

Strong (D)QBF Dependency Schemes via Tautology-free Resolution Paths[★]

Olaf Beyersdorff, Joshua Blinkhorn, and Tomáš Peitl

Friedrich-Schiller-Universität, Jena, Germany

Abstract. We suggest a general framework to study dependency schemes for dependency quantified Boolean formulas (DQBF). As our main contribution, we exhibit a *new tautology-free DQBF dependency scheme* that generalises the reflexive resolution path dependency scheme. We establish soundness of the tautology-free scheme, implying that it can be used in any DQBF proof system. We further explore the power of DQBF resolution systems parameterised by dependency schemes and show that our new scheme results in exponentially shorter proofs in comparison to the reflexive resolution path dependency scheme when used in the expansion DQBF system $\forall\text{Exp}+\text{Res}$.

On QBFs, we demonstrate that our new scheme is exponentially stronger than the reflexive resolution path dependency scheme when used in Q-Resolution, thus resulting in the strongest QBF dependency scheme known to date.

Keywords: DQBF · Dependency Schemes · Proof Complexity · QBF

1 Introduction

Quantified Boolean formulas (QBF) have been intensively studied in the past decade, both practically and theoretically. On the practical side, there have been huge improvements in QBF solving [28]. These build on the success of SAT solving [35], but also incorporate new ideas genuine to the QBF domain, such as expansion solving [19] and dependency schemes [31]. Due to its PSPACE completeness, QBF solving is relevant to many application domains that cannot be efficiently encoded into SAT [16,21,24]. On the theoretical side, there is a substantial body of QBF proof complexity results (e.g. [3,6,8–10]), which calibrates the strength of solvers while guiding their development.

In QBF solving, a severe technical complication is that *variable dependencies* stemming from the linear order of quantification¹ must be respected when assigning variables. In contrast, a SAT solver can assign variables in any order, granting complete freedom to decision heuristics, which are crucial for performance. As a remedy, QBF researchers have developed *dependency schemes*. Dependency schemes try to determine algorithmically which of the variable dependencies are essential, thereby identifying *spurious* dependencies which can be safely disregarded. The result is greater freedom for decision heuristics.

Practical QBF solving utilises dependency schemes, for example the solvers DepQBF [22] and Qute [25,26], and experiments show dependency-aware solving is particularly competitive on QBFs with high quantifier complexity [18,23].

The performance gains are also underlined by theoretical findings. There is a sequence of results [7,27,34] that establish how and when dependency schemes are sound to use with a QBF proof system, such as the central QBF resolution systems Q-Resolution [20] and long-distance Q-Resolution [2]. In [6] it is demonstrated that using the reflexive resolution path dependency scheme (\mathcal{D}^{rts} [34]) in Q-Resolution can exponentially shorten proofs.

[★] This research was supported by grants from FWF (J-4361), the John Templeton Foundation (grant no. 60842), and the Carl Zeiss Foundation.

¹ The standard input for solvers is a prenex QBF.

While dependency schemes aim to algorithmically determine spurious dependencies, *dependency quantified Boolean formulas* (DQBF) allow to directly express variable dependencies by specifying, for each existential variable x , a *dependency set* of universal variables on which x depends. This is akin to the use of Henkin quantifiers in first-order logic [17]. Compared to QBFs, DQBFs boost reasoning power and enable further applications (cf. [32] for an overview). The price of succinct encodings is an increase of the complexity of the satisfiability problem from PSPACE (for QBF) to NEXP (for DQBF) [1].

It seems natural that there should be a relationship between dependency schemes and DQBF, and indeed the paper [7] suggests that dependency schemes for QBF should be viewed as truth-preserving mappings from QBF to DQBF.

Now, is there even a need for *dependency schemes for DQBF*? The answer is yes: also for DQBFs it is possible that the dependency sets contain spurious dependencies, which can be safely eliminated [36]. Indeed, Wimmer et al. [36] showed that several dependency schemes for QBF, including \mathcal{D}^{rrs} , can be lifted to DQBF. They also demonstrate that using dependency schemes for DQBF preprocessing can have a significant positive impact on solving time.

However, in contrast to QBF, there are currently no results on how DQBF dependency schemes can be incorporated into DQBF proof systems, and how this affects their proof-theoretic strength.

This paper contributes to the theory of DQBF dependency schemes on three main fronts.

A. A proof complexity framework for DQBF dependency schemes. We extend the interpretation of QBF dependency schemes proposed in [7] to DQBF. The result is a framework in which a sound DQBF dependency scheme \mathcal{D} can be straightforwardly incorporated into an arbitrary DQBF proof system P , yielding the parametrised system $P(\mathcal{D})$. More precisely, in our framework a proof of Φ in $P(\mathcal{D})$ is simply a P proof of $\mathcal{D}(\Phi)$, where \mathcal{D} is a mapping between DQBFs.

A major benefit of this approach is that the rules of the proof system remain independent of the dependency scheme, which essentially plays the role of a preprocessor. Moreover, soundness of a dependency scheme is characterised by the natural property of full exhibition [4, 33], independently of proofs. This is a welcome feature, since even *defining* sound parameterisations on the QBF fragment has been fairly non-trivial, e.g. for the long-distance Q-Resolution calculus [4, 27].

We also extend the notion of *genuine* proof size lower bounds [12, 14] to DQBF proof systems. Since DQBF encompasses QBF, proof systems are susceptible to lower bounds from QBF proof complexity. We define a precise condition by which hardness from the QBF fragment is factored out. As such, our framework fosters the first dedicated DQBF proof complexity results.

B. The tautology-free dependency scheme. We define and analyse a new DQBF dependency scheme called the *tautology-free dependency scheme* (\mathcal{D}^{tf}). Our scheme builds on the reflexive resolution path dependency scheme (\mathcal{D}^{rrs}) [34], originally defined for QBFs, which prior to this paper was the strongest known DQBF scheme. \mathcal{D}^{tf} improves on \mathcal{D}^{rrs} by disallowing certain kinds of tautologies in resolution paths, thereby identifying further spurious dependencies.

We show that \mathcal{D}^{tf} is fully exhibited, and therefore sound, by reducing its full exhibition to that of \mathcal{D}^{rrs} . For this, we point out that the full exhibition of \mathcal{D}^{rrs} on DQBF is an immediate consequence of results of Wimmer et al. [36].

C. Exponential separations of (D)QBF proof systems. To demonstrate the strength of our new scheme \mathcal{D}^{tf} , we show that it can exponentially shorten proofs in DQBF proof systems. As a case study, we consider the expansion calculus $\forall\text{Exp}+\text{Res}$. The choice of $\forall\text{Exp}+\text{Res}$ is

motivated by two considerations: (1) it is a natural calculus, whose QBF fragment models expansion solving [19], and (2) other standard QBF resolution systems such as Q-Resolution and long-distance Q-Resolution do not lift to DQBF [11].

For $\forall\text{Exp}+\text{Res}$ parameterised by dependency schemes we show that

$$\forall\text{Exp}+\text{Res} < \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}}) < \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}}) \quad (1)$$

forms a hierarchy of DQBF proof systems of strictly increasing strength.

Since there exist no prior DQBF proof complexity results whatsoever, this entails proving exponential proof-size lower bounds in the first two systems. We obtain these by introducing two new DQBF versions of the equality formulas (originally QBFs [8, 13]). Together with the corresponding upper bounds, this yields the separations in (1). We highlight that these are *genuine* separations in the precise sense of our DQBF framework, whereby hardness due to the QBF fragment is factored out.

Finally, we show that our new dependency scheme \mathcal{D}^{tf} *is also relevant for QBFs*: we prove that Q-Resolution parameterised by \mathcal{D}^{tf} is exponentially stronger than Q-Resolution with \mathcal{D}^{rrs} . Thus \mathcal{D}^{tf} currently constitutes the strongest known dependency scheme for Q-Resolution.

Organisation. Section 2 defines DQBF preliminaries. In Section 3 we explain dependency schemes. Section 4 details how to parameterise DQBF proof systems by dependency schemes. In Section 5 we define our new scheme \mathcal{D}^{tf} and show its soundness. In Section 6 we prove the proof complexity upper and lower bounds needed for the strict hierarchy in (1). Section 7 applies \mathcal{D}^{tf} to QBF and shows that it is stronger than \mathcal{D}^{rrs} when used with Q-Resolution.

2 Preliminaries

DQBF syntax. We assume familiarity with the syntax of propositional logic and the notion of *Boolean formula* (simply *formula*). A *variable* is an element z of the countable set \mathbb{V} . A *literal* is a variable z or its *negation* \bar{z} . The negation of a literal a is denoted \bar{a} , where $\bar{\bar{z}} := z$ for any variable z . A *clause* is a disjunction of literals. A *conjunctive normal form formula* (CNF) is a conjunction of clauses. The set of variables appearing in a formula ψ is denoted $\text{vars}(\psi)$. For ease, we often write clauses as sets of literals, and CNFs as sets of clauses. For any clause C and any set of variables Z , we define $C \upharpoonright_Z := \{a \in C : \text{var}(a) \in Z\}$.

A *dependency quantified Boolean formula* (DQBF) is a sentence of the form $\Psi := \Pi \cdot \psi$, where $\Pi := \forall u_1 \cdots \forall u_m \exists x_1(S_{x_1}) \cdots \exists x_n(S_{x_n})$ is the *quantifier prefix* and ψ is a CNF called the *matrix*. In the quantifier prefix, each existential variable x_i is associated with a *dependency set* S_{x_i} , which is a subset of the universal variables $\{u_1, \dots, u_m\}$. With $\text{vars}_{\forall}(\Psi)$ and $\text{vars}_{\exists}(\Psi)$ we denote the universal and existential variable sets of Ψ , and with $\text{vars}(\Psi)$ their union. We deal only with *closed* DQBFs, in which $\text{vars}(\psi) \subseteq \text{vars}(\Psi)$. We define a relation $\text{deps}(\Psi)$ on $\text{vars}_{\forall}(\Psi) \times \text{vars}_{\exists}(\Psi)$, where $(u, x) \in \text{deps}(\Psi)$ if, and only if, $u \in S_x$.

The set of all DQBFs is denoted DQBF. A QBF is a DQBF whose dependency sets are linearly ordered with respect to set inclusion, i.e. $S_{x_1} \subseteq \cdots \subseteq S_{x_n}$. The prefix of a QBF can be written as a linear order in the conventional way. The set of all QBFs is denoted QBF.

DQBF semantics. An *assignment* α to a set Z of Boolean variables is a function from Z into the set of *Boolean constants* $\{0, 1\}$. The *domain restriction* of α to a subset $Z' \subseteq Z$ is written $\alpha \upharpoonright_{Z'}$. The set of all assignments to Z is denoted $\langle Z \rangle$. The *restriction* of a formula ψ by α , denoted $\psi[\alpha]$, is the result of substituting each variable z in Z by $\alpha(z)$, followed by applying the standard simplifications for Boolean constants, i.e. $\bar{0} \mapsto 1$, $\bar{1} \mapsto 0$, $\phi \vee 0 \mapsto \phi$, $\phi \vee 1 \mapsto 1$, $\phi \wedge 1 \mapsto \phi$, and $\phi \wedge 0 \mapsto 0$. We say that α *satisfies* ψ when $\psi[\alpha] = 1$, and *falsifies* ψ when $\psi[\alpha] = 0$.

A *model* for a DQBF $\Psi := \Pi \cdot \psi$ is a set of functions $f := \{f_x : x \in \text{vars}_\exists(\Psi)\}$, $f_x : \langle S_x \rangle \rightarrow \langle \{x\} \rangle$, for which, for each $\alpha \in \langle \text{vars}_\forall(\Psi) \rangle$, the combined assignment $\alpha \cup \{f_x(\alpha|_{S_x}) : x \in \text{vars}_\exists(\Psi)\}$ satisfies ψ . A DQBF is called *true* when it has a *model*, otherwise it is called *false*. When two DQBFs share the same truth value, we write $\Psi \stackrel{\text{tr}}{\equiv} \Psi'$.

DQBF expansion. *Universal expansion* is a syntactic transformation that removes a universal variable from a DQBF. Let Ψ be a DQBF, let u be a universal, and let y_1, \dots, y_k be the existentials for which $u \in S_{y_i}$. The *expansion* of Ψ by u is obtained by creating two ‘copies’ of Ψ . In the first copy, u is assigned 0 and each y_i is renamed $y_i^{\bar{u}}$. In the second copy, u is assigned 1 and each y_i is renamed y_i^u . The two copies are then combined, and u is removed completely from the prefix. Formally, $\text{exp}(\Psi, u) := \Pi' \cdot \psi'$, where Π' is obtained from Π by removing $\forall u$ and replacing each $\exists y_i(S_{y_i})$ with $\exists y_i^{\bar{u}}(S_{y_i} \setminus \{u\}) \exists y_i^u(S_{y_i} \setminus \{u\})$, and

$$\psi' := \psi[u \mapsto 0, y_1 \mapsto y_1^{\bar{u}}, \dots, y_k \mapsto y_k^{\bar{u}}] \wedge \psi[u \mapsto 1, y_1 \mapsto y_1^u, \dots, y_k \mapsto y_k^u].$$

Universal expansion is known to preserve the truth value, i.e. $\Psi \stackrel{\text{tr}}{\equiv} \text{exp}(\Psi, u)$. Expansion by a set of universal variables U is defined as the successive expansion by each $u \in U$ (the order is irrelevant), and is denoted $\text{exp}(\Psi, U)$. Expansion by the whole set $\text{vars}_\forall(\Psi)$ is denoted $\text{exp}(\Psi)$, and referred to as the *total expansion* of Ψ . The superscripts in the renamed existential variables are known as *annotations*. Annotations grow during successive expansions. In the total expansion, each variable is annotated with a total assignment to its dependency set.

3 DQBF Dependency Schemes and Full Exhibition

In this section, we lift the ‘DQBF-centric’ interpretation of QBF dependency schemes [7] to the DQBF domain, and recall the definition of full exhibition.

How should we interpret variable dependence? Dependency schemes [31] were originally introduced to identify so-called *spurious dependencies*: sometimes the order of quantification implies that z depends on z' , but forcing z to be independent preserves the truth value. Technically, a dependency scheme \mathcal{D} was defined to map a QBF Φ to a set of pairs $(z', z) \in \text{vars}(\Phi) \times \text{vars}(\Phi)$, describing an overapproximation of the dependency structure: $(z', z) \in \mathcal{D}(\Phi)$ means that the dependence of z on z' should not be ignored, whereas $(z', z) \notin \mathcal{D}(\Phi)$ means that it can be. The definition was tailored to QBF solving, in which variable dependencies for both true and false formulas come into play.

The DQBF-centric interpretation [7] followed somewhat later. There, the goal was a dependency scheme framework tailored to refutational QBF proof systems. Refutational systems work only with false formulas, and this allows a broad refinement: the dependence of universals on existentials can be ignored. As such, it makes sense to consider merely the effect of deleting some universal variables from the existential dependency sets. Thus, a dependency scheme becomes a mapping from QBF into DQBF.

Likewise, in this work we seek a framework tailored towards refutational proof systems. Hence we advocate the same approach for the *whole domain* DQBF. A DQBF dependency scheme will be viewed as a mapping to and from DQBF, in which the dependency sets may shrink. The notion of shrinking dependency sets is captured by the relation following.

Definition 1. *We define the relation \leq on $\text{DQBF} \times \text{DQBF}$ as follows: $\Pi' \cdot \phi \leq \Pi \cdot \psi$ if, and only if, $\phi = \psi$, $\text{vars}_\exists(\Psi') = \text{vars}_\exists(\Psi)$, and the dependency set of each existential in Π' is a subset of that of Π .*

In this paper, we only consider poly-time computable dependency schemes.

Definition 2 (dependency scheme). A dependency scheme is a polynomial-time computable function $\mathcal{D} : \text{DQBF} \rightarrow \text{DQBF}$ for which $\mathcal{D}(\Psi) \leq \Psi$ for all Ψ .

Under this definition, a *spurious dependency according to \mathcal{D}* is a pair (u, x) such that u is in the dependency set for x in Ψ , but not in $\mathcal{D}(\Psi)$. A natural property of dependency schemes, identified in [36], is *monotonicity*.²

Definition 3 (monotone (adapted from [36])). We call a dependency scheme \mathcal{D} monotone when $\Psi' \leq \Psi$ implies $\mathcal{D}(\Psi') \leq \mathcal{D}(\Psi)$, for all Ψ and Ψ' .

A fundamental concept in the DQBF-centric framework, which has strong connections to soundness in related proof systems [6], is *full exhibition*. First used by Slivovsky [33], the name was coined later in [4], describing the fact that there should be a model which ‘fully exhibits’ all spurious dependencies. ‘Full exhibition’ is synonymous with ‘truth-value preserving’.

Definition 4 (full exhibition [4, 33]). A dependency scheme \mathcal{D} is called fully exhibited when $\Psi \stackrel{tr}{=} \mathcal{D}(\Psi)$, for all Ψ .

4 Parametrising DQBF Calculi by Dependency Schemes

In this section we show how to incorporate dependency schemes into DQBF proof systems. In the spirit of so-called ‘genuine’ lower bounds [12], we also introduce a notion of genuine DQBF hardness.

Refutational DQBF proof systems. We first define what we mean by a DQBF proof system. With FDQBF we denote the set of false DQBFs. We consider only *refutational* proof systems, which try to show that a given formula is false. Hence, ‘proof’ and ‘refutation’ can be considered synonymous.

Following [15], a *DQBF proof system* over an alphabet Σ is a polynomial-time computable onto function $P : \Sigma^* \rightarrow \text{FDQBF}$. In practice, we do not always want to define a proof system explicitly as a function on a domain of strings. Instead, we define what constitutes a refutation in the proof system P , and then show: (1) *Soundness*: if Ψ has a refutation, it is false (the codomain of P is FDQBF); (2) *Correctness*: every false DQBF has a refutation (P is onto); (3) *Checkability*: refutations can be checked efficiently (P is polynomial-time computable).

To date, at least three DQBF proof systems have been proposed: the fundamental expansion-based system $\forall\text{Exp}+\text{Res}$ [7], the more sophisticated expansion system IR-calc [7], and the system Fork Resolution [29] which is based on the concept of information forks.

Incorporating dependency schemes. A dependency scheme, interpreted as a DQBF mapping as in Definition 2, can be combined with an arbitrary proof system in a straightforward manner.

Definition 5 ($P(\mathcal{D})$). Let P be a DQBF proof system and let \mathcal{D} be a dependency scheme. A $P(\mathcal{D})$ refutation of a DQBF Ψ is a P refutation of $\mathcal{D}(\Psi)$.

The proof system $P(\mathcal{D})$ essentially utilises the dependency scheme as a preprocessing step, mapping its input Ψ to the image $\mathcal{D}(\Psi)$ before proceeding with the refutation. In this way, the application of the dependency scheme \mathcal{D} is separated from the rules of the proof system P , and consequently the definition of P need not be explicitly modified to incorporate \mathcal{D} (cf. [4, 34]).

Of course, we must ensure that our preprocessing step is correct; we do not want to map a true formula to a false one, which would result in an unsound proof system. Now it becomes clear why full exhibition is central for soundness.

² A different notion of monotonicity for dependency schemes is defined in [27].

Proposition 6. *Given a DQBF proof system P and a dependency scheme \mathcal{D} , $P(\mathcal{D})$ is sound if, and only if, \mathcal{D} is fully exhibited.*

Proof. Suppose that \mathcal{D} is fully exhibited. Let π be a $P(\mathcal{D})$ refutation of a DQBF Ψ . Then π is a P refutation of $\mathcal{D}(\Psi)$, which is false by the soundness of P . Hence Ψ is false by the full exhibition of \mathcal{D} , so $P(\mathcal{D})$ is sound.

Suppose now that \mathcal{D} is not fully exhibited. Since \mathcal{D} preserves falsity by definition, there must exist a true DQBF Ψ for which $\mathcal{D}(\Psi)$ is false. Then there exists a P refutation of $\mathcal{D}(\Psi)$ by the completeness of P , so $P(\mathcal{D})$ is not sound. \square

Note that completeness and checkability of P are preserved trivially by any dependency scheme, so we can even say that $P(\mathcal{D})$ is a DQBF proof system if, and only if, \mathcal{D} is fully exhibited. Thus full exhibition characterises exactly the dependency schemes whose incorporation preserves the proof system.

Simulations, separations and genuine lower bounds. Of course, the rationale for utilising a dependency scheme as a preprocessor lies in the potential for shorter refutations. We first recall the notion of p -simulation from [15]. Let P and Q be DQBF proof systems. We say that P p -simulates Q (written $Q \leq_p P$) when there exists a polynomial-time computable function from Q refutations to P refutations that preserves the refuted formula.

Since a p -simulation is computed in polynomial time, the translation from Q into P incurs at most a polynomial size blow-up. As such, the conventional approach to proving the non-existence of a p -simulation is to exhibit a family of formulas $\{\Psi_n\}_{n \in \mathbb{N}}$ that has polynomial-size refutations in Q , while requiring super-polynomial size in P .

Now, it is of course possible that the hard formulas $\{\Psi_n\}_{n \in \mathbb{N}}$ are QBFs. While this suffices to show that $Q \not\leq_p P$, it is not what we want from a study of DQBF proof complexity; it is rather a statement about the QBF fragments of the systems P and Q . In reality the situation is even more complex. The lower bound may stem from QBF proof complexity even when $\{\Psi_n\}_{n \in \mathbb{N}}$ are not QBFs. More precisely, there may exist an ‘embedded’ QBF family $\{\Phi_n\}_{n \in \mathbb{N}}$ which is already hard for P , where ‘embedded’ means $\Phi_n \leq \Psi_n$. Under the reasonable assumption that decreasing dependency sets cannot increase proof size,³ any DQBF family in which $\{\Phi_n\}_{n \in \mathbb{N}}$ is embedded will be hard for P .

For that reason, we introduce a notion of *genuine* DQBF hardness that dismisses all embedded QBF lower bounds.

Definition 7. *Let P and Q be DQBF proof systems. We write $Q \not\leq_p^* P$ when there exists a DQBF family $\{\Psi_n\}_{n \in \mathbb{N}}$ such that:*

- (a) $\{\Psi_n\}_{n \in \mathbb{N}}$ has polynomial-size Q refutations;
- (b) $\{\Psi_n\}_{n \in \mathbb{N}}$ requires superpolynomial-size P refutations;
- (c) every QBF family $\{\Phi_n\}_{n \in \mathbb{N}}$ with $\Phi_n \leq \Psi_n$ has polynomial-size P refutations.

We write $P <_p^* Q$ when both $P \leq_p Q$ and $Q \not\leq_p^* P$ hold.

Hence, $P <_p^* Q$ means that Q simulates P , but P does not simulate Q , and the hardness result for P is a genuine DQBF lower bound. Prior to this paper, there were no such hardness results in the DQBF literature.

³ This holds for all known DQBF proof systems.

5 The Tautology-free Dependency Scheme

In this section we define the *tautology-free dependency scheme* \mathcal{D}^{tf} and show that it is fully exhibited.

For any DQBF Ψ , we denote by $I_{\exists}(\Psi)$ the set of independent existential variables, i.e. $I_{\exists}(\Psi) := \{x \in \text{vars}_{\exists}(\Psi) : S_x = \emptyset\}$ is the set of existentials whose dependency sets are empty.

Definition 8 (\mathcal{D}^{rrs} [34] and \mathcal{D}^{tf}). *The reflexive resolution path dependency scheme (\mathcal{D}^{rrs}) is defined as the mapping $\Psi \mapsto \Psi'$, where*

$$\begin{aligned}\Psi &:= \forall u_1 \cdots \forall u_m \exists x_1(S_{x_1}) \cdots \exists x_n(S_{x_n}) \cdot \psi, \\ \Psi' &:= \forall u_1 \cdots \forall u_m \exists x_1(S'_{x_1}) \cdots \exists x_n(S'_{x_n}) \cdot \psi,\end{aligned}$$

and S'_i is the set of universal variables $u \in S_i$ for which there exists a sequence C_1, \dots, C_k of clauses in ψ and a sequence p_1, \dots, p_{k-1} of existential literals satisfying the following conditions:

- (a) $u \in C_1$ and $\bar{u} \in C_k$;
- (b) for some $j \in [k-1]$, $x_i = \text{var}(p_j)$;
- (c) for each $j \in [k-1]$, $p_j \in C_j$, $\bar{p}_j \in C_{j+1}$, and $u \in S_{\text{var}(p_i)}$;
- (d) for each $j \in [k-2]$, $\text{var}(p_j) \neq \text{var}(p_{j+1})$.

The tautology-free dependency scheme (\mathcal{D}^{tf}) adds to \mathcal{D}^{rrs} the condition

- (e) for each $j \in [k-1]$, $(C_j \cup C_{j+1}) \upharpoonright_{I_{\exists}(\Psi)}$ is non-tautological.

Let us give an example, illustrating that \mathcal{D}^{tf} is stronger than \mathcal{D}^{rrs} .

Example 9. Consider the DQBF $\Psi = \exists x \forall u \exists z \cdot C_1 \wedge C_2$, where $C_1 = x \vee u \vee z$ and $C_2 = \bar{x} \vee \bar{u} \vee \bar{z}$. The sequence of clauses C_1, C_2 and the sequence consisting of the single literal $p_1 = z$ show that $(u, z) \in \text{deps}(\mathcal{D}^{\text{rrs}}(\Psi))$. However, the same sequence of clauses violates condition (e) of Definition 8 because $(C_1 \cup C_2) \upharpoonright_{I_{\exists}(\Psi)}$ is a tautology on $x \in I_{\exists}(\Psi)$. Since the reverse sequence has the same tautology and there are no other sequences to consider, we conclude that $(u, z) \notin \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$. \square

Proposition 10. \mathcal{D}^{tf} is a monotone dependency scheme.

Proof. It is easy to see that $\mathcal{D}^{\text{tf}}(\Psi) \leq \Psi$ for each Ψ . It remains to verify polynomial-time computability and monotonicity.

Polynomial-time computability. As there are polynomially many pairs, it suffices to show that whether (u, x) is in $\text{deps}(\Psi)$ can be decided in polynomial time for each pair (u, x) . Consider the directed graph $G_{\Psi}^u = (V_{\Psi}, E_{\Psi}^u)$ with the vertex set $V_{\Psi} = \{(C, a) : C \in \Psi, a \in C\}$ and with an edge from (C, a) to (D, e) if $\bar{e} \in C$, $u \in S_{\text{var}(e)}$, $\text{var}(a) \neq \text{var}(e)$, and $(C \cup D) \upharpoonright_{I_{\exists}(\Psi)}$ is non-tautological.

We claim that $(u, x) \in \text{deps}(\Psi)$ if, and only if, there is a literal a , $\text{var}(a) = x$, and clauses C, C', C'' such that $\bar{u} \in C''$, (C', a) is reachable from (C, u) and (C'', e) is reachable from (C', a) for some e . Indeed, it is easy to verify that the concatenation of a pair of such paths directly translates to the required sequences from Definition 8, and vice versa. Clearly, G_{Ψ}^u can be constructed in polynomial time, hence we can test all candidates (C, u) , compute all middle points (C', a) reachable from them, and check whether some (C'', e) is reachable from any of them, all in polynomial time.

Monotonicity. Let Ψ, Ψ' be DQBFs with $\Psi' \leq \Psi$, let $(u, x) \in \text{deps}(\mathcal{D}^{\text{tf}}(\Psi'))$. We show that $(u, x) \in \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$. It follows that $\mathcal{D}^{\text{tf}}(\Psi') \leq \mathcal{D}^{\text{tf}}(\Psi)$.

There exists a sequence of clauses C_1, \dots, C_k and a sequence of literals p_1, \dots, p_{k-1} satisfying conditions (a) to (e) in Definition 8 with respect to $(u, x) \in \text{deps}(\Psi')$. We show that the same sequences satisfy conditions (a) to (e) with respect to $(u, x) \in \text{deps}(\Psi)$, which implies $(u, x) \in \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$.

Conditions (a), (b) and (d) are satisfied trivially. Since $\Psi' \leq \Psi$, each dependency set $S_{\text{var}(p_i)}$ in Ψ is a superset of the corresponding dependency set $S'_{\text{var}(p_i)}$ in Ψ' , so condition (c) is satisfied. Condition (e) is satisfied since the set of independent variables $I_{\exists}(\Psi)$ is a subset of $I_{\exists}(\Psi')$. \square

Wimmer et al. [36] essentially showed that \mathcal{D}^{rrs} is fully exhibited, even though they did not use that term. Theorems 3 and 4 in [36] together imply that all spurious dependencies can be removed one by one in any order without changing the truth value (as is remarked at the start of Section 3.1 in that paper).

Theorem 11 (Wimmer et al. [36]). \mathcal{D}^{rrs} is fully exhibited.

We show full exhibition of \mathcal{D}^{tf} by reduction to full exhibition of \mathcal{D}^{rrs} .

Theorem 12. \mathcal{D}^{tf} is fully exhibited.

Proof. Since $\mathcal{D}^{\text{tf}}(\Psi) \leq \Psi$, we only need to show that if Ψ is true, then $\mathcal{D}^{\text{tf}}(\Psi)$ is true. Assume Ψ is true; then there is an assignment $\sigma \in \langle I_{\exists}(\Psi) \rangle$ such that $\Psi[\sigma]$ is true. We claim that $(u, x) \in \text{deps}(\mathcal{D}^{\text{rrs}}(\Psi[\sigma]))$ implies $(u, x) \in \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$. Consider sequences C_1, \dots, C_k and p_1, \dots, p_{k-1} witnessing $(u, x) \in \text{deps}(\mathcal{D}^{\text{rrs}}(\Psi[\sigma]))$. For each C_i there is $C'_i \in \Psi$, such that $C_i = C'_i[\sigma]$, i.e. $C'_i \subseteq C_i \cup \bar{\sigma}$, where $\bar{\sigma}$ is the largest clause falsified by σ . It is readily verified that the sequences C'_1, \dots, C'_k and p_1, \dots, p_{k-1} witness $(u, x) \in \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$. In particular, no tautologies can appear among $(C'_i \cup C'_{i+1}) \upharpoonright_{I_{\exists}(\Psi)}$, because all C'_i agree with $\bar{\sigma}$ on the variables of $I_{\exists}(\Psi)$. Hence, we get $\mathcal{D}^{\text{rrs}}(\Psi[\sigma]) \leq \mathcal{D}^{\text{tf}}(\Psi)[\sigma]$. By full exhibition of \mathcal{D}^{rrs} , we have that $\mathcal{D}^{\text{rrs}}(\Psi[\sigma])$ is true, which means $\mathcal{D}^{\text{tf}}(\Psi)[\sigma]$ is true, and hence $\mathcal{D}^{\text{tf}}(\Psi)$ is true. \square

Example 13. Consider Ψ from Example 9. It is easy to see that Ψ is true. As shown in Example 9, $\mathcal{D}^{\text{tf}}(\Psi) = \exists x \exists z \forall u. (x \vee z \vee u) \wedge (\bar{x} \vee \bar{z} \vee \bar{u})$. We can see that the assignment $x \mapsto 1, z \mapsto 0$ is a model of $\mathcal{D}^{\text{tf}}(\Psi)$, which is therefore true, in line with full exhibition of \mathcal{D}^{tf} . \square

6 Proof Complexity of $\forall\text{Exp}+\text{Res}(\mathcal{D})$

Among the first DQBF proof systems to be introduced, the expansion based system $\forall\text{Exp}+\text{Res}$ [7, 19] is arguably the most natural. In this section we investigate its proof complexity under parametrisation by dependency schemes; that is, we investigate the proof complexity of $\text{P}(\mathcal{D})$ where P is $\forall\text{Exp}+\text{Res}$. Our main result is the following theorem.

Theorem 14. $\forall\text{Exp}+\text{Res} <^*_p \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}}) <^*_p \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$.

The simulations present in Theorem 14 follow from two observations, namely (1) $\mathcal{D}^{\text{tf}}(\Psi) \leq \mathcal{D}^{\text{rrs}}(\Psi)$ (by definition), and (2) $\Psi' \leq \Psi$ guarantees that $\forall\text{Exp}+\text{Res}$ refutations of Ψ' are no larger than those of Ψ . Indeed, given a refutation of Ψ , restricting the annotations to the dependency sets of Ψ' produces a refutation of Ψ' of the same size. We refer to this property as the *monotonicity* of $\forall\text{Exp}+\text{Res}$.

The challenge is to show the genuine separations (Theorems 20 and 26). We note that the QBF analogue of the first separation is known [6]. The question (and indeed the notion) of a genuine separation was not previously considered.

The DQBF proof system $\forall\text{Exp}+\text{Res}$. We recall the propositional Resolution proof system [30]. A *Resolution refutation* of a CNF ψ is a sequence C_1, \dots, C_k of clauses where C_k is empty and each C_i is derived by one of the following rules:

A Axiom: C_i is a clause in ψ ;

R Resolution: $C_i = A \vee B$, where $C_r = A \vee x$ and $C_s = B \vee \bar{x}$, for some $r, s < i$.

The DQBF proof system $\forall\text{Exp}+\text{Res}$, with which we shall concern ourselves for the remainder of the section, is built upon Resolution. Perhaps the most obvious way to decide DQBF is to reduce it to propositional logic by expanding out all the universal variables, based on the fact that Ψ is true if, and only if, the matrix of $\text{exp}(\Psi)$ is satisfiable. This is exactly how $\forall\text{Exp}+\text{Res}$ works. The input DQBF is first expanded, and then refuted in Resolution.

Definition 15 ($\forall\text{Exp}+\text{Res}$ [7, 19]). *A $\forall\text{Exp}+\text{Res}$ refutation of a DQBF Ψ is a Resolution refutation of the matrix of $\text{exp}(\Psi)$.*

It is known that $\forall\text{Exp}+\text{Res}$ is sound, complete and checkable on DQBFs [7]. Note that a $\forall\text{Exp}+\text{Res}$ refutation of Ψ may be small even if its expansion $\text{exp}(\Psi)$ is large, since the underlying Resolution refutation of $\text{exp}(\Psi)$ need not necessarily introduce every clause as an axiom.

Given that fully exhibited dependency schemes like \mathcal{D}^{tf} and \mathcal{D}^{rrs} (Theorem 12) can be incorporated into an arbitrary DQBF proof system P (Proposition 6), we obtain the DQBF proof systems $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ and $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$.

Next we show the two genuine separations that together constitute a proof of Theorem 14.

Separation of $\forall\text{Exp}+\text{Res}$ and $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$. Our separating formulas are DQBFs based on the equality QBFs [8]. Our modification exploits a refined dependency structure and utilises the following notation: the *matrix-clause product* of a CNF ψ and a clause C is the CNF $\psi \otimes C := \{D \cup C : D \in \psi\}$.

Definition 16 (EQ_n^0 (adapted from [8])). $\text{EQ}_n^0 := \Pi_n^{\text{EQ}} \cdot \psi_n^{\text{EQ}}$, where

$$\begin{aligned} \Pi_n^{\text{EQ}} &:= \forall u_1 \cdots \forall u_n \exists x_1(\emptyset) \cdots \exists x_n(\emptyset) \exists z_1(u_1) \cdots \exists z_n(u_n), \\ \psi_n^{\text{EQ}} &:= (\bar{z}_1 \vee \cdots \vee \bar{z}_n) \wedge \bigwedge_{i=1}^n \left((\bar{x}_i \vee \bar{u}_i \vee z_i) \wedge (x_i \vee u_i \vee z_i) \right). \end{aligned}$$

Since the dependency sets of EQ_n^0 are strict subsets of those of the original equality formulas (in which each z_i depends on *each* u_j), the QBF lower bound for $\forall\text{Exp}+\text{Res}$ [5] does not suffice for EQ_n^0 . Nonetheless, a similar argument works, based on the fact that no small subset of clauses in the expansion is unsatisfiable.

Theorem 17. $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ requires exponential-size $\forall\text{Exp}+\text{Res}$ refutations.

Proof. The total expansion of EQ_n^0 is the CNF $\psi \wedge \bigwedge_{i=1}^n \left((\bar{x}_i \vee z_i^{\bar{u}_i}) \wedge (x_i \vee z_i^{u_i}) \right)$, where ψ is the conjunction of all clauses of the form $(\bar{z}_1^{a_1} \vee \cdots \vee \bar{z}_n^{a_n})$ with $\text{var}(a_i) = u_i$. We show that removing any of the 2^n clauses from ψ makes the total expansion satisfiable. It follows that any Resolution refutation of $\text{exp}(\text{EQ}_n^0)$ must have 2^n axiom clauses.

Suppose that some clause A is absent from ψ , and let us assume without loss of generality that $A := (\bar{z}_1^{u_1} \vee \cdots \vee \bar{z}_n^{u_n})$, i.e. the clause corresponding to $u_i \mapsto 1$ for each i (the general case is symmetrical). Now, assigning each $z_i^{u_i} \mapsto 1$ satisfies every clause in ψ except A . Assigning each $z_i^{\bar{u}_i} \mapsto 0$ satisfies each clause $(\bar{x}_i \vee z_i^{\bar{u}_i})$. Finally, assigning each $x_i \mapsto 1$ satisfies each clause $(x_i \vee z_i^{u_i})$. \square

The corresponding upper bound for EQ_n^0 in $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ does follow from that of the original equality QBFs (by the monotonicity of \mathcal{D}^{rrs} and $\forall\text{Exp}+\text{Res}$). We give a full proof nonetheless, since we will use the details later. The main point is that \mathcal{D}^{rrs} identifies all pairs as spurious dependencies.

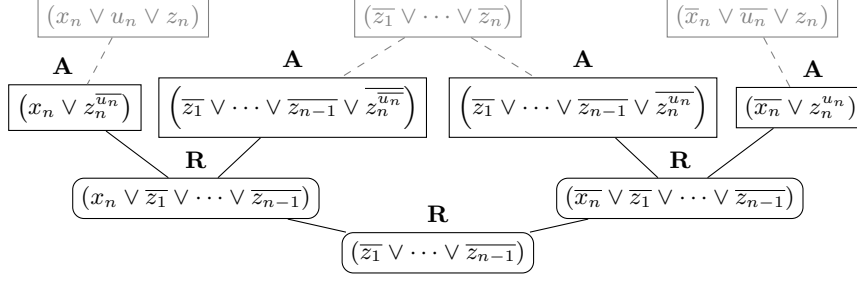


Fig. 1. The prelude to a linear-size $\forall\text{Exp}+\text{Res}$ refutation of Φ_n^0 . In order to reduce $\text{exp}(\Phi_n^0)$ to $\text{exp}(\mathcal{D}^{\text{rrs}}(\text{EQ}_{n-1}^0))$, we need only derive the clause $(\bar{z}_1 \vee \dots \vee \bar{z}_{n-1})$.

Proposition 18 ([6]). *For all n , the dependency sets of $\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0)$ are empty.*

Proof. Aiming for contradiction, suppose that there exists a sequence of clauses C_1, \dots, C_k and a sequence of literals p_1, \dots, p_{k-1} satisfying conditions (a) to (d) of Definition 8 with respect to $(u_i, z_i) \in \text{deps}(\text{EQ}_n^0)$. Since z_i is the unique variable whose dependency set contains u_i , we must have $k = 2$, by conditions (c) and (d). By condition (a), we have $u_i \in C_1$, so $C_1 = (x_i \vee u_i \vee z_i)$, and by condition (c) we have $p_1 = z_i$. Also by condition (c) we have $\bar{z}_i \in C_2$, so $C_2 = (\bar{z}_1 \vee \dots \vee \bar{z}_n)$. We therefore reach a contradiction, since $\bar{u}_i \notin C_2$ violates condition (a). \square

Theorem 19 ([6]). $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ has linear-size $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ refutations.

Proof. By Proposition 18, the total expansion of $\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0)$ is obtained simply by removing the universal literals; that is, the matrix of $\text{exp}(\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0))$ is

$$(\bar{z}_1 \vee \dots \vee \bar{z}_n) \wedge \bigwedge_{i=1}^n \left((\bar{x}_i \vee z_i) \wedge (x_i \vee z_i) \right). \quad (2)$$

It is easy to see that this CNF has linear-size Resolution refutations. First, resolve each pair $(x_i \vee z_i), (\bar{x}_i \vee z_i)$ over x_i , and resolve the resulting unit clauses (z_i) with the remaining clause to obtain the empty clause. \square

Theorems 17 and 19 together imply that $\forall\text{Exp}+\text{Res}$ does not p -simulate $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$. It remains to show that the lower bound is genuine.

Theorem 20. $\forall\text{Exp}+\text{Res} \not\leq_p^* \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$.

Proof. It is easy to see that the largest QBF Φ_n^0 that is smaller than EQ_n^0 has exactly one non-empty dependency set. Let us assume without loss of generality that this is $S_{z_n} = \{u_n\}$. We will show that Φ_n^0 has a linear-size $\forall\text{Exp}+\text{Res}$ refutation. Hence, by the monotonicity of $\forall\text{Exp}+\text{Res}$, any family of QBFs smaller than $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ has linear-size $\forall\text{Exp}+\text{Res}$ refutations. Thus, by Theorems 17 and 19, $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ satisfies all the conditions of Definition 7.

It remains to show that Φ_n^0 has a linear-size $\forall\text{Exp}+\text{Res}$ refutation, or equivalently, that $\text{exp}(\Phi_n^0)$ has a linear-size Resolution refutation. It is readily verified that $\text{exp}(\Phi_n^0)$ contains every clause in $\text{exp}(\mathcal{D}^{\text{rrs}}(\text{EQ}_{n-1}^0))$ except $(\bar{z}_1 \vee \dots \vee \bar{z}_{n-1})$. Figure 1 illustrates that this clause can be derived from $\text{exp}(\Phi_n^0)$ in a constant number of resolution steps. Since $\text{exp}(\mathcal{D}^{\text{rrs}}(\text{EQ}_{n-1}^0))$ has a linear-size Resolution refutation by Theorem 19, so does $\text{exp}(\Phi_n^0)$. \square

Separation of $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ and $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$. For our second separation, we introduce another DQBF family. This time, we refine the prefix of an existing modification of the equality formulas.

Definition 21 (EQ_n^1 (adapted from [13])). For each natural number n ,

$$\text{EQ}_n^1 := \Pi_n^{\text{EQ}} \exists r(\emptyset) \exists s(\{u_1, \dots, u_n\}) \cdot \\ (\psi_n^{\text{EQ}} \otimes (r \vee s)) \wedge (\psi_n^{\text{EQ}} \otimes (\bar{r} \vee \bar{s})) \wedge (r \vee \bar{s}) \wedge (\bar{r} \vee s).$$

The main idea is that the addition of the fresh variables r and s is enough to obfuscate all the spurious dependencies for \mathcal{D}^{rfs} . As such, preprocessing with \mathcal{D}^{rfs} has no effect, and hardness can be proved via the $\forall\text{Exp}+\text{Res}$ lower bound for EQ_n^0 (Theorem 17).

Proposition 22. For each n , $\mathcal{D}^{\text{rfs}}(\text{EQ}_n^1) = \text{EQ}_n^1$.

Proof. To prove the proposition, we must find sequences satisfying conditions (a) to (d) of Definition 8 with respect to both $(u_i, z_i), (u_i, s) \in \text{deps}(\text{EQ}_n^1)$ for each i . In fact, for both pairs (u_i, z_i) and (u_i, s) , the sequence of clauses

$$(r \vee x_i \vee u_i \vee z_i \vee s), (r \vee \bar{z}_1 \vee \dots \vee \bar{z}_n \vee s), (\bar{r} \vee \bar{x}_i \vee \bar{u}_i \vee z_i \vee \bar{s})$$

and the sequence of literals z_i, s suffice. \square

Theorem 23. $\{\text{EQ}_n^1\}_{n \in \mathbb{N}}$ requires exponential-size $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rfs}})$ refutations.

Proof. Consider the assignment α defined by $r \mapsto 0, s \mapsto 0$. It is easy to see that $\text{EQ}_n^1[\alpha] = \text{EQ}_n^0$. Now consider the ‘expanded’ assignment α^U defined by $r \mapsto 0, s^\sigma \mapsto 0$ for each $\sigma \in \langle \{u_1, \dots, u_n\} \rangle$. It is less easy to see, but readily verified, that $\text{exp}(\text{EQ}_n^1)[\alpha^U] = \text{exp}(\text{EQ}_n^1[\alpha]) = \text{exp}(\text{EQ}_n^0)$. Let π be a $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rfs}})$ refutation of EQ_n^1 ; that is, a Resolution refutation of $\text{exp}(\mathcal{D}^{\text{rfs}}(\text{EQ}_n^1))$. By Proposition 22, π is a Resolution refutation of $\text{exp}(\text{EQ}_n^1)$. Since Resolution is closed under restrictions, $\pi[\alpha^U]$ is a Resolution refutation of $\text{exp}(\text{EQ}_n^1)[\alpha^U] = \text{exp}(\text{EQ}_n^0)$ with $|\pi[\alpha^U]| \leq |\pi|$. By Theorem 17, $2^n \leq |\pi[\alpha^U]| \leq |\pi|$. \square

The situation is quite different for the tautology-free dependency scheme \mathcal{D}^{tf} . Here, the simple detection of consecutive-clause tautologies in the variable r is enough to identify all spurious dependencies, resulting in linear-size refutations.

Proposition 24. For each n , the dependency sets of $\mathcal{D}^{\text{tf}}(\text{EQ}_n^1)$ are all empty.

Proof. Aiming for contradiction, suppose that there exists a sequence of clauses C_1, \dots, C_k and a sequence of literals p_1, \dots, p_{k-1} satisfying conditions (a) to (e) of Definition 8 with respect to $(u_j, y) \in \text{deps}(\text{EQ}_n^1)$, for some $y \in \{z_j, s\}$.

By condition (c), none of the $\text{var}(p_i)$ is r . Hence, if some C_i is either $(r \vee \bar{s})$ or $\bar{s} \vee r$, we must have $i = 1$ or $i = k$, violating condition (a). Therefore those clauses do not appear in the sequence. It follows that none of the $\text{var}(p_i)$ is s , for otherwise we would have consecutive clauses C_i and C_{i+1} whose resolvent over s contains complementary literals in r , violating condition (e).

Hence each $\text{var}(p_i) = z_j$, and we must have $k = 2$, by conditions (c) and (d). Now we reach a contradiction as in the proof of Proposition 18, despite the addition of literals in r and s . By condition (a), we have $u_i \in C_1$, and we deduce that $\bar{u}_i \notin C_2$, contradicting condition (a). \square

Theorem 25. $\{\text{EQ}_n^1\}_{n \in \mathbb{N}}$ has linear-size $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$ refutations.

Proof. By Proposition 24, the total expansion of $\mathcal{D}^{\text{tf}}(\text{EQ}_n^1)$ is obtained by removing universal literals, hence $\text{exp}(\mathcal{D}^{\text{tf}}(\text{EQ}_n^1))$ is the CNF

$$(r \vee \bar{z}_1 \vee \dots \vee \bar{z}_n \vee s) \wedge \bigwedge_{i=1}^n \left((r \vee \bar{x}_i \vee z_i \vee s) \wedge (r \vee x_i \vee z_i \vee s) \right) \wedge (r \vee \bar{s}) \wedge \\ (\bar{r} \vee \bar{z}_1 \vee \dots \vee \bar{z}_n \vee \bar{s}) \wedge \bigwedge_{i=1}^n \left((\bar{r} \vee \bar{x}_i \vee z_i \vee \bar{s}) \wedge (\bar{r} \vee x_i \vee z_i \vee \bar{s}) \right) \wedge (\bar{r} \vee s).$$

By resolution of $(r \vee \bar{s})$ over r with each clause containing \bar{r} , and likewise of $(\bar{r} \vee s)$ with each clause containing r , we obtain all clauses in the CNF

$$(\exp(\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0)) \otimes (s)) \wedge (\exp(\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0)) \otimes (\bar{s})),$$

where $\exp(\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0))$ is the CNF (2) from the proof of Theorem 19. By resolution over s we obtain $\exp(\mathcal{D}^{\text{rrs}}(\text{EQ}_n^0))$ itself, which has a linear-size Resolution refutation by Theorem 19. It is easy to see that the whole refutation of $\exp(\mathcal{D}^{\text{tf}}(\text{EQ}_n^1))$ is of linear size. \square

Theorem 26. $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}}) \not\leq_p^* \forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$.

Proof. It is easy to see that the largest QBF Φ_n^1 that is smaller than EQ_n^1 has $S_s = \{u_1, \dots, u_n\}$ and exactly one other non-empty dependency set $S_{z_i} = \{u_i\}$, where $i = n$ without loss of generality. We will prove that Φ_n^1 has linear-size $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ refutations. We therefore prove the theorem, since by Theorems 23 and 25, and the monotonicity of $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$, $\{\text{EQ}_n^1\}_{n \in \mathbb{N}}$ satisfies all the conditions of Definition 7.

A $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ refutation of Φ_n^1 is a $\forall\text{Exp}+\text{Res}$ refutation of $\mathcal{D}^{\text{rrs}}(\Phi_n^1)$. By definition, $\mathcal{D}^{\text{rrs}}(\Phi_n^1) \leq \Phi_n^1$. Now, Φ_n^1 has linear-size $\forall\text{Exp}+\text{Res}$ refutations: it is readily verified that $\exp(\text{EQ}_n^1)$, which has linear-size Resolution refutations, can be derived from $\exp(\Phi_n^1)$ in a linear number of Resolution steps. Hence $\mathcal{D}^{\text{rrs}}(\Phi_n^1)$ has linear-size $\forall\text{Exp}+\text{Res}$ refutations, by the monotonicity of $\forall\text{Exp}+\text{Res}$; i.e. Φ_n^1 has linear-size $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{rrs}})$ refutations. \square

7 Tautology-free Dependencies for QBF

We now turn our attention to dedicated QBF proof complexity, in particular to the QBF proof systems $\text{Q-Res}(\mathcal{D})$ [34] that were introduced to model dependency-aware QBF solving. We show the following.

Theorem 27. $\text{Q-Res}(\mathcal{D}^{\text{tf}})$ is exponentially stronger than $\text{Q-Res}(\mathcal{D}^{\text{rrs}})$.

Since \mathcal{D}^{rrs} was state-of-the-art for $\text{Q-Res}(\mathcal{D})$, Theorem 27 shows that \mathcal{D}^{tf} is currently the strongest known dependency scheme applicable to dependency-aware QBF solving. We recall the definition of the QBF proof system $\text{Q-Res}(\mathcal{D})$.

Definition 28 ($\text{Q-Res}(\mathcal{D})$ [20, 34]). A Q-Res refutation of a QBF Φ is a sequence C_1, \dots, C_k of clauses in which C_k is empty and each C_i is derived by one of the following rules:

- A** Axiom: C_i is a non-tautological clause in the matrix of Φ ;
- R** Resolution: $C_i = A \vee B$, where $C_r = A \vee x$ and $C_s = B \vee \bar{x}$, for some $r, s < i$ and some $x \in \text{vars}_{\exists}(\Phi)$, and C_i is not a tautology.
- U** Universal reduction: $C_i \vee a = C_r$ for some $r < i$ and some literal a with $\text{var}(a) = u \in \text{vars}_{\forall}(\Phi)$ and $(u, x) \notin \text{deps}(\Phi)$ for each $x \in \text{vars}(C_i)$.

Given a QBF dependency scheme \mathcal{D} , a $\text{Q-Res}(\mathcal{D})$ refutation of Φ is a Q-Res refutation of $\mathcal{D}(\Phi)$.

$\text{Q-Res}(\mathcal{D}^{\text{tf}})$ is complete for QBF by [20], and soundness follows by full exhibition.

Theorem 29. $\text{Q-Res}(\mathcal{D}^{\text{tf}})$ is a QBF proof system.

QBF separation of $\text{Q-Res}(\mathcal{D}^{\text{rrs}})$ and $\text{Q-Res}(\mathcal{D}^{\text{tf}})$. Our separating formulas are the QBFs on which our DQBF modification EQ_n^1 (Definition 21) was based. An exponential lower bound for these formulas in $\text{Q-Res}(\mathcal{D}^{\text{rrs}})$ was shown in [13].

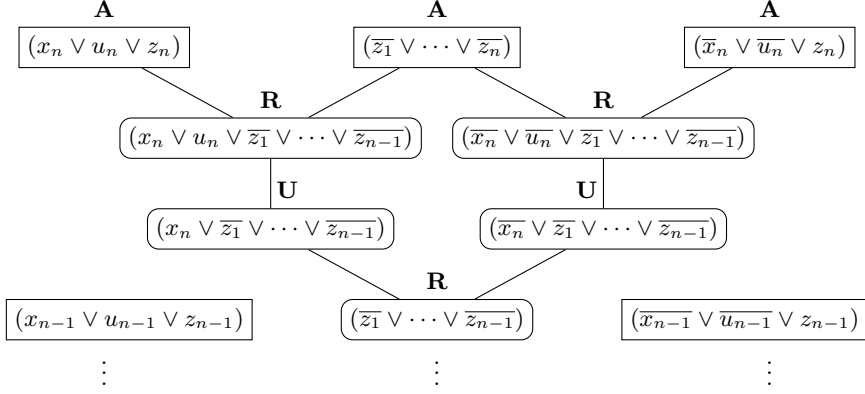


Fig. 2. Linear-size Q-Res refutation of the DQBF EQ_n^0 . In a constant number of steps, EQ_n^0 is reduced to EQ_{n-1}^0 .

Definition 30 (EQ_n^2 [13]). For each natural number n ,

$$\text{EQ}_n^2 := \exists r \exists x_1 \dots \exists x_n \forall u_1 \dots \forall u_n \exists z_1 \dots \exists z_n \exists s \cdot \\ (\psi_n^{\text{EQ}} \otimes (r \vee s)) \wedge (\psi_n^{\text{EQ}} \otimes (\bar{r} \vee \bar{s})) \wedge (r \vee \bar{s}) \wedge (\bar{r} \vee s)$$

Theorem 31 ([6]). EQ_n^2 require exponential-size Q-Res(\mathcal{D}^{rrs}) refutations.

We show that EQ_n^2 have linear-size refutations in Q-Res(\mathcal{D}^{tf}). The proof is along similar lines as our upper bound for EQ_n^1 in $\forall\text{Exp}+\text{Res}(\mathcal{D}^{\text{tf}})$. We first show that \mathcal{D}^{tf} identifies the full set of spurious dependencies, which gives rise naturally to short refutations.

Proposition 32. For each n , the dependency sets of $\mathcal{D}^{\text{tf}}(\text{EQ}_n^2)$ are all empty.

Proof. Aiming for contradiction once again, suppose that there exists a sequence of clauses C_1, \dots, C_k and a sequence of literals p_1, \dots, p_{k-1} satisfying conditions (a) to (e) of Definition 8 with respect to $(u_i, y) \in \text{deps}(\text{EQ}_n^1)$, for some variable $y \in \{z_1, \dots, z_n, s\}$.

As in the proof of Proposition 24, we can deduce that variables r and s do not appear in the sequence of literals. By condition (a) we have $u \in C_1$. By condition (c) we have $p_1 = z_i$ and $C_2 = (\bar{z}_1 \vee \dots \vee \bar{z}_n \vee a)$, with $\text{var}(a) = s$. By conditions (c) and (d), we have $p_2 = \bar{z}_j$ for some $j \neq i$. By condition (c) $z_j \in C_2$, and by conditions (c) and (d) we must have $k = 2$. This violates condition (a), since $\bar{u}_i \notin C_2$. \square

Theorem 33. $\{\text{EQ}_n^2\}_{n \in \mathbb{N}}$ has linear-size Q-Res(\mathcal{D}^{tf}) refutations.

Proof. By Proposition 32, $\text{deps}(\mathcal{D}^{\text{tf}}(\text{EQ}_n^2))$ is the empty relation. It follows that all universal literals in the matrix may be removed by universal reduction. Hence, with a single axiom and universal reduction step per clause, we derive the clauses of $\text{exp}(\mathcal{D}^{\text{tf}}(\text{EQ}_n^1))$ from the proof of Theorem 25. Each step of the linear-size Resolution refutation described there is also available in Q-Res(\mathcal{D}^{tf}). \square

8 Conclusions

We conclude with an interesting observation and a question for future research. The family $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ from Definition 16 is an adaptation of the equality QBFs $\{\text{EQ}_n\}_{n \in \mathbb{N}}$ from [8], obtained by shrinking the dependency set of each z_i to just $\{u_i\}$. While in QBF $\{\text{EQ}_n\}_{n \in \mathbb{N}}$

requires exponentially long proofs in both $\forall\text{Exp}+\text{Res}$ and Q-Res [5, 8], in DQBF $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ remains hard only for $\forall\text{Exp}+\text{Res}$. Indeed, even though Q-Res is incomplete for DQBF, it is sound, and $\{\text{EQ}_n^0\}_{n \in \mathbb{N}}$ has linear-size Q-Res refutations, as shown in Figure 2. This suggests that there may be some hidden proof-complexity relationship between $\forall\text{Exp}+\text{Res}$ and Q-Res in DQBF, even though Q-Res is incomplete there.

We have presented the strongest known dependency scheme \mathcal{D}^{tf} . A natural question is whether some even stronger dependency schemes for (D)QBF exist.

References

1. Azhar, S., Peterson, G., Reif, J.: Lower bounds for multiplayer non-cooperative games of incomplete information. *Journal of Computers and Mathematics with Applications* **41**, 957 – 992 (2001)
2. Balabanov, V., Jiang, J.R.: Unified QBF certification and its applications. *Formal Methods in System Design* **41**(1), 45–65 (2012)
3. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Sinz, C., Egly, U. (eds.) *International Conference on Theory and Practice of Satisfiability Testing (SAT)*. Lecture Notes in Computer Science, vol. 8561, pp. 154–169. Springer (2014)
4. Beyersdorff, O., Blinkhorn, J.: Dependency schemes in QBF calculi: Semantics and soundness. In: Rueher, M. (ed.) *International Conference on Principles and Practice of Constraint Programming (CP)*. Lecture Notes in Computer Science, vol. 9892, pp. 96–112. Springer (2016)
5. Beyersdorff, O., Blinkhorn, J.: Genuine lower bounds for QBF expansion. In: Niedermeier, R., Vallée, B. (eds.) *International Symposium on Theoretical Aspects of Computer Science (STACS)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 96, pp. 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)
6. Beyersdorff, O., Blinkhorn, J.: Dynamic QBF dependencies in reduction and expansion. *ACM Transactions on Computational Logic* **21**(2), 1–27 (2019)
7. Beyersdorff, O., Blinkhorn, J., Chew, L., Schmidt, R.A., Suda, M.: Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. *Journal of Automated Reasoning* **63**(3), 597–623 (2019)
8. 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)
9. Beyersdorff, O., Bonacina, I., Chew, L.: Lower bounds: From circuits to QBF proof systems. In: Sudan, M. (ed.) *ACM Conference on Innovations in Theoretical Computer Science (ITCS)*. pp. 249–260. ACM (2016)
10. Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. *ACM Transactions on Computation Theory* **11**(4), 26:1–26:42 (2019)
11. Beyersdorff, O., Chew, L., Schmidt, R.A., Suda, M.: Lifting QBF resolution calculi to DQBF. In: Creignou, N., Berre, D.L. (eds.) *International Conference on Theory and Practice of Satisfiability Testing (SAT)*. Lecture Notes in Computer Science, vol. 9710, pp. 490–499. Springer (2016)
12. Beyersdorff, O., Hinde, L., Pich, J.: Reasons for hardness in QBF proof systems. *ACM Transactions on Computation Theory* **12**(2), 10:1–10:27 (2020)
13. Blinkhorn, J., Beyersdorff, O.: Shortening QBF proofs with dependency schemes. In: Gaspers, S., Walsh, T. (eds.) *International Conference on Theory and Practice of Satisfiability Testing (SAT)*. Lecture Notes in Computer Science, vol. 10491, pp. 263–280. Springer (2017)
14. Chen, H.: Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. *ACM Transactions on Computation Theory* **9**(3), 15:1–15:20 (2017)
15. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. *Journal of Symbolic Logic* **44**(1), 36–50 (1979)
16. Egly, U., Kronegger, M., Lonsing, F., Pfandler, A.: Conformant planning as a case study of incremental QBF solving. *Annals of Mathematics and Artificial Intelligence* **80**(1), 21–45 (2017)
17. Henkin, L.: Some remarks on infinitely long formulas. *Infinitistic Methods* pp. 167 – 183 (1961)
18. Hoos, H.H., Peitl, T., Slivovsky, F., Szeider, S.: ,portfolio-based algorithm selection for circuit QBFs. In: Hooker, J.N. (ed.) *International Conference on Principles and Practice of Constraint Programming (CP)*. Lecture Notes in Computer Science, vol. 11008, pp. 195–209. Springer (2018)
19. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science* **577**, 25–42 (2015)
20. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. *Information and Computation* **117**(1), 12–18 (1995)
21. Kontchakov, R., Pulina, L., Sattler, U., Schneider, T., Selmer, P., Wolter, F., Zakharyashev, M.: Minimal module extraction from DL-lite ontologies using QBF solvers. In: Boutilier, C. (ed.) *International Joint Conference on Artificial Intelligence (IJCAI)*. pp. 836–841. AAAI Press (2009)

22. Lonsing, F., Egly, U.: DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In: de Moura, L. (ed.) Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Proceedings. Lecture Notes in Computer Science, vol. 10395, pp. 371–384. Springer (2017)
23. Lonsing, F., Egly, U.: Evaluating QBF solvers: Quantifier alternations matter. In: Hooker, J.N. (ed.) International Conference on Principles and Practice of Constraint Programming (CP). Lecture Notes in Computer Science, vol. 11008, pp. 276–294. Springer (2018)
24. Mangassarian, H., Veneris, A.G., Benedetti, M.: Robust QBF encodings for sequential circuits with applications to verification, debug, and test. *IEEE Transactions on Computers* **59**(7), 981–994 (2010)
25. Peitl, T., Slivovsky, F., Szeider, S.: Combining resolution-path dependencies with dependency learning. In: International Conference on Theory and Practice of Satisfiability Testing (SAT). pp. 306–318 (2019)
26. Peitl, T., Slivovsky, F., Szeider, S.: Dependency learning for QBF. *Journal of Artificial Intelligence Research* **65**, 181–208 (2019)
27. Peitl, T., Slivovsky, F., Szeider, S.: Long-distance Q-resolution with dependency schemes. *Journal of Automated Reasoning* **63**(1), 127–155 (2019)
28. Pulina, L., Seidl, M.: The 2016 and 2017 QBF solvers evaluations (QBFEVAL’16 and QBFEVAL’17). *Artif. Intell.* **274**, 224–248 (2019)
29. Rabe, M.N.: A resolution-style proof system for DQBF. In: Gaspers, S., Walsh, T. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 10491, pp. 314–325. Springer (2017)
30. Robinson, J.A.: A machine-oriented logic based on the resolution principle. *Journal of the ACM* **12**(1), 23–41 (1965)
31. Samer, M., Szeider, S.: Backdoor sets of quantified Boolean formulas. *Journal of Automated Reasoning* **42**(1), 77–97 (2009)
32. Scholl, C., Wimmer, R.: Dependency quantified boolean formulas: An overview of solution methods and applications - extended abstract. In: Beyersdorff, O., Wintersteiger, C.M. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 10929, pp. 3–16. Springer (2018)
33. Slivovsky, F.: Structure in #SAT and QBF. Ph.D. thesis, Vienna University of Technology (2015)
34. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. *Theoretical Computer Science* **612**, 83–101 (2016)
35. Vardi, M.Y.: Boolean satisfiability: Theory and engineering. *Communications of the ACM* **57**(3), 5 (2014)
36. Wimmer, R., Scholl, C., Wimmer, K., Becker, B.: Dependency schemes for DQBF. In: Creignou, N., Berre, D.L. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 9710, pp. 473–489. Springer (2016)