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 existentials. 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) [22] 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 [13,24], and expansion of universal variables [3].

* Supported by the Austrian Science Fund (FWF) under grant S11409-N23. This article will appear in the proceedings of IJCAR 2018, LNCS, Springer, 2018.
A QRA T 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 QRA T redundancy property. Informally, checking whether a clause \( C \) has QRA T 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 QRA T 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 QRA T is unaware of the quantifier structure, which is entirely ignored in UP.

We generalize redundancy checking in QRA T 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 QRA T system which we call QRA T+. In contrast to QRA T, redundancy checking in QRA T+ is aware of the quantifier structure of a QBF. We show that (2) the redundancy property in QRA T+ based on QUP is more powerful than the one in QRA T based on UP. QRA T+ can detect redundancies which QRA T cannot. As a formal foundation, we introduce (3) a theory of QBF abstractions used in QRA T+. Redundancy elimination by QRA T+ or QRA T 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 QRA T or QRA T+ 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 QRA T+ and QRA T for QBF preprocessing. Our implementation of QRA T+ and QRA T 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 \( \Pi := Q_1B_1 \ldots Q_nB_n \) and a quantifier free CNF \( \psi \) not containing tautological clauses. The prefix consists of quantifier blocks \( Q_iB_i \), 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 \( \psi \) is defined precisely over
the variables \( \text{vars}(\phi) = \text{vars}(\psi) := B_1 \cup \ldots \cup B_n \) in \( \Pi \) so that all variables are quantified, i.e., \( \phi \) is closed. The quantifier \( Q(\Pi, l) \) of literal \( l \) is \( Q_i \) if the variable \( \text{var}(l) \) of \( l \) appears in \( B_i \). The set of variables in a clause \( C \) is \( \text{vars}(C) := \{ x \mid l \in C, \text{var}(l) = x \} \). A literal \( l \) is existential if \( Q(\Pi, l) = \exists \) and universal if \( Q(\Pi, l) = \forall \). If \( Q(\Pi, l) = Q_i \) and \( Q(\Pi, k) = Q_j \), then \( l \leq_H k \) iff \( i \leq j \). We extend the ordering \( \leq_H \) to an arbitrary but fixed ordering on the variables in every block \( B_i \).

An assignment \( \tau : \text{vars}(\phi) \rightarrow \{ \top, \bot \} \) maps the variables of a QBF \( \phi \) to truth constants \( \top \) (true) or \( \bot \) (false). Assignment \( \tau \) is complete if it assigns every variable in \( \phi \), otherwise \( \tau \) is partial. By \( \tau(\phi) \) we denote \( \phi \) under \( \tau \), where each occurrence of variable \( x \) in \( \phi \) is replaced by \( \tau(x) \) and \( x \) is removed from the prefix of \( \phi \), followed by propositional simplifications on \( \tau(\phi) \). We consider \( \tau \) 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 \( \textbf{10} \) \( T \) of a QBF \( \phi \) is a complete binary tree of depth \( |\text{vars}(\phi)| + 1 \) where the internal (non-leaf) nodes of each level are associated with a variable of \( \phi \). 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 \( \leq_H \) of the prefix \( \Pi \) of \( \phi \). 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 \( \tau \) in \( T \) from the root to an internal node (leaf) represents a partial (complete) assignment. A leaf at the end of \( \tau \) is labelled by \( \tau(\phi) \), i.e., the value of \( \phi \) under \( \tau \). 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 \( \phi \) is satisfiable (unsatisfiable) iff the root of \( T \) is labelled with \( \top \) (\( \bot \)).

Given a QBF \( \phi \) and its assignment tree \( T \), a subtree \( T' \) of \( T \) is a pre-model \( \textbf{10} \) of \( \phi \) 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 \( \phi \) is a model \( \textbf{10} \) of \( \phi \), denoted by \( T' \models_{\text{I}} \phi \), if each node in \( T' \) is labelled with \( \top \). A QBF \( \phi \) is satisfiable iff it has a model. Given a QBF \( \phi \) 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 \( \psi \) defined over a set \( B \) of variables without an explicit quantifier prefix. A model of a CNF \( \psi \) is a model \( \tau \) of the QBF \( \exists B \psi \) which consists only of the single path \( \tau \). We write \( \tau \models \psi \) if \( \tau \) is a model of \( \psi \). For CNFs \( \psi \) and \( \psi', \psi' \) is implied by \( \psi \) (\( \psi \models \psi' \)) if, for all \( \tau \), it holds that if \( \tau \models \psi \) then \( \tau \models \psi' \). Two CNFs \( \psi \) and \( \psi' \) are equivalent (\( \psi \equiv \psi' \)), iff \( \psi \models \psi' \) and \( \psi' \models \psi \). We define notation to explicitly refer to QBF models. For QBFs \( \phi \) and \( \phi' \), \( \phi' \) is implied by \( \phi \) (\( \phi \models_{\text{I}} \phi' \)) if, for all \( T \), it holds that if \( T \models_{\text{I}} \phi \) then \( T \models_{\text{I}} \phi' \). QBFs \( \phi \) and \( \phi' \) are equivalent (\( \phi \equiv_{\text{I}} \phi' \)) iff \( \phi \models_{\text{I}} \phi' \) and \( \phi' \models_{\text{I}} \phi \), and satisfiability equivalent (\( \phi \equiv_{\text{sat}} \phi' \)) iff \( \phi \) is satisfiable whenever \( \phi' \) is satisfiable. Satisfiability equivalence of CNFs is defined analogously and denoted by the same symbol ‘\( \equiv_{\text{sat}} \).’
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 \(\Pi\) is the clause \(OC(\Pi, C, l) := \{k \mid k \in C, k \leq_H l, k \neq l\}\).

The outer clause \(OC(\Pi, C, l) \subset C\) of \(C\) on \(l \in C\) contains only literals that are smaller than or equal to \(l\) in the variable ordering of prefix \(\Pi\), 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 \(\Pi\) is the clause \(QIOR(\Pi, C, D, l) := (C \setminus \{l\}) \cup OC(\Pi, D, \bar{l})\).

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

Definition 3 \([10]\). Clause \(C\) has property QIOR (quantified implied outer resolvent) on literal \(l \in C\) with respect to QBF \(\Pi.\psi\) iff \(\phi \models QIOR(\Pi, C, D, l)\) 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 \(\psi\) 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 \(\phi_0\). 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 $\psi$ and clause $C$, let $\psi \land \overline{C} \vdash \emptyset$ denote the fact that propositional unit propagation (UP) applied to $\psi \land \overline{C}$ produces the empty clause, where $\overline{C}$ is the conjunction of the negation of all the literals in $C$. If $\psi \land \overline{C} \vdash \emptyset$ then we write $\psi \vdash C$ to denote that $C$ can be derived from $\psi$ by UP (since $\psi \models C$).

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

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 4 by unit propagation ($\vdash$) 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 $\overline{l} \in D$, the outer resolvent $\text{OR}(\Pi, C, D, l)$ has AT with respect to CNF $\psi$.

Example 2. Consider $\phi := \exists x_1 \forall u \exists x_2. (C \land D)$ with $C := (x_1 \lor u \lor x_2)$ and $D := (x_1 \lor \overline{u} \lor \overline{x_2})$ 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 $\text{OR}(\Pi, C, D, x_2) = (x_1 \lor u \lor \overline{x_1} \lor \overline{u})$ 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 ($\vdash$), 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 ($\vdash$) 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 [20]: if we attach a quantifier prefix $\Pi$ to equivalent CNFs $\psi$ and $\psi'$, then the resulting QBFs are equivalent.
Proposition 1 (20). Given CNFs \( \psi \) and \( \psi' \) such that \( \text{vars}(\psi) = \text{vars}(\psi') \) and a quantifier prefix \( \Pi \) defined precisely over \( \text{vars}(\psi) \). If \( \psi \equiv \psi' \), then \( \Pi.\psi \equiv \Pi.\psi' \).

Proposition 2. If clause \( C \) has QIOR on literal \( l \in C \) with respect to QBF \( \Pi.\psi \), then \( \Pi.\psi \equiv_1 \Pi.(\psi \land \text{OR}(\Pi,C,D,l)) \) 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 \( \psi \models \text{OR}(\Pi,C,D,l) \) for all \( D \in \psi \) with \( \bar{l} \in D \), and further also \( \psi \equiv \psi \land \text{OR}(\Pi,C,D,l) \). Then \( \Pi.\psi \equiv_1 \Pi.(\psi \land \text{OR}(\Pi,C,D,l)) \) by Proposition 4.

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 (\( \equiv_1 \)), i.e., OR is implied by the QBF \( \Pi.\psi \) (\( \equiv_1 \)). 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 \( \Pi := Q_1B_1\ldots Q_iB_iQ_{i+1}B_{i+1}\ldots Q_nB_n \). For a clause \( C \), 
\[
\text{levels}(\Pi,C) := \{ i \mid \exists l \in C, Q(\Pi,l) = Q_i \}
\]

The abstraction of \( \Pi \) with respect to \( i \) with \( 0 \leq i \leq n \) produces the abstracted QBF \( \text{Abs}(\Pi,i) := \Pi \) for \( i = 0 \) and otherwise \( \text{Abs}(\Pi,i) := \exists (B_1 \cup \ldots \cup B_i)Q_{i+1}B_{i+1}\ldots Q_nB_n \). The abstraction of \( \phi \) with respect to \( i \) with \( 0 \leq i \leq n \) produces the abstracted QBF \( \text{Abs}(\phi,i) := \text{Abs}(\Pi,i).\psi \) with prefix \( \text{Abs}(\Pi,i) \).

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 \( \text{Abs}(\phi,0) = \phi \), \( \text{Abs}(\phi,1) = \text{Abs}(\phi,2) = \exists (B_1 \cup B_2) \forall B_3 \exists B_4 \psi \), \( \text{Abs}(\phi,3) = \text{Abs}(\phi,4) = \exists (B_1 \cup B_2 \cup B_3 \cup B_4).\psi \).

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

Proposition 3. Given a QBF \( \phi := \Pi.\psi \) with prefix \( \Pi := Q_1B_1\ldots Q_iB_i\ldots Q_nB_n \) and \( \text{Abs}(\phi,i) \) for some arbitrary \( i \) with \( 0 \leq i \leq n \). For all \( T \) and \( T^A \) we have that if \( T \models_1 \phi \) and \( T^A \subseteq T \) is a pre-model of \( \text{Abs}(\phi,i) \), then \( T^A \models_1 \text{Abs}(\phi,i) \).

\(^1\)In general, clauses \( C \) are always (implicitly) interpreted under a quantifier prefix \( \Pi \).
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 \( T \models T^A \subseteq T \) is a pre-model of \( \text{Abs}(\phi, i) \), then \( T^A \models \text{Abs}(\phi, i) \). Consider \( \text{Abs}(\phi, j) \) for \( j := i + 1 \), which is an abstraction of \( \text{Abs}(\phi, i) \). We have to show that, for all \( T \) and \( T^B \), we have that if \( T \models \phi \) and \( T^B \subseteq T \) is a pre-model of \( \text{Abs}(\phi, j) \), then \( T^B \models \text{Abs}(\phi, j) \).

We distinguish cases by the type of \( Q \) which are existential in \( \phi \). These existential variables appear along a single branch of the variables in \( \text{Abs} \) universal with respect to \( \varnothing \). Since \( \text{Abs} \) is a CNF, i.e., it does not contain any universal variables. Note that the full abstraction \( \text{Abs}(\phi, i) \) for \( i := n \) of a QBF \( \phi \) is a CNF, i.e., it does not contain any universal variables.

Lemma 1. Let \( \phi := II.\psi \) and \( \phi' := II.\psi' \) be QBFs with the same prefix \( II := Q_1B_1 \ldots Q_nB_n \). Then for all \( i \), if \( \text{Abs}(\phi, i) \equiv T \text{Abs}(\phi', i) \) 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 \( 0 \leq i < n \), i.e., if \( \text{Abs}(\phi, i) \equiv T \text{Abs}(\phi', i) \) then \( \phi \equiv T \phi' \). Let \( j := i + 1 \) and consider \( \text{Abs}(\phi, j) \) and \( \text{Abs}(\phi', j) \), which are abstractions of \( \text{Abs}(\phi, i) \) and \( \text{Abs}(\phi', i) \). We have \( \text{Abs}(II, i) = \exists(B_1 \cup \ldots \cup B_i)Q_1B_1 \ldots Q_nB_n \) and \( \text{Abs}(II, j) = \exists(B_1 \cup \ldots \cup B_j)Q_1B_1 \ldots Q_nB_n \). We show that if \( \text{Abs}(\phi, j) \equiv T \text{Abs}(\phi', j) \) then \( \phi \equiv T \phi' \) by IH. Assume that \( \text{Abs}(\phi, j) \equiv T \text{Abs}(\phi', j) \). We distinguish cases by the type of \( Q_j \) in \( \text{Abs}(II, i) \). If \( Q_j = \exists \) then \( \text{Abs}(II, i) = \text{Abs}(II, j) = \exists(B_1 \cup \ldots \cup B_j)Q_1B_1 \ldots Q_nB_n \), and hence \( \phi \equiv T \phi' \).

If \( Q_j = \forall \), then towards a contradiction, assume that \( \text{Abs}(\phi, j) \equiv T \text{Abs}(\phi', j) \) but \( \text{Abs}(\phi, i) \not\equiv T \text{Abs}(\phi', i) \). Then there exists \( T \) such that \( T \models \text{Abs}(\phi, i) \) but \( T \not\models \text{Abs}(\phi', i) \). Since \( T \not\models \text{Abs}(\phi', i) \) there exists a pre-model \( T^A \subseteq T \) of \( \text{Abs}(\phi', j) \) 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 \( \text{Abs}(II, j) \) are also labelled with \( \bot \). These existential variables appear along a single branch of the variables in \( \text{Abs} \) universal with respect to \( \varnothing \). Therefore we have \( T^A \not\models \text{Abs}(\phi', j) \). Since \( T \models \text{Abs}(\phi, i) \)
and $T^A \subseteq T$, we have $T^A \models \phi$ by Proposition 8 which contradicts the assumption that $Abs(\phi, j) \equiv_t Abs(\phi', j)$. □

The converse of Lemma 1 does not hold. From the equivalence of two QBFs $\phi$ and $\phi'$ 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 $UR(\Pi, C) := C \setminus \{i \in C \mid Q(\Pi, i) = \forall, \forall' \in C, Q(\Pi, i') = \exists : \var(i') \leq_H \var(l)\}.$

Definition 9 (QBF unit propagation, QUP). QBF unit propagation (QUP) extends UP (Definition 7) by applications of UR. For a QBF $\phi := \Pi.\psi$ and a clause $C$, let $\Pi.\psi \models \phi$ denote the fact that QUP applied to $\Pi.\psi$ produces the empty clause, where $\psi$ is the conjunction of the negation of all the literals in $C$. If $\Pi.\psi \models \phi$ and additionally $\Pi.\psi \models \Pi.\psi \wedge C$ then we write $\phi \models \Pi.\psi \wedge C$ to denote that $C$ can be derived from $\phi$ by QUP.

In contrast to UP (Definition 1), deriving the empty clause by QUP by propagating $\overline{C}$ on a QBF $\phi$ is not sufficient to conclude that $C$ is implied by $\phi$.

Example 4. Given the QBF $\Pi.\psi$ with prefix $\Pi := \forall u \exists x$ and CNF $\psi := (u \lor \overline{x}) \land (\overline{u} \lor x)$ and the clause $C := (x)$. We have $\Pi.((u \lor \overline{x}) \land (\overline{u} \lor x) \land \overline{x}) \models \phi$ since propagating $\overline{C} = (\overline{x})$ produces $\overline{u}$, 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 $\Pi.\psi \models \overline{\Pi.\psi}$.

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 $\phi$ with respect to $C$. Let $i = \max(\text{levels}(\Pi, C))$ 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., $Abs(\Pi.\psi \wedge C, i) \models \emptyset$, 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 [10].

Lemma 2. Let $\Pi.\psi$ be a QBF with prefix $\Pi := Q_1B_1 \ldots Q_nB_n$ and $C$ a clause such that $\var(C) \subseteq B_1$. If $\Pi.\psi \models \emptyset$ then $\Pi.\psi \models_t \Pi.\psi \wedge C$.

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 $\var(C) \subseteq B_1$ and $\Pi.\psi \wedge C \models \emptyset$, the QBF $\Pi.\psi \wedge C$ is unsatisfiable and in particular $T \not\models_t \Pi.\psi \wedge C$. Since $\tau(C) = \bot$, we have $\tau(\overline{C}) = \top$ and hence $T \models_t \Pi.\psi \wedge \overline{C}$, which is a contradiction. □
Lemma 3. Let $\Pi, \psi$ be a QBF, $C$ a clause, and $i = \text{max} (\text{levels}(\Pi, C))$. If $\text{Abs}(\Pi, (\psi \land \overline{C}), i) \models \emptyset$ then $\text{Abs}(\Pi, \psi, i) \equiv \text{Abs}(\Pi, (\psi \land C), i)$.

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

Lemma 4. Let $\Pi, \psi$ be a QBF, $C$ a clause, and $i = \text{max} (\text{levels}(\Pi, C))$. If $\text{Abs}(\Pi, (\psi \land \overline{C}), i) \models \emptyset$ then $\Pi, \psi \equiv \Pi, (\psi \land C)$.

Proof. By Lemma 3 and Lemma 1 □

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 QBFs. The abstractions are constructed depending on the maximum nesting level of variables in the clause we want to check.

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

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., $\text{Abs}(\Pi, (\psi \land \overline{C}), i) \models \emptyset$). Further, if $C$ has QAT then we have $\Pi, \psi \equiv \Pi, (\psi \land C)$ 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 $\psi := (u_1 \lor \overline{x}_3) \land (u_1 \lor \overline{x}_3 \lor x_4) \land (\overline{u}_2 \lor \overline{x}_4)$. Clause $(u_1 \lor x_3)$ has QAT with respect to $\text{Abs}(\phi, 2)$ with $\text{max} (\text{levels}(C)) = 2$ since $\forall u_2$ is still universal in the abstraction. By QUP clause $(u_1 \lor \overline{x}_3 \lor x_4)$ becomes unit and clause $(\overline{u}_2 \lor \overline{x}_4)$ becomes empty by UR. However, clause $(u_1 \lor x_3)$ does not have AT since $\forall u_2$ is treated as an existential variable in UP, hence clause $(\overline{u}_2 \lor \overline{x}_4)$ 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 $l \in D$, 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 $\Pi, \psi \equiv \Pi, (\psi \land \text{OR}(\Pi, C, D, l))$ for all $D \in \psi$ with $l \in D$.

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 the appendix for full proofs.
Definition 13 (prefix/suffix assignment [10]). For a QBF $\phi := \Pi.\psi$ and a complete assignment $\tau$ in the assignment tree of $\phi$, the partial prefix and suffix assignments of $\tau$ with respect to variable $x$, denoted by $\tau^x$ and $\tau_x$, respectively, are defined as $\tau^x := \{ y \mapsto \tau(y) \mid y \leq_H x, y \neq x \}$ and $\tau_x := \{ y \mapsto \tau(y) \mid y \leq_H x \}$.

For a variable $x$ from block $B_i$ of a QBF, Definition [13] allows us to split a complete assignment $\tau$ into three parts $\tau^x|\tau|\tau_x$, where the prefix assignment $\tau^x$ assigns variables (excluding $x$) from blocks smaller than or equal to $B_i$, $l$ is a literal of $x$, and the suffix assignment $\tau_x$ 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 $\text{var}(l) = x$. Let $T$ be a model of $\phi$ and $\tau \subseteq T$ be a path in $T$. If $r(C \setminus \{l\}) = \bot$ then $\tau^x(D) = \top$ for all $D \in \psi$ with $l \in D$.

Proof (sketch, see appendix). Let $D \in \psi$ be a clause with $\bar{l} \in D$ and $R := \text{OR}(\Pi, C, D, l) = (C \setminus \{l\}) \cup \text{OC}(\Pi, D, \bar{l})$. By Definition 12 we have $\Pi.\psi \equiv_{\Pi} \Pi.((\psi \land \text{OR}(\Pi, C, D, l))$ for all $D \in \psi$ with $\bar{l} \in D$. The rest of the proof considers a path $\tau$ 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 $\phi \equiv_{\text{sat}} \phi'$.

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

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 $\phi_0$. Let $\phi' := \Pi.(\psi \cup \{C'\})$ with $C' := C \setminus \{l\}$. Then $\phi \equiv_{\text{sat}} \phi'$.

Proof (sketch, see appendix). The proof relies on Lemma 5 and works in the same way as the proof of Theorem 8 in [10]. A model $T'$ of $\phi'$ is obtained from a model $T$ of $\phi$ by modifying the subtree under the node associated to variable $x = \text{var}(l)$. Suffix assignments of some paths $\tau$ 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 [9]), 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 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 QRAT$^+$ and QRATU$^+$ in the QRAT$^+$ system can eliminate certain redundancies that their counterparts QRAT 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.

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

$B_1 := \{ x_{4i+1}, x_{4i+2} \mid 0 \leq i < n \}$

$B_2 := \{ u_{2i+1} \mid 0 \leq i < n \}$

$B_3 := \{ x_{4i+3} \mid 0 \leq i < n \}$

$B_4 := \{ u_{2i+2} \mid 0 \leq i < n \}$

$B_5 := \{ x_{4i+4} \mid 0 \leq i < n \}$

$C_{i,0} := (x_{4i+1} \lor u_{2i+1} \lor \neg x_{4i+3})$

$C_{i,1} := (x_{4i+2} \lor \neg u_{2i+1} \lor x_{4i+3})$

$C_{i,2} := (\neg x_{4i+1} \lor \neg u_{2i+1} \lor \neg x_{4i+3})$

$C_{i,3} := (\neg x_{4i+2} \lor u_{2i+1} \lor x_{4i+3})$

$C_{i,4} := (\neg x_{4i+2} \lor \neg u_{2i+1} \lor \neg x_{4i+3})$

$C_{i,5} := (\neg u_{2i+2} \lor \neg x_{4i+4})$

$C_{i,6} := (\neg x_{4i+1} \lor u_{2i+2} \lor \neg x_{4i+4})$

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 $\psi_C(n) := C(0)$ with $C(0) := \bigwedge_{j=0}^{6} C_{0,j}$ as follows.

$C_{0,0} := (x_1 \lor u_1 \lor \neg x_3)$

$C_{0,1} := (x_2 \lor \neg u_1 \lor x_3)$

$C_{0,2} := (\neg x_1 \lor \neg u_1 \lor \neg x_3)$

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

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

$C_{0,5} := (\neg u_2 \lor \neg x_4)$

$C_{0,6} := (\neg x_1 \lor u_2 \lor \neg x_4)$

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


Proof (sketch). For i and k with i ≠ k, the sets of variables in \( C(i) \) and \( C(k) \) are disjoint. Thus it suffices to prove the claim for an arbitrary \( C(i) \). Clause \( C_{i,0} \) has QRAT\(^+\) on literal \( x_{4i+1} \) and can be removed. The relevant outer resolvents are \( OR_{2,2} = OR(\Pi_C(n), C_{i,0}, C_{i,2}, x_{4i+1}) \) and \( OR_{0,6} = OR(\Pi_C(n), C_{i,0}, C_{i,6}, x_{4i+1}) \), and we have \( OR_{2,2} = OR_{0,6} = (u_{2i+1} \lor \neg x_{4i+3}) \). Since \( \max(levels(OR_{0,2})) = max(levels(OR_{0,6})) = 3 \), we apply QUP to the abstraction \( Abs(\Phi_C(n), 3) \).

Propagating \( OR_{0,2} \) and \( OR_{0,6} \), 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 \( \Phi_L(n) := \Pi_L(n) \) be a class of QBFs with prefix \( \Pi_L(n) \) and CNF \( \psi_L(n) \) defined as follows.

\[
\Pi_L(n) := \forall B_1 \exists B_2 \forall B_3 \exists B_4:
\begin{align*}
B_1 &:= \{ u_{3i+1}, u_{3i+2} \mid 0 \leq i < n \} \quad B_3 := \{ u_{3i+3} \mid 0 \leq i < n \} \\
B_2 &:= \{ x_{3i+1}, x_{3i+2} \mid 0 \leq i < n \} \quad B_4 := \{ x_{3i+3} \mid 0 \leq i < n \}
\end{align*}
\]

\[
\psi_L(n) := \bigwedge_{i=0}^{n-1} C(i) \text{ with } C(i) := \bigwedge_{j=0}^{7} C_{i,j}:
\begin{align*}
C_{i,0} &:= (\neg u_{3i+2} \lor \neg x_{3i+1} \lor x_{3i+3}) \quad C_{i,4} := (\neg x_{3i+1} \lor \neg x_{3i+2} \lor x_{3i+3}) \\
C_{i,1} &:= (\neg u_{3i+1} \lor \neg x_{3i+1} \lor x_{3i+2}) \quad C_{i,5} := (u_{3i+3} \lor \neg x_{3i+3}) \\
C_{i,2} &:= (u_{3i+1} \lor x_{3i+1} \lor \neg x_{3i+2}) \quad C_{i,6} := (\neg x_{3i+1} \lor x_{3i+1} \lor \neg x_{3i+3}) \\
C_{i,3} &:= (u_{3i+2} \lor x_{3i+1} \lor x_{3i+2}) \quad C_{i,7} := (\neg u_{3i+3} \lor x_{3i+3})
\end{align*}
\]

**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). 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 \[9\], 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 \[10\] 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\(^+\).
LQU\(^+\)-resolution is a calculus that generalizes traditional Q-resolution. 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, 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 \(\Phi_\Omega(n) := \Pi_\Omega(n) \cdot \psi_\Omega(n)\) be the QUParity QBFs with \(\Pi_\Omega(n) := \exists x_1, \ldots, x_n \forall z_1, z_2 \exists t_2, \ldots, t_n\) and \(\psi_\Omega(n) := C_0 \land C_1 \land \land_{i=2}^{n} C(i)\) where \(C_0 := (z_1 \lor z_2 \lor t_n)\), \(C_1 := (\bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_n)\), and \(C(i) := \land_{j=0}^{7} C_{i,j}\):

\[
C_{2,0} := (\bar{x}_1 \lor \bar{x}_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \\
C_{2,1} := (x_1 \lor x_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \\
C_{2,2} := (\bar{x}_1 \lor x_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \\
C_{2,3} := (x_1 \lor \bar{x}_2 \lor z_1 \lor z_2 \lor \bar{t}_2) \\
C_{2,4} := (\bar{x}_1 \lor \bar{x}_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2) \\
C_{2,5} := (x_1 \lor x_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2) \\
C_{2,6} := (\bar{x}_1 \lor x_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2) \\
C_{2,7} := (x_1 \lor \bar{x}_2 \lor \bar{z}_1 \lor \bar{z}_2 \lor \bar{t}_2)
\]

Any refutation of \(\Phi_\Omega(n)\) in LQU\(^+\)-resolution is exponential in \(n\). The QUParity formulas are a modification of the related LQParity formulas. An LQParity formula is obtained from a QUParity formula \(\Phi_\Omega(n)\) by replacing \(\forall z_1, z_2\) in prefix \(\Pi_\Omega(n)\) 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 \(\psi_\Omega(n)\) by the literal \(z\) and \(\bar{z}\), respectively.

**Proposition 6.** QRA\(TU\) can eliminate either variable \(z_1\) or \(z_2\) from a QUParity formula \(\Phi_\Omega(n)\) 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 QRA\(TU\) steps. Every clause \(C\) with \(z_2 \in C\) has QRA\(TU\) on \(z_2\) since \(\{z_1, \bar{z}_1\} \subseteq \text{OR}\) for all outer resolvents \(\text{OR}\). UP immediately detects a conflict when propagating \(\text{OR}\). After eliminating all literals \(z_2\), the clauses containing \(\bar{z}_2\) trivially have QRA\(TU\) on \(\bar{z}_2\), which can be eliminated. Finally, \(z_1\) including all of its occurrences is renamed to \(z\). \(\square\)

In the proof above the universal literals can be eliminated by QRA\(TU\) in any order. Hence in this case the non-confluence of rewrite rules in the QRA and QRA\(^+\) systems is not an issue. LQU\(^+\)-resolution has polynomial proofs for LQParity formulas. Hence the combination of QRA\(TU\) and LQU\(^+\)-resolution results in a calculus that is more powerful than LQU\(^+\)-resolution. A related result was obtained for the combination of QRA\(TU\) and the weaker QU-resolution calculus.

### 7 Experiments

We implemented QRA\(^+\) redundancy removal in a tool called QRA\(Pre^+\) for QBF preprocessing\(^2\). It applies rules QRATE\(^+\) and QRA\(TU^+\) to remove redundant

\(^2\) Source code of QRA\(Pre^+\): [https://github.com/lonsing/qratpreplus](https://github.com/lonsing/qratpreplus)
Table 1. Solved instances (S), solved unsatisfiable (⊥) and satisfiable ones (⊤), and total wall clock time in kiloseconds (K) including timeouts on instances from QBFEVAL’17. Different combinations of preprocessing by Bloqqer, HQSpre, and our tool QRA TPre+.

(a) Original instances (no prepr.).

| Solver | S | ⊥ | ⊤ | Time |
|--------|---|---|---|------|
| CAQE   | 170 | 128 | 42 | 656K |
| RAReQS | 167 | 133 | 34 | 660K |
| DepQBF | 152 | 108 | 44 | 690K |
| Qute   | 130 | 91  | 39 | 720K |

(b) Prepr. by QRA TPre+ only.

| Solver | S | ⊥ | ⊤ | 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 | ⊥ | ⊤ | 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 QRA TPre+.

| Solver | S | ⊥ | ⊤ | 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 | ⊥ | ⊤ | 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 QRA TPre+.

| Solver | S | ⊥ | ⊤ | Time |
|--------|---|---|---|------|
| CAQE   | 314 | 200 | 114 | 407K |
| RAReQS | 300 | 195 | 105 | 418K |
| DepQBF | 262 | 177 | 85  | 488K |
| Qute   | 250 | 169 | 81  | 500K |

We did not implement clause addition (QRATA+) or extended universal reduction (EUR). QRA TPre+ is the first implementation of QRAT+ and QRAT for QBF preprocessing. The preprocessors HQSpre [23] 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 QRATE+ and QRATU+, we have not yet optimized QRA TPre+. Moreover, in general large numbers of clauses in formulas may cause runtime 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 QRA TPre+ 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 QRA TPre+ to the preprocessed formulas with a soft wall clock time limit of 600 seconds. When QRA TPre+ 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 RAREQS [11], CAQE [19], and Qute [18] 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 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 full power of the QRAT* system, it may be beneficial to also add redundant clauses or universal literals to a formula.

3 We excluded the top-performing solver AIGSolve due to observed assertion failures.
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: ARCADE. 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 Preprocessing. J. Autom. Reasoning 58(1), 97–125 (2017)
11. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.: Solving QBF with Counterexample 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. Peitl, T., Slivovsky, F., Szeider, S.: Dependency Learning for QBF. In: SAT. LNCS, vol. 10491, pp. 298–313. Springer (2017)
19. Rabe, M.N., Tentrup, L.: CAQE: A Certifying QBF Solver. In: FMCAD. pp. 136–143. IEEE (2015)
20. Samulowitz, H., Davies, J., Bacchus, F.: Preprocessing QBF. In: CP. LNCS, vol. 4204, pp. 514–529. Springer (2006)
21. Van Gelder, A.: Contributions to the Theory of Practical Quantified Boolean Formula Solving. In: CP. LNCS, vol. 7514, pp. 647–663. Springer (2012)
22. 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)
23. 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)
24. Zhang, L., Malik, S.: Conflict Driven Learning in a Quantified Boolean Satisfiability Solver. In: ICCAD. pp. 442–449. ACM / IEEE Computer Society (2002)
A Appendix

A.1 Example: Formula $\Phi_L(1)$

The following example shows formula $\Phi_L(1)$ from the class $\Phi_L(n)$, which illustrates that QRATU$^+$ is more powerful than QRATU.

Example 7 (related to Definition [12] on page [12]). For $n := 1$, we have $\Phi_L(n)$ with prefix $\Pi_L(n) := \forall u_1, u_2 \exists x_1, x_2 \forall u_3 \exists x_3$ and CNF $\psi_L(n) := C(0)$ with $C(0) := \bigwedge_{j=0}^{7} C_{0,j}$ as follows.

$C_{0,0} := (\neg u_2 \lor \neg x_1 \lor \neg x_2)$
$C_{0,1} := (\neg u_1 \lor \neg x_1 \lor x_2)$
$C_{0,2} := (u_1 \lor x_1 \lor \neg x_2)$
$C_{0,3} := (u_2 \lor x_1 \lor x_2)$
$C_{0,4} := (\neg x_1 \lor x_2 \lor x_3)$
$C_{0,5} := (u_3 \lor \neg x_3)$
$C_{0,6} := (\neg x_1 \lor x_2 \lor \neg x_3)$
$C_{0,7} := (\neg u_3 \lor x_3)$

A.2 Proofs

Proof of Lemma [5] on page [10]

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

Proof (similar to proof of Lemma 6 in [10]). Let $D \in \psi$ be a clause with $\bar{l} \in D$ and consider $R := \text{OR}(\Pi, C, D, \bar{l}) = (C \{ l \}) \cup \text{OC}(\Pi, D, \bar{l})$. By Definition [12] we have $\Pi.\psi \equiv_I \Pi.\psi \cup \text{OR}(\Pi, C, D, \bar{l})$ for all $D \in \psi$ with $\bar{l} \in D$. Let $T$ be a model of $\Pi.\psi$ and $\tau \subseteq T$ a path in $T$. Since $T \models \Pi.\psi$ and $T \models \Pi.\psi \cup \text{OR}(\Pi, C, D, \bar{l})$, we have $\tau(\psi) = \top$ and $\tau(R) = \top$. Assuming that $\tau(C \{ l \}) = \bot$, we have $\tau(\text{OC}(\Pi, D, \bar{l})) = \top$ since $\tau(R) = \top$. The clause $\text{OC}(\Pi, D, \bar{l})$ does not contain $\bar{l}$ and it contains only literals of variables from blocks smaller than or equal to the block containing $x$. Hence we have $\tau^x(\text{OC}(\Pi, D, \bar{l})) = \top$ for the prefix assignment $\tau^x$, and further $\tau^x(D) = \top$ since $\text{OC}(\Pi, D, \bar{l}) \subseteq D$. $\square$

Proof of Theorem [3] on page [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 $\psi' := \psi \cup \{C\}$ and $\Pi'$ is the same as $\Pi$ with variables and respective quantifiers removed that no longer appear in $\psi'$. Then $\phi \equiv_{\text{sat}} \phi'$.

Proof (similar to proof of Theorem 7 in [10]). We can adapt the prefix $\Pi'$ of $\phi'$ to be the same as the prefix of $\phi$ in a satisfiability-preserving way. If $\phi$ is satisfiable then $\phi'$ is also satisfiable since every model of $\phi$ is also a model of $\phi'$. Let $T'$ be a model of $\phi'$ and $T^P := T'$ a pre-model of $\phi$. Consider paths $\tau \subseteq T^P$ in $T^P$ for which we have $\tau(\psi') = \top$ but $\tau(C) = \bot$, where $\tau = \tau^x l_x$ for $\text{var}(l) = x$ and $l \in C$. Since $\tau(C) = \bot$ also $\tau(C \{ l \}) = \bot$, and due to Lemma [4] we have $\tau^x(D) = \top$ for all $D \in \psi$ with $\bar{l} \in D$. We construct a pre-model $T$ of $\phi$ from $T^P$ by modifying all such paths $\tau \subseteq T^P$ by flipping the
assignment of \( x \) to obtain \( \tau' := \tau^2 l x \) such that \( \tau' \subseteq T \). (If we process multiple redundant clauses \( C \), then cyclic modifications by assignment flipping cannot occur if we do the modifications in reverse chronological ordering as the clauses were detected redundant. This is the same principle of reconstructing solutions when using blocked clause elimination, for example.) Now \( \tau'(C) = \top \) and also \( \tau'(D) = \top \) since \( \tau^2(D) = \top \), and \( \tau \) and \( \tau' \) have the same prefix assignment \( \tau^x \). Hence \( T \models_1 \phi \) and thus \( \phi \) is satisfiable. \( \Box \)

Proof of Theorem 4 on page 10:

**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 \( \phi_0 \). Let \( \phi' := \Pi.(\psi \cup \{C'\}) \) with \( C' := C \setminus \{l\} \). Then \( \phi \equiv_{sat} \phi' \).

**Proof** (similar to proof of Theorem 8 in [10]). If \( \phi' \) is satisfiable then \( \phi \) is also satisfiable since every model of \( \phi' \) is also a model of \( \phi \). Let \( T \) be a model of \( \phi \) and \( T^P := T \) be a pre-model of \( \phi' \). Consider paths \( \tau \subseteq T^P \) in \( T^P \) for which we have \( \tau(\psi) = \top \) and \( \tau(C) = \top \) but \( \tau(C') = \bot \). Since \( C' = C \setminus \{l\} \), we have \( \tau = \tau^2 l x \) for \( \text{var}(l) = x \) and \( l \in C \). Since \( C \) is universal, for every such \( \tau \) there exists a path \( \tau' \subseteq T^P \) with \( \tau' = \tau^2 \rho_x \), with \( \tau_x \) and \( \rho_x \) being different suffix assignments of \( \tau \) and \( \tau' \), respectively. We have \( \tau'(\psi) = \top \) and \( \tau'(C) = \top \) since \( \tau' \subseteq T \) because \( T^P = T \), and also \( \tau'(C') = \bot \) because \( l \in C \) but \( l \in \tau' \). Hence \( C' \) is satisfied by \( \tau' \) due to some assignment \( k \in \rho_x \). Due to \( \tau(C') = \top \) and Lemma 5 we have \( \tau(D) = \top \) for all \( D \in \psi \) with \( l \in D \) and hence also \( \tau(D) = \top \) since \( \tau \) and \( \tau' \) have the same prefix assignment \( \tau^x \). We construct a pre-model \( T' \) of \( \phi' \) from \( T^P \) by modifying all paths \( \tau = \tau^2 l x \) for which \( \tau(C') = \bot \) to be \( \tau'' := \tau^2 \rho_x \), where \( \rho_x \) is the suffix assignment of path \( \tau' = \tau^2 \rho_x \) that corresponds to the other branch \( l \) of the universal literal \( l \). These modifications in fact are a replacement of the subtree under \( \tau^x l \). (As noted in the proof of Theorem 8 above, cyclic modifications cannot occur if we process multiple redundant clauses \( C \), provided that we do the modifications in reverse chronological ordering as the clauses were detected redundant.) We have \( \tau''(C') = \top \) due to its suffix assignment \( \rho_x \), and also \( \tau''(\psi) = \top \). Therefore, \( T' \models_1 \phi' \) and hence \( \phi' \) is satisfiable. \( \Box \)

Extended Proof Sketch of Proposition 5 on page 12:

**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).** Formulas \( \Phi_L(n) \) are constructed based on a similar principle as \( \Phi_C(n) \) in Definition 14. For \( i \) and \( k \) with \( i \neq k \), the sets of variables in \( C(i) \) and \( C(k) \) are disjoint. Thus it suffices to prove the claim for an arbitrary \( C(i) \). Clause \( C_{i,0} \) has QRAT\(^+\) on literal \( \neg u_{3i+2} \). The relevant outer resolvent is \( OR_{0,3} = OR(II_L(n), C_{i,0}, C_{i,3}, \neg u_{3i+2}) = (\neg x_{3i+1} \lor \neg x_{3i+2}) \). We have \( \max(levels(OR_{0,3})) = 2 \), and variable \( u_{3i+3} \) is universal in \( Abs(\Phi_L(n), 2) \). Propagating \( OR_{0,3} \) makes \( C_{i,4} \) unit, finally \( C_{i,5} \) becomes empty under the derived assignment \( x_{3i+3} \) and since UR reduces the literal \( u_{3i+3} \). After removing literal
\neg u_{3i+2} \text{ from } C_{i,0}, \text{ clause } C_{i,3} \text{ trivially has QRAT}^+ \text{ on } u_{3i+2}. \text{ The literals of variable } u_{3i+1} \text{ in } C_{i,1} \text{ and } C_{i,2} \text{ can be eliminated in a similar way, where clause } C_{i,7} \text{ becomes empty by UR in QUP. In contrast to QRATU}^+, \text{ QRATU cannot eliminate any universal literals in } \Phi_L(n). \text{ Clauses } C_{i,5} \text{ and } C_{i,7} \text{ in } \Phi_L(n) \text{ do not become empty. All variables are existential since UP is applied. The claim can be proved by case analysis of all possible outer resolvents.} \Box