Electronic Colloquium on Computational Complexity, Report No. 112 (2020)

Simulating DQBF Preprocessing Techniques with Resolution Asymmetric Tautologies

Joshua Blinkhorn

Friedrich-Schiller-Universität, Jena, Germany

Abstract. Dependency quantified Boolean formulas (DQBF) describe an NEXPTIME-complete generalisation of QBF, which in turn generalises SAT. DQBF solving is an emerging field with further applications in, among others, incomplete circuit design.
QRAT is a recently proposed proof system for quantified Boolean formulas (QBF), which simulates the full suite of QBF preprocessing techniques and thus forms a uniform proof checking format for solver verification.
In this work, we study QRAT in the more general DQBF context, obtaining a sound and complete refutational DQBF proof system that we call DQRAT. We show that DQRAT can simulate the full suite of dedicated DQBF preprocessing techniques, except those relying on defined variables, which we cover with the introduction of a new form of prefix modification. Our work enables generalisations of further QBF preprocessing techniques (e.g. blocked literal elimination) that were not previously considered for DQBF.

Keywords: DQBF · QBF · QRAT · solving · preprocessing · proof systems

1 Introduction

Dependency quantified Boolean formulas (DQBF) [50] enrich propositional logic with Henkin quantifiers [32], offering natural encodings of problems such as partial equivalence checking [30, 26] and synthesis of safe controllers [17]. The last few years has seen a surge of interest in practical DQBF solving [28, 31, 61] and its accompanying theory of associated proof systems [4, 6, 8]. Indeed, as of 2018 the annual QBF evaluation competition runs a dedicated DQBF track [51].

Formally a DQBF is a logical expression of the form

$$\forall u_1 \cdots \forall u_m \exists x_1(S_1) \cdots \exists x_n(S_n) \cdot \psi\,,$$

where ψ is a propositional formula and each *dependency set* S_i is a subset of the universally quantified variables $\{u_1, \dots, u_m\}$. The intended meaning is as follows: for all Boolean values for the u_i, there exist Boolean values for the x_i, each depending only S_i, such that ψ is satisfied. A traditional quantified Boolean formula (QBF) is obtained when the dependency sets are linearly ordered with respect to set inclusion, i.e. $S_1 \subseteq \cdots \subseteq S_n$. A SAT instance is obtained when the set of universal variables is empty.

Hence SAT is a strict subset of QBF, which in turn is a strict subset of DQBF. Moreover, the decision problems for these three logical formalisms form canonical complete languages for the complexity classes NP [21], PSPACE [60], and NEXPTIME [3], respectively.

An automated decision procedure for an NEXPTIME-hard language could scarcely have been conceivable prior to the breakthroughs with SAT technologies [63] powered by conflict-driven clause learning [58]. At the present time, however, the applicability of SAT-based techniques to further logics is a thriving research area [12, 13], whereby (D)QBF serves as a yardstick measuring progress in the algorithmic solution of increasingly harder problems.

The QBF landscape. Practitioners have been developing QBF solvers for a quarter of a century, and a host of competitive implementations (CAQE [52], DepQBF [46], RAReQs [39], Qute [49],

ISSN 1433-8092

and so on) have steadily raised the performance bar. Breakthroughs for QBF technologies, like those reported for SAT, are yet to be seen; nonetheless, a recent study showed that the QBF-based workflow outperforms SAT on particular bounded synthesis problems [25].

Empirical studies report improved performance when the standard QBF preprocessor bloqqer [15] is enabled. *Preprocessing* [55] involves the application of various satisfiability-preserving transformations with the ultimate goal of reducing the size of the problem encoding. QBF preprocessors empoy techniques such as blocked clause elimination [41, 15], blocked literal elimination [37], and universal expansion [2, 14].

On the theory side, there exist a host of QBF *proof systems* [20], several of which 'underpin' solving techniques in the sense that the solver trace can be interpreted as a proof [16]. The relative proof complexities of these systems is very well understood [5, 10]. Moreover, interpreting the trace as a proof allows the result of the solver to be verified.[1]

Consequently, arguably the most practically relevant QBF proof system is the general proof checking format QRAT [38]. QRAT simulates the full suite of QBF preprocessing techniques [38, 18], and aims at verifying the complete workflow independently of the chosen solver. Since its initial proposal [37], QRAT has stirred significant interest. It is known to simulate various other proof systems [44, 43], support the efficient extraction of Skolem functions [36] (but not Herbrand functions [19]), and forms the basis for further techniques in the preprocessor QRATPre+ [47, 48].

The DQBF landscape. Research into DQBF solving began in 2012 [27]. There are three leading solvers: iDQ [28], a solver based on first-order instantiation; HQS [31], which attempts to reduce the problem to an equisatisfiable QBF; and dCAQE [61], which generalises the clausal abstraction paradigm [52]

The developers of HQS also maintain the only publicly available DQBF preprocessor HQSPre [65], which implements generalised QBF techniques along with some DQBF-specific procedures. Preprocessing with HQSPre is becoming standard (e.g. [61]), and experiments demonstrate improved performance when preprocessing is enabled [65]. Nonetheless, DQBF solving is at a very early stage, and consists largely of SAT and QBF techniques, generalised with varying degrees of success. The desire for further preprocessing techniques has been acknowledged [57].

On the theory side, there have been mixed results. From the plethora of QBF proof systems, only two (the expansion-based systems $\forall$Exp+Res [40] and IR-calc [10]) have been successfully lifted to DQBF [6]. We now know that the subtleties of DQBF semantics, markedly different from QBF, give rise to unsoundness [4] and incompleteness [11] issues for several standard QBF proof systems in the DQBF setting [6, 16]. No further DQBF proof systems have been proposed.[2] Moreover, the practical relevance of the existing systems is unclear,[3] and there is no general proof checking format covering the preprocessor HQSPre.

1.1 Our contributions

One QBF proof system which has not been studied in the DQBF context is QRAT. This is the task to which we take in this paper, and we are able to report some positive results. Our contributions fall into three categories.

A new DQBF proof system. We show that the refutational QRAT proof system does indeed lift naturally to DQBF. The resulting system, which we call DQRAT, is only the third sound and complete refutational DQBF proof system proposed to date. To show that DQRAT

[1] Verifying solvers by means of a unified proof format is already standard practice in the SAT community [64].
[2] Fork Resolution [53] has been shown to be incomplete for DQBF [56].
[3] There appears to be a connection between iDQ and IR-calc, but this is yet to be formally established.

is sound, we prove two master theorems (Theorems 5 and 7) that generalise the correctness of clause addition and literal elimination under certain conditions [38]. In turn, this necessitates that we introduce non-trivial DQBF versions of the notion of 'outer variables' (Definitions 1 and 2). We show completeness via a polynomial-time simulation of $\forall$Exp+Res, generalising the known QBF simulation [44].

Simuating existing DQBF preprocessing techniques. We demonstrate how DQRAT (or more precisely, the underlying set of transformations that we call $\mathcal{R}$) is able to simulate the full suite of HQSPre preprocessing techniques, except those relying on *defined variables.* Using our master theorems, our simulations double as correctness proofs for individual preprocessing techniques. In several cases, this greatly simplifies the argument compared to its original direct proof (cf. the appendix of [66]).

In order to cover the remaining techniques, we introduce *advanced prefix modification* (APM), whose addition to $\mathcal{R}$ is sufficient to simulate the full suite with no exceptions. Correctness of APM is based on a new result (Theorem 30) showing how the existence of a variable definition may allow a tightening of the quantifier prefix.

Enabling further DQBF preprocessing techniques. Finally, we show that blocked literal elimination (BLE) and addition (BLA) are applicable to DQBFs. These techniques, currently absent from HQSPre, were previously considered only in the QBF context [37]. As a corollary to one of our master theorems (Theorem 7), we prove that both techniques are correct in the DQBF setting. Since the QBF preprocessor bloqqer benefits from BLE [37], it may be a welcome addition to HQSPre.

Organisation of the paper. Background on DQBF is covered in Section 2. In Sections 3 and 4, we consider 'outer variables' and 'quantified implied outer resolvents' in the DQBF context, culminating in the proof of our master theorems. In Section 5 we define $\mathcal{R}$, a set of truth-preserving DQBF transformations which functions as the basis for the proof system DQRAT, covered in Section 6. Simulation of preprocessing techniques is covered in Section 7. Finally, in Section 8, we offer some concluding thoughts and highlight some related open problems.

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 $\mathbb{U}$. 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 $C\lceil Z := \{a \in C : \mathrm{var}(a) \in Z\}$. The *size* of clause $|C|$ is its set cardinality, and the *size* of a CNF $|\psi|$ is the sum of the sizes of its clauses.

A *dependency quantified Boolean formula* (DQBF) is a sentence of the form $\Psi := \Pi \cdot \psi$, where $\Pi := \forall u_1 \cdots \forall u_m \exists x_1(S^{\Pi}_{x_1}) \cdots \exists x_n(S^{\Pi}_{x_n})$ 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* $S^{\Pi}_{x_i}$, which is a subset of the universal variables $\{u_1, \dots, u_m\}$. With $\mathrm{vars}_\forall(\Pi)$ and $\mathrm{vars}_\exists(\Pi)$ we denote the universal and existential variables appearing in Π, and with $\mathrm{vars}(\Pi)$ their union. We also put $\mathrm{vars}_\forall(\Psi) := \mathrm{vars}_\forall(\Pi)$, $\mathrm{vars}_\exists(\Psi) := \mathrm{vars}_\exists(\Pi)$, and $\mathrm{vars}(\Psi) := \mathrm{vars}(\Pi)$. We assume that a DQBF is *closed* by definition, that is $\mathrm{vars}(\psi) \subseteq \mathrm{vars}(\Pi)$. The *size* of a DQBF is that of its matrix, i.e. $|\Psi| = |\psi|$.

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 of quantified variable sets in the conventional way (see e.g. [1]).

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 $\alpha\lceil Z'$. 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 $\psi[\alpha] = 0$.

A *model* for a DQBF $\varPsi := \varPi \cdot \psi$ is a set of functions $f := \{f_x : x \in \mathrm{vars}_\exists(\varPsi)\}$, $f_x : \langle S_x\rangle \to \langle\{x\}\rangle$, for which, for each $\alpha \in \langle \mathrm{vars}_\forall(\varPsi)\rangle$, the combined assignment $\alpha \cup \{f_x(\alpha\lceil S_x) : x \in \mathrm{vars}_\exists(\varPsi)\}$ 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 $\varPsi \equiv_{\mathrm{tr}} \varPhi$.

3 Outer Variables in the DQBF Setting

Recall that a QBF is a DQBF whose dependency sets can be linearly ordered with respect to set inclusion. For example, consider a DQBF $\varPsi := \varPi \cdot \psi$ with prefix

$$\varPi := \forall u_1 \forall u_2 \, \exists x_1(\emptyset) \, \exists x_2(\{u_1\}) \, .$$

Since the dependency sets of $\varPi$ can indeed be linearly ordered, as in $S^{\varPi}_{x_1} = \emptyset \subseteq S^{\varPi}_{x_2} = \{u_1\}$, $\varPsi$ is in fact a QBF. Such QBF prefixes can be written in a simpler form in which their dependency sets need not be explicitly stated; for example, $\varPi$ would typically be written

$$\varPi := \exists x_1 \forall u_1 \exists x_2 \forall u_2 \, ,$$

where the dependency set for x_i can be recovered by collecting the preceding universal variables. In general, a QBF prefix takes the form $\exists X_1 \forall U_1 \cdots \exists X_k \forall U_k$, where the X_i, U_i are pairwise disjoint sets (also called *blocks*) of Boolean variables. The order of quantification within a block need not be fixed, insofar as all such orders produce logically equivalent QBFs.

Writing QBF prefixes in this way is not merely a syntactic convenience – it is an expression of an ordered structure of variable dependencies, upon which the success of all dedicated QBF techniques relies. Moreover, it gives rise to a natural notion of *outer variables*, which is frequently used in proofs of QBF-specific results. Formally, the variables 'outer' to a given variable are those quantified in *the same or preceding blocks*.[4] For example, the variables outer to $u \in U_i$ are those of the set $\bigcup_{j=1}^{i} X_i \cup \bigcup_{j=1}^{i} U_i$.

DQBF prefixes in general, however, do not possess this linear structure. The dependency sets are only partially ordered with respect to set inclusion, whereby the notion of outer variables appears to break down.

Presently we propose a DQBF generalisation of outer variables, which will play a key role throughout. In a departure from QBF, we use separate definitions for universal and existential variables, but both versions reduce to the familiar notion on the QBF fragment. Our definitions are motivated in part by existing work, namely the DQBF generalisation of QBF preprocessing techniques in [65].

The existential version. In a QBF, the variables outer to an existential are exactly those universals in its dependency set, together with those existentials whose dependency sets are no larger (again, with respect to set inclusion). Viewed this way, the following DQBF generalisation is quite natural.

[4] Many publications define 'outer' in terms of a relation $\leq_{\varPi}$ over the variables.

Definition 1 (outer variables, $\exists$-version). *Given a prefix Π and an existential $y \in \mathrm{vars}_\exists(\Pi)$, we define the* outer variables

$$\mathsf{OV}(\Pi, y) := S_y^\Pi \cup \{x \in \mathrm{vars}_\exists(\Pi) : S_x^\Pi \subseteq S_y^\Pi\}\,.$$

The universal version. A natural analogue for universal variables is not so easy to conceive, for the simple reason that universal variables do not have dependency sets. As a result, the universal version is a more complicated affair.

Definition 2 (outer variables, $\forall$-version). *Given a prefix Π and a universal $u \in \mathrm{vars}_\forall(\Pi)$, the set of existential variables* dependent on u *is*

$$D_u^\Pi := \{x \in \mathrm{vars}_\exists(\Pi) : u \in S_x^\Pi\}\,,$$

the set of existential variables independent of u *is its complement*

$$I_u^\Pi := \{x \in \mathrm{vars}_\exists(\Pi) : u \notin S_x^\Pi\}\,,$$

and the kernel *of u is*

$$K_u^\Pi := \bigcap_{x \in D_u^\Pi} S_x^\Pi\,.$$

We define the outer variables

$$\mathsf{OV}(\Pi, u) := K_u^\Pi \cup \{x \in I_u^\Pi : S_x^\Pi \subseteq K_u^\Pi\}\,.$$

The kernel K_u^Π operates rather like the dependency set S_y^Π in the existential version, in the sense that the chosen existential variables have dependency sets no larger than K_u^Π. It is readily verified that $\mathsf{OV}(\Pi, u)$ specifies the familiar notion on QBF, but this would not hold if existentials outside of I_u^Π were included. Note that u always belongs to its kernel.

4 Quantified Implied Outer Resolvents for DQBF

Our goal in this section is to prove two theorems (Theorems 5 and 7), constituting DQBF generalisations of known QBF results [38, Thms. 7, 8]. The theorems express the fact that *clause set modification* (i.e. clause addition/elimination) and *clause modification* (i.e. literal addition/elimination) are truth-value preserving, under particular semantic conditions.

Once again, we require separate existential and universal treatments. Common to both is the definition of *outer clause*,[5] which rests on the definition of outer variables (Section 3).

Definition 3 (outer clause). *Given a clause D under a prefix Π, and a literal $p \in D$, we define the* outer clause

$$\mathsf{OC}(\Pi, D, p) := \big\{a \in D : \mathrm{var}(a) \in \mathsf{OV}\big(\Pi, \mathrm{var}(p)\big)\big\}\,.$$

[5] Here, p belongs to the outer clause, in contrast to the original definition (cf. [38]). Whereas the existential literal $p = \overline{y}$ is inevitably removed (Definition 4), the univeral literal $p = \overline{u}$ need not be (Definition 6).

The existential treatment. Now we define the natural DQBF generalisation of the property known as *quantified implied outer resolvents* (QIOR, [38]). First we consider the case where the 'witnessing literal' is existential. We say that a clause C is *under* a prefix Π when $\text{vars}(C) \subseteq \text{vars}(\Pi)$.

Definition 4 ($\mathsf{DQIOR}_\exists$). *A clause C under a prefix Π has $\mathsf{DQIOR}_\exists$ on an existential literal $y \in C$ with respect to a DQBF $\Pi \cdot \psi$ when*

$$\psi \models C \cup \left(\mathsf{OC}(\Pi, D, \overline{y}) \setminus \{\overline{y}\}\right) , \quad \textit{for all } D \in \psi \textit{ with } \overline{y} \in D \, .$$

If a clause has $\mathsf{DQIOR}_\exists$ (on any existential literal) with respect to a DQBF, it can be added to the matrix while preserving truth value.

Theorem 5. *Let $\Pi \cdot \psi$ be a DQBF, and let C be a clause under Π. If C has $\mathsf{DQIOR}_\exists$ on an existential literal $p \in C$ with respect to $\Pi \cdot \psi$, then*

$$\Pi \cdot \psi \equiv_{tr} \Pi \cdot \psi \cup \{C\} \, .$$

Proof. We let $y := \text{var}(p)$, and assume without loss of generality that p is the positive literal y. We use aliases for the DQBFs in question, namely $\Psi := \Pi \cdot \psi$ and $\Psi_C := \Pi \cdot \psi \cup C$. Since the removal of clauses preserves the truth of a DQBF, truth of Ψ_C implies truth of Ψ. Thus, we need only show that the addition of C to Ψ preserves truth.

We introduce aliases $X := \text{vars}_\exists(\Pi)$ and $U := \text{vars}_\forall(\Pi)$. Let $f := \{f_x\}_{x \in X}$ be a model for Ψ, and let A be the set of assignments $\nu \in \langle U \rangle$ for which $\nu \cup f(\mu)$ falsifies C. Now, we define a set of dependency functions $g := \{g_x\}_{x \in X}$ as follows: when $x \neq y$, define $g_x = f_x$; otherwise,

$$\begin{aligned} g_y : \langle S_y^\Pi \rangle &\to \langle \{y\} \rangle \\ \rho &\mapsto \begin{cases} y \mapsto 1 & \text{if some assignment in } A \text{ extends } \rho \, , \\ f_y(\rho) & \text{otherwise} \, . \end{cases} \end{aligned}$$

We show that g models Ψ_C; that is, letting $\mu \in \langle U \rangle$, we show that $\mu \cup g(\mu)$ satisfies $\psi \cup \{C\}$. We consider two cases.

Case (a). Suppose that no assignment in A extends $\mu \restriction S_y^\Pi$. Then, since μ itself does not belong to A, $\mu \cup f(\mu)$ satisfies C. By construction we have

$$g_y(\mu \restriction S_y^\Pi) = f_y(\mu \restriction S_y^\Pi) \, ,$$

so $g(\mu) = f(\mu)$. Therefore $\mu \cup g(\mu)$ also satisfies ψ.

Case (b). On the other hand, suppose that some $\nu \in A$ extends $\mu \restriction S_y^\Pi$. Observe that $\mu \cup g(\mu)$ and $\mu \cup f(\mu)$ agree on the variables $\text{vars}(\Pi) \setminus \{y\}$, and $g_y(\mu \restriction S_y^\Pi)(y) = 1$ by construction. It is then easy to see that $\mu \cup g(\mu)$ satisfies C alongside all clauses $E \in \psi$ with $\overline{y} \notin E$.

It remains to show that $\mu \cup g(\mu)$ satisfies each $D \in \psi$ with $\overline{y} \in D$. As μ and ν agree on S_y^Π, they also agree on any $S_x^\Pi \subseteq S_y^\Pi$ with $x \neq y$. As a consequence, $\rho \cup g(\rho)$ and $\mu \cup f(\mu)$ agree on $\mathsf{OV}(\Pi, y) \setminus \{y\}$, and hence agree on the clause $D' := \mathsf{OC}(\Pi, D, y) \setminus \{\overline{y}\}$. As $\mu \cup f(\mu)$ satisfies ψ and falsifies C, it satisfies D' by definition of $\mathsf{DQIOR}_\exists$ (Definition 4). Therefore $\rho \cup g(\rho)$ satisfies D', and hence also satisfies the full clause D. $\square$

The universal treatment. The universal version $\mathsf{DQIOR}_\forall$ differs slightly from its existential counterpart.[6]

[6] In Definition 6, the universal literal u is deleted from C, while its negation $\overline{u}$ remains in $\mathsf{OC}(\Pi, D, \overline{u})$. In contrast, in Definition 4 the positive existential literal y was not deleted, whereas its negation $\overline{y}$ was. One could of course take the uniform approach (as in [38]) and delete both literals in a single definition, potentially defining a weaker property. As the upshot of such nuances is currently unclear, we stick to the most general definitions for which the proofs of Theorems 5 and 7 go through.

Definition 6 ($\mathsf{DQIOR}_\forall$)**.** *A clause C under a prefix Π has $\mathsf{DQIOR}_\forall$ on a universal literal $u \in C$ with respect to a DQBF $\Pi \cdot \psi$ when*

$$\psi \models \big(C \setminus \{u\}\big) \cup \mathsf{OC}(\Pi, D, \overline{u})\,, \quad \textit{for all } D \in \psi \textit{ with } \overline{u} \in D\,.$$

Literals witnessing $\mathsf{DQIOR}_\forall$ can be eliminated while preserving truth value.

Theorem 7. *Let $\Pi \cdot \psi$ be a DQBF, and let C be a clause under Π. If C has $\mathsf{DQIOR}_\forall$ on a universal literal $u \in C$ with respect to $\Pi \cdot \psi$, then*

$$\Pi \cdot \psi \cup \{C\} \equiv_{tr} \Pi \cdot \psi \cup \big\{C \setminus \{u\}\big\}\,.$$

Proof. We let $u := \mathrm{var}(p)$, and assume without loss of generality that p is the positive literal u. We use aliases for the DQBFs, namely $\Psi_C := \Pi \cdot \psi \cup \{C\}$ and $\Psi_{C\setminus\{u\}} := \Pi \cdot \psi \cup \big\{C \setminus \{u\}\big\}$. Since the addition of literals to clauses preserves the truth of a DQBF, truth of $\Psi_{C\setminus\{u\}}$ implies truth of Ψ_C. Hence we need only show the reverse implication, i.e. the removal of u from C preserves truth of Ψ_C.

For any assignment α whose domain contains u, we denote by α^{-u} the assignment obtained from α by flipping the assignment to u, while preserving the assignment on the remainder of the domain.

Once again we use the aliases $U := \mathrm{vars}_\forall(\Pi)$ and $X := \mathrm{vars}_\exists(\Pi)$, and we let $f := \{f_x\}_{x\in X}$ be a model for Ψ_C. Let A be the set of assignments $\nu \in \langle U \rangle$ for which $\nu \cup f(\nu)$ falsifies $C \setminus \{u\}$, and observe that every $\nu \in A$ assigns u to 1. We define a set of dependency functions $g := \{g_x\}_{x \in X}$ as follows: if $x \in I_u^\Pi$, we define $g_x := f_x$; if $x \in D_u^\Pi$, we define

$$\begin{aligned} g_x : \langle S_x^\Pi \rangle &\to \langle \{x\} \rangle \\ \rho &\mapsto \begin{cases} f_x\big(\rho^{-u}\big) & \text{if some } \nu \in A \text{ extends } \rho{\upharpoonright}K_u^\Pi\,, \\ f_x(\rho) & \text{otherwise}\,. \end{cases} \end{aligned}$$

We now show that g is a model for $\Psi_{C\setminus\{u\}}$; that is, letting $\mu \in \langle U \rangle$, we show that $\mu \cup g(\mu)$ satisfies $\psi \cup \big\{C \setminus \{u\}\big\}$. We consider two cases.

Case (a). Suppose that no assignment in A extends $\mu{\upharpoonright}K_u^\Pi$. For each $x \in D_u^\Pi$, we have $K_u^\Pi \subseteq S_x^\Pi$, which implies that no assignment in A extends $(\mu{\upharpoonright}S_x^\Pi){\upharpoonright}K_u^\Pi$ It follows that $g(\mu) = f(\mu)$. Since μ itself does not belong to A, $\mu \cup f(\mu)$ satisfies $\psi \cup \big\{C \setminus \{u\}\big\}$. Thus $\mu \cup g(\mu)$ also satisfies $\psi \cup \big\{C \setminus \{u\}\big\}$.

Case (b). Suppose that some $\nu \in A$ extends $\mu{\upharpoonright}K_u^\Pi$. In this case, for each $x \in D_u^\Pi$, ν extends $(\mu{\upharpoonright}S_x^\Pi){\upharpoonright}K_u^\Pi$, so we have

$$g_x\left(\mu{\upharpoonright}S_x^\Pi\right) = f_x\big((\mu{\upharpoonright}S_x^\Pi)^{-u}\big)\,.$$

It follows that $g(\mu) = f(\mu^{-u})$. This means that $\mu \cup g(\mu)$ and $\mu^{-u} \cup f(\mu^{-u})$ agree on all of the variables of Π except u. Note that $\mu^{-u}(u) = 0$, since $\nu(u) = 1$ and $u \in K_u^\Pi$, implying $\mu(u) = 1$. Hence, since $\mu^{-u} \cup f(\mu^{-u})$ satisfies C, it also satisfies $C \setminus \{u\}$. It is then easy to see that $\mu \cup g(\mu)$ satisfies $C \setminus \{u\}$ alongside each $E \in \psi$ with $\overline{u} \notin E$.

It remains to show that $\mu \cup g(\mu)$ satisfies each $D \in \psi$ with $\overline{u} \in D$. Recall that $g_x = f_x$ for each $x \in I_u^\Pi$. Since μ and ν agree on K_u^Π, $g(\mu)$ and $f(\nu)$ agree on each $x \in I_u^\Pi$ for which $S_x^\Pi \subseteq K_u^\Pi$. Therefore $\mu \cup g(\mu)$ and $\nu \cup f(\nu)$ agree on $\mathsf{OV}(\Pi, u)$. Since $\nu \cup f(\nu)$ satisfies ψ and falsifies $C \setminus \{u\}$, it satisfies $D' := \mathsf{OC}(\Pi, D, \overline{u})$ by definition of $\mathsf{DQIOR}_\forall$ (Definition 6). It follows that $\mu \cup g(\mu)$ satisfies D', and hence satisfies D also. □

clause set modification	**clause modification**	**prefix modification**
AT	UR	BPM
DQRAT$_\exists$	DQRAT$_\forall$	DRRS

Fig. 1. The six transformations of $\mathcal{R}$.

5 Truth-preserving DQBF Transformations

With the intention of generalising the QRAT proof system, we define a set of six truth-value-preserving DQBF transformations that we call $\mathcal{R}$ (Figure 1). $\mathcal{R}$ consists of two clause set modification rules, two clause modification rules, and two prefix modification rules. Over the course of this section, we will show the application of each $\mathcal{R}$ rule is both sound and tractable, in the following sense.

Theorem 8. *Each of the six rules of $\mathcal{R}$ is polynomial-time checkable and preserves DQBF truth value.*

In all cases, polynomial-time checkability is fairly self-evident, so we omit the complexity details. Preservation of truth value is discussed per rule at the appropriate point.

Our presentation differs from the original QRAT system in two ways: first, with the addition of the prefix modification category; and second, we present the transformations as *reversible* – for example, AT handles *both* clause addition *and* elimination. With this we intend merely to enhance the clarity of the presentation. We do not artificially inflate the strength of $\mathcal{R}$ beyond a natural generalisation.

Asymmetric tautologies. Several of the rules of $\mathcal{R}$ rely on the notion of an *asymmetric tautology*, which in turn relies on *unit propagation.*

Definition 9 (unit propagation). *The application of* unit propagation *to a CNF ψ produces the CNF* $\mathsf{UP}(\psi)$ *defined as the output of the following routine.*

```
1. $\mathsf{UP}(\psi) \leftarrow \psi$
2. while $\mathsf{UP}(\psi)$ contains a unit clause $\{a\}$
3.    $\mathsf{UP}(\psi) \leftarrow \mathsf{UP}(\psi)[\{\overline{a}\}]$
4. return $\mathsf{UP}(\psi)$
```

In summary, unit propagation is the assignment of *unit literals* until fixpoint. We note that line 2 in the routine in Definition 9 introduces an ambiguity (the output $\mathsf{UP}(\psi)$ may depend upon the choice of unit clause $\{a\}$), but this need not concern us. We are only interested in cases where unit propagation produces the empty clause (i.e. $\emptyset \in \mathsf{UP}(\psi)$), and this occurs independently of the choice of unit clause.

Definition 10 (AT [35, 38]). *We call a clause $C := \{a_1, \ldots, a_k\}$ an* asymmetric tautology *with respect to a CNF ψ (written $\psi \underset{\mathrm{U}}{\models} C$) when $\mathsf{UP}\big(\psi \cup \big\{\{\overline{a_1}\}, \cdots, \{\overline{a_k}\}\big\}\big)$ contains the empty clause.*

It is well known that an asymmetric tautology is always an implicant of the CNF [35].

Proposition 11. *For each CNF ψ and clause C, $\psi \underset{\mathrm{U}}{\models} C$ implies $\psi \models C$.*

Moreover, unit propagation clearly runs in time polynomial in $|C| + |\psi|$. Hence, identification of asymmetric tautologies via unit propagation serves as an efficiently computable underapproximation to propositional entailment.

Resolution asymmetric tautologies. Unit propagation also fosters efficiently computable underapproximations of the properties $\mathsf{DQIOR}_\exists$ and $\mathsf{DQIOR}_\forall$ (Section 4).

Definition 12 **($\mathsf{DQRAT}_\exists$).** *A clause C under a prefix Π has $\mathsf{DQRAT}_\exists$ on an existential literal $y \in C$ with respect to a DQBF $\Pi \cdot \psi$ when*

$$\psi \ \underset{\mathrm{U}}{\models}\ C \cup \big(\mathsf{OC}(\Pi, D, y) \setminus \{\overline{y}\}\big)\,, \quad \textit{for all } D \in \psi \textit{ with } \overline{y} \in D\,.$$

Definition 13 **($\mathsf{DQRAT}_\forall$).** *A clause C under a prefix Π has $\mathsf{DQRAT}_\forall$ on a universal literal $u \in C$ with respect to a DQBF $\Pi \cdot \psi$ when*

$$\psi \ \underset{\mathrm{U}}{\models}\ \big(C \setminus \{u\}\big) \cup \mathsf{OC}(\Pi, D, u)\,, \quad \textit{for all } D \in \psi \textit{ with } \overline{u} \in D\,.$$

Note that $\mathsf{DQRAT}_\exists$ is identical to $\mathsf{DQIOR}_\exists$, except that entailment ('$\models$') is approximated by unit propagation ('$\underset{\mathrm{U}}{\models}$'), and similarly for $\mathsf{DQRAT}_\forall$. Both properties will be used within $\mathcal{R}$. It should be noted that Definitions 12 and 13 constitute DQBF generalisations of *resolution asymmetric tautology* [42].

5.1 Clause set modification

$\mathcal{R}$ contains two *clause set modification* rules that allow the addition (or elimination) of clauses to (or from) a DQBF matrix. The first works with asymmetric tautologies.

Definition 14 **(AT clause set modification).** AT clause set modification *is the reversible transformation*

$$\Pi \cdot \psi \ \xleftrightarrow{\mathsf{AT}}\ \Pi \cdot \psi \cup \{C\}\,,$$

where $\Pi \cdot \psi \cup \{C\}$ is a DQBF, and $\psi \underset{\mathrm{U}}{\models} C$.

It is easy to see that whenever $\psi \models C$, any two DQBFs $\Pi \cdot \psi$ and $\Pi \cdot \psi \cup \{C\}$ have exactly the same set of models [55]. It follows that AT preserves truth value, by Proposition 11.

Our second clause set modification rule works with *resolution* asymmetric tautologies.

Definition 15 **($\mathsf{DQRAT}_\exists$ clause set modification).** $\mathsf{DQRAT}_\exists$ clause set modification *is the reversible transformation*

$$\Pi \cdot \psi \ \xleftrightarrow{\mathsf{DQRAT}_\exists}\ \Pi \cdot \psi \cup \{C\}\,,$$

where $\Pi \cdot \psi \cup \{C\}$ is a DQBF, and C has $\mathsf{DQRAT}_\exists$ on an existential literal with respect to $\Pi \cdot \psi$.

By Proposition 11, if C has $\mathsf{DQRAT}_\exists$ on y with respect to Ψ, then it has $\mathsf{DQIOR}_\exists$ on y with respect to Ψ. Hence $\mathsf{DQRAT}_\exists$ preserves truth value by Theorem 5.

5.2 Clause modification

$\mathcal{R}$ contains two *clause modification* transformations that allow the elimination (or addition) of literals from (or to) a clause in a DQBF matrix. The first of these is universal reduction, a staple ingredient of many QBF solvers and proof systems.

Definition 16 **(UR).** Universal reduction (UR) *is the reversible transformation*

$$\Pi \cdot \psi \cup \{C\} \ \xleftrightarrow{\mathsf{UR}}\ \Pi \cdot \psi \cup \{C \setminus \{u\}\}\,,$$

where $\Pi \cdot \psi \cup \{C\}$ is a DQBF, $u \in C$ is a universal literal, and $\mathrm{var}(u) \notin S_x^\Pi$ *for each existential variable $x \in C$.*

The deletion of universal literals by universal reduction is known to preserve truth [4]. Since deletion of literals trivially preserves falsity, UR preserves truth value.

The second clause modification rule is applicable when a universal literal *witnesses* a resolution asymmetric tautology.

Definition 17 (DQRAT$_\forall$ clause modification). DQRAT$_\forall$ clause modification *is the reversible transformation*

$$\Pi \cdot \psi \cup \{C\} \xleftrightarrow{\mathsf{DQRAT}_\forall} \Pi \cdot \psi \cup \{C \setminus \{u\}\}\,,$$

where $\Pi \cdot \psi \cup \{C\}$ is a DQBF, and C has DQRAT$_\forall$ *on the universal literal $u \in C$ with respect to $\Pi \cdot \psi$.*

Again by Proposition 11, if C has DQRAT$_\forall$ on u with respect to Ψ, then it has DQIOR$_\forall$ on u with respect to Ψ. Hence DQRAT$_\forall$ preserves truth value by Theorem 7.

5.3 Prefix modification

The remaining two $\mathcal{R}$ rules manipulate the quantifier prefix, while leaving the matrix unchanged.

Basic prefix modification. Basic prefix modification is merely a device which allows arbitrary fresh variables to be introduced, quantified either universally or existentially, and in the latter case with an arbitrary dependency set.

Definition 18 (basic prefix modification). Basic prefix modification *(*BPM*) is the reversible transformation*

$$\Pi \cdot \psi \xleftrightarrow{\mathsf{BPM}} \Omega \cdot \psi\,,$$

where $\Pi \cdot \psi$ is a DQBF, and the prefixes satisfy $\mathrm{vars}_\forall(\Pi) \subseteq \mathrm{vars}_\forall(\Omega)$, $\mathrm{vars}_\exists(\Pi) \subseteq \mathrm{vars}_\forall(\Omega)$, *and $S^\Pi_x = S^\Omega_x$ for each $x \in \mathrm{vars}_\exists(\Pi)$.*

Note that the reverse transformation allows redundant variables (i.e. those that do not appear in the matrix ψ) to be deleted from the prefix.

To see that BPM preserves truth value, let $\Pi \cdot \psi$ and Ω conform to the definition. It is easy to see that a model for $\Omega \cdot \psi$ is transformed into a model for $\Pi \cdot \psi$ by deleting the functions corresponding to the variables $X := \mathrm{vars}_\exists(\Omega) \setminus \mathrm{vars}_\exists(\Pi)$. Moreover, a model for $\Omega \cdot \psi$ is obtained from a model for $\Pi \cdot \psi$ by adding an arbitrary functions for X.

We note that QRAT was originally presented without explicit prefix modification. Instead, prefix manipulations were built into the other rules wherever needed (cf. [38]). Here, we find it more convenient to single out prefix modification. It allows all other rules to be stated under a single, unmutable prefix, and thus simplifies definitions and proofs throughout the paper.

The reflexive resolution path dependency scheme. Our second prefix modification rule DRRS is based on the *reflexive resolution path dependency scheme* ($\mathcal{D}^{\mathrm{rrs}}$) [59]. A *dependency scheme* [54] is a kind of truth-preserving prefix modification [67, 9], presented as a function from the set of DQBFs to itself. The particular scheme $\mathcal{D}^{\mathrm{rrs}}$ is defined as follows.

Definition 19 ($\mathcal{D}^{\mathrm{rrs}}$, adapted from [59]). *The* reflexive resolution path dependency scheme ($\mathcal{D}^{\mathrm{rrs}}$) *is the mapping $\Pi \cdot \psi \mapsto \Omega \cdot \psi$, where, for each $x \in \mathrm{vars}_\exists(\Pi)$, S^Ω_x is the set of universal variables $u \in S^\Pi_x$ for which there exists a sequence $C_1, \dots, C_k$ of clauses in ψ and a sequence $p_1, \dots, p_{k-1}$ of existential literals satisfying the following conditions:*

(a) *$u \in C_1$ and $\overline{u} \in C_k$,*
(b) *for some $i \in [k-1]$, $x = \mathrm{var}(p_i)$,*
(c) *for each $i \in [k-1]$, $p_i \in C_i$, $\overline{p}_i \in C_{i+1}$, and $u \in S^{\Pi}_{\mathrm{var}(p_i)}$,*
(d) *for each $i \in [k-2]$, $\mathrm{var}(p_i) \neq \mathrm{var}(p_{i+1})$.*

Theorem 20 ([67]). *For each DQBF Ψ, $\Psi \equiv_{tr} \mathcal{D}^{\mathrm{rrs}}(\Psi)$.*

$\mathcal{R}$ accesses the reflexive resolution path dependency scheme via the rule DRRS.

Definition 21 ($\mathcal{D}^{\mathrm{rrs}}$ prefix modification). $\mathcal{D}^{\mathrm{rrs}}$ prefix modifictation (DRRS) *is the reversible transformation*

$$\Psi \xleftrightarrow{\mathsf{DRRS}} \Phi,$$

where Ψ is a DQBF and $\mathcal{D}^{\mathrm{rrs}}(\Psi) \leq \Phi \leq \Psi$.

As we will see, DRRS is particularly important for the simulation of universal expansion [38].

Finally, we note that *extended universal resolution*, a feature of the original QRAT system, is essentially universal reduction with respect to the prefix given by $\mathcal{D}^{\mathrm{rrs}}$ (cf. the treatment in [44]). Loosely speaking we have broken up extended universal reduction into its component parts, namely the prefix modification by $\mathcal{D}^{\mathrm{rrs}}$ (DRRS) and the clause modification by universal reduction (UR). One benefit of our approach is that the truth-value preservation of the components is already established; in comparison, a formal proof of the soundness of extended universal reduction was lacking.[7]

6 DQRAT as a Refutational DQBF Proof System

In this section, we show how our set $\mathcal{R}$ of truth-value-preserving transformations yields a sound-and-complete refutational DQBF proof system which we call DQRAT. As one might expect, the general setup follows that of the QBF proof system QRAT [38], to which DQRAT reduces on the QBF fragment.

6.1 DQRAT refutations

To cast the rules of $\mathcal{R}$ as a refutational proof system, we essentially define a refutation as a sequence of $\mathcal{R}$-transformations of an input DQBF. A refutation concludes when the matrix contains the empty clause, and the DQBF is trivially false.

In addition to $\mathcal{R}$-transformations, we allow refutations to use *arbitrary clause deletion*, which trivially preserves truth and is therefore a natural and sound addition to a refutational system like DQRAT.

Definition 22 (DQRAT refutation). *A* DQRAT refutation *of a DQBF Ψ is a sequence $\pi := \Pi_1 \cdot \psi_1, \dots, \Pi_k \cdot \psi_k$ of DQBFs in which $\Pi_1 \cdot \psi_1 = \Psi$, ψ_k contains the empty clause, and, for each for each $i \in [k-1]$, either* (a) there is an $\mathcal{R}$-transformation from Ψ_i to Ψ_{i+1}, or (b) $\Pi_{i+1} = \Pi_i$ and $\psi_{i+1} \subseteq \psi_i$. The size of π is $|\pi| := \Sigma_{i=1}^{k} |\Pi_i \cdot \psi_i|$.

To show that DQRAT is a proof system in the formal sense [23], we must show three things:

(a) *Soundness.* If a DQBF has a DQRAT refutation, it is false.
(b) *Completeness.* Every false DQBF has a DQRAT refutation.
(c) *Checkability.* Given a DQBF Ψ and a sequence π of DQBFs, it can be determined in time polynomial in $|\pi|$ whether π is a DQRAT refutation of Ψ.

[7] [38] refers to a result in [29], but the precise connection was not stated.

Proving soundness and checkability is straightforward. The proof of completeness is deferred to the following subsection.

Theorem 23. DQRAT *is a proof system for the language of false DQBFs.*

Proof. Soundness. Let $\Psi_1, \dots, \Psi_k$ be a DQRAT refutation of $\Psi = \Psi_1$. Aiming for contradiction, suppose that Ψ is true. By Theorem 8 and the fact that

$$\Pi \cdot \psi \text{ is true and } \phi \subseteq \psi \implies \Pi \cdot \phi \text{ is true}\,,$$

we deduce that Ψ_k is true. This is a contradiction, since Ψ_k, whose matrix contains the empty clause, is trivially false. Therefore Ψ is false. *Completeness.* Established by Corollary 26 (Section 6.2). *Checkability.* The validity of each transformation from Ψ to Φ using some $\mathsf{r} \in \mathcal{R}$, along with the validity of clause deletion yielding Φ from Ψ, can be determined in time polynomial in $|\Psi| + |\Phi|$. Thus DQRAT refutations can be checked in polynomial time.

6.2 Completeness by simulation of $\forall$Exp+Res

It remains to establish the completeness of DQRAT (Corollary 26), which we demonstrate via a simulation of $\forall\mathsf{Exp{+}Res}$ (Theorem 25). The proof method closely follows that of the anologous QBF result [44]. We first provide a brief summary of $\forall\mathsf{Exp{+}Res}$ [40], followed by the details of the simulation. For a gentle introduction to $\forall\mathsf{Exp{+}Res}$ and the expansion paradigm, see [16].

$\forall$Exp+Res Preliminaries. For a given DQBF $\Psi := \Pi \cdot \psi$, a $\forall\mathsf{Exp{+}Res}$ refutation works on clauses over the *variable* set

$$\mathrm{vars}_{\mathrm{ex}}(\Psi) := \{x^\mu : x \in \mathrm{vars}_\exists(\Psi), \mu \in \langle S_x^\Pi \rangle\}\,.$$

We emphasise that $\mathrm{vars}(\Psi)$ and $\mathrm{vars}_{\mathrm{ex}}(\Psi)$ are disjoint. For any variable $x^\mu \in \mathrm{vars}_{\mathrm{ex}}(\Psi)$, we refer to $x \in \mathrm{vars}_\exists(\Psi)$ as the *base* of x^μ (written $\mathrm{base}(x^\mu)$), and to μ as its *annotation*. The *expansion* of Ψ is the CNF

$$\mathrm{ex}(\Psi) \;:=\; \bigcup\nolimits_{\mu \in \langle \mathrm{vars}_\forall(\Psi)\rangle} \psi[\mu][s_\mu]\,,$$

where s_μ substitutes each $x \in \mathrm{vars}_\exists(\Psi)$ with $x^{\mu \restriction S_x^\Pi} \in \mathrm{vars}_{\mathrm{ex}}(\Psi)$. The *resolvent* of two clauses C and D over a variable p with $p \in C$ and $\overline{p} \in D$ is the clause $\mathrm{res}(C, D, p) := \big(C \setminus \{p\}\big) \cup \big(D \setminus \{\overline{p}\}\big)$.

Definition 24 ($\forall\mathsf{Exp{+}Res}$ [40])**.** *A* $\forall\mathsf{Exp{+}Res}$ refutation *of a DQBF Ψ is a sequence of clauses $C_1, \dots, C_k$ in which C_k is the empty clause and at least one of the following holds for each $i \in [k]$:*

(a) Axiom: *C_i is a clause in the expansion of Ψ;*
(b) Resolution: *$C_i = \mathrm{res}(C_r, C_s, p)$, for some $r, s < i$ and $p \in \mathrm{vars}_{\mathrm{ex}}(\Psi)$.*

For the sake of clarity, we emphasise that a $\forall\mathsf{Exp{+}Res}$ refutation of a DQBF is merely a resolution refutation of its expansion, and the expansion is merely a *propositional CNF representation* of the DQBF on a fresh variable set. It is well known that a DQBF is false if, and only if, its expansion is unsatisfiable. Hence $\forall\mathsf{Exp{+}Res}$ is a sound and complete refutational DQBF proof system.

Theorem 25. DQRAT *p-simulates* $\forall\mathsf{Exp{+}Res}$ *on the language of false DQBFs.*

Proof. Let $\pi := A_1, \dots, A_k, R_1, \dots, R_l$ be a $\forall$Exp+Res refutation of a DQBF $\Psi := \Pi \cdot \psi$, in which each A_i is introduced as an axiom and each R_i is derived by resolution (that is, we us assume without loss of generality that every axiom in π precedes every resolution step). Moreover, we assume without loss of generality that no clause in ψ is a tautology. We construct a DQRAT refutation of Ψ in five distinct phases.

Phase 1: *extension of prefix.* The DQRAT refutation must work with the annotated variables $\mathrm{vars}_{\mathrm{ex}}(\Psi)$, more precisely with the subset of $\mathrm{vars}_{\mathrm{ex}}(\Psi)$ which appears in π. In the first phase we extend the prefix Π to incorporate the annotated variables. Let us write the variables appearing in π as the set $Z_\pi := \{z_1, \dots, z_v\}$, and let us define the prefix

$$\Pi_\pi \;:=\; \exists z_1(S^{\Pi}_{\mathrm{base}(z_1)}) \cdots \exists z_v(S^{\Pi}_{\mathrm{base}(z_v)}) \,.$$

Note that $Z_\pi \subseteq \mathrm{vars}_{\mathrm{ex}}(\Psi)$ consists exclusively of annotated variables, and hence that Z_π and $\mathrm{vars}(\Psi)$ are disjoint variable sets.

Claim 1. A DQRAT derivation of $\Pi\Pi_\pi \cdot \psi$ from Ψ can be constructed in time polynomial in $|\pi|$.

Phase 2: *introduction of definitions.* In the second phase we introduce 'defining clauses' for the annotated variables Z_π, in the form of the CNF

$$\psi_{\mathrm{def}} \;:=\; \big\{\{\neg \mathrm{base}(z_i), z_i\}, \{\mathrm{base}(z_i), \neg z_i\} : i \in [v]\big\} \,.$$

Claim 2. A DQRAT derivation of $\Pi\Pi_\pi \cdot \psi \cup \psi_{\mathrm{def}}$ from $\Pi\Pi_\pi \cdot \psi$ can be constructed in time polynomial in $|\pi|$.

Phase 3: *preparation of axioms.* In the third phase, we use ψ_{def} to substitute the existential variables in ψ with the appropriate annotated variables from Z_π. For each $i \in [k]$, let $D_i \in \psi$ be a clause for which $A_i = D_i[\mu][s_\mu]$ for some μ (that is, $D_i \in \psi$ *corresponds* to the axiom A_i), and let $D_i^{\forall}$ be its universal subclause. We define the CNF $\psi_{\mathrm{prep}} := \{\{A_i \cup D_i^{\forall}\} : i \in [k]\}$.

Claim 3. A DQRAT derivation of $\Pi\Pi_\pi \cdot \psi_{\mathrm{prep}}$ from $\Pi\Pi_\pi \cdot \psi \cup \psi_{\mathrm{def}}$ can be constructed in time polynomial in $|\pi|$.

Phase 4: *derivation of axioms.* In the fourth phase, we eliminate all universal variables, leaving us with a fully existentially quantified DQBF whose matrix consists of the axioms of π. We define the prefix $\Pi_\pi^{\emptyset} := \exists z_1(\emptyset) \cdots \exists z_v(\emptyset)$.

Claim 4. A DQRAT derivation of $\Pi_\pi^{\emptyset} \cdot \{A_1, \dots, A_k\}$ from $\Pi\Pi_\pi \cdot \psi_{\mathrm{prep}}$ can be constructed in time polynomial in $|\pi|$.

Phase 5: *derivation of resolvents.* In the final phase, we add the remaining clauses of π, namely the resolvents $R_1, \dots, R_k$, to the matrix. The final addition of the empty clause R completes the DQRAT refutation.

Claim 5. A DQRAT derivation of $\Pi_\pi^{\emptyset} \cdot \{A_1, \dots, A_k, R_1, \dots, R_l\}$ from $\Pi_\pi^{\emptyset} \cdot \{A_1, \dots, A_k\}$ can be constructed in time polynomial in $|\pi|$.

From these five claims, it follows that a DQRAT refutation of Ψ can be constructed in time polynomial in $|\pi|$. It remains to prove them.

Proof of Claim 1. Since $\mathrm{vars}(\psi) \subseteq \mathrm{vars}(\Pi)$ and $\mathrm{vars}(\Pi) \cap \mathrm{vars}(\Pi_\pi) = \emptyset$, $\Pi\Pi_\pi \cdot \psi$ can be obtained from $\Pi \cdot \psi$ using BPM. Hence the sequence $\Psi, \Pi\Pi_\pi \cdot \psi$ is a DQRAT derivation, clearly constructible in time polynomial in $|\pi|$.

Proof of Claim 2. We let $\psi^0_{\mathrm{def}} := \emptyset$, and for each $i \in [v]$ we define the CNF

$$\psi^i_{\mathrm{def}} := \psi^{i-1}_{\mathrm{def}} \cup \big\{\{\neg\mathrm{base}(z_i), z_i\}, \{\mathrm{base}(z_i), \neg z_i\}\big\} .$$

Now, fixing $i \in [v]$, we introduce two aliases $C := \{\mathrm{base}(z_i), \neg z_i\}$ and $C' := \{\neg\mathrm{base}(z_i), z_i\}$. Since z_i does not appear at all in ψ_{i-1}, C' has $\mathsf{DQRAT}_\exists$ on z_i with respect to $\Pi\Pi_\pi \cdot \psi \cup \psi_{i-1}$ vacuously. Therefore

$$\Pi\Pi_\pi \cdot \psi \cup \psi^{i-1}_{\mathrm{def}} \xrightarrow{\mathsf{DQRAT}_\exists} \Pi\Pi_\pi \cdot \psi \cup \psi^{i-1}_{\mathrm{def}} \cup C' . \tag{1}$$

Also, C has $\mathsf{DQRAT}_\exists$ on $\neg z_i$ with respect to the right-hand side of (1). To see this, observe that the only matrix clause containing z_i is C', where $\mathsf{OC}(\Pi\Pi_\pi, D, \neg z_i) = C'$ and

$$\emptyset \ \underset{\mathrm{U}}{\models} \ C \cup \big(\mathsf{OC}(\Pi\Pi_\pi, C', \neg z_i) \setminus \{z_i\}\big)$$

holds trivially because the right-hand side is a tautology. Therefore

$$\Pi\Pi_\pi \cdot \psi \cup \psi^{i-1}_{\mathrm{def}} \cup C' \xrightarrow{\mathsf{DQRAT}_\exists} \Pi\Pi_\pi \cdot \psi \cup \psi^{i}_{\mathrm{def}} .$$

Since $\psi^w_{\mathrm{def}} = \psi_{\mathrm{def}}$, we obtain a derivation of $\Pi\Pi_\pi \cdot \psi$ from $\Pi\Pi_\pi \cdot \psi \cup \psi_{\mathrm{def}}$, clearly constructible in time polynomial in $|\pi|$.

Proof of Claim 3. We let $\psi^0_{\mathrm{prep}} := \emptyset$, and for each $i \in [k]$ we define the CNF

$$\psi^i_{\mathrm{prep}} := \psi^{i-1}_{\mathrm{prep}} \cup \{A_i \cup D^\forall_i\} .$$

Let $i \in [k]$. Observe that D_i is obtained from $A_i \cup D^\forall_i$ by replacing each annotated variable in A_i with its base (i.e. by inverting the substitution s_μ). Also, D_i belongs to ψ, and for each $z \in \mathrm{vars}(A_i)$, the clauses $\{\neg\mathrm{base}(z), z\}, \{\mathrm{base}(z), \neg z\}$ belong to ψ_{def}. Thus it is easy to see that $\{D_i\} \cup \psi_{\mathrm{def}} \underset{\mathrm{U}}{\models} \{A_i \cup D^\forall_i\}$; hence,

$$\Pi' \cdot \psi \cup \psi_{\mathrm{def}} \cup \psi^{i-1}_{\mathrm{prep}} \xrightarrow{\mathsf{AT}} \Pi' \cdot \psi \cup \psi_{\mathrm{def}} \cup \psi^{i}_{\mathrm{prep}} .$$

Since $\psi^k_{\mathrm{prep}} = \psi_{\mathrm{prep}}$, we obtain the required derivation, clearly constructible in time polynomial in $|\pi|$.

Proof of Claim 4. First, since $\mathrm{vars}_\exists(\Pi)$ and $\mathrm{vars}(\psi_{\mathrm{prep}})$ are disjoint, $\mathrm{vars}_\exists(\Pi)$ and the associated dependency sets can be removed via BPM; that is,

$$\Pi\Pi_\pi \cdot \psi_{\mathrm{prep}} \xrightarrow{\mathsf{BPM}} \forall u_1 \cdots \forall u_m \, \Pi_\pi \cdot \psi_{\mathrm{prep}} ,$$

where $\mathrm{vars}_\forall(\Pi) = \{u_1, \ldots, u_m\}$. Next, we observe that

$$\forall u_1 \cdots \forall u_m \, \Pi_\pi \cdot \psi_{\mathrm{prep}} \xrightarrow{\mathsf{DRRS}} \forall u_1 \cdots \forall u_m \, \Pi^\emptyset_\pi \cdot \psi_{\mathrm{prep}} .$$

To see that this step is valid, consider an arbitrary $u_j \in \mathrm{vars}_\forall(\Pi)$. Aiming for contradiction, suppose that there exists a sequence of clauses $C_1, \ldots, C_l \in \psi_{\mathrm{prep}}$ and a sequence of existential literals $p_1, \ldots, p_{l-1}$ satisfying conditions (a) to (d) of Definition 19 with respect to the prefix

$\forall u_1 \cdots u_m \,\Pi_\pi$. Since $u_j \in C_1$ by condition (a), the annotation of p_1 contains $\overline{u_j}$. Moreover, since the annotations in any given C_i are consistent, we deduce by condition (c) that the annotation of each p_i contains $\overline{u_j}$. In particular the annotation of p_{l-1} contains $\overline{u_j}$, and since $\overline{p_{l-1}} \in C_l$ by condition (c), we must have $u_j \in C_l$. Since $\overline{u_j} \in C_l$ by condition (a), C_l is a tautology. But this implies that some clause in ψ is a tautology, a contradiction. Therefore each dependency set in $\mathcal{D}^{\text{rrs}}(\forall u_1 \cdots \forall u_m \,\Pi_\pi \cdot \psi_{\text{prep}})$ is empty.

Finally, we reduce all the universal literals from ψ_{prep}, then remove the universal variables from the prefix; that is,

$$\forall u_1 \cdots \forall u_m \,\Pi_\pi \cdot \psi_{\text{prep}} \xrightarrow{\mathsf{UR}^*} \forall u_1 \cdots \forall u_m \,\Pi_\pi^{\emptyset} \cdot \{A_1, \ldots, A_k\} \xrightarrow{\mathsf{BPM}} \Pi_\pi^{\emptyset} \cdot \{A_1, \ldots, A_k\}\,.$$

Note that this comprises at most $m \cdot |\pi|$ applications of UR. The complete derivation of $\Pi_\pi^{\emptyset} \cdot \{A_1, \ldots, A_k\}$ from $\Pi\Pi_\pi \cdot \psi_{\text{prep}}$ can clearly be constructed in time polynomial in $|\pi|$.

Proof of Claim 5. Let $i \in [l]$. By the definition of $\forall$Exp+Res refutation, there exist two clauses $C_r, C_s \in \{A_1, \ldots, A_k, R_1, \ldots, R_{i-1}\}$ and a variable $z \in \text{vars}_{\text{ex}}(\Psi)$ with $z \in C_r$, $\overline{z} \in C_s$, such that $R_i = \big(C_r \setminus \{z\}\big) \cup \big(C_s \setminus \{\overline{z}\}\big)$. It is readily verified that $\{C_r, C_s\} \models_{\text{U}} R_i$, and so

$$\Pi_\pi \cdot \{A_1, \ldots, A_k, R_1, \ldots, R_{i-1}\} \xrightarrow{\mathsf{AT}} \Pi_\pi \cdot \{A_1, \ldots, A_k, R_1, \ldots, R_i\}\,.$$

Thus, a DQRAT derivation of $\Pi_\pi \cdot \{A_1, \ldots, A_k, R_1, \ldots, R_l\}$ from $\Pi_\pi \cdot \{A_1, \ldots, A_k\}$ can clearly be constructed in time polynomial in $|\pi|$. □

Since $\forall$Exp+Res is itself complete for DQBF [7], completeness of DQRAT follows immediately from the simulation (Theorem 25).

Corollary 26. *Every false DQBF has a* DQRAT *refutation.*

7 Simulating DQBF Preprocessing Rules with $\mathcal{R}$

The primary purpose of the original QRAT proof system was to simulate all known QBF preprocessing techniques with a small set of inference rules. In this section we investigate the analogous question of how far $\mathcal{R}$ simulates the state-of-the-art in DQBF preprocessing.

As it turns out, $\mathcal{R}$ is capable of simulating almost all existing preprocessing techniques, and more. Only techniques involving so-called *variable definitions* appear difficult to simulate. As a solution, we introduce an advanced form of prefix modification, whose addition to $\mathcal{R}$ is sufficient to cover their simulation.

In Subsection 7.1, we cover the $\mathcal{R}$-simulation of the majority of known DQBF preprocessing. Advanced prefix modification and variable definitions are dealt with in Subsection 7.2. Further preprocessing techniques for DQBF are covered in Subsection 7.3.

7.1 Preprocessing without variable definitions

The preprocessing techniques with which we deal in this section are grouped into five categories. We discuss each category in turn, accompanied by a table in which the techniques, their preconditions, and their $\mathcal{R}$-simulations are formalised.[8]

Equivalent literals. HQSPre identifies *equivalent literals* by searching the binary implication graph of the DQBF matrix.

[8] We omit the table for dependency set reduction, whose $\mathcal{R}$-simulation is trivial.

preprocessing	**preconditions**	**$\mathcal{R}$-simulation**
$\Pi \cdot \psi \overset{\mathsf{EL1}}{\Longrightarrow} \Pi \cdot \{\emptyset\}$	$\mathrm{var}(p) \in \mathrm{vars}_\forall(\Pi)$ $\mathrm{var}(q) \in \mathrm{vars}_\forall(\Pi)$	$\Pi \cdot \psi$ $\xrightarrow{\mathsf{AT}} \Pi \cdot \psi \cup \{\{p, \overline{q}\}\}$ $\xrightarrow{\mathsf{UR}^*} \Pi \cdot \psi \cup \{\emptyset\}$
$\Pi \cdot \psi \overset{\mathsf{EL2}}{\Longrightarrow} \Pi \cdot \{\emptyset\}$	$\mathrm{var}(p) \in \mathrm{vars}_\forall(\Pi)$ $\mathrm{var}(q) = x \in \mathrm{vars}_\exists(\Pi)$ $\mathrm{var}(q) \notin S^\Pi_x$	$\Pi \cdot \psi$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{\{p, \overline{q}\}, \{\overline{p}, q\}\}$ $\xrightarrow{\mathsf{UR}^*} \Pi \cdot \psi \cup \{\{p\}, \{\overline{p}\}\}$ $\xrightarrow{\mathsf{AT}} \Pi \cdot \psi \cup \{\emptyset\}$
$\Pi \cdot \psi \overset{\mathsf{EL3}}{\Longrightarrow} \Pi \setminus \{\exists x(S^\Pi_x)\} \cdot \psi[p/q]$	$\mathrm{var}(p) \in \mathrm{vars}_\forall(\Pi)$ $\mathrm{var}(q) = x \in \mathrm{vars}_\exists(\Pi)$ $\mathrm{var}(p) \in S^\Pi_x$	$\Pi \cdot \psi$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{\{p, \overline{q}\}, \{\overline{p}, q\}\}$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{\{p, \overline{q}\}, \{\overline{p}, q\}\} \cup \psi[p/q]$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \{\{p, \overline{q}\}, \{\overline{p}, q\}\} \cup \psi[p/q]$ $\xrightarrow{\mathsf{DQRAT}^*_\exists} \Pi \cdot \psi[p/q]$ $\xrightarrow{\mathsf{BPM}} \Pi \setminus \{\exists x(S^\Pi_x)\} \cdot \psi[p/q]$

Fig. 2. Summary of the $\mathcal{R}$-simulation of equivalent literal preprocessing steps. Literals p and q are binary-clause equivalent with respect to ψ.

Definition 27. *Given a CNF ψ, the* binary implication graph $BIG(\psi) = (V, E)$ *has vertex set $V := \{z, \overline{z} : z \in \mathrm{vars}(\psi)\}$ and edge set $E := \{(p, \overline{q}), (\overline{p}, q) : \{p, q\} \in \psi\}$.*

The following is easy to see: if there are paths P_1 (from p to q) and P_2 (from q to p) in $BIG(\psi)$, then ψ and $\psi[q/p]$ are equivalent [65, Lemma 3]. The paths P_1 and P_2 encode sequences of binary clauses from ψ of the form

$$\begin{aligned} S_1 &:= \{p, r_1\}, \{\overline{r_1}, r_2\}, \ldots, \{\overline{r_{k-1}}, r_k\}, \{\overline{r_k}, \overline{q}\}\,, \\ S_2 &:= \{q, s_1\}, \{\overline{s_1}, s_2\}, \ldots, \{\overline{s_{k'-1}}, s_{k'}\}, \{\overline{s_{k'}}, \overline{p}\}\,, \end{aligned}$$

We say that p and q are *binary clause equivalent* in ψ. Note that both $\{p, \overline{q}\}$ and $\{\overline{p}, q\}$ are asymmetric tautologies with respect to ψ.

HQSPre implements four preprocessing steps based on equivalent literals, which we denote EL1 through EL4. The $\mathcal{R}$-simulation of EL1 through EL3 is summarised in Figure 2; EL4 involves defined variables, and is covered in Subsection 7.2.

EL1 determines that equivalent *universal* literals imply the falsity of the DQBF. The appropriate $\mathcal{R}$-simulation derives a trivially false DQBF, i.e. one whose matrix contains the empty clause. This is as simple as adding $\{p, \overline{q}\}$ by AT and applying universal reduction (UR).

EL2 determines that the DQBF is false when p is universal, q is existential, and the dependency set for $\mathrm{var}(q)$ does not contain $\mathrm{var}(p)$. Once again, the $\mathcal{R}$-simulation derives a trivially false DQBF with a combination of AT and UR.

EL3 covers the sister case where the dependency set for $\mathrm{var}(q)$ *does* contain $\mathrm{var}(p)$, in which case p can be substituted for q and $\mathrm{var}(q)$ removed from the prefix. After introducing $\{p, \overline{q}\}$ and $\{\overline{p}, q\}$, it is easy to see that ψ can be transformed into $\psi[p/q]$ by successive applications of AT, first introducing the clauses in the latter one by one, then deleting those of the former. Note that $\{p, \overline{q}\}$ has $\mathsf{DQRAT}_\exists$ on $\overline{q}$ with respect to $\Pi \cdot \{\overline{p}, q\} \cup \psi[p/q]$, and $\{\overline{p}, q\}$ has $\mathsf{DQRAT}_\exists$ (trivally) on q with respect to $\Pi \cdot \psi[p/q]$, hence both clauses can be removed via $\mathsf{DQRAT}_\exists$.

preprocessing	$\Pi \cdot \psi \cup \{C_1 \cup \{\overline{p}\}, \dots, C_r \cup \{\overline{p}\}, D_1 \cup \{p\}, \dots, D_s \cup \{p\}\}$ $\xRightarrow{\mathsf{EVE1}} \Pi \setminus \{\exists x(S_x^{\Pi})\} \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}$
preconditions	$x \in \text{vars}_{\exists}(\Pi)$ $x \notin \text{vars}(\psi)$ $\text{var}(p) = x$ $\mathsf{OC}(\Pi, D_i \cup \{p\}, x) = D_i \cup \{p\}$, for each $i \in [r]$
$\mathcal{R}$-simulation	$\Pi \cdot \psi \cup \{C_1 \cup \{\overline{p}\}, \dots, C_r \cup \{\overline{p}\}, D_1 \cup \{p\}, \dots, D_s \cup \{p\}\}$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{C_1 \cup \{\overline{p}\}, \dots, C_r \cup \{\overline{p}\}, D_1 \cup \{p\}, \dots, D_s \cup \{p\}\}$ $\quad \cup \{C_i \cup D_j : i \in [r], j \in [s]\}$ $\xrightarrow{\mathsf{DQRAT}_{\exists}^*} \Pi \cdot \psi \cup \{D_1 \cup \{p\}, \dots, D_s \cup \{p\}\} \cup \{C_i \cup D_j : i \in [r], j \in [s]\}$ $\xrightarrow{\mathsf{DQRAT}_{\exists}^*} \Pi \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}$ $\xrightarrow{\mathsf{BPM}} \Pi \setminus \{\exists x(S_x^{\Pi})\} \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}$

Fig. 3. The $\mathcal{R}$-simulation of existential variable elimination by resolution.

Then existential variable $\text{var}(q)$, which no longer appears in the matrix $\psi[p/q]$, is removed from the prefix with BPM.

Dependency set reduction. HQSPre implements *dependency set reduction* (DSR) using the *standard dependency scheme* ($\mathcal{D}^{\text{std}}$). Since $\mathcal{D}^{\text{std}}$ is *less general* than $\mathcal{D}^{\text{rrs}}$ [59, 67], that is to say

$$\mathcal{D}^{\text{rrs}}(\Psi) \leq \mathcal{D}^{\text{std}}(\Psi) \leq \Psi\,, \qquad \text{for each DQBF } \Psi\,,$$

application of $\mathcal{D}^{\text{std}}$ is trivially simulated by DRRS.

Existential variable elimination by resolution. *Davis-Putnam resolution* [24] is an early CNF-based decision procedure. In summary, variables are eliminated one by one from a CNF by adding all resolvents over a given variable while deleting the clauses in which it appears. Davis-Putnam resolution is indeed a decision procedure, in the precise sense that a CNF is unsatisfiable if, and only if, the described procedure generates the empty clause. The elimination of *individual*, *innermost* existential variables via Davis-Putnam resolution is a standard QBF preprocessing technique [38].

HQSPre implements existential variable elimination via Davis-Putnam resolution under two scenarios, which we denote EVE1 and EVE2. The latter, which involves variable definitions, is covered in Subsection 7.2.

EVE1 allows an existential variable x to be eliminated provided that, for either literal p with $\text{var}(p) = x$, all clauses containing p contain only variables outer to x. In essence, this allows Davis-Putnam resolution to be performed over x as if it were an innermost existential variable in a QBF prefix.

The $\mathcal{R}$-simulation of EVE1, shown in Figure 3, follows the analogous QBF simulation [38]. All x-resolvents can be introduced as asymmetric tautologies with respect to the matrix. Then, clauses containing $\overline{p}$ always have $\mathsf{DQRAT}_{\exists}$ on $\overline{p}$ with respect to the remaining DQBF, and can be deleted one by one. Thereafter, since literal $\overline{p}$ no longer appears, clauses containing p have

preprocessing	**preconditions**	**$\mathcal{R}$-simulation**
$\Pi \cdot \psi \cup \{C \cup \{p\}\} \overset{\mathsf{BCE}}{\Longrightarrow} \Pi \cdot \psi$	$\mathrm{var}(p) \in \mathrm{vars}_\exists(\Pi)$ for each $D \in \psi$ with $\overline{p} \in D$, $(C \setminus \{p\}) \cup (\mathsf{OC}(\Pi, D, \overline{p}) \setminus \{\overline{p}\})$ is a tautology.	$\Pi \cdot \psi \cup \{C \cup \{p\}\} \xrightarrow{\mathsf{DQRAT}_\exists} \Pi \cdot \psi$
$\Pi \cdot \psi \cup \{C \cup \{\overline{p}\}, D\} \overset{\mathsf{HLA}}{\Longrightarrow} \Pi \cdot \psi \cup \{C \cup \{\overline{p}\}, D \cup \{p\}\}$	$C \subseteq D$	$\Pi \cdot \psi \cup \{C \cup \{\overline{p}\}, D\} \xrightarrow{\mathsf{AT}} \Pi \cdot \psi \cup \{C \cup \{\overline{p}\}, D, D \cup \{p\}\} \xrightarrow{\mathsf{AT}} \Pi \cdot \psi \cup \{C \cup \{\overline{p}\}, D \cup \{p\}\}$
$\Pi \cdot \psi \cup \{C \cup \{p\}\} \overset{\mathsf{CLA}}{\Longrightarrow} \Pi \cdot \psi \cup \{C \cup \{p, q\}\}$	for each $D \in \psi$ with $\overline{p} \in D$, $(C \setminus \{p\}) \cup (\mathsf{OC}(\Pi, D, \overline{p}) \setminus \{\overline{p}\})$ is a tautology, or else contains q	$\Pi \cdot \psi \xrightarrow{\mathsf{AT}} \Pi \cdot \psi \cup \{C \cup \{p\}, C \cup \{p, q\}\} \xrightarrow{\mathsf{DQRAT}_\exists} \Pi \cdot \psi \cup \{C \cup \{p, q\}\}$

Fig. 4. Summary of the $\mathcal{R}$-simulation of blocked clause elimination, hidden literal addition, and covered literal addition.

$\mathsf{DQRAT}_\exists$ (trivially) with respect to the remaining DQBF, and can also be deleted one by one. Finally, since x no longer appears in the matrix, it is eliminated via BPM.

Blocked clause elimination (including hidden and covered literal addition). The elimination of blocked clauses [45] is standard in SAT [41] and QBF [15] preprocessing. Blocked clause elimination (BLE) is often performed in conjunction with literal addition techniques such as hidden (HLA) [33] and covered (CLA) [34] literal addition. HQSPre implements BCE in the DQBF setting, and exploits HLA and CLA to strengthen clause eliminability.

The $\mathcal{R}$-simulations of these three preprocessing steps is shown in Figure 4. As in the QBF case, BCE is simulated merely by a single application of $\mathsf{DQRAT}_\exists$, since a blocked clause always has $\mathsf{DQRAT}_\exists$ on the *blocking literal* p with respect to the remaining DQBF. The simulation of HLA, which is a purely propositional preprocessing technique, is simulated easily by AT, while the simulation of CLA is identical to its QBF counterpart [38].

Universal expansion. The expansion of individual universal variables is a standard QBF technique [14, 38]. HQSPre utilises universal expansion (UE) in the DQBF setting in much the same way.

As opposed the complete expansion of all universal variables as in the proof system $\forall\mathsf{Exp{+}Res}$ (Section 6), the goal here is to remove a single universal variable u from the DQBF. To account for the dependencies on u, we must introduce a fresh copy x' of each existential $x \in D_u^\Pi$, and replace the matrix ψ with

$$\psi[0/u] \cup \psi[1/u, x'/x \text{ for each } x \in D_u^\Pi] \,.$$

The $\mathcal{R}$-simulation in Figure 5 is lengthy, but the individual steps are relatively simple to verify, and the overall structure of the simulation is a straightforward generalisation from the QBF setting [38].

preprocessing	$\Pi \cdot \psi \cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1, \dots, E_t\}$ $\xRightarrow{\mathsf{UE}} \big(\Pi \setminus (\{\forall u\} \cup \{\exists x(S_x^\Pi) : x \in D_u^\Pi\})\big) \cup \big\{\exists x\big(S_x^\Pi \setminus \{u\}\big)\, \exists x'\big(S_x^\Pi \setminus \{u\}\big) : x \in D_u^\Pi\big\} \cdot$ $\psi \cup \{C_1, \dots, C_r, E_1, \dots, E_t\} \cup \{D_1, \dots, D_s, E_1, \dots, E_t\}[x'/x : x \in D_u^\Pi]$
preconditions	$\mathrm{vars}(\psi) \cap D_u^\Pi = \emptyset$ $\mathrm{vars}(E_i) \cap D_u^\Pi \neq \emptyset$, for each $i \in [t]$ $x' \notin \mathrm{vars}(\Pi)$, for each $x \in D_u^\Pi$
$\mathcal{R}$-simulation	$\Pi \cdot \psi \cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1, \dots, E_t\}$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1, \dots, E_t\}$ $\cup \{E_1 \cup \{u\}, \dots, E_t \cup \{u\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}$ $\xrightarrow{\mathsf{AT}^*} \Pi \cdot \psi \cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}\}$ $\cup \{E_1 \cup \{u\}, \dots, E_t \cup \{u\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}$ $\xrightarrow{\mathsf{BPM}} \Pi \cup \big\{\exists x'(S_x^\Pi) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}\}$ $\cup \{E_1 \cup \{u\}, \dots, E_t \cup \{u\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}$ $\xrightarrow{\mathsf{DQRAT}^*_\exists} \Pi \cup \big\{\exists x'(S_x^\Pi) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}\}$ $\cup \{E_1 \cup \{u\}, \dots, E_t \cup \{u\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}$ $\cup \big\{\{\overline{u}, x, \overline{x'}\}, \{\overline{u}, \overline{x}, x'\} : x \in D_u^\Pi\big\}$ $\xrightarrow{\mathsf{AT}^*} \Pi \cup \big\{\exists x'(S_x^\Pi) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}\}$ $\cup \{E_1 \cup \{u\}, \dots, E_t \cup \{u\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}$ $\cup \big\{\{\overline{u}, x, \overline{x'}\}, \{\overline{u}, \overline{x}, x'\} : x \in D_u^\Pi\big\}$ $\cup \{D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}[x'/x : x \in D_u^\Pi]$ $\xrightarrow{\mathsf{AT}^*} \Pi \cup \big\{\exists x'(S_x^\Pi) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, E_1 \cup \{u\}, \dots, E_t \cup \{u\}\}$ $\cup \big\{\{\overline{u}, x, \overline{x'}\}, \{\overline{u}, \overline{x}, x'\} : x \in D_u^\Pi\big\}$ $\cup \{D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}[x'/x : x \in D_u^\Pi]$ $\xrightarrow{\mathsf{DQRAT}^*_\exists} \Pi \cup \big\{\exists x'(S_x^\Pi) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, E_1 \cup \{u\}, \dots, E_t \cup \{u\}\}$ $\cup \{D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}[x'/x : x \in D_u^\Pi]$ $\xrightarrow{\mathsf{DRRS}^*} \big(\Pi \setminus \{\exists x(S_x^\Pi) : x \in D_u^\Pi\}\big) \cup \big\{\exists x\big(S_x^\Pi \setminus \{u\}\big), \exists x'\big(S_x^\Pi \setminus \{u\}\big) : x \in D_u^\Pi\big\} \cdot \psi$ $\cup \{C_1 \cup \{u\}, \dots, C_r \cup \{u\}, E_1 \cup \{u\}, \dots, E_t \cup \{u\}\}$ $\cup \{D_1 \cup \{\overline{u}\}, \dots, D_s \cup \{\overline{u}\}, E_1 \cup \{\overline{u}\}, \dots, E_t \cup \{\overline{u}\}\}[x'/x : x \in D_u^\Pi]$ $\xrightarrow{\mathsf{UR}^*} \big(\Pi \setminus \{\exists x(S_x^\Pi) : x \in D_u^\Pi\}\big) \cup \big\{\exists x\big(S_x^\Pi \setminus \{u\}\big), \exists x'\big(S_x^\Pi \setminus \{u\}\big) : x \in D_u^\Pi\big\} \cdot$ $\psi \cup \{C_1, \dots, C_r, E_1, \dots, E_t\} \cup \{D_1, \dots, D_s, E_1, \dots, E_t\}[x'/x : x \in D_u^\Pi]$ $\xrightarrow{\mathsf{BPM}} \big(\Pi \setminus (\{\forall u\} \cup \{\exists x(S_x^\Pi) : x \in D_u^\Pi\})\big) \cup \big\{\exists x\big(S_x^\Pi \setminus \{u\}\big)\, \exists x'\big(S_x^\Pi \setminus \{u\}\big) : x \in D_u^\Pi\big\} \cdot$ $\psi \cup \{C_1, \dots, C_r, E_1, \dots, E_t\} \cup \{D_1, \dots, D_s, E_1, \dots, E_t\}[x'/x : x \in D_u^\Pi]$

Fig. 5. The $\mathcal{R}$-simulation of universal expansion.

Note that the application of DRRS is valid, since (at that point in the simulation) the variables depending on u can be partitioned into the two sets D_u^{Π} and $\{x' : x \in D_u^{\Pi}\}$, where the former appear only in clauses containing the positive literal u, and the latter only in clauses containing the negative literal $\overline{u}$. Hence, in the image under $\mathcal{D}^{\text{rrs}}$, u does not appear in any dependency set.

7.2 Advanced prefix modification: working with variable definitions

In the previous subsection, we saw that $\mathcal{R}$ is capable of simulating the full suite of DQBF preprocessing techniques, except those involving defined variables, namely EL4 and EVE2. In this subsection we propose *advanced prefix modification* (APM), whose addition to $\mathcal{R}$ is sufficient to simulate both techniques.

The *defined variable* is a well-studied CNF (i.e. propositional) concept. Defined variables most commonly arise via Tseitin transformation [62] of circuit representations into CNF, but can also occur in any given instance independently of such transformations.

As an example, consider the CNF

$$\psi := \{\{u, v, \overline{x}\}, \{\overline{u}, x\}, \{\overline{v}, x\}\} .$$

It is easy to see that any satisfying assignment α for ψ must set $\alpha(x) \equiv \alpha(u) \wedge \alpha(v)$; moreover, any satisfying assignment for any larger CNF $\phi \supseteq \psi$ must also have this property. It is said that ψ constitutes a *functional definition* of x, which means (informally) that ψ encodes a relationship $x \equiv f(u, v)$, and in our example f is the Boolean OR function.

For our current purposes, we define 'defined variable' so as to abstract away the defining function f. We require only that such a function exists.

Definition 28 (defined variable). *A variable z is* defined *in a CNF ψ by a variable set $Z \subseteq \text{vars}(\psi)$ when, for each $\alpha \in \langle Z \rangle$, either $\{x\} \in \psi[\alpha]$ or $\{\overline{x}\} \in \psi[\alpha]$.*

We will concern ourselves only with defined *existential* variables. For particular 'defining sets' Z, the dependency set for a defined variable can be reduced via the following transformation.

Definition 29 (advanced prefix modification). Advanced prefix modification *(*APM*) is the reversible transformation*

$$\Pi \cdot \psi \xleftrightarrow{\mathsf{APM}} \Omega \cdot \psi ,$$

where $\Pi \cdot \psi$ is a DQBF, $x \in \text{vars}_\exists(\Pi)$ is defined by Z in ψ, and the prefixes satisfy

(a) $\text{vars}_\forall(\Pi) = \text{vars}_\forall(\Omega)$ *and* $\text{vars}_\exists(\Pi) = \text{vars}_\forall(\Omega)$*,*
(b) $S_y^{\Pi} = S_y^{\Omega}$ *for each* $y \in \text{vars}_\exists(\Pi) \setminus \{x\}$*,*
(c) *putting* $Z_\forall := Z \cap \text{vars}_\forall(\Psi)$ *and* $Z_\exists := Z \cap \text{vars}_\exists(\Psi)$*,*

$$\left(Z_\forall \cup \bigcup_{y \in Z_\exists} S_y^{\Pi} \right) \cap S_x^{\Pi} \;\subseteq\; S_x^{\Omega} \;\subseteq\; S_x^{\Pi} .$$

Intuitively, APM preserves truth value for the following reason: in any model for Ψ, the particular function f_x must match any functional definition f for x in ψ. As such, f_x can only depend on variables on which f depends (i.e. the set $Z_\forall \cup \bigcup_{y \in Z_\exists} S_y^{\Pi}$), meanwhile it may not depend on any variable outside of S_x^{Π}. A formal proof follows.

Theorem 30. APM *preserves DQBF truth value.*

preprocessing	$\Pi \cdot \psi \overset{\mathsf{EL4}}{\Longrightarrow} (\Pi \setminus \{\exists x(S_x^\Pi), \exists y(S_y^\Pi)\}) \cup \{\exists x(S_x^\Pi \cap S_y^\Pi)\} \cdot \psi[p/q]$
preconditions	$\mathrm{var}(p) = x \in \mathrm{vars}_\exists(\Pi)$ $\mathrm{var}(q) = y \in \mathrm{vars}_\exists(\Pi)$ p and q are binary-clause equivalent in ψ
$\mathcal{R}$-simulation	$\Pi \cdot \psi$ $\xrightarrow{\mathsf{APM}} (\Pi \setminus \{\exists x(S_x^\Pi)\}) \cup \{\exists x(S_x^\Pi \cap S_y^\Pi)\} \cdot \psi$ $\xrightarrow{\mathsf{AT}^*} (\Pi \setminus \{\exists x(S_x^\Pi)\}) \cup \{\exists x(S_x^\Pi \cap S_y^\Pi)\} \cdot \psi \cup \psi[p/q]$ $\xrightarrow{\mathsf{AT}^*} (\Pi \setminus \{\exists x(S_x^\Pi)\}) \cup \{\exists x(S_x^\Pi \cap S_y^\Pi)\} \cdot \psi[p/q]$ $\xrightarrow{\mathsf{BPM}} (\Pi \setminus \{\exists x(S_x^\Pi), \exists y(S_y^\Pi)\}) \cup \{\exists x(S_x^\Pi \cap S_y^\Pi)\} \cdot \psi[p/q]$

Fig. 6. The $\mathcal{R}'$-simulation of preprocessing using equivalent existential literals.

Proof. Let $\Psi := \Pi \cdot \psi$ and $\Phi := \Omega \cdot \psi$ be DQBFs whose prefixes conform to the definition of APM (Definition 29). We will show that $\Psi \equiv_{\mathrm{tr}} \Phi$.

Suppose on the one hand that Ψ is false. From conditions (a)–(c) of Definition 29, it follows immediately that that $\Phi \leq \Psi$. Therefore Φ is also false.

Suppose on the other hand that Ψ is true, and has a model $f := \{f_y\}_{y \in \mathrm{vars}_\exists(\Pi)}$. Let $h_x : \langle Z \rangle \to \langle \{x\} \rangle$ be some function corresponding to the definition of x in ψ by Z, i.e.

$$\begin{aligned} h_x : \langle Z \rangle &\to \langle \{x\} \rangle \\ \alpha &\mapsto \begin{cases} \{x\} & \text{if } \{x\} \in \psi[\alpha]\,, \\ \{\overline{x}\} & \textit{otherwise}\,. \end{cases} \end{aligned}$$

Further, let $U := Z_\forall \cup \bigcup_{y \in Z_\exists} S_y^\Pi$, and consider the function $h'_x : \langle U \rangle \to \langle \{x\} \rangle$ obtained from h by substituting the existential assignments for the outputs from the model f; that is,

$$\begin{aligned} h'_x : \langle U \rangle &\to \langle \{x\} \rangle \\ \mu &\mapsto h_x\big(\mu \restriction Z_\forall \cup \{f_y(\mu \restriction S_y^\Pi) : y \in Z_\exists\}\big)\,. \end{aligned}$$

We claim that $f_x(\nu \restriction S_x^\Pi) = h'_x(\nu \restriction U)$, for each $\nu \in \langle \mathrm{vars}_\forall(\Pi) \rangle$. It follows immediately that f_x depends on no variable outside of $U \cap S_x^\Pi$. Hence, viewing f_x as a function whose domain is $\langle U \cap S_x^\Pi \rangle$, f becomes a model for Φ. Thus Φ is true.

To prove the claim, let $\nu \in \langle \mathrm{vars}_\forall(\Psi) \rangle$, and consider the CNF

$$\psi' \;:=\; \psi[\nu \cup \{f_y(\nu \restriction S_y^\Pi) : y \in \mathrm{vars}_\exists(\Psi) \setminus \{x\}\}]\,.$$

Now, if $\{x\} \in \psi'$, it is easy to see that we must have $f_x(\nu \restriction S_x^\Pi) = \{x\}$ (by definition of model), and $h'_x(\nu \restriction U) = \{x\}$ (by definition of h_x and h'_x). On the other hand, if $\{x\} \notin \psi'$, we have must have $\{\overline{x}\} \in \psi'$ by Definition 28, and $f_x(\nu \restriction S_x^\Pi) = h'_x(\nu \restriction U) = \{\overline{x}\}$ follows similarly. $\square$

For the remainder of the section, we let $\mathcal{R}' := \mathcal{R} \cup \{\mathsf{APM}\}$ denote $\mathcal{R}$ supplemented by advance prefix modification.

Equivalent literals revisited. Consider binary-equivalent literals p and q in some CNF ψ. In conjunction with AT (via the introduction of $\{p, \overline{q}\}$ and $\{\overline{p}, q\}$) the equivalence always gives

preprocessing	$\begin{aligned}&\Pi \cdot \psi \cup \{C_1 \cup \{x\}, \dots, C_r \cup \{x\}, D_1 \cup \{\overline{x}\}, \dots, D_s \cup \{\overline{x}\}\}\\ &\xRightarrow{\mathsf{EVE2}} \Pi \setminus \{\exists x(S^{\Pi}_x)\} \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}\end{aligned}$
preconditions	$x \in \text{vars}_{\exists}(\Pi)$ is defined in ψ by $Z \subseteq \mathsf{OV}(\Pi, x)$
$\mathcal{R}$-simulation	$\begin{aligned}&\Pi \cdot \psi \cup \{C_1 \cup \{x\}, \dots, C_r \cup \{x\}, D_1 \cup \{\overline{x}\}, \dots, D_s \cup \{\overline{x}\}\}\\ &\xrightarrow{\mathsf{APM}} \big(\Pi \setminus \{\exists x(S^{\Pi}_x)\}\big) \cup \{\exists x(\text{vars}_{\forall}(\Pi))\} \cdot\\ &\qquad \psi \cup \{C_1 \cup \{x\}, \dots, C_r \cup \{x\}, D_1 \cup \{\overline{x}\}, \dots, D_s \cup \{\overline{x}\}\}\\ &\xrightarrow{\mathsf{AT}^*} \big(\Pi \setminus \{\exists x(S^{\Pi}_x)\}\big) \cup \{\exists x(\text{vars}_{\forall}(\Pi))\} \cdot\\ &\qquad \psi \cup \{C_1 \cup \{x\}, \dots, C_r \cup \{x\}, D_1 \cup \{\overline{x}\}, \dots, D_s \cup \{\overline{x}\}\}\\ &\qquad \cup \{C_i \cup D_j : i \in [r], j \in [s]\}\\ &\xrightarrow{\mathsf{DQRAT}^*_{\exists}} \big(\Pi \setminus \{\exists x(S^{\Pi}_x)\}\big) \cup \{\exists x(\text{vars}_{\forall}(\Pi))\} \cdot\\ &\qquad \psi \cup \{D_1 \cup \{\overline{x}\}, \dots, D_s \cup \{\overline{x}\}\} \cup \{C_i \cup D_j : i \in [r], j \in [s]\}\\ &\xrightarrow{\mathsf{DQRAT}^*_{\exists}} \big(\Pi \setminus \{\exists x(S^{\Pi}_x)\}\big) \cup \{\exists x(\text{vars}_{\forall}(\Pi))\} \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}\\ &\xrightarrow{\mathsf{BPM}} \Pi \setminus \{\exists x(S^{\Pi}_x)\} \cdot \psi \cup \{C_i \cup D_j : i \in [r], j \in [s]\}\end{aligned}$

Fig. 7. The $\mathcal{R}'$-simulation of the elimination by resolution of a defined existential variable.

rise to some functional definition $\text{var}(q) \equiv f(\text{var}(p))$, where f is either the identity or Boolean negation. Hence $\text{var}(q)$ is defined by $\{\text{var}(p)\}$ in ψ (and vice versa).

Therefore, in the particular case where $\text{var}(p) = x$ and $\text{var}(q) = y$ are both existential, advanced prefix modification allows either dependency set S^{Π}_x or S^{Π}_y to be replaced with their intersection. Meanwhile, either variable may be eliminated due to the equivalence as in EL3 (Subsection 7.1).

In fact, this describes exactly how HQSPre uses EL4 to handle equivalent existential literals. The formal definition of EL4 and its $\mathcal{R}'$-simulation appear in Figure 6.

Existential variable elimination revisited. We may now cover to the second scenario in which HQSPre implements existential variable elimination via Davis-Putnam resolution, namely EVE2. In this scenario, the eliminated existential variable x is defined in the DQBF matrix *by some subset of* $\mathsf{OV}(\Pi, x)$ – this precondition is vital for soundness, and allows APM to simulate the variable elimination.

Details are given in Figure 7. Notice that APM is applied here *in reverse*: the dependency set for the defined variable *x grows maximally* to $\text{vars}_{\forall}(\Pi)$. To see that this application is valid, we observe that

$$Z \subseteq \mathsf{OV}(\Pi, x) \quad \Longrightarrow \quad Z_{\forall} \cup \bigcup_{y \in Z_{\exists}} S^{\Pi}_y \subseteq S^{\Pi}_x \,,$$

from which it follows that

$$\left(Z_{\forall} \cup \bigcup_{y \in Z_{\exists}} \text{vars}_{\forall}(\Pi) \right) \cap S^{\Pi}_x \;\subseteq\; S^{\Pi}_x \;\subseteq\; \text{vars}_{\forall}(\Pi) \,.$$

Hence APM (applied in the forward direction) would allow a maximal dependency set for x to be reduced to S^{Π}_x.

preprocessing	**preconditions**	**$\mathcal{R}$-simulation**
$\Pi \cdot \psi \cup \{C \cup \{p\}\} \overset{\mathsf{BCE}}{\Longrightarrow} \Pi \cdot \psi \cup \{C\}$	$\mathrm{var}(p) \in \mathrm{vars}_\forall(\Pi)$ for each $D \in \psi$ with $\overline{p} \in D$, $(C \setminus \{p\}) \cup (\mathsf{OC}(\Pi, D, \overline{p}) \setminus \{\overline{p}\})$ is a tautology.	$\Pi \cdot \psi \cup \{C \cup \{p\}\} \xrightarrow{\mathsf{DQRAT}_\forall} \Pi \cdot \psi \cup \{C\}$

Fig. 8. The $\mathcal{R}$-simulation of blocked literal elimination.

Complexity of APM. We do not prove that APM is efficiently checkable - indeed, it presumably isn't, since checking for defined variables (as in Definition 28) is clearly coNP-complete. Nonetheless, given the clear connection between defined variables and satisfiability, an implemented search for defined variables is likely to be simulated by (efficiently-checkable) SAT preprocessing techniques. Hence, an appropriate formulation of 'efficiently checkable variable definitions' based on current implementations can presumably be substituted. We choose to leave Definition 28 as is, such that Theorem 30 describes the maximal scope of APM.

7.3 Further $\mathcal{R}$-enabled DQBF preprocessing techniques

The alert reader may have noticed that $\mathsf{DQRAT}_\forall$ did not feature whatsoever in the foregoing preprocessing simulations, nor did it feature in the simulation of ∀Exp+Res, so DQRAT remains a complete proof system without it. It is therefore reasonable to enquire as to its purpose.

In fact, $\mathsf{DQRAT}_\forall$ fosters the DQBF generalisation of a QBF preprocessing concept known as *blocked literals* [37]. To the best of our knowledge, blocked literals (as apposed to blocked clauses) have not been considered in the DQBF context, presumably because soundness of blocked literal elimination (BLE) and addition (BLA) for DQBF (based on a natural generalisation of outer variables for a universal) had not been established prior to this paper. Since BLE has a clear positive impact on QBF preprocessing and solver performance [37], it is natural to conjecture that a DQBF preprocessor such as HQSPre would benefit from its inclusion.

Figure 8 details the natural DQBF generalisation of BLE. Informally, a universal literal is *blocked* when it satisfies the same criterion as for a *blocking* existential literal in a blocked clause (BCE, Subsection 7.1), namely all 'outer resolvents' are tautologous.[9] In fact, a universal literal p is blocked in a clause C only if C has $\mathsf{DQRAT}_\forall$ on p. As such, blocked literals are a special case of $\mathsf{DQRAT}_\forall$, which in turn is a special case of $\mathsf{DQIOR}_\forall$. Thus the soundness of BLE can be considered a corollary of Theorem 7.

Since BLE is simulated by reversible $\mathcal{R}$ transformations, the reverse procedure BLA is also truth-value preserving. While the impact of BLA in QBF preprocessing is not comparable with that of BLE, it can serve to reduce the blow-up due to universal expansion [37]. For the same reason it may also prove useful in the DQBF setting.

8 Conclusions

The award of the infamous 'prepended D' is a high point in the career of any QBF concept. In this paper, we showed that QRAT is eligible for the honour. In other words, the natural DQBF generalisation DQRAT is a sound and complete proof system for false DQBFs. With

[9] There is a nomenclative distinction between *blocked* (universal) and *blocking* (existential) literals [37].

the addition of an advanced form of prefix modification, it is able to simulate the full suite of state-of-the-art DQBF preprocessing techniques.

We also showed that blocked literal elimination (BLE) is applicable to DQBFs. Empirical studies have shown that BLE is beneficial in QBF preprocessing [38]. We suggest that its impact on DQBF preprocessing (HQSPre in particular) deserves a similar empirical investigation.

Our work highlights several related open problems:

- We showed that DQRAT simulates $\forall$Exp+Res, but does it simulate the stronger expansion-based proof system IR-calc? This question is still unsolved even on the QBF fragment.
- Since NP $\neq$ NEXPTIME by the Non-deterministic Time Hierarchy Theorem [22], refutational DQBF proof systems cannot be polynomially bounded. Hence DQRAT is not polynomially bounded – but can we find *concrete* DQBF families which require large refutations? Such DQBF families would be practically relevant, since they define hard problems for any solver simulated by DQRAT.
- In the QBF setting, QRAT forms not only a proof system for false QBFs, but also a dual system for true QBFs which supports the efficient extraction of Skolem function models [36]. The analogous situation for DQRAT and true DQBFs remains unclear.

Acknowledgements. I would like to thank Tomáš Peitl, David Sherratt and Friedrich Slivovsky for inspiring discussions at Dagstuhl seminar 20061 "*SAT and Interactions*", February 2020.

References

1. Arora, S., Barak, B.: Computational Complexity - A Modern Approach. Cambridge University Press (2009)
2. Ayari, A., Basin, D.A.: QUBOS: deciding quantified Boolean logic using propositional satisfiability solvers. In: Aagaard, M., O'Leary, J.W. (eds.) Conference on Formal Methods in Computer-aided Design (FMCAD). Logical Methods in Computer Science, vol. 2517, pp. 187–201. Springer (2002)
3. Azhar, S., Peterson, G., Reif, J.: Lower bounds for multiplayer non-cooperative games of incomplete information. Journal of Computers and Mathematics with Applications **41**, 957 – 992 (2001)
4. Balabanov, V., Chiang, H.K., Jiang, J.R.: Henkin quantifiers and Boolean formulae: A certification perspective of DQBF. Theoretical Computer Science **523**, 86–100 (2014)
5. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Sinz, C., Egly, U. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 8561, pp. 154–169. Springer (2014)
6. Beyerdorff, O., Blinkhorn, J., Chew, L., Schmidt, R., Suda, M.: Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. Journal of Automated Reasoning **63**(3), 597–623 (2019)
7. Beyersdorff, O., Blinkhorn, J., Chew, L., Schmidt, R.A., Suda, M.: Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. Journal of Automated Reasoning **63**(3), 597–623 (2019)
8. Beyersdorff, O., Blinkhorn, J., Mahajan, M.: Building strategies into QBF proofs. Journal of Automated Reasoning (2020). https://doi.org/10.1007/s10817-020-09560-1
9. Beyersdorff, O., Blinkhorn, J., Peitl, T.: Strong (D)QBF dependency schemes via tautology-free resolution paths. In: International Conference on Theory and Practice of Satisfiability Testing (SAT) (2020), in press.
10. Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. ACM Transactions on Computation Theory **11**(4), 26:1–26:42 (2019)
11. Beyersdorff, O., Chew, L., Schmidt, R.A., Suda, M.: Lifting QBF resolution calculi to DQBF. In: Creignou, N., Berre, D.L. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 9710, pp. 490–499. Springer (2016)
12. Beyersdorff, O., Creignou, N., Egly, U., Vollmer, H.: SAT and interactions (dagstuhl seminar 16381). Dagstuhl Reports **6**(9), 74–93 (2016)
13. Beyersdorff, O., Egly, U., Mahajan, M., Nalon, C.: SAT and interactions (dagstuhl seminar 20061). Dagstuhl Reports **6**(9), 74–93 (2020), in press.
14. Biere, A.: Resolve and expand. Lecture Notes in Computer Science, vol. 3542, pp. 59–70. Springer (2004)
15. Biere, A., Lonsing, F., Seidl, M.: Blocked clause elimination for QBF. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) International Conference on Automated Deduction (CADE). Lecture Notes in Computer Science, vol. 6803, pp. 101–115. Springer (2011)

16. Blinkhorn, J.: Quantified Boolean Formulas: Proof Complexity and Models of Solving. Ph.D. thesis, University of Leeds (2019)
17. Bloem, R., Könighofer, R., Seidl, M.: SAT-based synthesis methods for safety specs. In: McMillan, K.L., Rival, X. (eds.) International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI). Lecture Notes in Computer Science, vol. 8318, pp. 1–20. Springer (2014)
18. Chew, L., Clymo, J.: The equivalences of refutational QRAT. In: International Conference on Theory and Practice of Satisfiability Testing (SAT). pp. 100–116 (2019)
19. Chew, L., Clymo, J.: How QBF expansion makes strategy extraction hard. International Joint Conference on Automated Reasoning (IJCAR) (2020), in press.
20. Chew, L.N.: QBF proof complexity. Ph.D. thesis, University of Leeds, UK (2017)
21. Cook, S.A.: The complexity of theorem-proving procedures. In: Harrison, M.A., Banerji, R.B., Ullman, J.D. (eds.) ACM Symposium on Theory of Computing (STOC). pp. 151–158. ACM (1971)
22. Cook, S.A.: A hierarchy for nondeterministic time complexity. In: Fischer, P.C., Zeiger, H.P., Ullman, J.D., Rosenberg, A.L. (eds.) ACM Symposium on Theory of Computing (STOC). pp. 187–192. ACM (1972)
23. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. Journal of Symbolic Logic **44**(1), 36–50 (1979)
24. Davis, M., Putnam, H.: A computing procedure for quantification theory. Journal of the ACM **7**(3), 201–215 (1960)
25. Faymonville, P., Finkbeiner, B., Rabe, M.N., Tentrup, L.: Encodings of bounded synthesis. In: Legay, A., Margaria, T. (eds.) International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Lecture Notes in Computer Science, vol. 10205, pp. 354–370. Springer (2017)
26. Finkbeiner, B., Tentrup, L.: Fast DQBF refutation. In: Sinz, C., Egly, U. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 8561, pp. 243–251. Springer (2014)
27. Fröhlich, A., Kovásznai, G., Biere, A.: A DPLL algorithm for solving DQBF (2012)
28. Fröhlich, A., Kovásznai, G., Biere, A., Veith, H.: iDQ: Instantiation-based DQBF solving. In: Berre, D.L. (ed.) Workshop on Pragmatics of SAT (POS). EPiC Series in Computing, vol. 27, pp. 103–116. EasyChair (2014)
29. Gelder, A.V.: Variable independence and resolution paths for quantified Boolean formulas. In: Lee, J.H. (ed.) International Conference on Principles and Practice of Constraint Programming (CP). Lecture Notes in Computer Science, vol. 6876, pp. 789–803. Springer (2011)
30. Gitina, K., Reimer, S., Sauer, M., Wimmer, R., Scholl, C., Becker, B.: Equivalence checking of partial designs using dependency quantified Boolean formulae. In: International Conference on Computer Design (ICCD). pp. 396–403. IEEE Computer Society (2013)
31. Gitina, K., Wimmer, R., Reimer, S., Sauer, M., Scholl, C., Becker, B.: Solving DQBF through quantifier elimination. In: Nebel, W., Atienza, D. (eds.) Design, Automation & Test in Europe Conference (DATE). pp. 1617–1622. ACM (2015)
32. Henkin, L.: Some remarks on infinitely long formulas. Infinitistic Methods pp. 167 – 183 (1961)
33. Heule, M., Järvisalo, M., Biere, A.: Clause elimination procedures for CNF formulas. In: Fermüller, C.G., Voronkov, A. (eds.) International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR). Lecture Notes in Computer Science, vol. 6397, pp. 357–371. Springer (2010)
34. Heule, M., Järvisalo, M., Biere, A.: Covered clause elimination. In: Voronkov, A., Sutcliffe, G., Baaz, M., Fermüller, C.G. (eds.) International Conference on Logic for Programming, Artificial Intelligence and Reasoning - Short Papers (LPAR). EPiC Series in Computing, vol. 13, pp. 41–46. EasyChair (2010)
35. Heule, M., Järvisalo, M., Lonsing, F., Seidl, M., Biere, A.: Clause elimination for SAT and QSAT. Journal of Artificial Intelligence Research **53**, 127–168 (2015)
36. Heule, M., Seidl, M., Biere, A.: Efficient extraction of skolem functions from QRAT proofs. In: Conference on Formal Methods in Computer-aided Design (FMCAD). pp. 107–114. IEEE (2014)
37. Heule, M., Seidl, M., Biere, A.: Blocked literals are universal. In: Havelund, K., Holzmann, G.J., Joshi, R. (eds.) NASA International Symposium on Formal Methods (NFM). Lecture Notes in Computer Science, vol. 9058, pp. 436–442. Springer (2015)
38. Heule, M.J.H., Seidl, M., Biere, A.: Solution validation and extraction for QBF preprocessing. Journal of Automated Reasoning **58**(1), 97–125 (2017)
39. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. Artificial Intelligence **234**, 1–25 (2016)
40. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. Theoretical Computer Science **577**, 25–42 (2015)
41. Järvisalo, M., Biere, A., Heule, M.: Blocked clause elimination. In: Esparza, J., Majumdar, R. (eds.) International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Lecture Notes in Computer Science, vol. 6015, pp. 129–144. Springer (2010)
42. Järvisalo, M., Heule, M., Biere, A.: Inprocessing rules. In: Gramlich, B., Miller, D., Sattler, U. (eds.) International Joint Conference on Automated Reasoning (IJCAR). Lecture Notes in Computer Science, vol. 7364, pp. 355–370. Springer (2012)

43. Kiesl, B., Heule, M.J.H., Seidl, M.: A little blocked literal goes a long way. In: Gaspers, S., Walsh, T. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 10491, pp. 281–297. Springer (2017)
44. Kiesl, B., Seidl, M.: QRAT polynomially simulates $\forall\mathsf{Exp}+\mathsf{Res}$. In: Janota, M., Lynce, I. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 11628, pp. 193–202. Springer (2019)
45. Kullmann, O.: On a generalization of extended resolution. Discrete Applied Mathematics **96-97**, 149–176 (1999)
46. Lonsing, F., Biere, A.: DepQBF: A dependency-aware QBF solver. Journal of Satisfiability, Boolean Modeling and Computation **7**(2-3), 71–76 (2010)
47. Lonsing, F., Egly, U.: QRAT+: generalizing QRAT by a more powerful QBF redundancy property. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) International Joint Conference on Automated Reasoning (IJCAR). Lecture Notes in Computer Science, vol. 10900, pp. 161–177. Springer (2018)
48. Lonsing, F., Egly, U.: QRATPre+: Effective QBF preprocessing via strong redundancy properties. In: Janota, M., Lynce, I. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 11628, pp. 203–210. Springer (2019)
49. Peitl, T., Slivovsky, F., Szeider, S.: Dependency learning for QBF. Journal of Artificial Intelligence Research **65**, 180–208 (2019)
50. Peterson, G.L., Reif, J.H.: Multiple-person alternation. In: Symposium on Foundations of Computer Science (FOCS). pp. 348–363. IEEE Computer Society (1979)
51. Pulina, L., Seidl, M.: The 2016 and 2017 QBF solvers evaluations (QBFEVAL'16 and QBFEVAL'17). Artificial Intelligence **274**, 224–248 (2019)
52. Rabe, M.N., Tentrup, L.: CAQE: A certifying QBF solver. In: Kaivola, R., Wahl, T. (eds.) Conference on Formal Methods in Computer-aided Design (FMCAD). pp. 136–143. IEEE (2015)
53. Rabe, M.N.: A resolution-style proof system for DQBF. In: Gaspers, S., Walsh, T. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 10491, pp. 314–325. Springer (2017)
54. Samer, M., Szeider, S.: Backdoor sets of quantified Boolean formulas. Journal of Automated Reasoning **42**(1), 77–97 (2009)
55. Samulowitz, H., Davies, J., Bacchus, F.: Preprocessing QBF. In: Benhamou, F. (ed.) International Conference on Principles and Practice of Constraint Programming (CP). Lecture Notes in Computer Science, vol. 4204, pp. 514–529. Springer (2006)
56. Scholl, C., Jiang, J.R., Wimmer, R., Ge-Ernst, A.: A PSPACE subclass of dependency quantified Boolean formulas and its effective solving. In: National Conference on Artificial Intelligence (AAAI). pp. 1584–1591. AAAI Press (2019)
57. Scholl, C., Wimmer, R.: Dependency quantified boolean formulas: An overview of solution methods and applications - extended abstract. In: Beyersdorff, O., Wintersteiger, C.M. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 10929, pp. 3–16. Springer (2018)
58. Silva, J.P.M., Sakallah, K.A.: GRASP - a new search algorithm for satisfiability. In: Rutenbar, R.A., Otten, R.H.J.M. (eds.) International Conference on Computer-Aided Design (ICCAD). pp. 220–227. IEEE Computer Society / ACM (1996)
59. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. Theoretical Computer Science **612**, 83–101 (2016)
60. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time: Preliminary report. In: Aho, A.V., Borodin, A., Constable, R.L., Floyd, R.W., Harrison, M.A., Karp, R.M., Strong, H.R. (eds.) ACM Symposium on Theory of Computing (STOC). pp. 1–9. ACM (1973)
61. Tentrup, L., Rabe, M.N.: Clausal abstraction for DQBF. In: Janota, M., Lynce, I. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 11628, pp. 388–405. Springer (2019)
62. Tseitin, G.S.: On the complexity of derivation in propositional calculus. Studies in Constructive Mathematics and Mathematical Logic, Part 2 pp. 115–125 (1968)
63. Vardi, M.Y.: Boolean satisfiability: Theory and engineering. Communications of the ACM **57**(3), 5 (2014)
64. Wetzler, N., Heule, M., Jr., W.A.H.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: Sinz, C., Egly, U. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 8561, pp. 422–429. Springer (2014)
65. Wimmer, R., Gitina, K., Nist, J., Scholl, C., Becker, B.: Preprocessing for DQBF. In: International Conference on Theory and Practice of Satisfiability Testing (SAT). pp. 173–190 (2015)
66. Wimmer, R., Scholl, C., Wimmer, K., Becker, B.: Dependency schemes for DQBF (AVACS technical report no. 110) (2014)
67. Wimmer, R., Scholl, C., Wimmer, K., Becker, B.: Dependency schemes for DQBF. In: Creignou, N., Berre, D.L. (eds.) International Conference on Theory and Practice of Satisfiability Testing (SAT). Lecture Notes in Computer Science, vol. 9710, pp. 473–489. Springer (2016)

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il