

8 Concluding Remarks

We have lifted the local and the global homomorphism rule from propositional resolution to the quantified case, introducing several generalizations, including the use of dependency schemes. Although we have established an exponential lower bound for the most general system LH without a dependency scheme, we left open to prove an exponential lower bound for $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$.

The systems introduced here are incomparable with the proof systems LQU+ [1] and IR-calc [4]. Since they are stronger than GS, there are classes of formulas that are easy for our systems and hard for LQU+ and IR-calc [11]. For the converse, we can apply our construction to the QParity [4] formulas and make them rigid, so that they are hard for LH. Both LQU+ and IR-calc can derive the original formula and then proceed with short refutations of QParity.

There are several possibilities for further strengthening $\mathsf{LH}(\mathrm{D}^{\mathrm{rrs}})$. One possibility is to consider a suitably defined *dynamic homomorphism rule* [19] which considers homomorphisms between sets of *derived* clauses. Neither of the lower bounds established in this paper applies to proof systems that use such a dynamic rule: all the modifications made to the input formula to achieve rigidity can be undone by a polynomial number of resolution steps so that after these steps symmetries and homomorphisms can be exploited to get short proofs.

Another possibility, somewhat related to the dynamic systems discussed above, is based on the idea of *symmetry recomputation*, as considered by Blinkhorn and Beyersdorff [5], which exploits symmetries of the input formula after the application of a partial assignment. We think that this idea can be combined with our homomorphism systems.

All these ideas for even stronger proof systems for QBF give rise to challenging theoretical questions that include separation results, as well as lower and upper bounds. Another interesting line of research is concerned with the possibility of utilizing the strength of the various homomorphism rules considered in this paper within a QBF solver.

References

1. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Sinz, C., Egly, U. (eds.) Theory and Applications of Satisfiability

Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8561, pp. 154–169. Springer Verlag (2014)
2. Beyersdorff, O., Blinkhorn, J.: Dynamic QBF dependencies in reduction and expansion. ACM Trans. Comput. Log. **21**(2), 8:1–8:27 (2020)
3. Beyersdorff, O., Blinkhorn, J., Hinde, L.: Size, cost, and capacity: A semantic technique for hard random qbfs. Logical Methods in Computer Science **15**(1) (2019)
4. Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. TOCT **11**(4), 26:1–26:42 (2019)
5. Blinkhorn, J., Beyersdorff, O.: Proof complexity of QBF symmetry recomputation. In: Janota, M., Lynce, I. (eds.) Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11628, pp. 36–52. Springer Verlag (2019)
6. Büning, H.K., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Information and Computation **117**(1), 12–18 (1995)
7. Cadoli, M., Schaerf, M., Giovanardi, A., Giovanardi, M.: An algorithm to evaluate Quantified Boolean Formulae and its experimental evaluation. Journal of Automated Reasoning **28**(2) (2002)
8. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM **5**, 394–397 (1962)
9. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8312, pp. 291–308. Springer Verlag (2013)
10. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause/term resolution and learning in the evaluation of Quantified Boolean Formulas. J. Artif. Intell. Res. **26**, 371–416 (2006)
11. Kauers, M., Seidl, M.: Short proofs for some symmetric quantified Boolean formulas. Inform. Process. Lett. **140**, 4–7 (2018)
12. Krishnamurthy, B.: Short proofs for tricky formulas. Acta Informatica **22**, 253–275 (1985)
13. Lonsing, F.: Dependency Schemes and Search-Based QBF Solving: Theory and Practice. Ph.D. thesis, Johannes Kepler University, Linz, Austria (Apr 2012)
14. Lonsing, F., Biere, A.: Integrating dependency schemes in search-based QBF solvers. In: Strichman, O., Szeider, S. (eds.) Theory and Applications of Satisfiability Testing - SAT 2010. Lecture Notes in Computer Science, vol. 6175, pp. 158–171. Springer Verlag (2010)
15. Samer, M., Szeider, S.: Backdoor sets of quantified Boolean formulas. Journal of Automated Reasoning **42**(1), 77–97 (2009)
16. Silva, J.P.M.: The impact of branching heuristics in propositional satisfiability algorithms. In: Barahona, P., Alferes, J.J. (eds.) Progress in Artificial Intelligence, 9th Portuguese Conference on Artificial Intelligence, EPIA '99, Évora, Portugal, September 21-24, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1695, pp. 62–74. Springer Verlag (1999)
17. Slivovsky, F., Szeider, S.: Quantifier reordering for QBF. Journal of Automated Reasoning **56**(4), 459–477 (2016). https://doi.org/10.1007/s10817-015-9353-1, http://dx.doi.org/10.1007/s10817-015-9353-1

18. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. Theoretical Computer Science **612**, 83–101 (2016). https://doi.org/10.1016/j.tcs.2015.10.020, http://dx.doi.org/10.1016/j.tcs.2015.10.020
19. Szeider, S.: The complexity of resolution with generalized symmetry rules. Theory Comput. Syst. **38**(2), 171–188 (2005)
20. Van Gelder, A.: Variable independence and resolution paths for quantified boolean formulas. In: Lee, J. (ed.) Principles and Practice of Constraint Programming - CP 2011. Lecture Notes in Computer Science, vol. 6876, pp. 789–803. Springer Verlag (2011)
21. Zhang, L., Malik, S.: The quest for efficient boolean satisfiability solvers. In: Brinksma, D., Larsen, K.G. (eds.) Computer Aided Verification: 14th International Conference (CAV 2002). Lecture Notes in Computer Science, vol. 2404, pp. 17–36 (2002)