QRAT Polynomially Simulates ∀-Exp+Res*

Benjamin Kiesl[1,2] and Martina Seidl[3]

[1] Institute of Logic and Computation, TU Wien
[2] CISPA Helmholtz Center for Information Security
[3] Institute for Formal Models and Verification, JKU Linz

Abstract. The proof system ∀-Exp+Res formally captures expansion-based solving of quantified Boolean formulas (QBFs) whereas the QRAT proof system captures QBF preprocessing. From previous work it is known that certain families of formulas have short proofs in QRAT but not in ∀-Exp+Res. However, it was not known if the two proof systems were incomparable (i.e., if there also existed QBFs with short ∀-Exp+Res proofs but without short QRAT proofs), or if QRAT polynomially simulates ∀-Exp+Res. We close this gap of the QBF-proof-complexity landscape by presenting a polynomial simulation of ∀-Exp+Res in QRAT. Our simulation shows how definition introduction combined with extended-universal reduction can mimic the concept of universal expansion.

1 Introduction

Proof systems for quantified Boolean formulas (QBFs) have been extensively studied to obtain a better understanding of the strengths and limitations of different QBF-solving approaches (e.g., [5, 10, 8, 3, 13, 25]). Much is known about instantiation-based proof systems [16, 4, 5], which provide the foundation for expansion-based solvers [15, 7], and about Q-resolution systems [13, 20, 28, 1, 26, 2, 6, 14, 23], which provide the foundation for search-based solvers [21, 22]. There is, however, one other practically useful proof system that is quite different from the aforementioned ones and whose exact place in the complexity landscape is still unclear: the QRAT proof system [12].

The QRAT proof system is a generalization of DRAT [27] (the de-facto standard for proofs in practical SAT solving) that has its strengths when it comes to preprocessing: Many QBF solvers benefit from preprocessing techniques to simplify a QBF before they actually evaluate its truth. With the QRAT system, it is possible to certify the correctness of virtually all preprocessing simplifications performed by state-of-the-art QBF solvers and preprocessors. Recently, it has been shown that QRAT can polynomially simulate the long-distance-resolution calculus, a strictly stronger extension of the Q-Resolution calculus [18]. So far, however, it has not been known if QRAT can also polynomially simulate the instantiation-based calculus ∀-Exp+Res [16]. In this short paper, we show that this is indeed the case by providing a simulation whose resulting QRAT proof is only linear in the size of the original ∀-Exp+Res proof.

* This work has been supported by the Austrian Science Fund (FWF) projects W1255-N23 and S11408-N23 and the LIT AI Lab funded by the State of Upper Austria.

2 Preliminaries

We consider *quantified Boolean formulas* in *prenex conjunctive normal form* (PCNF), which are of the form $\mathcal{Q}.\psi$, where $\mathcal{Q}$ is a *quantifier prefix* and ψ, called the *matrix* of the QBF, is a propositional formula in conjunctive normal form (CNF); we define propositional formulas and quantifier prefixes in the following.

Propositional formulas in CNF are built from variables and logical operators as follows. A *literal* is either a variable x (a *positive literal*) or the negation $\bar{x}$ of a variable x (a *negative literal*). The *complement* $\bar{l}$ of a literal l is defined as $\bar{l} = \bar{x}$ if $l = x$ and $\bar{l} = x$ if $l = \bar{x}$. A *clause* is a finite disjunction of the form $(l_1 \vee \cdots \vee l_n)$ where $l_1, \dots, l_n$ are literals. We denote the empty clause by $\bot$. A clause with exactly one literal is a *unit clause*. A *formula* is a finite conjunction of the form $C_1 \wedge \cdots \wedge C_m$ where $C_1, \dots, C_m$ are clauses. Clauses can be viewed as sets of literals, and formulas can be viewed as sets of clauses. For an expression (i.e., a literal, formula, etc.) E, we denote the set of variables occurring in E by $var(E)$. If $var(E)$ is a singleton set, we sometimes treat it like a variable.

A *quantifier prefix* has the form $\mathcal{Q}_1 X_1 \dots \mathcal{Q}_q X_q$ where all the X_i are mutually disjoint sets of variables, $\mathcal{Q}_i \in \{\forall, \exists\}$, and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$. The quantifier of a literal l is $\mathcal{Q}_i$ if $var(l) \in X_i$. Given a literal l with quantifier $\mathcal{Q}_i$ and a literal k with quantifier $\mathcal{Q}_j$, we write $l \leq_{\mathcal{Q}} k$ if $i \leq j$, and $l <_{\mathcal{Q}} k$ if $i < j$. We sometimes write $l \leq k$ instead of $l \leq_{\mathcal{Q}} k$, and we write $l < k$ instead of $l <_{\mathcal{Q}} k$ if $\mathcal{Q}$ is clear from the context. If $l \leq k$, we say that l occurs *left of* k.

Given a literal l and a propositional formula ψ, we define $\psi[l]$ to be the formula obtained from ψ by first removing all clauses that contain l and then removing $\bar{l}$ from all remaining clauses. The result of applying the *unit-clause rule* to ψ is the formula $\psi[l]$ where (l) is a unit clause in F. The iterated application of the unit-clause rule, until either the empty clause is derived or no unit clauses are left, is called *unit propagation*. In case unit propagation derives the empty clause, we say that unit propagation derived a *conflict* on ψ.

Given a propositional formula ψ and a clause $(l_1 \vee \cdots \vee l_k)$, we say that ψ implies $(l_1 \vee \cdots \vee l_k)$ via unit propagation—denoted by $\psi \vdash_1 (l_1 \vee \cdots \vee l_k)$—if unit propagation derives a conflict on $\psi \wedge (\bar{l}_1) \wedge \cdots \wedge (\bar{l}_k)$. For example, the formula $(\bar{x} \vee z) \wedge (\bar{y} \vee \bar{z})$ implies the clause $(\bar{x} \vee \bar{y})$ via unit propagation since unit propagation derives a conflict on $(\bar{x} \vee z) \wedge (\bar{y} \vee \bar{z}) \wedge (x) \wedge (y)$.

A QBF $\exists x \mathcal{Q}.\psi$ is true if at least one of $\mathcal{Q}.\psi[x]$ and $\mathcal{Q}.\psi[\bar{x}]$ is true, otherwise it is false. Respectively, a QBF $\forall x \mathcal{Q}.\psi$ is true if both $\mathcal{Q}.\psi[x]$ and $\mathcal{Q}.\psi[\bar{x}]$ are true, otherwise it is false. If the matrix ψ of a QBF $\mathcal{Q}.\psi$ is the empty formula, then $\mathcal{Q}.\psi$ is true. If ϕ contains the empty clause, then $\mathcal{Q}.\psi$ is false.

An *assignment* is a function from variables to the truth values 1 (*true*) and 0 (*false*). We denote assignments by the sequences of literals they satisfy. E.g., $x\bar{y}$ denotes the assignment that assigns 1 to x and 0 to y.

Finally, for the formal definition of polynomial simulations between proof systems we refer to Cook and Reckhow [9]. An informal summary is this: A proof system f polynomially simulates a proof system g if there exists a polynomial-time procedure that transforms g-proofs into f-proofs.

3 The QRAT Proof System

Here, we introduce the basics of the QRAT proof system [12]. The two main concepts behind QRAT are QRAT *literals* and universal reduction via the *reflexive-resolution-path dependency scheme* [11].

The definition of QRAT literals is based on the notion of an *outer resolvent.* Given two clauses $C \lor l, D \lor \bar{l}$ of a QBF $\mathcal{Q}.\psi$, the outer resolvent $C \lor l \bowtie^{l}_{\mathcal{Q}} D \lor \bar{l}$ of $C \lor l$ with $D \lor \bar{l}$ upon l is the clause consisting of all literals in C together with those literals of D that occur left of l, i.e., the clause $C \cup \{k \mid k \in D \text{ and } k \leq_{\mathcal{Q}} l\}$. If all outer resolvents upon a literal are implied via unit propagation, then that literal is a QRAT *literal* [12]:

Definition 1. *A literal l is a* QRAT literal *in a clause $C \lor l$ with respect to a QBF $\mathcal{Q}.\psi$ if, for every clause $D \lor \bar{l} \in \psi \setminus \{C \lor l\}$, it holds that $\psi \vdash_1 C \lor l \bowtie^{l}_{\mathcal{Q}} D \lor \bar{l}$.*

Example 1. Let $C = (b \lor x \lor y)$ and let $\phi = \exists ab \forall x \mathsf{Q} y \exists c.(\bar{b} \lor \bar{y} \lor c) \land (a \lor \bar{y} \lor c) \land (a \lor b \lor x)$, where $\mathsf{Q} \in \{\exists, \forall\}$. The literal y is a QRAT literal in C with respect to ϕ since there are two outer resolvents: the tautology $(b \lor \bar{b} \lor x)$, obtained by resolving with $(\bar{b} \lor \bar{y} \lor c)$, and the clause $(a \lor b \lor x)$, obtained by resolving with $(a \lor \bar{y} \lor c)$. The matrix of ϕ implies both outer resolvents via unit propagation.

Let $\phi = \mathcal{Q}.\psi$ be a QBF. If a universal literal u is a QRAT literal in a clause $C \in \psi$, the removal of u from C is called QRAT-*literal elimination.* If, after adding a universal literal u to a clause $C \in \psi$, u becomes a QRAT literal, then this addition is called QRAT-*literal addition.* If a clause contains an existential QRAT literal, it is called a QRAT clause (or simply a QRAT) with respect to ϕ; its addition to a QBF is called QRAT *addition* and its removal is called QRAT *elimination.* It can be shown that QRAT-literal addition and elimination as well as QRAT-clause addition and elimination preserve the truth value of a QBF.

The introduction of definition clauses of the form $(\bar{x} \lor y), (x \lor \bar{y})$ (where x is a fresh variable not occurring in ϕ), is an instance of QRAT addition if we put x into the same quantifier block as y: $(\bar{x} \lor y)$ is a QRAT since x is fresh and thus there are no outer resolvents upon $\bar{x}$; $(x \lor \bar{y})$ is then a QRAT since the only outer resolvent upon x is the tautology $(\bar{y} \lor y)$, obtained by resolving with $(\bar{x} \lor y)$.

The reflexive-resolution-path dependency scheme (short, $\mathcal{D}^{\text{rrs}}$) is based on the notion of a *resolution path* [11]. Intuitively, a QBF contains a resolution path between a universal literal u and an existential literal e if we can start with a clause that contains u and perform a number of resolution steps over existential literals that occur right of u to obtain a clause that contains both u and e. An example of a resolution path is given in Fig. 1.

$$\underbrace{u \lor \cdots \lor e_1}_{C_1} \quad \longrightarrow \quad \underbrace{\bar{e}_1 \lor \cdots \lor e_2}_{C_2} \quad \longrightarrow \quad \underbrace{\bar{e}_2 \lor \cdots \lor e_3}_{C_3} \quad \longrightarrow \quad \underbrace{\bar{e}_3 \lor \cdots \lor e_4}_{C_4}$$

Fig. 1. A resolution path from u to e_4.

Definition 2. *Given a QBF $\phi = \mathcal{Q}.\psi$, a universal literal u, and an existential literal e_n, ϕ contains a* resolution path *from u to e_n if there exists a sequence $C_1, \ldots, C_n$ of clauses and a sequence $e_1, \ldots, e_{n-1}$ of existential literals such that*

(1) $u \in C_1$ *and* $e_n \in C_n$,
(2) $e_1, \ldots, e_n$ *occur right of* u,
(3) $e_i \in C_i$, $\bar{e}_i \in C_{i+1}$, *for* $i \in 1, \ldots, n-1$, *and*
(4) $var(e_i) \neq var(e_{i+1})$ *for* $i \in 1, \ldots, n-1$.

The reflexive-resolution-path dependency scheme defines that a literal e *depends* on a literal u if and only if e is existential, u is universal, and at least one of the following conditions holds: (1) There exist resolution paths from u to e and from $\bar{u}$ to $\bar{e}$. (2) There exist resolution paths from u to $\bar{e}$ and from $\bar{u}$ to e.

Next we define the QRAT proof system. In the QRAT proof system, a *derivation* for a QBF $\phi = \mathcal{Q}.\psi$ is a sequence $M_1, \ldots, M_n$ of proof steps. Starting with $\phi_0 = \phi$, every M_i modifies ϕ_{i-1} in one of the following five ways, which results in a new formula $\phi_i = \mathcal{Q}_i.\psi_i$, which we call the *accumulated formula* at step i:

(1) Add a clause that is implied by ψ_{i-1} via unit propagation.
(2) Add a clause that is a QRAT clause with respect to ϕ_{i-1}.
(3) Remove an arbitrary clause from ϕ_{i-1}.
(4) Remove a QRAT literal from a clause in ϕ_{i-1}.
(5) Remove a universal literal u from a clause $C \vee u \in \phi_{i-1}$ where all $l \in C$ are independent of u according to $\mathcal{D}^{\text{rrs}}$ ("extended universal reduction").

A QRAT derivation $M_1, \ldots, M_n$ thus derives new formulas $\phi_1, \ldots, \phi_n$ from ϕ. If the final formula ϕ_n contains $\perp$, then the derivation is a (*refutation*) *proof* of ϕ. To simplify the presentation, we do not specify in detail how the modification steps M_i are represented syntactically, but it should be clear that their size needs to be at most linear with respect to the involved clauses and literals. Note that certain proof steps can modify the quantifier prefix.

4 The ∀-Exp+Res Proof System

A ∀-Exp+Res proof of a QBF $\phi = \mathcal{Q}.\psi$ is a sequence $C_1, \ldots, C_n$ of clauses where each clause is obtained either via the *axiom rule* or the *resolution rule*. The axiom rule is as follows:

$$\frac{C}{\{l^{\tau_l} \mid l \in C, l \text{ is existential}\}}\ (\text{Ax})$$

Here, C is a clause of ψ, τ is an assignment that falsifies all universal literals of C, and τ_l denotes the assignment τ restricted to the universal variables u with $u < l$. Intuitively, τ_l can be seen as an annotation of the literal l. For example, the axiom rule allows us to use the assignment $\tau = u\bar{v}$ for deriving the clause $x^u \vee \bar{y}^{u\bar{v}}$ from the formula $\forall u \exists x \forall v \exists y.(\bar{u} \vee x \vee v \vee \bar{y})$. The resolution rule of ∀-Exp+Res is just the usual resolution rule from propositional logic—it derives a new clause C_k from two earlier clauses C_i, C_j with $i, j < k$:

$$\frac{C \vee l^\tau \qquad D \vee \bar{l}^\tau}{C \vee D} \text{ (Res)}$$

We next illustrate the intuition behind the simulation of ∀-Exp+Res by QRAT.

5 Simulating ∀-Exp+Res by QRAT: Intuition

To simulate ∀-Exp+Res by QRAT, we need to find a way to simulate applications of the axiom rule. Intuitively, the axiom rule introduces multiple instantiations of a single existential variable because, in satisfying assignments of the formula, this variable might take different truth values depending on the truth values of the universal variables that occur left of it. We can introduce these instantiations in QRAT by first adding definitions of the new variables and then eliminating the superfluous universal variables with extended-universal reduction. Once this is done, we can just straightforwardly perform the remaining resolution steps in QRAT since resolvents are implied via unit propagation. Assume a ∀-Exp+Res proof uses the axiom rule as follows, where the quantifier prefix is $\forall u \exists x \forall v \exists y$:

$$\frac{\bar{u} \vee x \vee v \vee y}{x^u \vee y^{u\bar{v}}} \text{ (Ax)}$$

We simulate the derivation of $(x^u \vee y^{u\bar{v}})$ in QRAT as follows:

(1) Add definitions for the new variables x^u and $y^{u\bar{v}}$, where x^u goes to the same quantifier block as x and $y^{u\bar{v}}$ goes to the same quantifier block as y. The new clauses are $(\bar{x} \vee x^u), (x \vee \bar{x}^u), (\bar{y} \vee y^{u\bar{v}}), (y \vee \bar{y}^{u\bar{v}})$.
(2) Add a clause that is similar to the original clause $(\bar{u} \vee x \vee v \vee y)$, with the only difference that we now use the new annotated variables instead of the original ones. Observe that we can resolve $(\bar{x} \vee x^u)$ with $(\bar{u} \vee x \vee v \vee y)$ to replace x by x^u; likewise for y and $y^{u\bar{v}}$. Because of this, $(\bar{u} \vee x \vee v \vee y)$ and the definition clauses together imply the new clause, $(\bar{u} \vee x^u \vee v \vee y^{u\bar{v}})$, via unit propagation.
(3) Eliminate $(\bar{u} \vee x \vee v \vee y)$ and the definition clauses introduced in step 1.
(4) Eliminate the universal literals $\bar{u}$ and v from $(\bar{u} \vee x^u \vee v \vee y^{u\bar{v}})$ by extended universal reduction, resulting in the clause $(x^u \vee y^{u\bar{v}})$.

The correctness of the fourth step is a consequence of Lemma 1, which we prove in the next section, where we define our simulation.

6 Simulating ∀-Exp+Res by QRAT

We start with a QBF $\mathcal{Q}.\psi$ and a ∀-Exp+Res proof π of $\mathcal{Q}.\psi$. We then construct a QRAT proof Π of $\mathcal{Q}.\psi$ as follows:

Step 1 (Introduction of Definitions): For each annotated variable x^τ in the ∀-Exp+Res proof π, we introduce a definition of the form $(\bar{x} \vee x^\tau), (x \vee \bar{x}^\tau)$. We also put x^τ into the same quantifier block as x. Note that each annotated variable must have been obtained by an application of the axiom rule. The definition introductions are QRAT additions, as explained on page 3. We denote the resulting accumulated formula by $\mathcal{Q}'.\psi_1$.

Step 2 (Introduction of Annotated Clauses): For each clause $C^\tau \in \pi$ that was obtained from a clause $C \in \psi$ by applying the axiom rule with the assignment τ, we add the clause $C^\tau \vee \bar{u}_1 \vee \cdots \vee \bar{u}_k$. Since C and the definitions of the annotated literals of C^τ are in ψ_1, the clause $C^\tau \vee \bar{u}_1 \vee \cdots \vee \bar{u}_k$ is implied via unit propagation and thus it can be added as a QRAT. We denote the accumulated formula after performing all these QRAT additions by $\mathcal{Q}'.\psi_2$

Step 3 (Elimination of Input Clauses and Definitions): We now eliminate all clauses of ψ as well as the definitions introduced in step 1 since we don't need them anymore. Note that QRAT allows the elimination of arbitrary clauses. We thus obtain the accumulated formula $\mathcal{Q}'.\psi_3$ with $\psi_3 := \psi_2 \setminus \psi_1$.

Step 4 (Removal of Universal Literals): We now remove all universal literals from the clauses in ψ_3. We start by removing the occurrences of the right-most variable u and apply extended universal reduction on all clauses in which it occurs. Once u is eliminated, we move on to the new right-most variable and eliminate it. We also remove eliminated variables from the quantifier prefix. We repeat this for all universal literals and denote the resulting accumulated formula by $\mathcal{Q}''.\psi_4$. It remains to show that all the removal steps are valid extended-universal-reduction steps. This is a consequence of the following lemma:

Lemma 1. *If $\mathcal{Q}'.\psi_3$ contains a resolution path from u to e, then e must be an annotated literal of the form l^τ where the assignment τ falsifies u.*

Proof. Suppose there exists a resolution path $C_1, \ldots, C_n$ from u to e. We show by induction on n that e is of the form l^τ where τ falsifies u.

Base Case ($n = 1$): C_1 contains both u and e. Hence, all the existential literals of C_1 must have been obtained by instantiating with an assignment that falsifies all universal literals of C_1. Moreover, by the definition of resolution paths, e must occur right of u. Hence, e must be of the form l^τ where τ falsifies u.

Induction Step ($n > 1$): Since $C_1, \ldots, C_n$ is a resolution path from u to e, we know that $e \in C_n$ and that $C_1, \ldots, C_{n-1}$ is a resolution path from u to some literal e_{n-1} such that $e_{n-1} \in C_{n-1}$ and $\bar{e}_{n-1} \in C_n$. By the induction hypothesis, e_{n-1} is of the form l^τ_{n-1} where τ falsifies u. But then, since $\bar{e}_{n-1} \in C_n$, we know that C_n must have been obtained by instantiating it with an assignment that falsifies u. It follows that e is of the form l^τ where τ falsifies u. ☐

Thus, whenever we eliminate a universal literal u from a clause C in step 4, then $\mathcal{D}^{\text{rrs}}$ defines each existential literal $e \in C$ that occurs right of u to be independent of u (literals to the left of u are trivially independent of u): Since $e \in C$, we know that e is of the form l^τ where τ falsifies u. Thus, there cannot exist resolution paths from $\bar{u}$ to e or to $\bar{e}$, for otherwise Lemma 1 would tell us that e is of the form l^σ where σ falsifies $\bar{u}$. Hence, e is independent of u according to $\mathcal{D}^{\text{rrs}}$. Note that, strictly speaking, Lemma 1 would only guarantee that the

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a \vee x \vee b \vee y \vee c}{a \vee b^{\bar{x}} \vee c^{\bar{x}\bar{y}}} \qquad \dfrac{a \vee x \vee b \vee y \vee \bar{c}}{a \vee b^{\bar{x}} \vee \bar{c}^{\bar{x}\bar{y}}}}{a \vee b^{\bar{x}}} \qquad \dfrac{x \vee \bar{b}}{\bar{b}^{\bar{x}}}}{a} \qquad \dfrac{\dfrac{\bar{y} \vee c}{c^{xy}} \qquad \dfrac{\bar{a} \vee \bar{x} \vee b \vee \bar{c}}{\bar{a} \vee b^{x} \vee \bar{c}^{xy}}}{\bar{a} \vee b^{x}}}{b^{x}} \qquad \dfrac{\bar{x} \vee \bar{b}}{\bar{b}^{x}}}{\bot}
$$

Fig. 2. Example of a $\forall$-Exp+Res refutation.

first elimination of a universal literal is a valid extended-universal-reduction step (because the elimination modifies the formula $\mathcal{Q}'.\psi_3$). However, since the elimination of universal literals does not introduce additional resolution paths, all eliminations of universal literals are valid extended-universal-reduction steps.

Step 5 (Resolution Proof): In this last step, we perform all resolution steps of π as QRAT additions to derive the empty clause. This is possible since ψ_4 contains all clauses that are involved in the resolution proof. We thus conclude:

Theorem 2. *Π is a* QRAT *refutation of $\mathcal{Q}.\psi$.*

We illustrate our simulation on an example before showing that it is polynomial:

Example 2. Fig. 2 shows a $\forall$-Exp+Res refutation of $\exists a \forall x \exists b \forall y \exists c.\psi$ with

$$\psi = (a \vee x \vee b \vee y \vee c) \wedge (a \vee x \vee b \vee y \vee \bar{c}) \wedge (x \vee \bar{b}) \wedge (\bar{y} \vee c) \wedge (\bar{a} \vee \bar{x} \vee b \vee \bar{c}) \wedge (\bar{x} \vee \bar{b}).$$

For simulating this proof in QRAT, we proceed as follows.

(1) We introduce definitions of the annotated variables by adding the following eight QRAT clauses:

$$
\begin{array}{cccc}
(\bar{b} \vee b^{\bar{x}}) & (\bar{b} \vee b^{x}) & (\bar{c} \vee c^{\bar{x}\bar{y}}) & (\bar{c} \vee c^{xy}) \\
(b \vee \bar{b}^{\bar{x}}) & (b \vee \bar{b}^{x}) & (c \vee \bar{c}^{\bar{x}\bar{y}}) & (c \vee \bar{c}^{xy})
\end{array}
$$

(2) We then introduce the following QRAT clauses, which correspond to applications of the axiom rule in $\forall$-Exp+Res:

$$
\begin{array}{lll}
(a \vee x \vee b^{\bar{x}} \vee y \vee c^{\bar{x}\bar{y}}) & (\bar{y} \vee c^{xy}) & (x \vee \bar{b}^{\bar{x}}) \\
(a \vee x \vee b^{\bar{x}} \vee y \vee \bar{c}^{\bar{x}\bar{y}}) & (\bar{a} \vee \bar{x} \vee b^{x} \vee \bar{c}^{xy}) & (\bar{x} \vee \bar{b}^{x})
\end{array}
$$

(3) We remove the original clauses and the clauses introduced in step 1. Only the clauses introduced in step 2 remain.

(4) From the remaining clauses, we first remove all occurrences of y and then all occurrences of x via extended universal reduction. We obtain the clauses introduced by applications of the axiom rule in the $\forall$-Exp+Res proof.

(5) Finally, we can simply perform the resolution steps of the $\forall$-Exp+Res proof to obtain a QRAT refutation of the input formula.

This concludes the example. $\square$

It remains to show that the simulation is polynomial. We first bound the size (measured by the number of symbols) of the resulting QRAT proof:

Lemma 3. *Π is linear in the size of π.*

Proof. In step 1 (definition introduction), we perform two QRAT additions for each annotated variable in the ∀-Exp+Res proof π. The size of the corresponding QRAT derivation is clearly linear with respect to π. In step 2, we perform one QRAT addition for each application of the axiom rule in π, again resulting in a linear-size QRAT derivation. In step 3, we eliminate clauses of ψ and definitions—also clearly linear. In step 4, we remove universal literals from existing clauses. All these universal literals are contained in π as their respective clauses are involved in applications of the axiom rule. Hence, also this step yields a QRAT derivation of linear size. Finally, the resolution proof derived in step 5 is part of π and thus also of linear size with respect to π. We conclude that the size of the final QRAT proof is linear with respect to the size of π. □

It should now be clear that our simulation can be performed in polynomial time:

Theorem 4. QRAT *polynomially simulates* ∀-Exp+Res.

7 Conclusion

We filled an empty spot in the QBF-proof-complexity landscape by showing that QRAT polynomially simulates universal expansion in general, and the proof system ∀-Exp+Res in particular. Our approach is similar to the approach in [12], which mimics the expansion of inner-most universal variables in QRAT.

There are, however, some subtle but important differences to [12]. First, in [12] the universal variables are fully expanded, which could potentially duplicate the whole formula. In contrast, we expand arbitrary variables and focus only on the clauses that are used as axioms in the ∀-Exp+Res proof. By only deriving these clauses (using new definitions), we can ensure that the resulting proof is small. Second, in [12] the QRAT proof is generated during proof search, when it is still unclear if the formula is true or false. In our simulation here, the proof of unsatisfiability is given as input and therefore we know from the beginning that the formula is false. This allows us to delete clauses eagerly (deletion doesn't have to preserve satisfiability), which is not the case in [12].

A closer look at our simulation shows that the only features of QRAT needed for the simulation are Q-resolution, definition introduction, and extended universal reduction. A system that uses only these features could be seen as *extended* $\mathrm{Q}(\mathcal{D}^{\mathrm{rrs}})$*-resolution* in the dependency framework of Slivovsky and Szeider [24]. In propositional logic, we know that extended resolution polynomially simulates DRAT [19] but it is not known if extended Q-resolution [17] or extended $\mathrm{Q}(\mathcal{D}^{\mathrm{rrs}})$-resolution can polynomially simulate QRAT. It is also still unclear how QRAT is related to the stronger expansion-based systems IR-calc and IRM-calc [4]. Finally, since there exist efficient proof checkers for QRAT and since the size increase induced by our simulation is only linear, our simulation could be used in practice to validate the results of expansion-based solvers.

References

1. Balabanov, V., Jiang, J.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012)
2. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Proc. of the 17th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2014). LNCS, vol. 8561, pp. 154–169. Springer (2014)
3. Beyersdorff, O., Bonacina, I., Chew, L.: Lower bounds: From circuits to QBF proof systems. In: Proc. of the 2016 ACM Conference on Innovations in Theoretical Computer Science (ITCS 2016). pp. 249–260. ACM (2016)
4. Beyersdorff, O., Chew, L., Janota, M.: On unification of QBF resolution-based calculi. In: Proc. of the 39th Int. Symposium on Mathematical Foundations of Computer Science (MFCS 2014). LNCS, vol. 8635, pp. 81–93. Springer (2014)
5. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. In: Proc. of the 32nd Int. Symposium on Theoretical Aspects of Computer Science (STACS 2015). LIPIcs, vol. 30, pp. 76–89. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015)
6. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Are short proofs narrow? QBF resolution is not simple. In: Proc. of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016). LIPIcs, vol. 47, pp. 15:1–15:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
7. Bloem, R., Braud-Santoni, N., Hadzic, V., Egly, U., Lonsing, F., Seidl, M.: Expansion-based QBF solving without recursion. In: Proc. of the Int. Conference on Formal Methods in Computer Aided Design (FMCAD 2018). pp. 1–10. IEEE (2018)
8. Chen, H.: Proof Complexity Modulo the Polynomial Hierarchy: Understanding Alternation as a Source of Hardness. In: Proc. of the 43rd Int. Colloquium on Automata, Languages, and Programming (ICALP 2016). LIPIcs, vol. 55, pp. 94:1–94:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
9. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. Journal of Symbolic Logic 44(1), 36–50 (1979)
10. Egly, U.: On stronger calculi for QBFs. In: Proc. of the 19th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2016). LNCS, vol. 9710, pp. 419–434. Springer (2016)
11. Gelder, A.V.: Variable independence and resolution paths for quantified boolean formulas. In: Proc. of the 17th Int. Conference on Principles and Practice of Constraint Programming (CP 2011). LNCS, vol. 6876, pp. 789–803. Springer (2011)
12. Heule, M.J.H., Seidl, M., Biere, A.: Solution validation and extraction for QBF preprocessing. Journal of Automated Reasoning 58(1), 1–29 (2016)
13. Janota, M.: On Q-Resolution and CDCL QBF solving. In: Proc. of the 19th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2016). LNCS, vol. 9710, pp. 402–418. Springer (2016)
14. Janota, M., Grigore, R., Marques-Silva, J.: On QBF proofs and preprocessing. In: Proc. of the 19th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19). LNCS, vol. 8312, pp. 473–489. Springer (2013)
15. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. Artificial Intelligence 234, 1–25 (2016)
16. Janota, M., Marques-Silva, J.: On propositional QBF expansions and Q-resolution. In: Proc. of the 16th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2013). LNCS, vol. 7962, pp. 67–82. Springer (2013)

17. Jussila, T., Biere, A., Sinz, C., Kröning, D., Wintersteiger, C.M.: A first step towards a unified proof checker for QBF. In: Proc. of the 10th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2007). LNCS, vol. 4501, pp. 201–214. Springer (2007)
18. Kiesl, B., Heule, M.J.H., Seidl, M.: A little blocked literal goes a long way. In: Proc. of the 20th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2017). LNCS, vol. 10491, pp. 281–297. Springer (2017)
19. Kiesl, B., Rebola-Pardo, A., Heule, M.J.H.: Extended resolution simulates DRAT. In: Proc. of the 9th Int. Joint Conference on Automated Reasoning (IJCAR 2018). LNCS, vol. 10900, pp. 516–531. Springer (2018)
20. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified boolean formulas. Information and Computation 117(1), 12–18 (1995)
21. Lonsing, F., Egly, U.: DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL. In: Proc. of the 26th Int. Conference on Automated Deduction (CADE-26). LNCS, vol. 10395, pp. 371–384. Springer (2017)
22. Peitl, T., Slivovsky, F., Szeider, S.: Dependency learning for QBF. In: Proc. of the 20th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2017). LNCS, vol. 10491, pp. 298–313. Springer (2017)
23. Slivovsky, F., Szeider, S.: Variable dependencies and Q-resolution. In: Proc. of the 17th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2014). LNCS, vol. 8561, pp. 269–284. Springer (2014)
24. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. Theoretical Computer Science 612, 83–101 (2016)
25. Tentrup, L.: On expansion and resolution in CEGAR based QBF solving. In: Proc. of the 29th Int. Conference on Computer Aided Verification (CAV 2017). LNCS, vol. 10427, pp. 475–494. Springer (2017)
26. Van Gelder, A.: Contributions to the theory of practical quantified boolean formula solving. In: Proc. of the 18th Int. Conference on Principles and Practice of Constraint Programming (CP 2012). LNCS, vol. 7514, pp. 647–663. Springer (2012)
27. Wetzler, N., Heule, M.J.H., Hunt Jr., W.A.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: Proc. of the 17th Int. Conference on Theory and Applications of Satisfiability Testing (SAT 2014). LNCS, vol. 8561, pp. 422–429. Springer, Cham (2014)
28. Zhang, L., Malik, S.: Conflict driven learning in a quantified boolean satisfiability solver. In: Proc. of the 2002 IEEE/ACM Int. Conference on Computer-aided Design (ICCAD 2002). pp. 442–449. ACM/IEEE Computer Society (2002)