Electronic Colloquium on Computational Complexity, Report No. 135 (2021)

Strong (D)QBF Dependency Schemes via Implication-free Resolution Paths

OLAF BEYERSDORFF, JOSHUA BLINKHORN, and TOMÁŠ PEITL, Friedrich-Schiller-Universität Jena, Germany

We suggest a general framework to study dependency schemes for dependency quantified Boolean formulas (DQBF). As our main contribution, we exhibit a new *infinite collection* of *implication-free* DQBF dependency schemes that generalise the reflexive resolution path dependency scheme. We establish soundness of all these schemes, implying that they can be used in any DQBF proof system. We further explore the power of QBF and DQBF resolution systems parameterised by implication-free dependency schemes and show that the hierarchical structure naturally present among the dependency schemes translates isomorphically to a hierarchical structure of parameterised proof systems with respect to p-simulation.

As a special case, we demonstrate that our new schemes are exponentially stronger than the reflexive resolution path dependency scheme when used in Q-resolution, thus resulting in the strongest QBF dependency schemes known to date.

Additional Key Words and Phrases: 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 [35]. These build on the success of SAT solving [41], but also incorporate new ideas genuine to the QBF domain, such as expansion solving [26] and dependency schemes [37]. Due to its PSPACE completeness, QBF solving is relevant to many application domains that cannot be efficiently encoded into SAT [22, 28, 31]. On the theoretical side, there is a substantial body of QBF proof complexity results (e.g. [3, 6, 8, 10, 11]), 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 quantification[1] 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 uses dependency schemes, for example the solvers DepQBF [29] and Qute [32, 33], and experiments show dependency-aware solving is particularly competitive on QBFs with high quantifier complexity [25, 30]. The performance gains are also underlined by theoretical findings. There is a sequence of results [7, 34, 40] 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 [27] and long-distance Q-resolution [2]. In [6] it is demonstrated that using the reflexive resolution path dependency scheme ($\mathcal{D}^{\text{rrs}}$ [40]) in Q-resolution can exponentially shorten proofs.

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 [23]. Compared to QBFs, DQBFs boost reasoning power and enable further applications (cf. [38] for an overview).

[1] The standard input for solvers is a prenex QBF.

Authors' address: Olaf Beyersdorff, olaf.beyersdorff@uni-jena.de; Joshua Blinkhorn, joshua.blinkhorn@uni-jena.de; Tomáš Peitl, Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743, Jena, Thüringen, Germany, tomas.peitl@uni-jena.de.

ISSN 1433-8092

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 Beyersdorff et al. [7] suggest 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 [42]. Indeed, Wimmer et al. [42] showed that several dependency schemes for QBF, including $\mathcal{D}^{\mathrm{rrs}}$, can be lifted to DQBF. They also demonstrated 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 parameterised system $\mathsf{P}(\mathcal{D})$. In fact, in our framework a proof of Φ in $\mathsf{P}(\mathcal{D})$ 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, 39], 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, 34].

We also extend the notion of *genuine* proof size lower bounds [13, 15] 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 implication-free dependency schemes. We define and analyse a new *infinite* class of *implication-free* dependency schemes $\mathcal{D}^F(R, \lambda)$, parameterised by an *implication relation* R, and an integer λ. Each implication-free dependency scheme generalises the reflexive resolution-path dependency scheme $\mathcal{D}^{\mathrm{rrs}}$ [40] by discarding resolution paths subsumed by certain propositional implicants of the input formula, thereby identifying more spurious dependencies through missing resolution-path connections. Which implicants can be used in this step is dictated by the implication relation—the more implicants, the more spurious dependencies found, though possibly at an increased computational cost. How long a segment of each path is checked for subsumption is controlled by λ—again with a trade-off between effectiveness and efficiency.

$\mathcal{D}^{\mathrm{rrs}}$ itself is in fact also an implication-free dependency scheme, as is $\mathcal{D}^{\mathrm{tf}}$, the *tautology-free* dependency scheme, which we introduced in the conference version of this paper [9], and which uses tautologies in place of implicants and a λ value of 1. Prior to this paper, $\mathcal{D}^{\mathrm{rrs}}$ was the strongest known DQBF dependency scheme—now it is superseded by the implication-free schemes.

We show that all implication-free schemes are fully exhibited, and therefore sound, by reducing their full exhibition to that of $\mathcal{D}^{\mathrm{rrs}}$. For this, we point out that the full exhibition of $\mathcal{D}^{\mathrm{rrs}}$ on DQBF is an immediate consequence of results of Wimmer et al. [42].

C. Exponential separations of (D)QBF proof systems. To demonstrate the strength of our new schemes, we show that they can exponentially shorten proofs in DQBF, as well as QBF, proof systems. As a case study, we consider the expansion calculus ∀Exp+Res and the basic QBF system Q-resolution (Q-Res) [27]. These two proof systems form the foundations of QBF proof theory: Q-Res, the first QBF proof system defined, is the foundation for proof systems that model search-based solvers; ∀Exp+Res is the basis for expansion-based proof systems and solvers [26]. Of the two, only ∀Exp+Res lifts to a DQBF proof system—attempts to lift Q-Res and its derivatives to DQBF run into issues with either soundness or completeness [12].

We show that parametrizing these proof systems by a particular class of implication-free dependency schemes (which we call *normal*) operates homomorphically: the ordering of the dependency schemes with respect to their strength carries over to the ordering of the proof systems with respect to p-simulation.

Since there exist no prior DQBF proof complexity results whatsoever, this entails proving exponential proof-size lower bounds in an infinite number of proof systems. We obtain these by introducing a parameterised class of DQBF versions of the equality formulas (originally QBFs [8, 14]). 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.

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 implication-free schemes $\mathcal{D}^F(R, \lambda)$ and show their soundness. In Section 6 we prove the proof complexity upper and lower bounds needed to establish the hierarchy of proof systems corresponding to the hierarchical structure of implication-free schemes. Finally, in Section 7 we discuss the computational cost of applying implication-free schemes.

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* $\overline{z}$. The negation of a literal a is denoted $\overline{a}$, where $\overline{\overline{z}} := z$ 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 $\mathrm{vars}(\psi)$. 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 $C\restriction_Z := \{a \in C : \mathrm{var}(a) \in Z\}$.

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 $\mathrm{vars}_\forall(\Psi)$ and $\mathrm{vars}_\exists(\Psi)$ we denote the universal and existential variable sets of Ψ, and with $\mathrm{vars}(\Psi)$ their union. We deal only with *closed* DQBFs, in which $\mathrm{vars}(\psi) \subseteq \mathrm{vars}(\Psi)$. We define a relation $\mathrm{deps}(\Psi)$ on $\mathrm{vars}_\forall(\Psi) \times \mathrm{vars}_\exists(\Psi)$, where $(u, x) \in \mathrm{deps}(\Psi)$ 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 $\alpha\restriction_{Z'}$. 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. $\overline{0} \mapsto 1$, $\overline{1} \mapsto 0$, $\phi \vee 0 \mapsto \phi$, $\phi \vee 1 \mapsto 1$, $\phi \wedge 1 \mapsto \phi$, and $\phi \wedge 0 \mapsto 0$. We say that α *satisfies* ψ when $\psi[\alpha] = 1$, and *falsifies* ψ when $\psi[\alpha] = 0$. For a non-tautological clause C, we denote by $\overline{C}$ the unique assignment $\overline{C} : \text{vars}(C) \to \{0, 1\}$ that falsifies C.

A *model* for a DQBF $\Psi := \Pi \cdot \psi$ is a set of functions $f := \{f_x : x \in \text{vars}_\exists(\Psi)\}$, $f_x : \langle S_x \rangle \to \langle \{x\} \rangle$, for which, for each $\alpha \in \langle \text{vars}_\forall(\Psi) \rangle$, the combined assignment $\alpha \cup \{f_x(\alpha \restriction_{S_x}) : x \in \text{vars}_\exists(\Psi)\}$ 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 $\Psi \overset{\text{tr}}{\equiv} \Psi'$.

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 $y_1, \ldots, y_k$ 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 $y_i^{\overline{u}}$. 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, $\exp(\Psi, u) := \Pi' \cdot \psi'$, where Π' is obtained from Π by removing $\forall u$ and replacing each $\exists y_i(S_{y_i})$ with $\exists y_i^{\overline{u}}(S_{y_i} \setminus \{u\}) \, \exists y_i^u(S_{y_i} \setminus \{u\})$, and

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

Universal expansion is known to preserve the truth value, i.e. $\Psi \overset{\text{tr}}{\equiv} \exp(\Psi, u)$. 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 $\exp(\Psi, U)$. Expansion by the whole set $\text{vars}_\forall(\Psi)$ is denoted $\exp(\Psi)$, 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 [37] 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 following relation.

Definition 3.1. We define the relation $\leq$ on DQBF $\times$ 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 3.2 (dependency scheme). A *dependency scheme* is a polynomial-time computable function $\mathcal{D}$: DQBF $\rightarrow$ 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 [42], is *monotonicity*.[2]

Definition 3.3 (monotone (adapted from [42])). 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'.

Dependency schemes can be compared on their *generality*—the more general a dependency scheme, the more independence it detects.

Definition 3.4 (more general). We say that a dependency scheme $\mathcal{D}$ is *more general* than a dependency scheme $\mathcal{D}'$, written $\mathcal{D}' \leq \mathcal{D}$, if $\mathcal{D}(\Psi) \leq \mathcal{D}'(\Psi)$ for all 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 [39], 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 3.5 (full exhibition [4, 39]). A dependency scheme $\mathcal{D}$ is called *fully exhibited* when $\Psi \overset{\text{tr}}{\equiv} \mathcal{D}(\Psi)$, for all Ψ.

Dependency schemes preserve falsity by definition; that is, if Ψ is false, so is $\mathcal{D}(\Psi)$, since $\mathcal{D}(\Psi) \leq \Psi$. Therefore fully exhibited dependency schemes can also be characterised as those that preserve truth. It is easy to see that full exhibition carries over to less general schemes.

PROPOSITION 3.6. *If $\mathcal{D}$ is fully exhibited and more general than $\mathcal{D}'$, then $\mathcal{D}'$ is fully exhibited.*

4 PARAMETERISING 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 [13], 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 [16], a *DQBF proof system* over an alphabet Σ is a polynomial-time computable onto function $\mathsf{P} : \Sigma^* \rightarrow$ 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) *Completeness:* every false DQBF has a refutation (P is onto); (3) *Checkability:* refutations can be checked efficiently (P is polynomial-time computable).

[2] A different notion of monotonicity for dependency schemes is defined in [34].

Two concrete examples of DQBF proof systems from the literature are the fundamental expansion-based system $\forall$Exp+Res [7], and the more sophisticated instantiation-based system IR-calc [7].

Incorporating dependency schemes. A dependency scheme, interpreted as a DQBF mapping as in Definition 3.2, can be combined with an arbitrary proof system in a straightforward manner.

Definition 4.1 ($\mathsf{P}(\mathcal{D})$). Let P be a DQBF proof system and let $\mathcal{D}$ be a dependency scheme. A $\mathsf{P}(\mathcal{D})$ refutation of a DQBF Ψ is a P refutation of $\mathcal{D}(\Psi)$.

The proof system $\mathsf{P}(\mathcal{D})$ 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, 40]).

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.

Proposition 4.2. *Given a DQBF proof system P and a dependency scheme $\mathcal{D}$, $\mathsf{P}(\mathcal{D})$ is sound if, and only if, $\mathcal{D}$ is fully exhibited.*

Proof. Suppose that $\mathcal{D}$ is fully exhibited. Let π be a $\mathsf{P}(\mathcal{D})$ 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 $\mathsf{P}(\mathcal{D})$ 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 $\mathsf{P}(\mathcal{D})$ is not sound. □

Note that completeness and checkability of P are preserved trivially by any dependency scheme, so we can even say that $\mathsf{P}(\mathcal{D})$ 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 [16]. Let P and Q be DQBF proof systems. We say that P *p-simulates* Q (written $\mathsf{Q} \leq_p \mathsf{P}$) 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 $\mathsf{Q} \not\leq_p \mathsf{P}$, 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.

[3] This holds for all known DQBF proof systems.

Definition 4.3. 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.

5 THE IMPLICATION-FREE DEPENDENCY SCHEMES

In this section we will define an infinite parameterised class of dependency schemes that generalize the *reflexive resolution-path dependency scheme* $\mathcal{D}^{\mathrm{rrs}}$ [40], until now the most general (D)QBF dependency scheme. We introduce $\mathcal{D}^{\mathrm{rrs}}$ formally later, in Definition 5.4, as a special case. In the conference version of this paper [9], we proposed a generalization of $\mathcal{D}^{\mathrm{rrs}}$ called the *tautology-free dependency scheme*, $\mathcal{D}^{\mathrm{tf}}$, which is also a special case in our new class; $\mathcal{D}^{\mathrm{tf}}$ is formally presented in Definition 5.5.

Before we proceed with technical details, let us discuss informally the motivation and intuition for these generalizations of $\mathcal{D}^{\mathrm{rrs}}$. The main building stone of $\mathcal{D}^{\mathrm{rrs}}$ are *resolution paths*, which are sequences of clauses where each consecutive pair contains a pair of complementary existential literals.

$$\cdots \longrightarrow C \vee p \longrightarrow \overline{p} \vee D \longrightarrow \cdots$$

A resolution path that consists of binary clauses establishes a chain of implications—falsifying the first literal will propagate values throughout the chain and set the value of the last literal. Hence, such a resolution path makes its endpoints depend on each other. Even if a resolution path consists of clauses of larger width, it holds the potential, under the right partial assignment, to reduce to a width-2 chain and enforce a dependency.

$\mathcal{D}^{\mathrm{rrs}}$ identifies independent pairs of variables based on the non-existence of resolution paths connecting them. Suppose x depends on u syntactically ($u \in S_x$). This dependence is only real if a change of the value of u can enforce a change of the value of x, and moreover if both values can be enforced. Such a change of value, in turn, can only be enforced through a resolution path. Therefore, there must be a pair of resolution paths that connects complementary literals on u with complementary literals on x, for the dependency to be realizable. This pair of paths can in fact be concatenated at the endpoints that contain x and $\overline{x}$, creating a single joint path from u through x and $\overline{x}$ (or $\overline{x}$ and x) to $\overline{u}$.

$$(u \vee C_1 \vee p_1) \rightarrow (\overline{p_1} \vee C_2 \vee p_2) \rightarrow \cdots \rightarrow (\overline{p_{i-1}} \vee C_i \vee x) \dashrightarrow (\overline{x} \vee C_{i+1} \vee p_{i+1}) \rightarrow \cdots \rightarrow (\overline{p_{k-1}} \vee C_k \vee \overline{u})$$

The existence of two paths is equivalent to the existence of one joint path, and this is also how we define resolution path dependencies in this paper (Definition 5.3). If there is no such joint resolution path, $\mathcal{D}^{\mathrm{rrs}}$ detects independence and removes u from the dependency set of x.

We build on this intuition by observing that some resolution paths are in fact unable to carry information between their endpoints because they can never reduce to a chain of binary clauses. Let us restrict ourselves to QBFs for a moment, and imagine we are evaluating whether x depends on u, and there is another existential variable y, which needs to be assigned *before* u (i.e. comes earlier in the prefix). If, for a resolution path connecting u and x, for each assignment to y a clause on the path is satisfied, then the path can never reduce to a chain of implications—at the 'moment' of assigning u one of the clauses will always have disappeared. Such resolution paths can therefore be safely disregarded when checking whether x depends on u.

The concept of the order in which variables are assigned is quite natural in QBF, where each formula represents a game and players take turns in the order of the quantifier prefix. It is not clear how we can get a full analogue of such an ordering in DQBF, where variables are ordered inherently non-linearly, but we can at least get an approximation in the form of the *independent existential variables* $I_\exists(\Psi)$—the variables in a DQBF Ψ with empty dependency sets. For all intents and purposes, we can also think of $I_\exists(\Psi)$ as *free* variables in the usual sense. These variables arguably need to be assigned before anything else. Along with the observation from the previous paragraph, this gives rise to the tautology-free dependency scheme $\mathcal{D}^{\text{tf}}$ defined in the conference version of this paper [9]. We simply check for each candidate resolution path whether or not it contains both literals of some independent existential variable $y \in I_\exists(\Psi)$—if (and only if) it does, then every assignment to $I_\exists(\Psi)$ satisfies a clause on the path and the path is discarded.

Here, we further strengthen the observation that leads to $\mathcal{D}^{\text{tf}}$, by not considering all assignments to $I_\exists(\Psi)$, but only those that are *propositionally consistent* with the matrix—those that can be extended to a full satisfying assignment of the matrix. This rests on the idea that under an inconsistent assignment to $I_\exists(\Psi)$, no real value propagation can occur, because the matrix is already propositionally unsatisfiable, and so further assignments, and indeed dependencies, do not matter anymore. Another way to look at this is to consider what unit propagation does—it extends an assignment with further, *forced* assignments that are necessary to maintain propositional consistency. In a propositionally inconsistent state, it no longer makes sense to speak of a semantically forced assignment. We capture this by collecting, for a candidate resolution path, all the independent existential literals appearing on the path, and asking whether the clause formed by them is implied by the matrix of our formula Ψ—if (and only if) not, then all those literals can be simultaneously falsified by a consistent assignment to $I_\exists(\Psi)$ and the path can be restricted without satisfying any of its clauses. If, on the other hand, this clause is implied, then every propositionally consistent assignment to $I_\exists(\Psi)$ must satisfy one of the clauses on the path, and the path can safely be discarded.

In practice, because full propositional entailment is coNP-complete, we will also ask whether the clause is a particular, polynomial-time identifiable kind of implied clause—which leads us to the notion of an *implication relation*.

Definition 5.1 (implication relation). An *implication relation* is a relation $R \subseteq \text{CNF} \times \mathcal{C}$ between CNFs and clauses such that for every CNF ψ and clause C, if $(\psi, C) \in R$, then $\psi \models C$. If $(\psi, C) \in R$, we say that C is R-implied by ψ.

We will be particularly interested in well-behaved, *normal*, implication relations.

Definition 5.2. We say that an implication relation R is *normal* if

(1) (weakening) whenever $(\psi, C) \in R$ and $\psi \subseteq \phi, C \subseteq D$, then $(\phi, D) \in R$;
(2) (restriction) for every partial assignment $\alpha \in \langle V \rangle$, $V \subseteq \text{var}(\psi)$, if $(\psi, C) \in R$, then $(\psi[\alpha], C[\alpha]) \in R$;
(3) (expansion) if C is non-tautological, then $(\psi[\overline{C}], \emptyset) \in R \implies (\psi, C) \in R$; and
(4) (tautology closure) $(\psi, C) \in R \iff (\phi, D) \in R$ for any ψ, ϕ and any tautological clauses C, D.

Several important examples of (normal) implication relations are listed below.

- $R_\emptyset = \emptyset$ is the empty implication relation.
- $R_\models$ is the full implication relation, i.e $R_\models = \{(\psi, C) : \psi \models C\}$. For every implication relation R, we have $R \subseteq R_\models$.
- $R_\top$ is the maximal implication relation where every clause is a tautology. Property 4. of normality can be restated as $R \cap R_\top \neq \emptyset \implies R_\top \subseteq R$. Together with Property 1., this can be strengthened to $R \neq R_\emptyset \implies R_\top \subseteq R$.

We are now ready to define our class of *implication-free* dependency schemes. Apart from an implication relation, they also have an integer parameter that controls the length of the path segment checked for the implication relation. We will later see (Theorem 7.2) that this has a significant impact on computational complexity.

Let $\mathbb{N}$ denote the set of positive integers, and $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$.

Definition 5.3 (implication-free dependency schemes $\mathcal{D}^F(R, \lambda)$). Let R be an implication relation. For any $\lambda \in \mathbb{N}$, the *λ-locally R-free dependency scheme* $\mathcal{D}^F(R, \lambda)$ 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'_{x_i} is the set of universal variables $u \in S_{x_i}$ for which there exists a *λ-locally R-free (u, x_i)-resolution path*, i.e. a sequence $C_1, \ldots, C_k$ of clauses in ψ and a sequence $p_1, \ldots, p_{k-1}$ of existential literals satisfying the following conditions:

(a) $u \in C_1$ and $\overline{u} \in C_k$;
(b) for some $j \in [k-1]$, $x_i = \mathrm{var}(p_j)$;
(c) for each $j \in [k-1]$, $p_j \in C_j$, $\overline{p}_j \in C_{j+1}$, and $u \in S_{\mathrm{var}(p_j)}$;
(d) for each $j \in [k-2]$, $\mathrm{var}(p_j) \neq \mathrm{var}(p_{j+1})$.
(e) for each $j \in [k-\lambda]$, $\left(\psi, (C_j \cup \cdots \cup C_{j+\lambda}) \restriction_{I_\exists(\Psi)}\right) \notin R$.

The *globally R-free dependency scheme* $\mathcal{D}^F(R, \infty)$ replaces condition (e) with

(e′) $\left(\psi, (C_1 \cup \cdots \cup C_k) \restriction_{I_\exists(\Psi)}\right) \notin R$.

For the empty implication relation $R_\emptyset$, conditions (e) and (e′) are satisfied vacuously, and so $\mathcal{D}^F(R_\emptyset, \lambda)$ reduces to just $\mathcal{D}^{\mathrm{rrs}}$ (c.f. the original definition of $\mathcal{D}^{\mathrm{rrs}}$).

Definition 5.4 ($\mathcal{D}^{\mathrm{rrs}}$ [40]). For each $\lambda \in \mathbb{N}^\infty$, $\mathcal{D}^{\mathrm{rrs}} := \mathcal{D}^F(R_\emptyset, \lambda)$.

The *tautology-free dependency scheme* $\mathcal{D}^{\mathrm{tf}}$, which we introduced in the conference version of this paper [9], is also a special case.

Definition 5.5. $\mathcal{D}^{\mathrm{tf}} := \mathcal{D}^F(R_\top, 1)$.

Wimmer et al. [42] essentially showed that $\mathcal{D}^{\mathrm{rrs}}$ is fully exhibited, even though they did not use that term. Theorems 3 and 4 in [42] 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 5.6 (Wimmer et al. [42]). *$\mathcal{D}^{\mathrm{rrs}}$ is fully exhibited.*

We show full exhibition of any $\mathcal{D}^F(R, \lambda)$ by reduction to full exhibition of $\mathcal{D}^{\mathrm{rrs}}$. In order to show that, it is sufficient to show that $\mathcal{D}^F(R_\vDash, \infty)$ is fully exhibited—it can be easily seen from Definition 5.3 that $\mathcal{D}^F(R_\vDash, \infty)$ is the *most general* implication-free dependency scheme.

Proposition 5.7. *For every implication relation R and $\lambda \in \mathbb{N}^\infty$, $\mathcal{D}^F(R, \lambda) \leq \mathcal{D}^F(R_\vDash, \infty)$.*

Proof. Conditions (a)-(d) of Definition 5.3 are independent of R and λ. Whenever condition (e′) is satisfied for $R_\vDash$, then, by definition, both (e) and (e′) are satisfied for any implication relation R and any $(\lambda + 1)$-sized segment. □

Similarly, $\mathcal{D}^{\mathrm{rrs}}$ is the *least general* implication-free dependency scheme.

Proposition 5.8. *For every implication relation R and $\lambda \in \mathbb{N}^\infty$, $\mathcal{D}^{\text{rrs}} \leq \mathcal{D}^F(R, \lambda)$.*

Proof. Conditions (e) and (e') are always satisfied for $\mathcal{D}^{\text{rrs}}$. □

Under the assumption of normality, we can similarly order any two implication-free dependency schemes whose implication relations are comparable.

Lemma 5.9. *Let $R \subseteq R'$ be normal implication relations and $\lambda \leq \lambda' \in \mathbb{N}^\infty$. Then $\mathcal{D}^F(R, \lambda) \leq \mathcal{D}^F(R', \lambda')$.*

Proof. All we need to observe is that by checking a potentially longer path segment (λ' instead of λ), we can collect a potentially larger clause $D \supseteq C$, but the weakening property ensures that whenever C is R-implied, so is D. □

Theorem 5.10. *$\mathcal{D}^F(R_\models, \infty)$ is fully exhibited.*

Proof. Since $\mathcal{D}^F(R_\models, \infty)(\Psi) \leq \Psi$, we only need to show that if Ψ is true, then $\mathcal{D}^F(R_\models, \infty)(\Psi)$ 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 $(u, x) \in \text{deps}(\mathcal{D}^F(R_\models, \infty)(\Psi))$.

Consider the sequences $C_1, \ldots, C_k$ and $p_1, \ldots, p_{k-1}$ that witness $(u, x) \in \text{deps}(\mathcal{D}^{\text{rrs}}(\Psi[\sigma]))$. For each C_i there is $C'_i \in \Psi$, such that $C_i = C'_i[\sigma]$, i.e. $C'_i \subseteq C_i \cup \overline{\sigma}$, where $\overline{\sigma}$ is the largest clause falsified by σ. It is readily verified that the sequences $C'_1, \ldots, C'_k$ and $p_1, \ldots, p_{k-1}$ witness $(u, x) \in \text{deps}(\mathcal{D}^F(R_\models, \infty)(\Psi))$. In particular, the clause $(C'_1 \cup \cdots \cup C'_k)\restriction_{I_\exists(\Psi)} \subseteq \overline{\sigma}$ is not implied, because the assignment σ falsifies it, but leaves the formula $\Psi[\sigma]$ true, and therefore leaves its matrix propositionally satisfiable.

Hence, we get $\mathcal{D}^{\text{rrs}}(\Psi[\sigma]) \leq \mathcal{D}^F(R_\models, \infty)(\Psi)[\sigma]$. By full exhibition of $\mathcal{D}^{\text{rrs}}$, we have that $\mathcal{D}^{\text{rrs}}(\Psi[\sigma])$ is true, which means $\mathcal{D}^F(R_\models, \infty)(\Psi)[\sigma]$ is true, and hence $\mathcal{D}^F(R_\models, \infty)(\Psi)$ is true. □

Corollary 5.11. *For any implication relation R and any $\lambda \in \mathbb{N}^\infty$, $\mathcal{D}^F(R, \lambda)$ is fully exhibited.*

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

Example 5.12. 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 $C_2 = \overline{x} \vee \overline{u} \vee \overline{z}$. 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 5.3 (with the implication relation $R = R_\top$) because $(C_1 \cup C_2)\restriction_{I_\exists(\Psi)}$ is a tautology on $x \in I_\exists(\Psi)$. Since there are no other sequences to consider, we conclude that $(u, z) \notin \text{deps}(\mathcal{D}^{\text{tf}}(\Psi))$.

As we have just shown $\mathcal{D}^{\text{tf}}(\Psi) = \exists x \exists z \forall u \cdot (x \vee z \vee u) \wedge (\overline{x} \vee \overline{z} \vee \overline{u})$. 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 $\mathcal{D}^{\text{tf}}$.

We remark that the parameters R and λ in Definition 5.3 could in principle also be turned into functions of the input formula. While in this paper we focus on the case where the parameters are fixed, it is straightforward to see that Proposition 5.7, and by extension Corollary 5.11, generalize to the case of functional parameters as well.

6 PROOF COMPLEXITY

In this section we turn to proof complexity of QBF and DQBF proof systems parametrized by implication-free dependency schemes. The base proof systems we analyze are ∀Exp+Res, which is a full-blown DQBF proof system, and Q-Res, which is sound on DQBF but only complete for QBF, and hence is treated as a QBF proof system (we define both proof systems in Subsection 6.1). The central result is Theorem 6.1, which lifts the generality ordering between implication-free dependency schemes established by Lemma 5.9 onto the level of proof systems.

Theorem 6.1. *Let $P \in \{\forall\mathsf{Exp+Res}, \mathsf{Q\text{-}Res}\}$, R, R' non-empty normal implication relations, and $\lambda, \lambda' \in \mathbb{N}^\infty$. Then*

$$P(\mathcal{D}^F(R,\lambda)) \quad \leq_p \quad P(\mathcal{D}^F(R',\lambda')) \quad \Longleftrightarrow \quad R \subseteq R' \text{ and } \lambda \leq \lambda',$$

and whenever $R \nsubseteq R'$ or $\lambda > \lambda'$, the separation $\forall\mathsf{Exp+Res}(\mathcal{D}^F(R,\lambda)) \nleq_p^* \forall\mathsf{Exp+Res}(\mathcal{D}^F(R',\lambda'))$ is DQBF-genuine.

Theorem 6.1 is phrased with elegance in mind, but as such requires some explanation. The part where $P = \mathsf{Q\text{-}Res}$ refers to the proof system on QBF; the part where $P = \forall\mathsf{Exp+Res}$ has a twofold interpretation—first as a DQBF proof system, and second as its QBF restriction. In effect, what we aim to say is that the simulations hold on all of DQBF, while the separations can already be obtained with QBF; and DQBF-genuine separations exist as well.

In spite of its elegance, Theorem 6.1 unfortunately cannot capture all of our results on implication-free dependency schemes. Part of the reason is that the empty implication relation $R_\emptyset$, which gives rise to $\mathcal{D}^F(R_\emptyset,\lambda) = \mathcal{D}^{\text{rrs}}$, requires special treatment, and part is that the non-parameterised versions of the proof systems (i.e., $\forall\mathsf{Exp+Res}$ and $\mathsf{Q\text{-}Res}$ themselves) cannot be expressed in a parameterised form, and must also be treated separately. We can capture the remaining cases with one extra theorem.

Theorem 6.2. *Let $P \in \{\forall\mathsf{Exp+Res}, \mathsf{Q\text{-}Res}\}$. Then $P <_p P(\mathcal{D}^{\text{rrs}}) <_p P(\mathcal{D}^{\text{tf}})$, and in the case of $\forall\mathsf{Exp+Res}$, the separations are DQBF-genuine.*

The separation between $\mathsf{Q\text{-}Res}$ and $\mathsf{Q\text{-}Res}(\mathcal{D}^{\text{rrs}})$ from Theorem 6.2 is known [5], the rest of Theorem 6.2 was the subject of the conference version of this paper [9] (and is also contained below in this version). The new content in this version is the generalisation of $\mathcal{D}^{\text{tf}}$ to $\mathcal{D}^F(R,\lambda)$ and, correspondingly, Theorem 6.1. A visual summary of both theorems is provided in Figure 1.

Before we dive into the technical details, let us give a high-level overview of the proof.

(1) The first ingredient is *monotonicity* of both $\forall\mathsf{Exp+Res}$ and $\mathsf{Q\text{-}Res}$, meaning that if $\Psi \leq \Psi'$, then any proof of Ψ' in either $\forall\mathsf{Exp+Res}$ or $\mathsf{Q\text{-}Res}$ can be transformed into a proof of Ψ in the same proof system in polynomial time. In other words, deleting some dependencies only makes a formula easier. For $\mathsf{Q\text{-}Res}$ the proof in fact does not even need to be changed, and for $\forall\mathsf{Exp+Res}$ one just needs to restrict annotations as necessary.
(2) Monotonicity together with an ordering of the dependency schemes (Lemma 5.9) gives us all the simulations.
(3) For the separations, we define a class of formula families parameterised by a propositional formula $\mathcal{W}$ and an integer λ. We show that if both parameters of an implication-free dependency scheme are sufficiently strong, the dependency scheme removes all dependencies, otherwise it leaves them intact.
(4) Finally, we show that our parameterised class is hard with dependencies in, and easy with dependencies removed. A separation follows, and we further demonstrate that for $\forall\mathsf{Exp+Res}$ it can be made DQBF-genuine.

The rest of the section is organised as follows. We first define the proof systems $\forall\mathsf{Exp+Res}$ and $\mathsf{Q\text{-}Res}$, and then proceed proving the separations in Theorems 6.1 and 6.2, starting with the latter.

6.1 $\forall\mathsf{Exp+Res}$ and $\mathsf{Q\text{-}Res}$

Among the first DQBF proof systems introduced, the expansion based system $\forall\mathsf{Exp+Res}$ [7, 26] is arguably the most natural.

We first recall the propositional resolution proof system [36]. A *resolution refutation* of a CNF ψ is a sequence $C_1, \ldots, C_k$ 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;

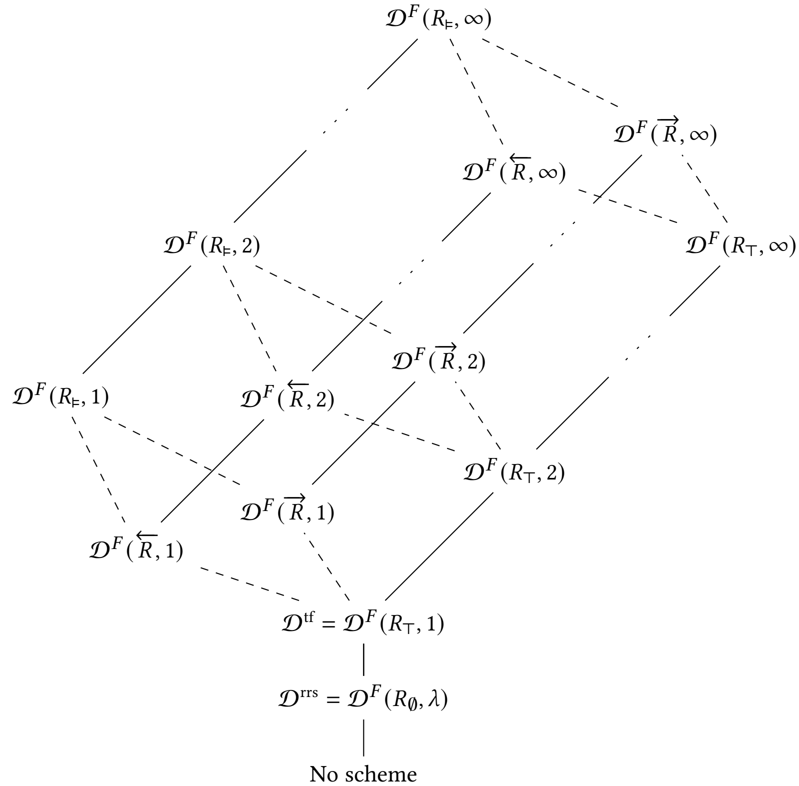

Fig. 1. A visualization of the proof complexity results from Theorems 6.1 and 6.2. The upper lattice shows Theorem 6.1, the bottom chain corresponds to Theorem 6.2. All types of lines and downward paths indicate strict p-simulations—lack of downward connection indicates incomparability. The line style only serves to highlight the presence of the two dimensions; the dots on solid lines indicate that there is an infinite omitted sequence of $\lambda = 3, \ldots,$. The dashed-line lattices with a fixed λ and varying implication relation are in general much larger, containing every non-empty normal implication relation; here we simplified them for the purpose of visual presentation assuming incomparable normal implication relations $\overleftarrow{R}$ and $\overrightarrow{R}$.

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

∀Exp+Res 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 $\exp(\Psi)$ is satisfiable. This is exactly how ∀Exp+Res works. The input DQBF is first expanded, and then refuted in resolution.

Definition 6.3 (∀Exp+Res *[7, 26]).* A ∀Exp+Res refutation of a DQBF Ψ is a resolution refutation of the matrix of $\exp(\Psi)$.

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

Given that fully exhibited dependency schemes like $\mathcal{D}^F(R, \lambda)$ (Theorem 5.10) can be incorporated into an arbitrary DQBF proof system P (Proposition 4.2), we obtain a class of DQBF proof systems of the form ∀Exp+Res($\mathcal{D}^{\mathsf{F}}(\mathsf{R}, \lambda)$).

Definition 6.4 (Q-Res($\mathcal{D}$)*[27, 40]).* A Q-Res *refutation* of a DQBF Ψ is a sequence $C_1, \ldots, C_k$ 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 Ψ;

R *Resolution:* $C_i = A \vee B$, where $C_r = A \vee x$ and $C_s = B \vee \overline{x}$, 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 $\text{var}(a) = u \in \text{vars}_\forall(\Psi)$ and $(u, x) \notin \text{deps}(\Psi)$ for each $x \in \text{vars}(C_i)$.

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

Q-Res($\mathcal{D}^{\mathsf{F}}(\mathsf{R}, \lambda)$) is complete for QBF because it generalizes Q-Res (which itself is sound and complete by [27]), and soundness follows by full exhibition as in the proof of Proposition 4.2.

6.2 Establishing Lower Bounds

In order to simplify the notation in some of the lemmas to come, we will write, for a proof system P and a formula Ψ, $h_P(\Psi)$ (for *hardness of* Ψ *in* P) to denote the length of the shortest P-proof of the formula Ψ. For a DQBF Ψ, we write $\delta(\Psi)$ for the DQBF that results from Ψ by removing all dependencies, i.e. $\delta(\Psi)$ is uniquely defined by the properties $\delta(\Psi) \leq \Psi$ and $\text{vars}_\exists(\delta(\Psi)) = I_\exists(\delta(\Psi))$.

Our separation results are based on various families of formulas derived by modifying the equality QBFs [8], defined below.

Definition 6.5 (EQ_n *(adapted from [8])).* $\mathrm{EQ}_n := \Pi_n^{\mathrm{EQ}} \cdot \psi_n^{\mathrm{EQ}}$, where

$$\Pi_n^{\mathrm{EQ}} := \exists x_1 \cdots \exists x_n \forall u_1 \cdots \forall u_n \, \exists z_1 \cdots \exists z_n \,,$$

$$\psi_n^{\mathrm{EQ}} := (\overline{z_1} \vee \cdots \vee \overline{z_n}) \wedge \bigwedge_{i=1}^{n} \Big((\overline{x_i} \vee \overline{u_i} \vee z_i) \wedge (x_i \vee u_i \vee z_i) \Big) \,.$$

Lemma 6.6. *ψ_n^{EQ} is propositionally satisfiable ($x_i \mapsto 0, u_i \mapsto 1, z_i \mapsto 0$).*

It is known that EQ_n is hard for both ∀Exp+Res and Q-Res.

Lemma 6.7 ([5, 8]). *For $P \in \{$∀Exp+Res, Q-Res$\}$, $h_P(\mathrm{EQ}_n) \geq 2^n$.*

While EQ_n would be a sufficient building stone for classical separations, in order to obtain DQBF-genuine separations we will additionally need a modified version, which we mark with a $\downarrow$ to evoke that the dependency sets shrink.

Definition 6.8 ($\mathrm{EQ}_n^{\downarrow}$ *(adapted from [8])).* $\mathrm{EQ}_n^{\downarrow} := \Pi_n^{\mathrm{EQ}\downarrow} \cdot \psi_n^{\mathrm{EQ}}$, where

$$\Pi_n^{\mathrm{EQ}\downarrow} := \forall u_1 \cdots \forall u_n \exists x_1(\emptyset) \cdots \exists x_n(\emptyset) \, \exists z_1(u_1) \cdots \exists z_n(u_n).$$

We may now begin proving Theorem 6.2, first establishing a DQBF-genuine separation between ∀Exp+Res and ∀Exp+Res($\mathcal{D}^{\mathrm{rrs}}$).

Since the dependency sets of $\mathrm{EQ}_n^{\downarrow}$ 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 ∀Exp+Res [5] does not suffice for $\mathrm{EQ}_n^{\downarrow}$. Nonetheless, a similar argument works, based on the fact that no small subset of clauses in the expansion is unsatisfiable.

Theorem 6.9. $h_{\forall\mathsf{Exp+Res}}(\mathrm{EQ}_n^{\downarrow}) \geq 2^n$.

Proof. The total expansion of $\mathrm{EQ}_n^{\downarrow}$ is the CNF $\psi \wedge \bigwedge_{i=1}^{n} \left((\overline{x_i} \vee z_i^{\overline{u_i}}) \wedge (x_i \vee z_i^{u_i}) \right)$, where ψ is the conjunction of all clauses of the form $(\overline{z_1^{a_1}} \vee \cdots \vee \overline{z_n^{a_n}})$ with $\mathrm{var}(a_i) = u_i$. We show that removing any of the 2^n clauses from ψ makes the total expansion satisfiable. It follows that any resolution refutation of $\exp(\mathrm{EQ}_n^{\downarrow})$ must have 2^n axiom clauses.

Suppose that some clause A is absent from ψ, and let us assume without loss of generality that $A := (\overline{z_1^{u_1}} \vee \cdots \vee \overline{z_n^{u_n}})$, 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 $z_i^{\overline{u_i}} \mapsto 0$ satisfies each clause $(\overline{x_i} \vee z_i^{\overline{u_i}})$. Finally, assigning each $x_i \mapsto 1$ satisfies each clause $(x_i \vee z_i^{u_i})$. □

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

Proposition 6.10 ([6]). $\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_n) = \mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_n^{\downarrow}) = \delta(\mathrm{EQ}_n^{\downarrow}) = \delta(\mathrm{EQ}_n)$.

Proof. It is sufficient to prove $\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_n) = \delta(\mathrm{EQ}_n)$, i.e that $\mathcal{D}^{\mathrm{rrs}}$ removes all dependencies from EQ_n; the rest follows from monotonicity of $\mathcal{D}^{\mathrm{rrs}}$. Aiming for contradiction, suppose that there exists a sequence of clauses $C_1, \ldots, C_k$ and a sequence of literals $p_1, \ldots, p_{k-1}$ satisfying conditions (a) to (d) of Definition 5.3 with respect to some pair $(u_i, z_{i'}) \in \mathrm{deps}(\mathrm{EQ}_n)$. Obviously $k \geq 2$; $k = 2$ is impossible due to condition (c), so $k \geq 3$. For $2 \leq j < k$, $p_j, \overline{p_{j-1}} \in C_j$ (c) and $\mathrm{var}(p_j) \neq \mathrm{var}(p_{j-1})$ (d), whence it follows that $C_j = \{\overline{z_1} \vee \cdots \vee \overline{z_n}\}$; but then $k = 3$ by condition (c). By (a) we must have $p_1 = z_i$ and $p_{k-1} = p_2 = \overline{z_i}$, which contradicts (d). □

Theorem 6.11 ([6]). $h_{\forall\mathsf{Exp+Res}(\mathcal{D}^{\mathrm{rrs}})}(\mathrm{EQ}_n^{\downarrow}) \in O(n)$.

Proof. By Proposition 6.10, the total expansion of $\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_n^{\downarrow})$ is obtained simply by removing the universal literals; that is, the matrix of $\exp(\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_n^{\downarrow}))$ is

$$(\overline{z_1} \vee \cdots \vee \overline{z_n}) \wedge \bigwedge_{i=1}^{n} \left((\overline{x_i} \vee z_i) \wedge (x_i \vee z_i) \right). \tag{1}$$

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

Theorems 6.9 and 6.11 together show that $\forall\mathsf{Exp+Res}$ does not p-simulate $\forall\mathsf{Exp+Res}(\mathcal{D}^{\mathrm{rrs}})$. It remains to show that the lower bound is genuine.

Theorem 6.12. $\forall\mathsf{Exp+Res} \not\leq_p^* \forall\mathsf{Exp+Res}(\mathcal{D}^{\mathrm{rrs}})$.

Proof. It is easy to see that any largest QBF $\Phi_n^{\downarrow}$ that is smaller than $\mathrm{EQ}_n^{\downarrow}$ 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 $\Phi_n^{\downarrow}$ has a linear-size $\forall\mathsf{Exp+Res}$ refutation. Hence, by the monotonicity of $\forall\mathsf{Exp+Res}$, any family of QBFs smaller than $\{\mathrm{EQ}_n^{\downarrow}\}_{n \in \mathbb{N}}$ has linear-size $\forall\mathsf{Exp+Res}$ refutations. Thus, by Theorems 6.9 and 6.11, $\{\mathrm{EQ}_n^{\downarrow}\}_{n \in \mathbb{N}}$ satisfies all the conditions of Definition 4.3.

It remains to show that $\Phi_n^{\downarrow}$ has a linear-size $\forall\mathsf{Exp+Res}$ refutation, or equivalently, that $\exp(\Phi_n^{\downarrow})$ has a linear-size resolution refutation. It is readily verified that $\exp(\Phi_n^{\downarrow})$ contains every clause in $\exp(\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_{n-1}^{\downarrow}))$ except $(\overline{z_1} \vee \cdots \vee \overline{z_{n-1}})$. Figure 2 illustrates that this clause can be derived from $\exp(\Phi_n^{\downarrow})$ in a constant number of resolution steps. Since $\exp(\mathcal{D}^{\mathrm{rrs}}(\mathrm{EQ}_{n-1}^{\downarrow}))$ has a linear-size resolution refutation by Theorem 6.11, so does $\exp(\Phi_n^{\downarrow})$. □

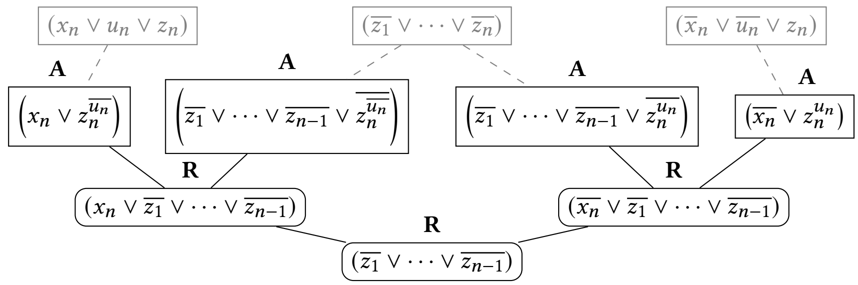

Fig. 2. The prelude to a linear-size ∀Exp+Res refutation of $\Phi_n^{\downarrow}$. In order to reduce $\exp(\Phi_n^{\downarrow})$ to $\exp(\mathcal{D}^{\text{rrs}}(\text{EQ}_{n-1}^{\downarrow}))$, we need only derive the clause $(\overline{z_1} \vee \cdots \vee \overline{z_{n-1}})$.

Now we turn to simulations and separations between the parameterised proof systems ∀Exp+Res$(\mathcal{D}^F(R,\lambda))$, during the course of which we will prove both Theorem 6.1 as well as the remainder of Theorem 6.2, i.e. a (genuine) separation between $P(\mathcal{D}^{\text{rrs}})$ and $P(\mathcal{D}^{\text{tf}})$. We will define a parameterised class of QBF and DQBF families derived from EQ_n and $\text{EQ}_n^{\downarrow}$.

We use the following notation: the *matrix-clause product* of a CNF ψ and a clause C is the CNF $\psi \otimes C := \{D \cup C : D \in \psi\}$. Note that as a special case, if $\psi = \emptyset$ is the empty CNF, then any matrix-clause product $\psi \otimes C = \emptyset$ is also empty.

Definition 6.13. Let $n, \lambda \in \mathbb{N}$, $b_+, b_-, s_1, \ldots, s_\lambda \in \mathbb{V}$, and let $\mathcal{W}$ be a CNF such that the sets $\text{var}(\mathcal{W})$, $\text{var}(\text{EQ}_n)$, and $\{b_+, b_-, s_1, \ldots, s_\lambda\}$ are pairwise disjoint. We define a modification $\text{EQ}_n(\mathcal{W}, \lambda)$ of the equality formulas as follows:

$$\Pi_n^{\text{EQ}} \quad \exists b_+(\emptyset)\, \exists b_-(\emptyset) \quad \exists_{o \in \text{var}(\mathcal{W})} o(\emptyset) \quad \exists s_1(U_n) \cdots s_\lambda(U_n) \cdot$$
$$\left(\{b_+, s_1\} \otimes \psi_n^{EQ}\right) \cup \left(\{b_-, \overline{s_\lambda}\} \otimes \psi_n^{EQ}\right) \cup \left(\{b_+, b_-\} \otimes \mathcal{W}\right) \cup \psi_\lambda^L,$$

where $\psi_\lambda^L := \left\{\{\overline{s_1}, s_2\}, \ldots, \{\overline{s_{\lambda-1}}, s_\lambda\}, \{\overline{b_+}, s_1\}, \{\overline{b_-}, \overline{s_\lambda}\}\right\}$. Furthermore, we define $\text{EQ}_n(\top, \lambda)$ as $\text{EQ}_n(\emptyset, \lambda)$ with the variable b_- removed and occurrences of its literals replaced by substituting $b_- = \overline{b_+}$. We denote by $\psi_n^{\text{EQ}}(\mathcal{W}, \lambda)$ and $\psi_n^{\text{EQ}}(\top, \lambda)$ the matrix of $\text{EQ}_n(\mathcal{W}, \lambda)$ and $\text{EQ}_n(\top, \lambda)$, respectively.

We further define $\text{EQ}_n^{\downarrow}(\mathcal{W}, \lambda)$ as the DQBF

$$\Pi_n^{\text{EQ}\downarrow}\, \exists b_+(\emptyset)\, \exists b_-(\emptyset)\, \exists_{o \in \text{var}(\mathcal{W})} o(\emptyset)\, \exists s_1(U_n) \cdots s_\lambda(U_n) \cdot \psi_n^{\text{EQ}}(\mathcal{W}, \lambda)$$

and $\text{EQ}_n^{\downarrow}(\top, \lambda)$ as $\text{EQ}_n^{\downarrow}(\emptyset, \lambda)$ with the variable b_- removed and occurrences of its literals replaced by substituting $b_- = \overline{b_+}$.

The intuition for these formulas is the following. Resolution paths that certify dependencies between the u_i and z_j must use both copies of ψ_n^{EQ}, collecting the independent-existential clause $\{b_+, b_-\}$ along the way. The CNF $\mathcal{W}$, which will typically be unsatisfiable, ensures that $\{b_+, b_-\}$ is implied—but in order to discover this implication, an implication relation has to have the power to refute $\mathcal{W}$ (i.e. $(\mathcal{W}, \emptyset) \in R$). The sub-formula ψ_λ^L is a gadget that prolongs paths—in order to collect both b_+ and b_- for an implied clause, a high value of λ is required to see a long segment at once. Deficiency in either of the parameters means that the implied clause $\{b_+, b_-\}$ will not be discovered.

We now proceed with the technical details of the proofs.

Since $\mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda)$ and $\mathrm{EQ}_n^{\downarrow}(\top,\lambda)$ can be restricted with the assignment $b_+ \mapsto 0, b_- \mapsto 1, s_i \mapsto 0$ to obtain $\mathrm{EQ}_n^{\downarrow}$, and any ∀Exp+Res or Q-Res proof can also be restricted with the same assignment at no cost in size, the lower bound from Theorem 6.9 lifts to $\mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda)$, and by monotonicity of ∀Exp+Res to $\mathrm{EQ}_n(\mathcal{W},\lambda)$.

Lemma 6.14. *Let $n,\lambda \in \mathbb{N}$, and $\mathcal{W}$ a CNF. For Ψ in $\{\mathrm{EQ}_n(\mathcal{W},\lambda), \mathrm{EQ}_n(\top,\lambda), \mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda), \mathrm{EQ}_n^{\downarrow}(\top,\lambda)\}$, $h_{\forall\mathsf{Exp+Res}}(\Psi) \geq 2^n$.*

The same argument lifts the Q-Res lower bound as well.

Lemma 6.15. *Let $n,\lambda \in \mathbb{N}$, $\mathcal{W}$ a CNF. For $\Phi \in \{\mathrm{EQ}_n(\mathcal{W},\lambda), \mathrm{EQ}_n(\top,\lambda)\}$, $h_{\mathsf{Q\text{-}Res}}(\Phi) \geq 2^n$.*

Lemma 6.16. *For every $n,\lambda \in \mathbb{N}$ and $\mathcal{W}$ a CNF, there is a propositional model of $\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda)$ where $b_+ \mapsto 1, b_- \mapsto 0$, and there is one where $b_+ \mapsto 0, b_- \mapsto 1$. In other words, $\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda) \not\models \{b_+\}, \{b_-\}, \{\overline{b_+}\}, \{\overline{b_-}\}$. On the other hand, $\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda) \models \{\overline{b_+}, \overline{b_-}\}$, and if $\mathcal{W}$ is unsatisfiable, then also $\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda) \models \{b_+, b_-\}$.*

Proof. Concatenate the assignment given by Lemma 6.6 with the appropriate assignment to b_+ and b_-. The implications are immediate. □

Proposition 6.17. *Let R be a non-empty normal implication relation, $(\mathcal{W},\emptyset) \in R$. For all $n,\lambda \in \mathbb{N}$:*

$$\mathcal{D}^F(R,\lambda)\big(\mathrm{EQ}_n(\mathcal{W},\lambda)\big) = \mathcal{D}^F(R,\lambda)\big(\mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda)\big) = \delta(\mathrm{EQ}_n(\mathcal{W},\lambda)),$$
$$\mathcal{D}^F(R_\top,\lambda)\big(\mathrm{EQ}_n(\top,\lambda)\big) = \mathcal{D}^F(R,\lambda)\big(\mathrm{EQ}_n^{\downarrow}(\top,\lambda)\big) = \delta(\mathrm{EQ}_n(\top,\lambda))$$

Proof. Any sequence of clauses that satisfies conditions (a)-(d) must use both copies of EQ_n—otherwise we reach a contradiction in the same way as in the proof of Proposition 6.10 (note that the added literals b_+ and s_1, or b_- and $\overline{s_\lambda}$ in the second copy, do not have any impact on the existence of paths, because they appear in one polarity only). On the other hand, consider a path that goes through both copies of EQ_n. It is easy to see that in order to transition from one copy to the other, other than going directly, the path has to go through the chain of clauses $\{\overline{s_1}, s_2\}, \ldots, \{\overline{s_{\lambda-1}}, s_\lambda\}$. This chain is $\lambda - 1$ long, and so there is a segment of length $\lambda + 1$ that contains both a clause from the first copy as well as a clause from the second copy. Together, these two clauses restricted to $I_\exists(\mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda))$ give the clause $\{b_+, b_-\}$. Because R is normal and $(\mathcal{W},\emptyset) \in R$, also $(\{b_+, b_-\} \otimes \mathcal{W}, \{b_+, b_-\}) \in R$, and thus $(\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda), \{b_+, b_-\}) \in R$, falsifying condition (e) of Definition 5.3. Similarly, in the case of $R_\top$, we collect the tautology $\{b_+, \overline{b_+}\}$ and falsify condition (e). We conclude there are no resolution paths satisfying all conditions (a)-(e), and hence all dependencies are removed. □

Corollary 6.18. *Let $P \in \{\forall\mathsf{Exp+Res}, \mathsf{Q\text{-}Res}\}$, let R be a non-empty normal implication relation, $(\mathcal{W},\emptyset) \in R$. For $\Psi \in \{\mathrm{EQ}_n(\mathcal{W},\lambda), \mathrm{EQ}_n(\top,\lambda), \mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda), \mathrm{EQ}_n^{\downarrow}(\top,\lambda)\}$, $h_{P(\mathcal{D}^F(R,\lambda))}(\Psi) \in O(n+\lambda)$.*

Proof. By Proposition 6.17, $\exp(\mathcal{D}^F(R,\lambda)(\mathrm{EQ}_n(\mathcal{W},\lambda)))$ is obtained simply by deleting all universal literals from $\psi_n^{\mathrm{EQ}}(\mathcal{W},\lambda))$. Using the chain ψ_λ^L we can derive $\exp(\delta(\mathrm{EQ}_n))$ from the two copies in a linear number of steps, and then we proceed like in the proof of Theorem 6.11. The case of $\mathrm{EQ}_n(\top,\lambda)$ is analogous, and the part about the $\mathrm{EQ}_n^{\downarrow}$ versions follows from monotonicity of ∀Exp+Res and Q-Res. □

Proposition 6.19. *Let R be a normal implication relation, $\mathcal{W}$ a CNF, $\lambda, \lambda' \in \mathbb{N}$. For $\Psi \in \{\mathrm{EQ}_n(\mathcal{W},\lambda), \mathrm{EQ}_n^{\downarrow}(\mathcal{W},\lambda)\}$,*

$$(\mathcal{W},\emptyset) \notin R \implies \mathcal{D}^F(R,\infty)\big(\Psi\big) = \Psi,$$
$$\lambda' < \lambda \implies \mathcal{D}^F(R_\models,\lambda')\big(\Psi\big) = \Psi,$$

and for $\Psi' \in \{\mathrm{EQ}_n(\top, \lambda), \mathrm{EQ}_n^{\downarrow}(\top, \lambda)\}$,

$$\mathcal{D}^{\mathrm{rrs}}(\Psi') = \Psi'.$$

Proof. To prove the proposition, we must find sequences of clauses and literals satisfying conditions (a) to (e) of Definition 5.3 with respect to both $(u_i, z_i), (u_i, s_j) \in \mathrm{deps}(\mathrm{EQ}_n^{\downarrow}(\mathcal{W}, \lambda))$, or $(u_i, z_j), (u_i, s_j) \in \mathrm{deps}(\mathrm{EQ}_n(\mathcal{W}, \lambda))$, for each $i \in [n]$ and $j \in [\lambda]$. We will show that for $\mathrm{EQ}_n(\mathcal{W}, \lambda)$ the sequence of clauses

$$P = \{b_+, x_i, u_i, z_i, s_1\}, \{b_+, \overline{z_1}, \ldots, \overline{z_n}, s_1\}, \{b_+, x_j, u_j, z_j, s_1\}, \{\overline{s_1}, s_2\}, \ldots, \{\overline{s_{\lambda-1}}, s_\lambda\}, \{b_-, \overline{x_i}, \overline{u_i}, z_i, \overline{s_\lambda}\},$$

and the sequence of literals $z_i, z_j, s_1, \ldots, s_\lambda$ suffice, and similarly, for $\mathrm{EQ}_n^{\downarrow}(\mathcal{W}, \lambda)$

$$P^{\downarrow} = \{b_+, x_i, u_i, z_i, s_1\}, \{b_+, \overline{z_1}, \ldots, \overline{z_n}, s_1\}, \{\overline{s_1}, s_2\}, \ldots, \{\overline{s_{\lambda-1}}, s_\lambda\}, \{b_-, \overline{x_i}, \overline{u_i}, z_i, \overline{s_\lambda}\}$$

with the literals $z_i, s_1, \ldots, s_\lambda$. It is readily verified that conditions (a) to (d) are satisfied regardless of R, $\mathcal{W}$, and λ, and in particular this serves to show the case with $\mathcal{D}^{\mathrm{rrs}}$ and Ψ' because condition (e) is vacuous there. Hence it remains to verify condition (e) in the other cases. Consider the independent existential literals that appear on the sequences P and $P^{\downarrow}$: b_+ and b_-.

If $\lambda' < \lambda$, on any segment of length $\lambda' + 1$, there is at most one of b_+ and b_-, and by Lemma 6.16 those unit clauses are not implied, hence condition (e) is satisfied for any implication relation R and P detects a dependency.

If $\lambda' \geq \lambda$, we collect the clause $\{b_+, b_-\}$, which is implied as long as $\mathcal{W}$ is unsatisfiable. Though, by normality of R,

$$(\mathcal{W}, \emptyset) \notin R \iff (\{b_+, b_-\} \otimes \mathcal{W} \cup \{\overline{b_+}, \overline{b_-}\}, \{b_+, b_-\}) \notin R \implies (\psi_n^{\mathrm{EQ}}(\mathcal{W}, \lambda), \{b_+, b_-\}) \notin R,$$

where the last implication follows from (the contrapositive of) the restriction property of normality using the satisfying assignment given by Lemma 6.6 extended by setting all s_i to 1. Hence, condition (e) is satisfied. □

Proposition 6.19 and Lemma 6.14 together imply hardness for parameterised versions of our proof systems. For the sake of brevity of the statement, we skip the lower bound for the EQ_n versions in ∀Exp+Res, which follows trivially by monotonicity.

Corollary 6.20. *Let R be a normal implication relation, $(\mathcal{W}, \emptyset) \notin R$, $\lambda \in \mathbb{N}_0$. Then*

$$h_{\forall\mathsf{Exp+Res}(\mathcal{D}^F(R,\infty))}(\mathrm{EQ}_n^{\downarrow}(\mathcal{W}, \lambda)) \geq 2^n;$$
$$h_{\forall\mathsf{Exp+Res}(\mathcal{D}^F(R_{\vDash},\lambda-1))}(\mathrm{EQ}_n^{\downarrow}(\mathcal{W}, \lambda)) \geq 2^n;$$
$$h_{\forall\mathsf{Exp+Res}(\mathcal{D}^{\mathrm{rrs}})}(\mathrm{EQ}_n^{\downarrow}(\top, \lambda)) \geq 2^n;$$

and for $P \in \{\forall\mathsf{Exp+Res}, \mathsf{Q\text{-}Res}\}$,

$$h_{P(\mathcal{D}^F(R,\infty))}(\mathrm{EQ}_n(\mathcal{W}, \lambda)) \geq 2^n;$$
$$h_{P(\mathcal{D}^F(R_{\vDash},\lambda-1))}(\mathrm{EQ}_n(\mathcal{W}, \lambda)) \geq 2^n;$$
$$h_{P(\mathcal{D}^{\mathrm{rrs}})}(\mathrm{EQ}_n(\top, \lambda)) \geq 2^n;$$

We are finally ready to prove Theorem 6.1 and the rest of Theorem 6.2.

Theorem 6.1. The simulations follow from Lemma 5.9 and monotonicity of the proof systems. The separations follow from Corollaries 6.18 and 6.20, for finite λ, as follows. By assumption $R \setminus R'$ contains some $(\mathcal{W}, C)$ for some non-tautological clause C, and by normality then $(\mathcal{W}[\overline{C}], \emptyset) \in R \setminus R'$. Then, we can invoke the two corollaries with $\mathrm{EQ}_n(\mathcal{W}[\overline{C}], \lambda)$ to obtain linear-size proofs in $P(\mathcal{D}^F(R,\lambda))$ and exponential lower bounds in $P(\mathcal{D}^F(R,\lambda'))$ and $P(\mathcal{D}^F(R',\lambda))$. To separate the global dependency schemes from their local counterparts, we can make a diagonalization argument. The family $(\mathrm{EQ}_n(\mathcal{W}, n))_{n\in\mathbb{N}}$ can be seen to have short proofs in $P(\mathcal{D}^F(R,\lambda))$ if, and only if $(\mathcal{W}, \emptyset) \in R$ and $\lambda = \infty$.

Finally, we need to show that the separations for ∀Exp+Res are genuine. That means that as soon as we transform a hard family into a QBF family by shrinking dependency sets, the hardness vanishes. Like in the proof of Theorem 6.12, any largest QBF embedded in $\mathrm{EQ}_n^{\downarrow}(\mathcal{W}, \lambda)$ has only one variable with a non-empty dependency set, other than the s-variables; but also like in that proof, such QBFs can easily be seen to have short ∀Exp+Res proofs even without any dependency scheme. □

6.3 Consequences of Theorem 6.1

One trivial consequence of Theorem 6.1, which is nevertheless worth pointing out explicitly, is that it implies the converse of Lemma 5.9.

Lemma 6.21. *Let R, R' be non-empty normal implication relations and $\lambda, \lambda' \in \mathbb{N}^\infty$. Then $\mathcal{D}^F(R',\lambda')$ is more general than $\mathcal{D}^F(R,\lambda)$ if, and only if, $R \subseteq R'$ and $\lambda \le \lambda'$.*

Proof. One direction is just Lemma 5.9, for the other direction, consider the separating formulas $\mathrm{EQ}_n(\mathcal{W}, \lambda)$ for $(\mathcal{W}, \emptyset) \in R \setminus R'$, where one scheme detects independence that the other misses. □

Together with Proposition 5.8, which says that $\mathcal{D}^{\mathrm{rrs}}$ is less general than any other implication-free scheme, this allows us to restate Theorem 6.1 slightly more compactly as follows. We say that an implication-free dependency scheme is *normal* if its implication relation is.

Theorem 6.22. *Let $P \in \{\forall\mathsf{Exp{+}Res}, \mathsf{Q\text{-}Res}\}$, $\mathcal{D}, \mathcal{D}'$ normal implication-free dependency schemes. Then*

$$P(\mathcal{D}) \quad \le_p \quad P(\mathcal{D}') \iff \mathcal{D} \le \mathcal{D}',$$

and whenever $\mathcal{D} \not\le \mathcal{D}'$, the separation $\forall\mathsf{Exp{+}Res}(\mathcal{D}) \not\le_p^* \forall\mathsf{Exp{+}Res}(\mathcal{D}')$ is DQBF-genuine.

This version, which also incorporates the upper half of the chain from Theorem 6.2, highlights that the parameterisation operator is an injective homomorphism from the partially ordered set of normal implication-free dependency schemes, into the partially ordered set of parameterised versions of P.

Theorem 6.1 speaks about an infinite ordered hierarchy of proof systems even when the implication relation R is fixed. As a potentially interesting application, we will now sketch one way to use the full power of Theorem 6.1 in order to extend this 1-dimensional hierarchy and obtain an infinite 2-dimensional grid hierarchy of DQBF proof systems parameterised by implication-free dependency schemes. The crucial component here are hierarchies of implication relations.

Let $\{R_k\}_{k\in\mathbb{N}}$ be a sequence of normal polynomial-time computable implication relations of increasing size, i.e. $R_k \subsetneq R_{k+1}$. Then, Theorem 6.1 says that

$$P(\mathcal{D}^F(R_k,\lambda)) \quad \le_p \quad P(\mathcal{D}^F(R_{k'},\lambda')) \iff k \le k' \text{ and } \lambda \le \lambda'.$$

One example of such a hierarchy of implication relations can be constructed from *bounded-depth DPLL*. DPLL [19, 20] is a well known algorithm for SAT, which proceeds by branching on indeterminate variables alternated with formula

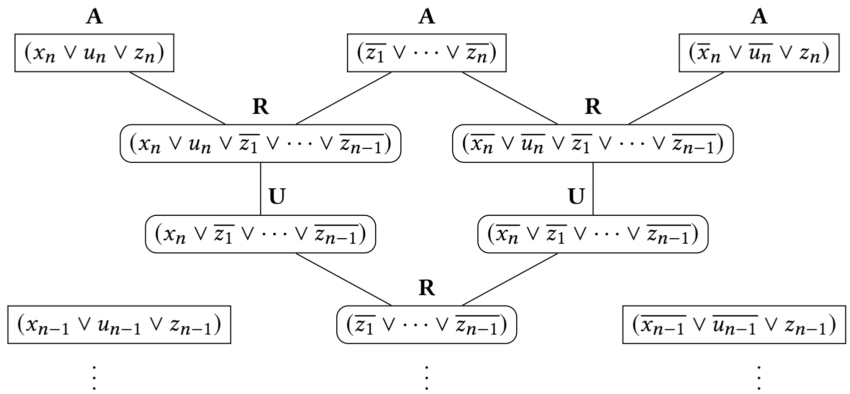

Fig. 3. Linear-size Q-Res refutation of the DQBF $\mathrm{EQ}_n^{\downarrow}$. In a constant number of steps, $\mathrm{EQ}_n^{\downarrow}$ is reduced to $\mathrm{EQ}_{n-1}^{\downarrow}$.

simplification by unit propagation.[4] If unsatisfiability is determined in a given branch, the last decision is reverted. Crucially, this reverted value is no longer considered a decision; it is effectively propagated by a deeper inference that just determined unsatisfiability of the other branch. A bounded-depth version simply requires that the number of active decisions at any time in the algorithm is bounded by a fixed constant k—as soon as that bound is to be exceeded, the algorithm terminates with the answer ‘don’t know’. We can define an implication relation $R_{\mathrm{DPLL}(k)}$ to consist of pairs (ψ, C) for which $\psi[\overline{C}]$ can be proved unsatisfiable by k-bounded-depth DPLL (for at least one choice of decision variables). This generalizes *reverse unit propagation*, which is just 0-bounded-DPLL. It can easily be seen that k-bounded-depth DPLL can be executed in worst-case $O(n^k) \cdot \mathrm{poly}(n)$ time, by going over all possible choices of decision variables and simplifying by unit propagation accordingly. Thus, $R_{\mathrm{DPLL}(k)}$ is indeed a polynomial-time computable implication relation, and it is not too hard to see that all conditions of normality are fulfilled. An example of a formula that is decidable with bounded depth k, but not with depth $k-1$, is the formula which contains all 2^{k+1} full clauses on some set of $k+1$ variables. Thus, the sequence of R_k is indeed strictly increasing, and Theorem 6.1 applies as described.

Until now $\mathcal{D}^{\mathrm{rrs}}$ was the state-of-the-art dependency scheme for Q-Res($\mathcal{D}$); Theorem 6.22 shows that each normal implication-free dependency scheme is stronger, pushing the state of the art in dependency-aware QBF solving.

7 COMPUTATIONAL COMPLEXITY

In view of Theorem 5.10, it is natural to ask why one should even consider less general implication-free schemes. The answer lies in the complexity of computing the dependency scheme. While Theorem 7.1 tells us that local schemes with polynomial-time computable implication relations are themselves polynomial-time computable, Theorems 7.2 and 7.3 provide negative results—both globality and hardness in computing the implication relation translate into hardness for the dependency scheme.

Theorem 7.1. *$\mathcal{D}^F(R, \lambda)$ is polynomial-time computable if R is polynomial-time computable and $\lambda < \infty$.*

[4]We do not consider other techniques, like pure literal elimination, to be part of core DPLL for the discussion here; though they could be added without affecting the gist of the idea.

Proof. Before we proceed with the technical details, let us discuss the intuition. In order to decide the existence of a clause path fulfilling conditions (a)-(e) of Definition 5.3, we must essentially perform a reachability search. However, unlike plain reachability, this search must keep track of the last λ clauses on the path being searched in order to be able to test condition (e). We show that this can be simulated by plain, memoryless reachability by considering a blown-up graph where the vertices are all λ-tuples of clauses, effectively encoding the λ-size memory into a single vertex. The formal proof follows.

Let $\Psi = \Pi \cdot \psi$ be a DQBF. There is only a polynomial number of variable pairs, so it is sufficient that we can test in polynomial time whether a given x depends on a given u. Let $u \in \text{vars}_\forall(\Psi), x \in \text{vars}_\exists(\Psi)$. As a preprocessing step, we go over all candidate paths of length at most λ, which can be done in $O(|\Psi|^\lambda)$ time simply by listing all λ-tuples of clauses. If we find a suitable path, we are done, so for the rest of the proof assume there is no path of length $\leq \lambda$.

Consider the directed graph $G^u_{\Psi,\lambda} = (V^u_{\Psi,\lambda}, E^u_{\Psi,\lambda})$ with the vertex set

$$V^u_{\Psi,\lambda} = \Big\{ \big[(C_1, a_1), \dots, (C_\lambda, a_\lambda)\big] : C_i \in \Psi, a_i \in C_i \text{ for } i \in [\lambda], \\ \overline{a_{i+1}} \in C_i, \text{var}(a_i) \neq \text{var}(a_{i+1}), u \in S_{\text{var}(a_{i+1})} \text{ for } i \in [\lambda - 1] \Big\}$$

and with an edge $\Big[(C_1, a_1), \dots, (C_\lambda, a_\lambda)\Big] \to \Big[(D_1, e_1), \dots, (D_\lambda, e_\lambda)\Big]$ if $(C_i, a_i) = (D_{i-1}, e_{i-1})$, $e_\lambda \in \text{vars}_\exists(\Psi)$, $u \in S_{\text{var}(e_\lambda)}$, and

$$\left(\psi, \bigcup_{i=1}^{\lambda} (C_i \cup D_i) \restriction_{I_\exists(\Psi)}\right) \notin R$$

In other words, the definitions of $V^u_{\Psi,\lambda}$ and $E^u_{\Psi,\lambda}$ are exactly such that for an edge, the sequences $C_1, \dots, C_\lambda, D_\lambda$ and $\overline{a_2}, \dots, \overline{a_\lambda}, \overline{e_\lambda}$ fulfill conditions (a)-(e) of Definition 5.3.

We claim that $(u, x) \in \text{deps}(\Psi)$ if, and only if, there are vertices

$$V = \big[(C_1, a_1), \dots, (C_\lambda, a_\lambda)\big], \ V' = \Big[(C'_1, a'_1), \dots, (C'_\lambda, a'_\lambda)\Big], \ V'' = \Big[(C''_1, a''_1), \dots, (C''_\lambda, a''_\lambda)\Big],$$

allowing $V = V'$ or $V' = V''$, such that $u = a_1, \overline{u} \in C''_\lambda, \exists i \in [\lambda - 1] : \ \text{var}(a'_{i+1}) = x$, and there are paths $V \to V' \to V''$. Indeed, it is easy to verify that the clauses and literals on the concatenated path $V \to V''$ constitute the required sequences from Definition 5.3, and vice versa, any sequences according to Definition 5.3 with more than λ clauses can be translated to a path $V \to V' \to V''$ in $G^u_{\Psi,\lambda}$.

Clearly, $G^u_{\Psi,\lambda}$ can be constructed in $O\left(t_R(|\Psi|)|\Psi|^{2\lambda}\right)$ time, where t_R is the time it takes to check R. Hence we can test all candidates for V, compute all suitable middle points V' reachable from them, and check whether some V'' is reachable from any of them, all in polynomial time. □

Theorem 7.1 gives a recipe for an algorithm that runs in time $\text{poly}(|\Psi|)^\lambda$. If we treat λ as a parameter as in parameterised complexity theory [18, 21], this is called an XP algorithm. A natural question from the point of view of parameterised complexity is whether there is a *fixed-parameter tractable (FPT)* in λ algorithm to compute $\mathcal{D}^F(R, \lambda)$ for a polynomial-time computable R, i.e. one whose running time scales as $f(\lambda) \cdot \text{poly}(|\Psi|)$ for some computable function f. We leave this question for future work.

Setting aside the XP vs FPT question, we can look at the limit case of $\lambda = \infty$. If global implication-free schemes were polynomially computable, we would not have to worry about the local schemes at all. Though, as the next theorem says, computing a global implication-free scheme is in fact NP-hard.

Theorem 7.2. *If R is an implication relation with $R_\top \subseteq R$, then $\mathcal{D}^F(R, \infty)$ is* NP*-hard.*

Proof. By reduction from SAT. Let $\psi = C_1 \wedge \cdots \wedge C_r$ be a CNF. Using the fresh variables $u, z_1, \ldots, z_{r+1} \notin \text{vars}(\psi)$ we define the following (D)QBF:

$$\Psi = \exists \text{vars}(\psi)\ \forall u\ \exists z_1, \ldots, z_{r+1} \cdot \phi,$$

where

$$\phi = (u \vee z_1) \wedge (\overline{u} \vee \overline{z_{r+1}}) \wedge \bigwedge_{i=1}^{r} \bigwedge_{a \in C_i} (a \vee \overline{z_i} \vee z_{i+1}).$$

Since ϕ can be satisfied by the fresh variables only, no non-tautological clause on $\text{vars}(\psi)$ is implied by ϕ. Because $I_\exists(\Psi) = \text{vars}(\psi)$, in condition (e′) applied to Ψ we will only ever test whether $(\phi, C) \in R$ for clauses C on $\text{vars}(\psi)$, which means that $\mathcal{D}^F(R, \infty)(\Psi) = \mathcal{D}^F(R_\top, \infty)(\Psi)$. To complete the proof, we will now show that ψ is satisfiable if and only if $(u, z_i) \in \text{deps}(\mathcal{D}^F(R_\top, \infty)(\Psi))$ for any $i \in [r+1]$.

Let $D_1, \ldots, D_k$ and $p_1, \ldots, p_{k-1}$ be sequences that satisfy conditions (a)-(d) of Definition 5.3 for (u, z_i) for some i. Because $u \in D_1$, we have $D_1 = (u, z_1)$, and so $p_1 = z_1$. Hence $\overline{z_1} \in D_2$, and so D_2 must be one of the clauses $(a \vee \overline{z_1} \vee z_2)$ corresponding to some $a \in C_1$, and so p_2 must be z_2. By induction, we have that D_i is one of the clauses corresponding to C_{i-1} for $1 < i < k$, and $p_i = z_i$ for $1 \le i < k$. Finally $D_k = (\overline{u}, \overline{z_{r+1}})$, and hence $k = r + 2$.

Therefore, $\alpha := (D_1 \cup \cdots \cup D_k)\restriction_{I_\exists(\Psi)}$ contains at least one literal from every clause C_i, and conversely for every set of literals α that hits every clause C_i, we can construct corresponding sequences $D_1, \ldots, D_{r+2}$ and $z_1, \ldots, z_{r+1}$ for which $D_1 \cup \cdots \cup D_{r+2}\restriction_{I_\exists(\Psi)} \subseteq \alpha$. Because α hits every clause of ψ, it is a satisfying assignment of ψ if and only if it is non-tautological. It follows that ψ is satisfiable if and only if condition (e′) of Definition 5.3 can be satisfied on top of (a)-(d), i.e. $(u, z_i) \in \text{deps}(\mathcal{D}^F(R_\top, \infty)(\Psi))$. □

Theorem 7.2 says that globality is a source of hardness. Next, we will show that hardness of a normal implication relation also translates into hardness of any implication-free dependency scheme that uses it, regardless of λ.

Theorem 7.3. *For any normal implication relation R, the co-problem of computing R can be reduced in polynomial time to computing $\mathcal{D}^F(R, 1)$ (or indeed any value of $\lambda \in \mathbb{N}^\infty$).*

Proof. Let R be a normal implication relation, ψ a CNF, C a clause. Using the fresh variables $u, z \notin \text{vars}(\psi) \cup \text{vars}(C)$ we define the following (D)QBF:

$$\Psi = \exists \text{vars}(C)\ \forall u\ \exists(\text{vars}(\psi) \setminus \text{vars}(C))\ \exists z \cdot \phi,$$

where $\phi = \psi \wedge C_1 \wedge C_2$ and $C_1 = (C \vee u \vee z)$, $C_2 = (\overline{u} \vee \overline{z})$. We claim that $(u, z) \in \text{deps}(\mathcal{D}^F(R, 1)(\Psi))$ if and only if $(\psi, C) \notin R$.

If $(u, z) \in \text{deps}(\mathcal{D}^F(R, 1)(\Psi))$, then this can only be due to the sequences C_1, C_2 and z satisfying all conditions (a)-(e) of Definition 5.3. In particular, $(\phi, (C_1 \cup C_2)\restriction_{I_\exists(\Psi)} = C) \notin R$. By the weakening property then also $(\psi, C) \notin R$.

Conversely, if $(\psi, C) \notin R$, then, since $\psi = \phi[\overline{u}, z]$ and $C = C[\overline{u}, z]$, by the restriction property $(\phi, C) \notin R$. Hence, the sequences C_1, C_2 and z satisfy conditions (a)-(e), and $(u, z) \in \text{deps}(\mathcal{D}^F(R, 1)(\Psi))$. □

As a corollary, we obtain NP-hardness for $\mathcal{D}^F(R_\models, \lambda)$, transferred from its coNP-complete normal full implication relation $R_\models$. In fact, since $R_\models$ is in coNP, the following observation additionally gives us NP membership as well.

Observation 1. *For any implication relation R in* coNP, *$\mathcal{D}^F(R_\models, \lambda)$ is in* NP *for any $\lambda \in \mathbb{N}^\infty$.*

Proof. To check that for a pair $(u, x) \in \text{vars}(\Psi)$ of variables $(u, x) \in \text{deps}(\mathcal{D}^F(R_\models, \lambda)(\Psi))$, we can simply guess the right sequences of clauses and literals, and verify that none of the corresponding restricted clause unions are in R with respect to the matrix of Ψ, which can all be done in non-deterministic polynomial time (we are checking that a clause is *not* in R). □

Corollary 7.4. *For any* $\lambda \in \mathbb{N}^{\infty}$, $\mathcal{D}^F(R_{\models}, \lambda)$ *is* NP-*complete.*

8 CONCLUSIONS

We conclude with an interesting observation and a question for future research. The family $\{\mathrm{EQ}_n^{\downarrow}\}_{n\in\mathbb{N}}$ from Definition 6.8 is an adaptation of the equality QBFs $\{\mathrm{EQ}_n\}_{n\in\mathbb{N}}$ from [8], obtained by shrinking the dependency set of each z_i to just $\{u_i\}$. While in QBF $\{\mathrm{EQ}_n\}_{n\in\mathbb{N}}$ requires exponentially long proofs in both ∀Exp+Res and Q-Res [5, 8], in DQBF $\{\mathrm{EQ}_n^{\downarrow}\}_{n\in\mathbb{N}}$ remains hard only for ∀Exp+Res. Indeed, even though Q-Res is incomplete for DQBF, it is sound, and $\{\mathrm{EQ}_n^{\downarrow}\}_{n\in\mathbb{N}}$ has linear-size Q-Res refutations, as shown in Figure 3. This suggests that there may be some hidden proof-complexity relationship between ∀Exp+Res and Q-Res in DQBF, even though Q-Res is incomplete there.

In the conference version of this paper, we introduced only the dependency scheme $\mathcal{D}^{\mathrm{tf}}$, at that time the strongest known dependency scheme. With this paper, we answer the open question posed there—whether dependency schemes even stronger than $\mathcal{D}^{\mathrm{tf}}$ exist—by showing that there is in fact a rich infinite multidimensional world of parameterised dependency schemes that naturally generalize $\mathcal{D}^{\mathrm{tf}}$. Nevertheless, our concluding question in this paper remains the same: do some even stronger dependency schemes for (D)QBF exist?

REFERENCES

[1] Salman Azhar, Gary Peterson, and John Reif. 2001. Lower Bounds for Multiplayer Non-cooperative Games of Incomplete Information. *Journal of Computers and Mathematics with Applications* 41 (2001), 957 – 992.

[2] Valeriy Balabanov and Jie-Hong R. Jiang. 2012. Unified QBF Certification and its Applications. *Formal Methods in System Design* 41, 1 (2012), 45–65.

[3] Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. 2014. QBF Resolution Systems and Their Proof Complexities. In *International Conference on Theory and Practice of Satisfiability Testing (SAT) (Lecture Notes in Computer Science, Vol. 8561)*, Carsten Sinz and Uwe Egly (Eds.). Springer, Vienna, 154–169.

[4] Olaf Beyersdorff and Joshua Blinkhorn. 2016. Dependency Schemes in QBF Calculi: Semantics and Soundness. In *International Conference on Principles and Practice of Constraint Programming (CP) (Lecture Notes in Computer Science, Vol. 9892)*, Michel Rueher (Ed.). Springer, Toulouse, 96–112.

[5] Olaf Beyersdorff and Joshua Blinkhorn. 2018. Genuine Lower Bounds for QBF Expansion. In *International Symposium on Theoretical Aspects of Computer Science (STACS) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 96)*, Rolf Niedermeier and Brigitte Vallée (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Caen, 12:1–12:15.

[6] Olaf Beyersdorff and Joshua Blinkhorn. 2019. Dynamic QBF Dependencies in Reduction and Expansion. *ACM Transactions on Computational Logic* 21, 2 (2019), 1–27.

[7] Olaf Beyersdorff, Joshua Blinkhorn, Leroy Chew, Renate A. Schmidt, and Martin Suda. 2019. Reinterpreting Dependency Schemes: Soundness Meets Incompleteness in DQBF. *Journal of Automated Reasoning* 63, 3 (2019), 597–623.

[8] Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. 2019. Size, Cost, and Capacity: A Semantic Technique for Hard Random QBFs. *Logical Methods in Computer Science* 15, 1 (2019).

[9] Olaf Beyersdorff, Joshua Blinkhorn, and Tomás Peitl. 2020. Strong (D)QBF Dependency Schemes via Tautology-Free Resolution Paths. In *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12178)*, Luca Pulina and Martina Seidl (Eds.). Springer, Alghero, 394–411. https://doi.org/10.1007/978-3-030-51825-7_28

[10] Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. 2016. Lower Bounds: From Circuits to QBF Proof Systems. In *ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, Madhu Sudan (Ed.). ACM, Cambridge, MA, USA, 249–260.

[11] Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. 2019. New Resolution-Based QBF Calculi and Their Proof Complexity. *ACM Transactions on Computation Theory* 11, 4 (2019), 26:1–26:42.

[12] Olaf Beyersdorff, Leroy Chew, Renate A. Schmidt, and Martin Suda. 2016. Lifting QBF Resolution Calculi to DQBF, See [17], 490–499.

[13] Olaf Beyersdorff, Luke Hinde, and Ján Pich. 2020. Reasons for Hardness in QBF Proof Systems. *ACM Transactions on Computation Theory* 12, 2 (2020), 10:1–10:27.

[14] Joshua Blinkhorn and Olaf Beyersdorff. 2017. Shortening QBF Proofs with Dependency Schemes. In *International Conference on Theory and Practice of Satisfiability Testing (SAT) (Lecture Notes in Computer Science, Vol. 10491)*, Serge Gaspers and Toby Walsh (Eds.). Springer, Melbourne, 263–280.

[15] Hubie Chen. 2017. Proof Complexity Modulo the Polynomial Hierarchy: Understanding Alternation as a Source of Hardness. *ACM Transactions on Computation Theory* 9, 3 (2017), 15:1–15:20.

[16] Stephen A. Cook and Robert A. Reckhow. 1979. The Relative Efficiency of Propositional Proof Systems. *Journal of Symbolic Logic* 44, 1 (1979), 36–50.

[17] Nadia Creignou and Daniel Le Berre (Eds.). 2016. *International Conference on Theory and Practice of Satisfiability Testing (SAT)*. Lecture Notes in Computer Science, Vol. 9710. Springer.

[18] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer, Cham. https://doi.org/10.1007/978-3-319-21275-3

[19] Martin Davis, George Logemann, and Donald W. Loveland. 1962. A machine program for theorem-proving. *Commun. ACM* 5, 7 (1962), 394–397.

[20] Martin Davis and Hilary Putnam. 1960. A Computing Procedure for Quantification Theory. *Journal of the ACM* 7, 3 (1960), 201–215.

[21] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer, Cham.

[22] Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. 2017. Conformant planning as a case study of incremental QBF solving. *Annals of Mathematics and Artificial Intelligence* 80, 1 (2017), 21–45.

[23] L. Henkin. 1961. Some Remarks on Infinitely Long Formulas. *Infinitistic Methods* (1961), 167 – 183.

[24] John N. Hooker (Ed.). 2018. *International Conference on Principles and Practice of Constraint Programming (CP)*. Lecture Notes in Computer Science, Vol. 11008. Springer.

[25] Holger H. Hoos, Tomáš Peitl, Friedrich Slivovsky, and Stefan Szeider. 2018. ,Portfolio-Based Algorithm Selection for Circuit QBFs, See [24], 195–209.

[26] Mikoláš Janota and João Marques-Silva. 2015. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science* 577 (2015), 25–42.

[27] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. 1995. Resolution for Quantified Boolean Formulas. *Information and Computation* 117, 1 (1995), 12–18.

[28] Roman Kontchakov, Luca Pulina, Ulrike Sattler, Thomas Schneider, Petra Selmer, Frank Wolter, and Michael Zakharyaschev. 2009. Minimal Module Extraction from DL-Lite Ontologies Using QBF Solvers. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Craig Boutilier (Ed.). AAAI Press, Pasadena, 836–841.

[29] Florian Lonsing and Uwe Egly. 2017. DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL. In *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Proceedings (Lecture Notes in Computer Science, Vol. 10395)*, Leonardo de Moura (Ed.). Springer, Gothenburg, 371–384.

[30] Florian Lonsing and Uwe Egly. 2018. Evaluating QBF Solvers: Quantifier Alternations Matter, See [24], 276–294.

[31] Hratch Mangassarian, Andreas G. Veneris, and Marco Benedetti. 2010. Robust QBF Encodings for Sequential Circuits with Applications to Verification, Debug, and Test. *IEEE Trans. Comput.* 59, 7 (2010), 981–994.

[32] Tomáš Peitl, Friedrich Slivovsky, and Stefan Szeider. 2019. Combining Resolution-Path Dependencies with Dependency Learning. In *International Conference on Theory and Practice of Satisfiability Testing (SAT)*. Springer, Lisbon, 306–318.

[33] Tomáš Peitl, Friedrich Slivovsky, and Stefan Szeider. 2019. Dependency Learning for QBF. *Journal of Artificial Intelligence Research* 65 (2019), 181–208.

[34] Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. 2019. Long-Distance Q-Resolution with Dependency Schemes. *Journal of Automated Reasoning* 63, 1 (2019), 127–155.

[35] Luca Pulina and Martina Seidl. 2019. The 2016 and 2017 QBF solvers evaluations (QBFEVAL'16 and QBFEVAL'17). *Artif. Intell.* 274 (2019), 224–248.

[36] John Alan Robinson. 1965. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM* 12, 1 (1965), 23–41.

[37] Marko Samer and Stefan Szeider. 2009. Backdoor Sets of Quantified Boolean Formulas. *Journal of Automated Reasoning* 42, 1 (2009), 77–97.

[38] Christoph Scholl and Ralf Wimmer. 2018. Dependency Quantified Boolean Formulas: An Overview of Solution Methods and Applications - Extended Abstract. In *International Conference on Theory and Practice of Satisfiability Testing (SAT) (Lecture Notes in Computer Science, Vol. 10929)*, Olaf Beyersdorff and Christoph M. Wintersteiger (Eds.). Springer, Oxford, 3–16.

[39] Friedrich Slivovsky. 2015. *Structure in #SAT and QBF*. Ph.D. Dissertation. Vienna University of Technology.

[40] Friedrich Slivovsky and Stefan Szeider. 2016. Soundness of Q-resolution with dependency schemes. *Theoretical Computer Science* 612 (2016), 83–101.

[41] Moshe Y. Vardi. 2014. Boolean Satisfiability: Theory and Engineering. *Commun. ACM* 57, 3 (2014), 5.

[42] Ralf Wimmer, Christoph Scholl, Karina Wimmer, and Bernd Becker. 2016. Dependency Schemes for DQBF, See [17], 473–489.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il