QRAT$^+$: Generalizing **QRAT** by a More Powerful QBF Redundancy Property*

Florian Lonsing and Uwe Egly

Research Division of Knowledge Based Systems
Institute of Logic and Computation, TU Wien, Austria
`firstname.lastname@tuwien.ac.at`

Abstract. The QRAT (quantified resolution asymmetric tautology) proof system simulates virtually all inference rules applied in state of the art quantified Boolean formula (QBF) reasoning tools. It consists of rules to rewrite a QBF by adding and deleting clauses and universal literals that have a certain redundancy property. To check for this redundancy property in QRAT, propositional unit propagation (UP) is applied to the quantifier free, i.e., propositional part of the QBF. We generalize the redundancy property in the QRAT system by QBF specific UP (QUP). QUP extends UP by the universal reduction operation to eliminate universal literals from clauses. We apply QUP to an abstraction of the QBF where certain universal quantifiers are converted into existential ones. This way, we obtain a generalization of QRAT we call QRAT$^+$. The redundancy property in QRAT$^+$ based on QUP is more powerful than the one in QRAT based on UP. We report on proof theoretical improvements and experimental results to illustrate the benefits of QRAT$^+$ for QBF preprocessing.

1   Introduction

In practical applications of propositional logic satisfiability (SAT), it is necessary to establish correctness guarantees on the results produced by SAT solvers by proof checking [7]. The DRAT (deletion resolution asymmetric tautology) [23] approach has become state of the art to generate and check propositional proofs.

The logic of quantified Boolean formulas (QBF) extends propositional logic by existential and universal quantification of the propositional variables. Despite the PSPACE-completeness of QBF satisfiability checking, QBF technology is relevant in practice due to the potential succinctness of QBF encodings [4].

DRAT has been lifted to QBF to obtain the QRAT (quantified RAT) proof system [8,10]. QRAT allows to represent and check (un)satisfiability proofs of QBFs and compute Skolem function certificates of satisfiable QBFs. The QRAT system simulates virtually all inference rules applied in state of the art QBF reasoning tools, such as Q-resolution [15] including its variant long-distance Q-resolution [13,25], and expansion of universal variables [3].

A QRAT proof of a QBF in prenex CNF consists of a sequence of inference steps that rewrite the QBF by adding and deleting clauses and universal literals that have the QRAT *redundancy property*. Informally, checking whether a clause C has QRAT amounts to checking whether all possible resolvents of C on a literal $l \in C$ (under certain restrictions) are *propositionally implied* by the quantifier-free CNF part of the QBF. The principle of redundancy checking by inspecting resolvents originates from the RAT property in propositional logic [12] and was generalized to first-order logic in terms of *implication modulo resolution* [14]. Instead of a complete (and thus computationally hard) propositional implication check on a resolvent, the QRAT system relies on an incomplete check by *propositional unit propagation* (UP). Thereby, it is checked whether UP can derive the empty clause from the CNF augmented by the negated resolvent. Hence redundancy checking in QRAT is unaware of the quantifier structure, which is entirely ignored in UP.

We generalize redundancy checking in QRAT by making it aware of the quantifier structure of a QBF. To this end, we check the redundancy of resolvents based on *QBF specific UP* (QUP). It extends UP by the *universal reduction* (UR) operation [15] and is a polynomial-time procedure like UP. UR is central in resolution based QBF calculi [1,15] as it shortens individual clauses by eliminating universal literals depending on the quantifier structure. We apply QUP to *abstractions* of the QBF where certain universal quantifiers are converted into existential ones. The purpose of abstractions is that if a resolvent is found redundant by QUP on the abstraction, then it is also redundant in the original QBF.

Our contributions are as follows: (1) by applying QUP and QBF abstractions instead of UP, we obtain a *generalization of the* QRAT *system* which we call QRAT$^+$. In contrast to QRAT, redundancy checking in QRAT$^+$ is aware of the quantifier structure of a QBF. We show that (2) the redundancy property in QRAT$^+$ based on QUP is *more powerful* than the one in QRAT based on UP. QRAT$^+$ can detect redundancies which QRAT cannot. As a formal foundation, we introduce (3) a *theory of QBF abstractions* used in QRAT$^+$. Redundancy elimination by QRAT$^+$ or QRAT can lead to (4) *exponentially shorter proofs* in certain resolution based QBF calculi, which we point out by a concrete example. Note that here we do not study the power of QRAT or QRAT$^+$ as proof systems themselves, but the impact of redundancy elimination. Finally, we report on experimental results (5) to illustrate the benefits of redundancy elimination by QRAT$^+$ and QRAT for QBF preprocessing. Our implementation of QRAT$^+$ and QRAT for preprocessing is the first one reported in the literature.

2 Preliminaries

We consider QBFs $\phi := \Pi.\psi$ in *prenex conjunctive normal form (PCNF)* with a *quantifier prefix* $\Pi := Q_1 B_1 \ldots Q_n B_n$ and a quantifier free CNF ψ not containing tautological clauses. The prefix consists of *quantifier blocks* $Q_i B_i$, where B_i are *blocks* (i.e., sets) of propositional variables and $Q_i \in \{\forall, \exists\}$ are *quantifiers*. We have $B_i \cap B_j = \emptyset$, $Q_i \neq Q_{i+1}$ and $Q_n = \exists$. The CNF ψ is defined precisely over the variables $vars(\phi) = vars(\psi) := B_1 \cup \ldots \cup B_n$ in Π so that all variables are

quantified, i.e., ϕ is *closed*. The *quantifier* $\mathsf{Q}(\Pi, l)$ of literal l is Q_i if the variable $\mathsf{var}(l)$ of l appears in B_i. The set of variables in a clause C is $vars(C) := \{x \mid l \in C, \mathsf{var}(l) = x\}$. A literal l is *existential* if $\mathsf{Q}(\Pi, l) = \exists$ and *universal* if $\mathsf{Q}(\Pi, l) = \forall$. If $\mathsf{Q}(\Pi, l) = Q_i$ and $\mathsf{Q}(\Pi, k) = Q_j$, then $l \leq_\Pi k$ iff $i \leq j$. We extend the ordering $\leq_\Pi$ to an arbitrary but fixed ordering on the variables in every block B_i.

An *assignment* $\tau : vars(\phi) \to \{\top, \bot\}$ maps the variables of a QBF ϕ to truth constants $\top$ (*true*) or $\bot$ (*false*). Assignment τ is *complete* if it assigns every variable in ϕ, otherwise τ is *partial*. By $\tau(\phi)$ we denote ϕ *under τ*, where each occurrence of variable x in ϕ is replaced by $\tau(x)$ and x is removed from the prefix of ϕ, followed by propositional simplifications on $\tau(\phi)$. We consider τ as a set of literals such that, for some variable x, $x \in \tau$ if $\tau(x) = \top$ and $\bar{x} \in \tau$ if $\tau(x) = \bot$.

An *assignment tree* [10] T of a QBF ϕ is a *complete binary tree* of depth $|vars(\phi)| + 1$ where the internal (non-leaf) nodes of each level are associated with a variable of ϕ. An internal node is universal (existential) if it is associated with a universal (existential) variable. The order of variables along every path in T respects the extended order $\leq_\Pi$ of the prefix Π of ϕ. An internal node associated with variable x has two outgoing edges pointing to its children: one labelled with $\bar{x}$ and another one labelled with x, denoting the assignment of x to false and true, respectively. Each path τ in T from the root to an internal node (leaf) represents a partial (complete) assignment. A leaf at the end of τ is labelled by $\tau(\phi)$, i.e., the value of ϕ under τ. An internal node associated with an existential (universal) variable is labelled with $\top$ iff one (both) of its children is (are) labelled with $\top$. The QBF ϕ is *satisfiable* (*unsatisfiable*) iff the root of T is labelled with $\top$ ($\bot$).

Given a QBF ϕ and its assignment tree T, a subtree T' of T is a *pre-model* [10] of ϕ if (1) the root of T is the root of T', (2) for every universal node in T' both children are in T', and (3) for every existential node in T' exactly one of its children is in T'. A pre-model T' of ϕ is a *model* [10] of ϕ, denoted by $T' \models_t \phi$, if each node in T' is labelled with $\top$. A QBF ϕ is *satisfiable* iff it has a model. Given a QBF ϕ and one of its *models* T', T'' is a *rooted subtree* of T' ($T'' \subseteq T'$) if T'' has the same root as T' and the leaves of T'' are a subset of the leaves of T'.

We consider CNFs ψ defined over a set B of variables without an explicit quantifier prefix. A *model* of a CNF ψ is a model τ of the QBF $\exists B.\psi$ which consists only of the single path τ. We write $\tau \models \psi$ if τ is a model of ψ. For CNFs ψ and ψ', ψ' is *implied* by ψ ($\psi \models \psi'$) if, for all τ, it holds that if $\tau \models \psi$ then $\tau \models \psi'$. Two CNFs ψ and ψ' are *equivalent* ($\psi \equiv \psi'$), iff $\psi \models \psi'$ and $\psi' \models \psi$. We define notation to explicitly refer to QBF models. For QBFs ϕ and ϕ', ϕ' is *implied* by ϕ ($\phi \models_t \phi'$) if, for all T, it holds that if $T \models_t \phi$ then $T \models_t \phi'$. QBFs ϕ and ϕ' are *equivalent* ($\phi \equiv_t \phi'$) iff $\phi \models_t \phi'$ and $\phi' \models_t \phi$, and *satisfiability equivalent* ($\phi \equiv_{sat} \phi'$) iff ϕ is satisfiable whenever ϕ' is satisfiable. Satisfiability equivalence of CNFs is defined analogously and denoted by the same symbol '$\equiv_{sat}$'.

3 The Original QRAT Proof System

Before we generalize QRAT, we recapitulate the original proof system [10] and emphasize that redundancy checking in QRAT is unaware of quantifier structures.

Definition 1 ([10]). *The* outer clause *of clause C on literal $l \in C$ with respect to prefix Π is the clause* $\mathsf{OC}(\Pi, C, l) := \{k \mid k \in C, k \leq_\Pi l, k \neq l\}$.

The outer clause $\mathsf{OC}(\Pi, C, l) \subset C$ of C on $l \in C$ contains only literals that are smaller than or equal to l in the variable ordering of prefix Π, excluding l.

Definition 2 ([10]). *Let C be a clause with $l \in C$ and D be a clause with $\bar{l} \in D$ occurring in QBF $\Pi.\psi$. The* outer resolvent *of C with D on l with respect to Π is the clause* $\mathsf{OR}(\Pi, C, D, l) := (C \setminus \{l\}) \cup \mathsf{OC}(\Pi, D, \bar{l})$.

Example 1. Given $\phi := \exists x_1 \forall u \exists x_2.(C \wedge D)$ with $C := (x_1 \vee u \vee x_2)$ and $D := (\bar{x}_1 \vee \bar{u} \vee \bar{x}_2)$, we have $\mathsf{OR}(\Pi, C, D, x_1) = (u \vee x_2)$, $\mathsf{OR}(\Pi, C, D, u) = (x_1 \vee \bar{x}_1 \vee x_2)$, $\mathsf{OR}(\Pi, C, D, x_2) = (x_1 \vee u \vee \bar{x}_1 \vee \bar{u})$, and $\mathsf{OR}(\Pi, D, C, \bar{u}) = (x_1 \vee \bar{x}_1 \vee \bar{x}_2)$. Computing outer resolvents is asymmetric since $\mathsf{OR}(\Pi, C, D, u) \neq \mathsf{OR}(\Pi, D, C, \bar{u})$.

Definition 3 ([10]). *Clause C has property* QIOR (quantified implied outer resolvent) *on literal $l \in C$ with respect to QBF $\Pi.\psi$ iff $\psi \models \mathsf{OR}(\Pi, C, D, l)$ for all $D \in \psi$ with $\bar{l} \in D$.*

Property QIOR relies on checking whether every possible outer resolvent OR of some clause C on a literal is redundant by checking if OR is *propositionally implied* by the *quantifier-free CNF ψ* of the given QBF $\Pi.\psi$. If C has QIOR on literal $l \in C$ then, depending on whether l is existential or universal and side conditions, either C is redundant and can be removed from QBF $\Pi.\psi$ or l is redundant and can be removed from C, respectively, resulting in a *satisfiability-equivalent* QBF.

Theorem 1 ([10]). *Given a QBF $\phi := \Pi.\psi$ and a clause $C \in \psi$ with QIOR on an* existential *literal $l \in C$ with respect to QBF $\phi' := \Pi.\psi'$ where $\psi' := \psi \setminus \{C\}$. Then $\phi \equiv_{sat} \phi'$.*

Theorem 2 ([10]). *Given a QBF $\phi_0 := \Pi.\psi$ and $\phi := \Pi.(\psi \cup \{C\})$ where C has QIOR on a* universal *literal $l \in C$ with respect to ϕ_0. Let $\phi' := \Pi.(\psi \cup \{C'\})$ with $C' := C \setminus \{l\}$. Then $\phi \equiv_{sat} \phi'$.*

Note that in Theorems 1 and 2 clause C is actually removed from the QBF for the check whether C has QIOR on a literal. Checking propositional implication ($\models$) as in Definition 3 is co-NP hard and hence intractable. Therefore, in practice a polynomial-time incomplete implication check based on propositional unit propagation (UP) is applied. The use of UP is central in the QRAT proof system.

Definition 4 (propositional unit propagation, UP). *For a CNF ψ and clause C, let $\psi \wedge \overline{C} \vdash_1 \emptyset$ denote the fact that propositional unit propagation (UP) applied to $\psi \wedge \overline{C}$ produces the empty clause, where $\overline{C}$ is the conjunction of the negation of all the literals in C. If $\psi \wedge \overline{C} \vdash_1 \emptyset$ then we write $\psi \vdash_1 C$ to denote that C can be derived from ψ by UP (since $\psi \models C$).*

Definition 5 ([10]). *Clause C has property* AT (asymmetric tautology) *with respect to a CNF ψ iff $\psi \vdash_1 C$.*

AT is a propositional clause redundancy property that is used in the QRAT proof system to check whether outer resolvents are redundant, thereby replacing propositional implication ($\models$) in Definition 3 by unit propagation ($\vdash_1$) as follows.

Definition 6 ([10]). *Clause C has property* QRAT *(quantified resolution asymmetric tautology) on literal $l \in C$ with respect to QBF $\Pi.\psi$ iff, for all $D \in \psi$ with $\bar{l} \in D$, the outer resolvent $\mathsf{OR}(\Pi, C, D, l)$ has* AT *with respect to CNF ψ.*

Example 2. Consider $\phi := \exists x_1 \forall u \exists x_2.(C \wedge D)$ with $C := (x_1 \vee u \vee x_2)$ and $D := (\bar{x}_1 \vee \bar{u} \vee \bar{x}_2)$ from Example 1. C does not have AT with respect to CNF D, but C has QRAT on x_2 with respect to QBF $\exists x_1 \forall u \exists x_2.(D)$ since $\mathsf{OR}(\Pi, C, D, x_2) = (x_1 \vee u \vee \bar{x}_1 \vee \bar{u})$ has AT with respect to CNF D.

QRAT is a restriction of QIOR, i.e., a clause that has QRAT also has QIOR but not necessarily vice versa. Therefore, the soundness of removing redundant clauses and literals based on QRAT follows right from Theorems 1 and 2.

Based on the QRAT redundancy property, the QRAT proof system [10] consists of rewrite rules to eliminate redundant clauses, denoted by QRATE, to add redundant clauses, denoted by QRATA, and to eliminate redundant universal literals, denoted by QRATU. In a QRAT *satisfaction proof (refutation)*, a QBF is reduced to the empty formula (respectively, to a formula containing the empty clause) by applying the rewrite rules. The QRAT proof systems has an additional rule to eliminate universal literals by *extended universal reduction* (EUR). We do not present EUR because it is not affected by our generalization of QRAT, which we define in the following. Observe that QIOR and AT (and hence also QRAT) are based on *propositional* implication ($\models$) and unit propagation ($\vdash_1$), i.e., the quantifier structure of the given QBF is not exploited.

4 QRAT$^+$: A More Powerful QBF Redundancy Property

We make redundancy checking of outer resolvents in QRAT aware of the quantifier structure of a QBF. To this end, we generalize QIOR and AT by replacing propositional implication ($\models$) and unit propagation ($\vdash_1$) by QBF implication ($\models_t$) and QBF unit propagation, respectively. Thereby, we obtain a more general and more powerful notion of the QRAT redundancy property, which we call QRAT$^+$.

First, in Proposition 2 we point out a property of QIOR (Definition 3) which is due to the following result from related work [21]: if we attach a quantifier prefix Π to equivalent CNFs ψ and ψ', then the resulting QBFs are equivalent.

Proposition 1 ([21]). *Given CNFs ψ and ψ' such that $vars(\psi) = vars(\psi')$ and a quantifier prefix Π defined precisely over $vars(\psi)$. If $\psi \equiv \psi'$ then $\Pi.\psi \equiv_t \Pi.\psi'$.*

Proposition 2. *If clause C has QIOR on literal $l \in C$ with respect to QBF $\Pi.\psi$, then $\Pi.\psi \equiv_t \Pi.(\psi \wedge \mathsf{OR}(\Pi, C, D, l))$ for all $D \in \psi$ with $\bar{l} \in D$.*

Proof. Since C has QIOR on literal $l \in C$ with respect to QBF $\Pi.\psi$, by Definition 3 we have $\psi \models \mathsf{OR}(\Pi, C, D, l)$ for all $D \in \psi$ with $\bar{l} \in D$, and further also $\psi \equiv \psi \wedge \mathsf{OR}(\Pi, C, D, l)$. Then $\Pi.\psi \equiv_t \Pi.(\psi \wedge \mathsf{OR}(\Pi, C, D, l))$ by Proposition 1. □

By Proposition 2 any outer resolvent OR of some clause C that has QIOR with respect to some QBF $\Pi.\psi$ is redundant in the sense that it can be *added to the QBF $\Pi.\psi$ in an equivalence preserving way* ($\equiv_t$), i.e., OR is *implied by the QBF $\Pi.\psi$* ($\models_t$). This is the central characteristic of our generalization QRAT^+ of QRAT. We develop a redundancy property used in QRAT^+ which allows to, e.g., remove a clause C from a QBF $\Pi.\psi$ in a satisfiability preserving way (like in QRAT, cf. Theorem 1.) if all respective outer resolvents of C are implied by the QBF $\Pi.(\psi \setminus \{C\})$. Since checking QBF implication is intractable just like checking propositional implication in QIOR, in practice we apply a polynomial-time incomplete QBF implication check based on *QBF unit propagation*.

In the following, we develop a theoretical framework of *abstractions* of QBFs that underlies our generalization QRAT^+ of QRAT. Abstractions are crucial for the soundness of checking QBF implication by QBF unit propagation.

Definition 7 (nesting levels, prefix/QBF abstraction). *Let $\phi := \Pi.\psi$ be a QBF with prefix $\Pi := Q_1 B_1 \ldots Q_i B_i Q_{i+1} B_{i+1} \ldots Q_n B_n$. For a clause C, $levels(\Pi, C) := \{i \mid \exists l \in C, \mathsf{Q}(\Pi, l) = Q_i\}$ is the set of* nesting levels *in C.[1] The abstraction of Π with respect to i with $0 \leq i \leq n$ produces the abstracted prefix $Abs(\Pi, i) := \Pi$ for $i = 0$ and otherwise $Abs(\Pi, i) := \exists(B_1 \cup \ldots \cup B_i)Q_{i+1}B_{i+1} \ldots Q_n B_n$. The abstraction of ϕ with respect to i with $0 \leq i \leq n$ produces the* abstracted QBF *$Abs(\phi, i) := Abs(\Pi, i).\psi$ with prefix $Abs(\Pi, i)$.*

Example 3. Given the QBF $\phi := \Pi.\psi$ with prefix $\Pi := \forall B_1 \exists B_2 \forall B_3 \exists B_4$. We have $Abs(\phi, 0) = \phi$, $Abs(\phi, 1) = Abs(\phi, 2) = \exists(B_1 \cup B_2)\forall B_3 \exists B_4.\psi$, $Abs(\phi, 3) = Abs(\phi, 4) = \exists(B_1 \cup B_2 \cup B_3 \cup B_4).\psi$.

In an abstracted QBF $Abs(\phi, i)$ universal variables from blocks smaller than or equal to B_i are converted into existential ones. If the original QBF ϕ has a model T, then *all* nodes in T associated to universal variables must be labelled with $\top$, in particular the universal variables that are existential in $Abs(\phi, i)$. Hence, for *all* models T of ϕ, every model T^A of $Abs(\phi, i)$ is a subtree of T.

Proposition 3. *Given a QBF $\phi := \Pi.\psi$ with prefix $\Pi := Q_1 B_1 \ldots Q_i B_i \ldots Q_n B_n$ and $Abs(\phi, i)$ for some arbitrary i with $0 \leq i \leq n$. For all T and T^A we have that if $T \models_t \phi$ and $T^A \subseteq T$ is a pre-model of $Abs(\phi, i)$, then $T^A \models_t Abs(\phi, i)$.*

Proof. By induction on i. The base case $i := 0$ is trivial.

As induction hypothesis (IH), assume that the claim holds for some i with $0 \leq i < n$, i.e., for all T and T^A we have that if $T \models_t \phi$ and $T^A \subseteq T$ is a pre-model of $Abs(\phi, i)$, then $T^A \models_t Abs(\phi, i)$. Consider $Abs(\phi, j)$ for $j = i + 1$, which is an abstraction of $Abs(\phi, i)$. We have to show that, for all T and T^B we have that if $T \models_t \phi$ and $T^B \subseteq T$ is a pre-model of $Abs(\phi, j)$, then $T^B \models_t Abs(\phi, j)$. We distinguish cases by the type of Q_j in the abstracted prefix $Abs(\Pi, i) = \exists(B_1 \cup \ldots \cup B_i)Q_j B_j \ldots Q_n B_n$ of $Abs(\phi, i)$.

If $Q_j = \exists$ then $Abs(\Pi, i) = Abs(\Pi, j) = \exists(B_1 \cup \ldots B_i \cup B_j) \ldots Q_n B_n$. Since $Abs(\phi, i) = Abs(\phi, j)$, the claim holds for $Abs(\phi, j)$ by IH.

---

[1] In general, clauses C are always (implicitly) interpreted under a quantifier prefix Π.

If $Q_j = \forall$ then, towards a contradiction, assume that, for some T and T^B, $T \models_t \phi$ and $T^B \subseteq T$ is a pre-model of $Abs(\phi, j)$, but $T^B \not\models_t Abs(\phi, j)$. Then the root of T^B is labelled with $\bot$, and in particular the nodes of all the variables which are existential in B_j with respect to $Abs(\Pi, j)$ are also labelled with $\bot$. These existential variables appear along a single branch τ' in T^B, i.e., τ' is a partial assignment of the variables in B_j. Since $T^B \subseteq T^A$ and $Q_j = \forall$ in $Abs(\Pi, i)$, the root of T^A is labelled with $\bot$ since there is the branch τ' containing the variables in B_j whose nodes are labelled with $\bot$ in T^A. Hence $T^A \not\models_t Abs(\phi, i)$, which is a contradiction to IH. Therefore, we conclude that $T^B \models_t Abs(\phi, j)$. $\qquad\square$

If an abstraction $Abs(\phi, i)$ is unsatisfiable then also the original QBF ϕ is unsatisfiable due to Proposition 3. We generalize Proposition 1 from CNFs to QBFs and their abstractions. Note that the full abstraction $Abs(\phi, i)$ for $i := n$ of a QBF ϕ is a CNF, i.e., it does not contain any universal variables.

Lemma 1. *Let $\phi := \Pi.\psi$ and $\phi' := \Pi.\psi'$ be QBFs with the same prefix $\Pi := Q_1 B_1 \ldots Q_i B_i \ldots Q_n B_n$. Then for all i, if $Abs(\phi, i) \equiv_t Abs(\phi', i)$ then $\phi \equiv_t \phi'$.*

Proof. By induction on $i := 0$ up to $i := n$. The base case $i := 0$ is trivial.

As induction hypothesis (IH), assume that the claim holds for some i with $0 \le i < n$, i.e., if $Abs(\phi, i) \equiv_t Abs(\phi', i)$ then $\phi \equiv_t \phi'$. Let $j = i + 1$ and consider $Abs(\phi, j)$ and $Abs(\phi', j)$, which are abstractions of $Abs(\phi, i)$ and $Abs(\phi', i)$. We have $Abs(\Pi, i) = \exists(B_1 \cup \ldots \cup B_i)Q_j B_j \ldots Q_n B_n$ and $Abs(\Pi, j) = \exists(B_1 \cup \ldots \cup B_j) \ldots Q_n B_n$. We show that if $Abs(\phi, j) \equiv_t Abs(\phi', j)$ then $Abs(\phi, i) \equiv_t Abs(\phi', i)$, and hence also $\phi \equiv_t \phi'$ by IH. Assume that $Abs(\phi, j) \equiv_t Abs(\phi', j)$. We distinguish cases by the type of Q_j in $Abs(\Pi, i)$. If $Q_j = \exists$ then $Abs(\Pi, i) = Abs(\Pi, j) = \exists(B_1 \cup \ldots B_i \cup B_j) \ldots Q_n B_n$, and hence $Abs(\phi, i) \equiv_t Abs(\phi', i)$.

If $Q_j = \forall$, then towards a contradiction, assume that $Abs(\phi, j) \equiv_t Abs(\phi', j)$ but $Abs(\phi, i) \not\equiv_t Abs(\phi', i)$. Then there exists T such that $T \models_t Abs(\phi, i)$ but $T \not\models_t Abs(\phi', i)$. Since $T \not\models_t Abs(\phi', i)$ there exists a pre-model $T^A \subseteq T$ of $Abs(\phi', j)$ such that the root of T^A is labelled with $\bot$, and in particular the nodes of all the variables which are existential in B_j with respect to $Abs(\Pi, j)$ (and universal with respect to $Abs(\Pi, i)$) are also labelled with $\bot$. These existential variables appear along a single branch τ' in T^A, i.e., τ' is a partial assignment of the variables in B_j. Therefore we have $T^A \not\models_t Abs(\phi', j)$. Since $T \models_t Abs(\phi, i)$ and $T^A \subseteq T$, we have $T^A \models_t Abs(\phi, j)$ by Proposition 3, which contradicts the assumption that $Abs(\phi, j) \equiv_t Abs(\phi', j)$. $\qquad\square$

The converse of Lemma 1 does not hold. From the equivalence of two QBFs ϕ and ϕ' we cannot conclude that the abstractions $Abs(\phi, i)$ and $Abs(\phi', i)$ are equivalent. In our generalization QRAT$^+$ of the QRAT system we check whether an outer resolvent of some clause C is implied ($\models_t$) by an *abstraction* of the given QBF. If so then by Lemma 1 the outer resolvent is also implied by the original QBF. Below we prove that this condition is sufficient for the soundness of redundancy removal in QRAT$^+$. To check QBF implication in an incomplete way and in polynomial time, in practice we apply *QBF unit propagation*, which is an extension of propositional unit propagation, to abstractions of the given QBF.

Definition 8 (universal reduction, UR [15]). *Given a QBF $\phi := \Pi.\psi$ and a non-tautological clause C,* universal reduction (UR) *of C produces the clause $UR(\Pi, C) := C \setminus \{l \in C \mid \mathsf{Q}(\Pi, l) = \forall, \forall l' \in C, \mathsf{Q}(\Pi, l') = \exists : \mathsf{var}(l') \leq_\Pi \mathsf{var}(l)\}$.*

Definition 9 (QBF unit propagation, QUP). QBF unit propagation (QUP) *extends UP (Definition 4) by applications of UR. For a QBF $\phi := \Pi.\psi$ and a clause C, let $\Pi.(\psi \wedge \overline{C}) \vDash_\forall \emptyset$ denote the fact that QUP applied to $\Pi.(\psi \wedge \overline{C})$ produces the empty clause, where $\overline{C}$ is the conjunction of the negation of all the literals in C. If $\Pi.(\psi \wedge \overline{C}) \vDash_\forall \emptyset$ and additionally $\Pi.\psi \models_t \Pi.(\psi \wedge C)$ then we write $\phi \vDash_\forall C$ to denote that C can be derived from ϕ by QUP.*

In contrast to UP (Definition 4), deriving the empty clause by QUP by propagating $\overline{C}$ on a QBF ϕ is not sufficient to conclude that C is implied by ϕ.

Example 4. Given the QBF $\Pi.\psi$ with prefix $\Pi := \forall u \exists x$ and CNF $\psi := (u \vee \bar{x}) \wedge (\bar{u} \vee x)$ and the clause $C := (x)$. We have $\Pi.((u \vee \bar{x}) \wedge (\bar{u} \vee x) \wedge (\bar{x})) \vDash_\forall \emptyset$ since propagating $\overline{C} = (\bar{x})$ produces $(\bar{u})$, which is reduced to $\emptyset$ by UR. However, $\Pi.\psi \not\models_t \Pi.(\psi \wedge C)$ since $\Pi.\psi$ is satisfiable whereas $\Pi.(\psi \wedge C)$ is unsatisfiable. Note that $Abs(\Pi.((u \vee \bar{x}) \wedge (\bar{u} \vee x) \wedge \bar{x}), 2) \not\vDash_\forall \emptyset$.

To correctly apply QUP for checking whether some clause C (e.g., an outer resolvent) is implied by a QBF $\phi := \Pi.\psi$ and thus avoid the problem illustrated in Example 4, we carry out QUP on a *suitable abstraction* of ϕ with respect to C. Let $i = \max(levels(\Pi, C))$ be the maximum nesting level of variables that appear in C. We show that if QUP derives the empty clause from the abstraction $Abs(\phi, i)$ augmented by the negated clause $\overline{C}$, i.e., $Abs(\Pi.(\psi \wedge \overline{C}), i) \vDash_\forall \emptyset$, then we can safely conclude that C is implied by the *original* QBF, i.e., $\Pi.\psi \models_t \Pi.(\psi \wedge C)$. This approach extends failed literal detection for QBF preprocessing [16].

Lemma 2. *Let $\Pi.\psi$ be a QBF with prefix $\Pi := Q_1 B_1 \ldots Q_n B_n$ and C a clause such that $vars(C) \subseteq B_1$. If $\Pi.(\psi \wedge \overline{C}) \vDash_\forall \emptyset$ then $\Pi.\psi \equiv_t \Pi.(\psi \wedge C)$.*

Proof. By contradiction, assume $T \models_t \Pi.\psi$ but $T \not\models_t \Pi.(\psi \wedge C)$. Then there is a path $\tau \subseteq T$ such that $\tau(C) = \bot$. Since $vars(C) \subseteq B_1$ and $\Pi.(\psi \wedge \overline{C}) \vDash_\forall \emptyset$, the QBF $\Pi.(\psi \wedge \overline{C})$ is unsatisfiable and in particular $T \not\models_t \Pi.(\psi \wedge \overline{C})$. Since $\tau(C) = \bot$, we have $\tau(\overline{C}) = \top$ and hence $T \models_t \Pi.(\psi \wedge \overline{C})$, which is a contradiction. $\square$

Lemma 3. *Let $\Pi.\psi$ be a QBF, C a clause, and $i = \max(levels(\Pi, C))$. If $Abs(\Pi.(\psi \wedge \overline{C}), i) \vDash_\forall \emptyset$ then $Abs(\Pi.\psi, i) \equiv_t Abs(\Pi.(\psi \wedge C), i)$.*

Proof. The claim follows from Lemma 2 since all variables that appear in C are existentially quantified in $Abs(\Pi.(\psi \wedge \overline{C}), i)$ in the leftmost quantifier block. $\square$

Lemma 4. *Let $\Pi.\psi$ be a QBF, C a clause, and $i = \max(levels(\Pi, C))$. If $Abs(\Pi.(\psi \wedge \overline{C}), i) \vDash_\forall \emptyset$ then $\Pi.\psi \equiv_t \Pi.(\psi \wedge C)$.*

Proof. By Lemma 3 and Lemma 1. $\square$

Lemma 4 provides us with the necessary theoretical foundation to lift AT (Definition 5) from UP, which is applied to CNFs, to QUP, which is applied to *suitable abstractions* of QBFs. The abstractions are constructed depending on the maximum nesting level of variables in the clause we want to check.

Definition 10 (QAT). *Let ϕ be a QBF, C a clause, and $i = \max(levels(\Pi, C))$ Clause C has property* QAT *(quantified asymmetric tautology) with respect to ϕ iff $Abs(\phi, i) \vDash_{\overline{1}\forall} C$.*

As an immediate consequence from the definition of QUP (Definition 9) and Lemma 3, we can conclude that a clause C has QAT with respect to a QBF $\Pi.\psi$ if QUP derives the empty clause from the suitable abstraction of $\Pi.\psi$ with respect to C (i.e., $Abs(\Pi.(\psi \wedge \overline{C}), i) \vDash_{\overline{1}\forall} \emptyset$). Further, if C has QAT then we have $\Pi.\psi \equiv_t \Pi.(\psi \wedge C)$ by Lemma 4, i.e., C is implied by the given QBF $\Pi.\psi$.

Example 5. Given the QBF $\phi := \Pi.\psi$ with $\Pi := \forall u_1 \exists x_3 \forall u_2 \exists x_4$ and $\psi := (u_1 \vee \bar{x}_3) \wedge (u_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{u}_2 \vee \bar{x}_4)$. Clause $(u_1 \vee \bar{x}_3)$ has QAT with respect to $Abs(\phi, 2)$ with $\max(levels(C)) = 2$ since $\forall u_2$ is still universal in the abstraction. By QUP clause $(u_1 \vee \bar{x}_3 \vee x_4)$ becomes unit and clause $(\bar{u}_2 \vee \bar{x}_4)$ becomes empty by UR. However, clause $(u_1 \vee \bar{x}_3)$ does *not* have AT since $\forall u_2$ is treated as an existential variable in UP, hence clause $(\bar{u}_2 \vee \bar{x}_4)$ does not become empty by UR.

In contrast to AT, QAT is aware of quantifier structures in QBFs as shown in Example 5. We now generalize QRAT to QRAT$^+$ by replacing AT by QAT. Similarly, we generalize QIOR to QIOR$^+$ by replacing propositional implication ($\models$) and equivalence (Proposition 1), by *QBF implication and equivalence* (Lemma 4).

Definition 11 (QRAT$^+$). *Clause C has property* QRAT$^+$ *on literal $l \in C$ with respect to QBF $\Pi.\psi$ iff, for all $D \in \psi$ with $\bar{l} \in D$, the outer resolvent $OR(\Pi, C, D, l)$ has* QAT *with respect to QBF $\Pi.\psi$.*

Definition 12 (QIOR$^+$). *Clause C has property* QIOR$^+$ *on literal $l \in C$ with respect to QBF $\Pi.\psi$ iff $\Pi.\psi \equiv_t \Pi.(\psi \wedge OR(\Pi, C, D, l))$ for all $D \in \psi$ with $\bar{l} \in D$.*

If a clause has QRAT then it also has QRAT$^+$. Moreover, due to Proposition 2, if a clause has QIOR then it also has QIOR$^+$. Hence QRAT$^+$ and QIOR$^+$ indeed are generalizations of QRAT and QIOR, which are strict, as we argue below. The soundness of removing redundant clauses and universal literals based on QIOR$^+$ (and on QRAT$^+$) can be proved by the *same* arguments as original QRAT, which we outline in the following. We refer to an appendix for full proofs [18].

Definition 13 (prefix/suffix assignment [10]). *For a QBF $\phi := \Pi.\psi$ and a complete assignment τ in the assignment tree of ϕ, the partial* prefix *and* suffix *assignments of τ with respect to variable x, denoted by τ^x and τ_x, respectively, are defined as $\tau^x := \{y \mapsto \tau(y) \mid y \leq_\Pi x, y \neq x\}$ and $\tau_x := \{y \mapsto \tau(y) \mid y \not\leq_\Pi x\}$.*

For a variable x from block B_i of a QBF, Definition 13 allows us to split a complete assignment τ into three parts $\tau^x l \tau_x$, where the prefix assignment τ^x

assigns variables (excluding x) from blocks smaller than or equal to B_i, l is a literal of x, and the suffix assignment τ_x assigns variables from blocks larger than B_i.

Prefix and suffix assignments are important for proving the soundness of satisfiability-preserving redundancy removal by QIOR^+ (and QIOR). Soundness is proved by showing that certain paths in a model of a QBF can safely be modified based on prefix and suffix assignments, as stated in the following.

Lemma 5 (cf. Lemma 6 in [10]). *Given a clause C with QIOR^+ with respect to QBF $\phi := \Pi.\psi$ on literal $l \in C$ with $\mathsf{var}(l) = x$. Let T be a model of ϕ and $\tau \subseteq T$ be a path in T. If $\tau(C \setminus \{l\}) = \bot$ then $\tau^x(D) = \top$ for all $D \in \psi$ with $\bar{l} \in D$.*

Proof (sketch, see appendix [18]). Let $D \in \psi$ be a clause with $\bar{l} \in D$ and $R := \mathsf{OR}(\Pi, C, D, l) = (C \setminus \{l\}) \cup \mathsf{OC}(\Pi, D, \bar{l})$. By Definition 12, we have $\Pi.\psi \equiv_t \Pi.(\psi \wedge \mathsf{OR}(\Pi, C, D, l))$ for all $D \in \psi$ with $\bar{l} \in D$. The rest of the proof considers a path τ in T and works in the same way as the proof of Lemma 6 in [10]. □

Theorem 3. *Given a QBF $\phi := \Pi.\psi$ and a clause $C \in \psi$ with QIOR^+ on an existential literal $l \in C$ with respect to QBF $\phi' := \Pi.\psi'$ where $\psi' := \psi \setminus \{C\}$. Then $\phi \equiv_{sat} \phi'$.*

Proof (sketch, see appendix [18]). The proof relies on Lemma 5 and works in the same way as the proof of Theorem 7 in [10]. A model T of ϕ is obtained from a model T' of ϕ' by flipping the assignment of variable $x = \mathsf{var}(l)$ on a path τ in T' to satisfy clause C. All $D \in \psi$ with $\bar{l} \in D$ are satisfied by such modified τ. □

Theorem 4. *Given a QBF $\phi_0 := \Pi.\psi$ and $\phi := \Pi.(\psi \cup \{C\})$ where C has QIOR^+ on a universal literal $l \in C$ with respect to ϕ_0. Let $\phi' := \Pi.(\psi \cup \{C'\})$ with $C' := C \setminus \{l\}$. Then $\phi \equiv_{sat} \phi'$.*

Proof (sketch, see appendix [18]). The proof relies on Lemma 5 and works in the same way as the proof of Theorem 8 in [10]. A model T' of ϕ' is obtained from a model T of ϕ by modifying the subtree under the node associated to variable $x = \mathsf{var}(l)$. Suffix assignments of some paths τ in T are used to construct modified paths in T' under which clause C' is satisfied. All $D \in \psi$ with $\bar{l} \in D$ are still satisfied after such modifications. □

Analogously to the QRAT proof system that is based on the QRAT redundancy property (Definition 6), we obtain the QRAT^+ *proof system* based on property QRAT^+ (Definition 11). The system consists of rewrite rules QRATE^+, QRATA^+, and QRATU^+ to eliminate or add redundant clauses, and to eliminate redundant universal literals. On a conceptual level, these rules in QRAT^+ are similar to their respective counterparts in the QRAT system. The extended universal reduction rule EUR is the same in the QRAT and QRAT^+ systems. In contrast to QRAT, QRAT^+ is aware of quantifier structures of QBFs because it relies on the QBF specific property QAT and QUP instead of on propositional AT and UP.

The QRAT^+ system has the same desirable properties as the original QRAT system. QRAT^+ *simulates* virtually all inference rules applied in QBF reasoning tools and it is based on redundancy property QRAT^+ that can be checked in

polynomial time by QUP. Further, QRAT^+ allows to represent proofs in the *same proof format* as QRAT. However, proof checking, i.e., checking whether a clause listed in the proof has QRAT^+ on a literal, must be adapted to the use of QBF abstractions and QUP. Consequently, the available QRAT proof checker $\mathsf{QRATtrim}$ [10] cannot be used out of the box to check QRAT^+ proofs.

Notably, *Skolem functions* can be extracted from QRAT^+ proofs of satisfiable QBFs in the *same* way as in QRAT (consequence of Theorem 3, cf. Corollaries 26 and 27 in [10]). Hence like QRAT, QRAT^+ can be integrated in complete QBF workflows that include preprocessing, solving, and Skolem function extraction [5].

5   Exemplifying the Power of QRAT^+

In the following, we point out that the QRAT^+ system is more powerful than QRAT in terms of redundancy detection. In particular, we show that the rules QRATE^+ and QRATU^+ in the QRAT^+ system can eliminate certain redundancies that their counterparts QRATE and QRATU cannot eliminate.

Definition 14. *For $n \geq 1$, let $\Phi_C(n) := \Pi_C(n).\psi_C(n)$ be a class of QBFs with prefix $\Pi_C(n)$ and CNF $\psi_C(n)$ defined as follows.*

$\Pi_C(n) := \exists B_1 \forall B_2 \exists B_3 \forall B_4 \exists B_5:$ $\psi_C(n) := \bigwedge_{i:=0}^{n-1} \mathcal{C}(i)$ *with* $\mathcal{C}(i) := \bigwedge_{j:=0}^{6} C_{i,j}$:

$B_1 := \{x_{4i+1}, x_{4i+2} \mid 0 \leq i < n\}$ $C_{i,0} := (x_{4i+1} \vee u_{2i+1} \vee \neg x_{4i+3})$

$B_2 := \{u_{2i+1} \mid 0 \leq i < n\}$ $C_{i,1} := (x_{4i+2} \vee \neg u_{2i+1} \vee x_{4i+3})$

$B_3 := \{x_{4i+3} \mid 0 \leq i < n\}$ $C_{i,2} := (\neg x_{4i+1} \vee \neg u_{2i+1} \vee \neg x_{4i+3})$

$B_4 := \{u_{2i+2} \mid 0 \leq i < n\}$ $C_{i,3} := (\neg x_{4i+2} \vee u_{2i+1} \vee x_{4i+3})$

$B_5 := \{x_{4i+4} \mid 0 \leq i < n\}$ $C_{i,4} := (u_{2i+1} \vee \neg x_{4i+3} \vee x_{4i+4})$

$\phantom{B_5 := \{x_{4i+4} \mid 0 \leq i < n\}}$ $C_{i,5} := (\neg u_{2i+2} \vee \neg x_{4i+4})$

$\phantom{B_5 := \{x_{4i+4} \mid 0 \leq i < n\}}$ $C_{i,6} := (\neg x_{4i+1} \vee u_{2i+2} \vee \neg x_{4i+4})$

Example 6. For $n := 1$, we have $\Phi_C(n)$ with prefix $\Pi_C(n) := \exists x_1, x_2 \forall u_1 \exists x_3 \forall u_2 \exists x_4$ and CNF $\psi_C(n) := \mathcal{C}(0)$ with $\mathcal{C}(0) := \bigwedge_{j:=0}^{6} C_{0,j}$ as follows.

$C_{0,0} := (x_1 \vee u_1 \vee \neg x_3)$ $C_{0,4} := (u_1 \vee \neg x_3 \vee x_4)$

$C_{0,1} := (x_2 \vee \neg u_1 \vee x_3)$ $C_{0,5} := (\neg u_2 \vee \neg x_4)$

$C_{0,2} := (\neg x_1 \vee \neg u_1 \vee \neg x_3)$ $C_{0,6} := (\neg x_1 \vee u_2 \vee \neg x_4)$

$C_{0,3} := (\neg x_2 \vee u_1 \vee x_3)$

Proposition 4. *For $n \geq 1$, QRATE^+ can eliminate all clauses in $\Phi_C(n)$ whereas QRATE cannot eliminate any clause in $\Phi_C(n)$.*

Proof (sketch). For i and k with $i \neq k$, the sets of variables in $\mathcal{C}(i)$ and $\mathcal{C}(k)$ are disjoint. Thus it suffices to prove the claim for an arbitrary $\mathcal{C}(i)$. Clause $C_{i,0}$ has QRAT^+ on literal x_{4i+1} and can be removed. The relevant outer resolvents are $\mathsf{OR}_{0,2} = \mathsf{OR}(\Pi_C(n), C_{i,0}, C_{i,2}, x_{4i+1})$ and $\mathsf{OR}_{0,6} = \mathsf{OR}(\Pi_C(n), C_{i,0}, C_{i,6}, x_{4i+1})$, and we have $\mathsf{OR}_{0,2} = \mathsf{OR}_{0,6} = (u_{2i+1} \vee \neg x_{4i+3})$. Since $\max(levels(\mathsf{OR}_{0,2})) = \max(levels(\mathsf{OR}_{0,6})) = 3$, we apply QUP to the abstraction $Abs(\Phi_C(n), 3)$. Note that variable u_{2i+2} from block B_4 still is universal in the prefix of $Abs(\Phi_C(n), 3)$.

11

Propagating $\overline{\mathsf{OR}_{0,2}}$ and $\overline{\mathsf{OR}_{0,6}}$, respectively, in either case makes $C_{i,4}$ unit, finally $C_{i,5}$ becomes empty under the derived assignment x_{4i+4} since UR reduces the literal $\neg u_{2i+2}$. After removing $C_{i,0}$, clauses $C_{i,2}$ and $C_{i,6}$ trivially have QRAT^+ on $\neg x_{4i+1}$. Then $C_{i,1}$ has QRAT^+ on x_{4i+3}. Finally, the remaining clauses trivially have QRAT^+. In contrast to that, QRATE cannot eliminate any clause in $\Phi_C(n)$. Clause $C_{i,5}$ does not become empty by UP since all variables are existential. The claim can be proved by case analysis of all possible outer resolvents. □

Definition 15. *For $n \geq 1$, let $\Phi_L(n) := \Pi_L(n).\psi_L(n)$ be a class of QBFs with prefix $\Pi_L(n)$ and CNF $\psi_L(n)$ defined as follows.*

$\Pi_L(n) := \forall B_1 \exists B_2 \forall B_3 \exists B_4$:
$B_1 := \{u_{3i+1}, u_{3i+2} \mid 0 \leq i < n\}$ $B_3 := \{u_{3i+3} \mid 0 \leq i < n\}$
$B_2 := \{x_{3i+1}, x_{3i+2} \mid 0 \leq i < n\}$ $B_4 := \{x_{3i+3} \mid 0 \leq i < n\}$

$\psi_L(n) := \bigwedge_{i:=0}^{n-1} \mathcal{C}(i)$ *with* $\mathcal{C}(i) := \bigwedge_{j:=0}^{7} C_{i,j}$:
$C_{i,0} := (\neg u_{3i+2} \vee \neg x_{3i+1} \vee \neg x_{3i+2})$ $\qquad$ $C_{i,4} := (\neg x_{3i+1} \vee \neg x_{3i+2} \vee x_{3i+3})$
$C_{i,1} := (\neg u_{3i+1} \vee \neg x_{3i+1} \vee x_{3i+2})$ $\qquad$ $C_{i,5} := (u_{3i+3} \vee \neg x_{3i+3})$
$C_{i,2} := (u_{3i+1} \vee x_{3i+1} \vee \neg x_{3i+2})$ $\qquad$ $C_{i,6} := (\neg x_{3i+1} \vee x_{3i+2} \vee \neg x_{3i+3})$
$C_{i,3} := (u_{3i+2} \vee x_{3i+1} \vee x_{3i+2})$ $\qquad$ $C_{i,7} := (\neg u_{3i+3} \vee x_{3i+3})$

Proposition 5. *For $n \geq 1$, QRATU^+ can eliminate the entire quantifier block $\forall B_1$ in $\Phi_L(n)$ whereas QRATU cannot eliminate any universal literals in $\Phi_L(n)$.*

Proof (sketch, see appendix [18]). Formulas $\Phi_L(n)$ are constructed based on a similar principle as $\Phi_C(n)$ in Definition 14. E.g., clauses $C_{i,0}$ and $C_{i,1}$ have QRAT^+ but not QRAT on literals $\neg u_{3i+2}$ and $\neg u_{3i+1}$. During QUP, clauses $C_{i,5}$ and $C_{i,7}$ become empty only due to UR, which is not possible when using UP. □

6 Proof Theoretical Impact of QRAT and QRAT^+

As argued in the context of *interference-based proof systems* [6], certain proof steps may become applicable in a proof system only after redundant parts of the formula have been eliminated. We show that redundancy elimination by QRAT^+ or QRAT can lead to exponentially shorter proofs in the resolution based LQU^+-resolution [1] QBF calculus. Note that we do not compare the power of QRAT or QRAT^+ as proof systems themselves, but the impact of redundancy elimination on other proof systems. The following result relies only on QRATU, i.e., it does not require the more powerful redundancy property QRATU^+ in QRAT^+.

$\quad$ LQU^+-resolution is a calculus that generalizes traditional Q-resolution [15]. It allows to generate resolvents on both existential and universal variables and admits tautological resolvents of a certain kind. LQU^+-resolution is among the strongest resolution calculi currently known [1,2], yet the following class of QBFs provides an exponential lower bound on the size of LQU^+-resolution proofs.

Definition 16 ([2]). *For $n > 1$, let $\Phi_Q(n) := \Pi_Q(n).\psi_Q(n)$ be the* QUParity *QBFs with $\Pi_Q(n) := \exists x_1, \ldots, x_n \forall z_1, z_2 \exists t_2, \ldots, t_n$ and $\psi_Q(n) := C_0 \wedge C_1 \wedge \bigwedge_{i:=2}^{n} \mathcal{C}(i)$ where $C_0 := (z_1 \vee z_2 \vee t_n)$, $C_1 := (\bar{z}_1 \vee \bar{z}_2 \vee \bar{t}_n)$, and $\mathcal{C}(i) := \bigwedge_{j:=0}^{7} C_{i,j}$:*

$$C_{2,0} := (\bar{x}_1 \lor \bar{x}_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \qquad C_{i,0} := (\bar{t}_{i-1} \lor \bar{x}_i \lor z_1 \lor z_2 \lor \bar{t}_i)$$
$$C_{2,1} := (x_1 \lor x_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \qquad C_{i,1} := (t_{i-1} \lor x_i \lor z_1 \lor z_2 \lor \bar{t}_i)$$
$$C_{2,2} := (\bar{x}_1 \lor x_2 \lor z_1 \lor z_2 \lor t_2) \qquad C_{i,2} := (\bar{t}_{i-1} \lor x_i \lor z_1 \lor z_2 \lor t_i)$$
$$C_{2,3} := (x_1 \lor \bar{x}_2 \lor z_1 \lor z_2 \lor t_2) \qquad C_{i,3} := (t_{i-1} \lor \bar{x}_i \lor z_1 \lor z_2 \lor t_i)$$
$$C_{2,4} := (\bar{x}_1 \lor \bar{x}_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2) \qquad C_{i,4} := (\bar{t}_{i-1} \lor \bar{x}_i \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_i)$$
$$C_{2,5} := (x_1 \lor x_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2) \qquad C_{i,5} := (t_{i-1} \lor x_i \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_i)$$
$$C_{2,6} := (\bar{x}_1 \lor x_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor t_2) \qquad C_{i,6} := (\bar{t}_{i-1} \lor x_i \lor \bar{z}_1 \lor \bar{z}_2 \lor t_i)$$
$$C_{2,7} := (x_1 \lor \bar{x}_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor t_2) \qquad C_{i,7} := (t_{i-1} \lor \bar{x}_i \lor \bar{z}_1 \lor \bar{z}_2 \lor t_i)$$

Any refutation of $\Phi_Q(n)$ in LQU$^+$-resolution is exponential in n [2]. The QUParity formulas are a modification of the related *LQParity* formulas [2]. An LQParity formula is obtained from a QUParity formula $\Phi_Q(n)$ by replacing $\forall z_1, z_2$ in prefix $\Pi_Q(n)$ by $\forall z$ and by replacing every occurrence of the literal pairs $z_1 \lor z_2$ and $\bar{z}_1 \lor \bar{z}_2$ in the clauses in $\psi_Q(n)$ by the literal z and $\bar{z}$, respectively.

Proposition 6. QRATU *can eliminate either variable z_1 or z_2 from a QUParity formula $\Phi_Q(n)$ to obtain a related LQParity formula in polynomial time.*

Proof. We eliminate z_2 (z_1 can be eliminated alternatively) in a polynomial number of QRATU steps. Every clause C with $z_2 \in C$ has QRAT on z_2 since $\{z_1, \bar{z}_1\} \subseteq$ OR for all outer resolvents OR. UP immediately detects a conflict when propagating $\overline{\text{OR}}$. After eliminating all literals z_2, the clauses containing $\bar{z}_2$ trivially have QRAT on $\bar{z}_2$, which can be eliminated. Finally, z_1 including all of its occurrences is renamed to z. ◻

In the proof above the universal literals can be eliminated by QRATU in any order. Hence in this case the non-confluence [9,14] of rewrite rules in the QRAT and QRAT$^+$ systems is not an issue. LQU$^+$-resolution has polynomial proofs for LQParity formulas [2]. Hence the combination of QRATU and LQU$^+$-resolution results in a calculus that is more powerful than LQU$^+$-resolution. A related result [13] was obtained for the combination of QRATU and the weaker QU-resolution calculus [22].

7 Experiments

We implemented QRAT$^+$ redundancy removal in a tool called QRATPre+ for QBF preprocessing.[2] It applies rules QRATE$^+$ and QRATU$^+$ to remove redundant clauses and universal literals. We did not implement clause addition (QRATA$^+$) or extended universal reduction (EUR). QRATPre+ is the *first implementation* of QRAT$^+$ and QRAT for QBF preprocessing. The preprocessors HQSpre [24] and Bloqqer [10] (which generates partial QRAT proofs to trace preprocessing steps) do not apply QRAT to eliminate redundancies. The following experiments were run on a cluster of Intel Xeon CPUs (E5-2650v4, 2.20 GHz) running Ubuntu 16.04.1. We used the benchmarks from the PCNF track of the QBFEVAL'17 competition. In terms of scheduling the non-confluent (cf. [9,14]) rewrite rules

---

[2] Source code of QRATPre+: https://github.com/lonsing/qratpreplus

Table 1. Solved instances (S), solved unsatisfiable ($\bot$) and satisfiable ones ($\top$), and total wall clock time in kiloseconds (K) including time outs on instances from QBFEVAL'17. Different combinations of preprocessing by Bloqqer, HQSpre, and our tool QRATPre+.

(a) Original instances (no prepr.).

Solver	S	$\bot$	$\top$	Time
CAQE	170	128	42	656K
RAReQS	167	133	34	660K
DepQBF	152	108	44	690K
Qute	130	91	39	720K

(b) Prepr. by QRATPre+ only.

Solver	S	$\bot$	$\top$	Time
CAQE	209	141	68	594K
RAReQS	203	152	51	599K
DepQBF	157	109	48	689K
Qute	131	98	33	724K

(c) Prepr. by Bloqqer only.

Solver	S	$\bot$	$\top$	Time
RAReQS	256	180	76	508K
CAQE	251	168	83	522K
DepQBF	187	121	66	630K
Qute	154	109	45	682K

(d) Prepr. by Bloqqer and QRATPre+.

Solver	S	$\bot$	$\top$	Time
RAReQS	262	178	84	492K
CAQE	255	172	83	507K
DepQBF	193	127	66	622K
Qute	148	107	41	688K

(e) Prepr. by HQSpre only.

Solver	S	$\bot$	$\top$	Time
CAQE	306	197	109	415K
RAReQS	294	194	100	429K
DepQBF	260	171	89	494K
Qute	255	171	84	497K

(f) Prepr. by HQSpre and QRATPre+.

Solver	S	$\bot$	$\top$	Time
CAQE	314	200	114	407K
RAReQS	300	195	105	418K
DepQBF	262	177	85	488K
Qute	250	169	81	500K

$QRATE^+$ and $QRATU^+$, we have not yet optimized QRATPre+. Moreover, in general large numbers of clauses in formulas may cause run time overhead. In this respect, our implementation leaves room for improvements.

We illustrate the impact of QBF preprocessing by $QRAT^+$ and QRAT on the performance of QBF solving. To this end, we applied QRATPre+ in addition to the state of the art QBF preprocessors Bloqqer and HQSpre. In the experiments, first we preprocessed the benchmarks using Bloqqer and HQSpre, respectively, with a generous limit of two hours wall clock time. We considered 39 and 42 formulas where Bloqqer and HQSpre timed out, respectively, in their original form. Then we applied QRATPre+ to the preprocessed formulas with a *soft wall clock time limit* of 600 seconds. When QRATPre+ reaches the limit, it prints the formula with redundancies removed that have been detected so far. These preprocessed formulas are then solved. Table 1 shows the performance of our solver DepQBF [17] in addition to the top-performing solvers[3] RAReQS [11], CAQE [20], and Qute [19] from QBFEVAL'17, using limits of 7 GB and 1800 seconds wall clock time. The solvers implement different solving paradigms such as expansion or resolution-based QCDCL. The results clearly indicate the benefits of preprocessing by QRATPre+. The number of solved instances increases. Qute is an exception to

---

[3] We excluded the top-performing solver AIGSolve due to observed assertion failures.

this trend. We conjecture that QRATPre+ blurs the formula structure in addition to Bloqqer and HQSpre, which may be harmful to Qute.

We emphasize that we hardly observed a difference in the effectiveness of redundancy removal by $QRAT^+$ and QRAT on the considered benchmarks. The benefits of QRATPre+ shown in Table 1 are due to redundancy removal by QRAT already, and not by $QRAT^+$. However, on additional 672 instances from class *Gent-Rowley* (encodings of the Connect Four game) available from QBFLIB, $QRATE^+$ on average removed 54% more clauses than QRATE. We attribute this effect to larger numbers of quantifier blocks in the Gent-Rowley instances (median 73, average 79) compared to QBFEVAL'17 (median 3, average 27). The advantage of QBF abstractions in the $QRAT^+$ system is more pronounced on instances with many quantifier blocks.

8 Conclusion

We presented $QRAT^+$, a generalization of the QRAT proof system, that is based on a more powerful QBF redundancy property. The key difference between the two systems is the use of QBF specific unit propagation in contrast to propositional unit propagation. Due to this, redundancy checking in $QRAT^+$ is aware of quantifier structures in QBFs, as opposed to QRAT. Propagation in $QRAT^+$ potentially benefits from the presence of universal variables in the underlying formula. This is exploited by the use of abstractions of QBFs, for which we developed a theoretical framework, and from which the soundness of $QRAT^+$ follows. By concrete classes of QBFs we demonstrated that $QRAT^+$ is more powerful than QRAT in terms of redundancy detection. Additionally, we reported on proof theoretical improvements of a certain resolution based QBF calculus made by QRAT (or $QRAT^+$) redundancy removal. A first experimental evaluation illustrated the potential of redundancy elimination by $QRAT^+$.

As future work, we plan to implement a workflow for checking $QRAT^+$ proofs and extracting Skolem functions similar to QRAT proofs [10]. In our $QRAT^+$ preprocessor QRATPre+ we currently do not apply a specific strategy to handle the non-confluence of rewrite rules. We want to further analyze the effects of non-confluence as it may have an impact on the amount of redundancy detected. In our tool QRATPre+ we considered only redundancy removal. However, to get closer to the the full power of the $QRAT^+$ system, it may be beneficial to also add redundant clauses or universal literals to a formula.

References

1. Balabanov, V., Widl, M., Jiang, J.R.: QBF Resolution Systems and Their Proof Complexities. In: SAT. LNCS, vol. 8561, pp. 154–169. Springer (2014)
2. Beyersdorff, O., Chew, L., Janota, M.: Proof Complexity of Resolution-based QBF Calculi. In: STACS. LIPIcs, vol. 30, pp. 76–89. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2015)
3. Biere, A.: Resolve and Expand. In: SAT. LNCS, vol. 3542, pp. 59–70. Springer (2004)

4. Faymonville, P., Finkbeiner, B., Rabe, M.N., Tentrup, L.: Encodings of Bounded Synthesis. In: TACAS. LNCS, vol. 10205, pp. 354–370. Springer (2017)

5. Fazekas, K., Heule, M., Seidl, M., Biere, A.: Skolem Function Continuation for Quantified Boolean Formulas. In: TAP. LNCS, vol. 10375, pp. 129–138. Springer (2017)

6. Heule, M., Kiesl, B.: The Potential of Interference-Based Proof Systems. In: AR-CADE. EPiC Series in Computing, vol. 51, pp. 51–54. EasyChair (2017)

7. Heule, M., Kullmann, O.: The science of brute force. Commun. ACM 60(8), 70–79 (2017)

8. Heule, M., Seidl, M., Biere, A.: A Unified Proof System for QBF Preprocessing. In: IJCAR. LNCS, vol. 8562, pp. 91–106. Springer (2014)

9. Heule, M., Seidl, M., Biere, A.: Blocked Literals Are Universal. In: NASA Formal Methods. LNCS, vol. 9058, pp. 436–442. Springer (2015)

10. Heule, M., Seidl, M., Biere, A.: Solution Validation and Extraction for QBF Pre-processing. J. Autom. Reasoning 58(1), 97–125 (2017)

11. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.: Solving QBF with Coun-terexample Guided Refinement. Artif. Intell. 234, 1–25 (2016)

12. Järvisalo, M., Heule, M., Biere, A.: Inprocessing Rules. In: IJCAR. LNCS, vol. 7364, pp. 355–370. Springer (2012)

13. Kiesl, B., Heule, M., Seidl, M.: A Little Blocked Literal Goes a Long Way. In: SAT. LNCS, vol. 10491, pp. 281–297. Springer (2017)

14. Kiesl, B., Suda, M.: A Unifying Principle for Clause Elimination in First-Order Logic. In: CADE. LNCS, vol. 10395, pp. 274–290. Springer (2017)

15. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for Quantified Boolean Formulas. Inf. Comput. 117(1), 12–18 (1995)

16. Lonsing, F., Biere, A.: Failed Literal Detection for QBF. In: SAT. LNCS, vol. 6695, pp. 259–272. Springer (2011)

17. Lonsing, F., Egly, U.: DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL. In: CADE. LNCS, vol. 10395, pp. 371–384. Springer (2017)

18. Lonsing, F., Egly, U.: QRAT$^+$: Generalizing QRAT by a More Powerful QBF Redundancy Property. CoRR abs/1804.02908 (2018), https://arxiv.org/abs/1804.02908, IJCAR 2018 proceedings version with appendix

19. Peitl, T., Slivovsky, F., Szeider, S.: Dependency Learning for QBF. In: SAT. LNCS, vol. 10491, pp. 298–313. Springer (2017)

20. Rabe, M.N., Tentrup, L.: CAQE: A Certifying QBF Solver. In: FMCAD. pp. 136–143. IEEE (2015)

21. Samulowitz, H., Davies, J., Bacchus, F.: Preprocessing QBF. In: CP. LNCS, vol. 4204, pp. 514–529. Springer (2006)

22. Van Gelder, A.: Contributions to the Theory of Practical Quantified Boolean Formula Solving. In: CP. LNCS, vol. 7514, pp. 647–663. Springer (2012)

23. Wetzler, N., Heule, M., Hunt, W.A.J.: DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs. In: SAT. LNCS, vol. 8561, pp. 422–429. Springer (2014)

24. Wimmer, R., Reimer, S., Marin, P., Becker, B.: HQSpre - An Effective Preprocessor for QBF and DQBF. In: TACAS. LNCS, vol. 10205, pp. 373–390. Springer (2017)

25. Zhang, L., Malik, S.: Conflict Driven Learning in a Quantified Boolean Satisfiability Solver. In: ICCAD. pp. 442–449. ACM / IEEE Computer Society (2002)