A Unified Proof System for QBF Preprocessing*

Marijn J.H. Heule[1], Martina Seidl[2], and Armin Biere[2]

[1] Department of Computer Science, The University of Texas at Austin
marijn@cs.utexas.edu
[2] Institute for Formal Models and Verification, JKU Linz
martina.seidl@jku.at biere@jku.at

Abstract. For quantified Boolean formulas (QBFs), preprocessing is essential to solve many real-world formulas. The application of a preprocessor, however, prevented the extraction of proofs for the original formula. Such proofs are required to independently validate correctness of the preprocessor's rewritings and the solver's result. Especially for universal expansion proof checking was not possible so far. In this paper, we introduce a unified proof system based on three simple and elegant *quantified resolution asymmetric tautology* (QRAT) rules. In combination with an extended version of universal reduction, they are sufficient to efficiently express *all* preprocessing techniques used in state-of-the-art preprocessors including universal expansion. Moreover, these rules give rise to new preprocessing techniques. We equip our preprocessor bloqqer with QRAT proof logging and provide a proof checker for QRAT proofs.

1 Introduction

Effectively checking the result returned by a QBF solver has been an open challenge for a long time [1,2,3,4,5,6,7]. The current state-of-the-art is to simply dump Q-resolution proofs and to validate their structure. This approach has two major drawbacks. On the one hand the proofs might get extremely large and cannot be produced due to technical limitations. On the other hand, there are solving and preprocessing techniques for which it is not known if and how they translate to Q-resolution.

Due to the diversity of techniques in state-of-the-art preprocessors [8,9], it is not straightforward to provide a checker which verifies the output of the preprocessor. In fact, it would be preferable to translate the different preprocessing techniques to a canonical representation which then can be checked easily. Some efforts go in this direction by using Q-resolution. If a resolution proof is available, then checking is polynomial w.r.t. the proof size. However, the proof itself might become exponentially large and already writing down the proof might be costly.

* This work was supported by the Austrian Science Fund (FWF) through the national research network RiSE (S11408-N23), Vienna Science and Technology Fund (WWTF) under grant ICT10-018, DARPA contract number N66001-10-2-4087, and the National Science Foundation under grant number CCF-1153558.

Furthermore, it is not known for all preprocessing techniques how to express them in terms of resolution, what is the case for universal expansion [10].

In propositional logic, the RUP proof checking format [11] is extremely successful because it simply logs the learnt clauses and provides an easy checking criterion. For optimization purposes, recently, the DRUP extension has been presented [12] which provides elimination criteria for redundant clauses. It has been recognized that RUP and DRUP can be characterized with the *resolution asymmetric tautology property* (RAT) which has been originally developed in the context of propositional preprocessing for characterizing and comparing the strength of the various techniques. In this paper, we extend RAT [13] to QRAT, the *quantified resolution asymmetric tautology property*, and introduce novel clause addition and elimination techniques using QRAT. On this basis, we capture the state-of-the-art preprocessing techniques in a uniform manner what allows us to develop a checker verifying the correctness of a QBF preprocessor. Moreover, checking QRAT proofs is polynomial in the proof size. We integrated QRAT-based tracing in our preprocessor bloqqer [8] and implemented an efficient checker for QRAT proofs.

2 Preliminaries

We consider QBFs in prenex conjunctive normal form (PCNF). A QBF in PCNF has the structure $\Pi.\psi$ where the prefix Π has the form $Q_1X_1Q_2X_2\ldots Q_nX_n$ with disjoint variable sets X_i and $Q_i \in \{\forall, \exists\}$. The formula ψ is a propositional formula in conjunctive normal form, i.e., a conjunction of clauses. A clause is a disjunction of literals and a literal is either a variable (positive literal) or a negated variable (negative literal). The variable of a literal is denoted by $\mathsf{var}(l)$ where $\mathsf{var}(l) = x$ if $l = x$ or $l = \bar{x}$. The negation of a literal l is denoted by $\bar{l}$. The quantifier $\mathsf{Q}(\Pi, l)$ of a literal l is Q_i if $\mathsf{var}(l) \in X_i$. Let $\mathsf{Q}(\Pi, l) = Q_i$ and $\mathsf{Q}(\Pi, k) = Q_j$, then $l \leq_\Pi k$ if $i \leq j$. We sometimes write formulas in CNF as sets of clauses and clauses as sets of literals. We consider only closed QBFs, so ψ contains only variables which occur in the prefix. The variables occurring in the prefix of ϕ are given by $\mathsf{vars}(\phi)$. The subformula ψ_l consisting of all clauses of matrix ψ containing literal l is defined by $\psi_l = \{C \mid l \in C, C \in \psi\}$. By $\top$ and $\bot$ we denote the truth constants true and false. QBFs are interpreted as follows: a QBF $\forall x\Pi.\psi$ is false iff $\Pi.\psi[x/\top]$ or $\Pi.\psi[x/\bot]$ is false where $\Pi.\psi[x/t]$ is the QBF obtained by replacing all occurrences of variable x by t. Respectively, a QBF $\exists x\Pi.\psi$ is false iff both $\Pi.\psi[x/\top]$ and $\Pi.\psi[x/\bot]$ are false. If the matrix ψ of a QBF ϕ contains the empty clause after eliminating the truth constants according to standard rules, then ϕ is false. Accordingly, if the matrix ψ of QBF ϕ is empty, then ϕ is true. Two QBFs are *satisfiability equivalent* iff they have the same truth value.

Models and countermodels of QBFs can either be described intensionally in form of Herbrand and Skolem functions [1] or extensionally in form of subtrees of assignment trees. An *assignment tree* of a QBF ϕ is a complete binary tree of depth $|\mathsf{vars}(\phi) + 1|$ where the non-leaf nodes of each level are *associated* with

a variable of ϕ. The order of the associated variables in the tree respects the order of the variables in the prefix of ϕ. A non-leaf node associated with variable x has one outgoing edge *labelled* with x and one outgoing edge *labelled* with $\bar{x}$. Each path starting from the root of the tree represents a (partial) variable assignment. We also write a path as a sequence of literals. A path τ from the root node to a leaf is a complete assignment and the leaf is labelled with the value of the QBF under τ. Nodes associated with existential variables act as OR-nodes, while universal nodes act as AND-nodes. Respectively, a node is labelled either with $\top$ or with $\bot$. A QBF is true (satisfiable) iff its root is labelled with $\top$. A QBF is false (unsatisfiable) iff its root is labelled with $\bot$. By τ^x and τ_x we denote the partial assignments obtained from the complete assignment τ with $\tau = \tau^x l \tau_x$ where $\mathsf{var}(l) = x$. A QBF ϕ with $\mathsf{vars}(\phi) = \{x_1, \ldots, x_n\}$ under (partial) assignment τ is the QBF $\phi[x_1/t_1, \ldots, x_n/t_n]$ where $t_i = \top$ if $x_i \in \tau$, $t_i = \bot$ if $\bar{x}_i \in \tau$, and $t_i = x_i$ otherwise.

Example 1. Consider the QBF $\exists a \forall b \exists c \forall d \exists e.(a \vee b \vee \bar{c} \vee \bar{d} \vee e)$ and the path from the root $\tau = a\bar{b}\bar{c}d\bar{e}$. Then we have partial assignments $\tau^c = a\bar{b}$ and $\tau_c = d\bar{e}$.

A *pre-model* M of QBF ϕ is a subtree of the assignment tree of ϕ such that (1) for each universal node in M, both children are in M; (2) for each existential node in M, exactly one of the children is in M; and (3) the root of the assignment tree is in M. A pre-model M of QBF ϕ is a *model* of ϕ if in addition each node in M is labelled with $\top$. Obviously, only a true QBF can have a model. A false QBF has at least one countermodel, which is defined dually as follows. In a *pre-countermodel* M existential nodes have two children, whereas universal nodes have only one and the root of the assignment tree is in M. A pre-countermodel M is a *countermodel* if each node is labelled with $\bot$. Two QBFs are *logically equivalent* iff they have the same set of (counter) models modulo variable names.

3 QRAT: Quantified Resolution Asymmetric Tautologies

The QRAT proof system, introduced below, provides the basis for satisfiability equivalence preserving clause addition, clause deletion, and clause modification techniques. To this end, we first have to recapitulate the notion of *QBF resolvents* (resolvents for short) and introduce the concept of *asymmetric literal addition*.

Definition 1 (Resolvent). *Given two non-tautological clauses C and D with $x \in C$ and $\bar{x} \in D$, the resolvent over pivot variable x is $(C \backslash \{x\}) \cup (D \backslash \{\bar{x}\})$.*

Note that we do not restrict the pivot element to existential variables as it is usually done in the literature. Furthermore, for the moment, we do not consider universal reduction rule necessary for the completeness of Q-resolution.

Definition 2 (Asymmetric Literal Addition). *Given a QBF $\Pi.\psi$ and a clause C. The clause $\mathsf{ALA}(\psi, C)$ is the unique clause obtained by repeatedly applying the extension rule $C := C \cup \{\bar{l}\}$ if $\exists l_1, \ldots, l_k \in C$ and $(l_1 \vee \ldots \vee l_k \vee l) \in \psi$ called* asymmetric literal addition *to C until fixpoint.*

Asymmetric literal addition is well understood for propositional logic [14]. For QBF, a variant called *hidden literal addition* has been described in [8] where it is (unnecessarily) required that the l_i occur to the left of l in the prefix.

The new definition for QBF used in this paper is the same in the propositional case. Thus $\phi[C/C']$ with $C' = \mathsf{ALA}(\psi, C)$ has exactly the same (propositional) models as ϕ, which lifts to QBF equivalence, since the values the leaves of assignment trees do not change. As consequence we have the following lemma.

Lemma 1. *Let $\phi = \Pi.\psi \cup \{C\}$ be a QBF and $C' = \mathsf{ALA}(\psi, C)$ be obtained from C by asymmetric literal addition. Further, let $\phi' = \phi[C/C']$. Then ϕ and ϕ' are logically equivalent.*

A clause C is called an *asymmetric tautology* (AT) w.r.t. ψ if $\mathsf{ALA}(\psi, C)$ is a tautology. ALA, AT, and resolution as introduced above are sufficient to define the RAT proof system for propositional logic. For QBFs, we must additionally consider quantifier dependencies which we capture by the notion of *outer clauses* and *outer resolvents*.

Definition 3 (Outer Clause). *Let C be a clause occurring in QBF $\Pi.\psi$. The* outer clause *of C on literal $l \in C$, denoted by $\mathcal{O}(\Pi, C, l)$, is given by the clause $\{k \mid k \in C, k \leq_\Pi l, k \neq l\}$.*

Definition 4 (Outer Resolvent). *Let C be a clause with $l \in C$ and and D a clause occurring in QBF $\Pi.\psi$ with $\bar{l} \in D$. The* outer resolvent *of C with D on literal l w.r.t. Π, denoted by $\mathcal{R}(\Pi, C, D, l)$, is given by the clause $O \cup (C \backslash \{l\})$ if $Q(\Pi, l) = \forall$ and by $O \cup C$ if $Q(\Pi, l) = \exists$ assuming $O = \mathcal{O}(\Pi, D, \bar{l})$.*

Definition 5 (Quantified Resolution Asymmetric Tautology (QRAT)). *Given a QBF $\Pi.\psi$ and a clause C. Then C has QRAT on literal $l \in C$ with respect to $\Pi.\psi$ iff it holds for all $D \in \psi_{\bar{l}}$ that $\mathsf{ALA}(\psi, R)$ is a tautology for the outer resolvent $R = \mathcal{R}(\Pi, C, D, l)$.*

The intuition behind these definitions is almost identical to the propositional case [15]: consider potential resolvents of a clause on a certain literal with resolution candidates containing the negation of the picked literal. If all of them are "redundant", or more precisely asymmetric tautologies in the context of this paper, then this clause is redundant too and can be added or removed.

The important difference to the propositional case is that inner variables, w.r.t. the pivot variable resolved upon, might have different values for different choices of universal literals, and thus one can not simply apply resolution blindly before checking for redundancy of the resolvent. Inner literals in the resolution candidates should be ignored. This is the same restriction as for quantified blocked clauses [8]. As it turns out, for existential pivots, it is possible to have a slightly more general version, i.e., the pivot literal can be included in the outer resolvent, while in previous work this was not the case, and for universal pivots, it is not allowed. The QRAT proof system uses this observation to establish syntactical redundancy detection criteria to safely add, remove, and modify clauses.

Lemma 2. *Given a clause C which has* QRAT *w.r.t. a QBF $\Pi.\psi$ on an existential literal $l \in C$ with $\mathsf{var}(l) = x$. If there is an assignment $\sigma = \tau^x \bar{l} \tau_x$ that falsifies C, but satisfies ψ then the assignment τ^x satisfies all $D \in \psi$ with $\bar{l} \in D$.*

Proof. Let $D \in \psi$ be a clause with $\bar{l} \in D$, $\sigma(C) = \bot$, and $O = \mathcal{O}(\Pi, D, \bar{l})$. In order to show $\tau^x(D) = \top$ by contradiction we assume that $\tau^x(O) = \bot$. This leads to $\sigma(R) = \bot$ for the outer resolvent $R = O \cup C$ too (note that we do not remove l from C). By induction on the order of literals added to R in computing $\mathsf{ALA}(\psi, R)$ we show that $\sigma(l') = \bot$ for all literals l' in $\mathsf{ALA}(\psi, R)$. This is clear for all $l' \in R$. Assume $l_1, \ldots, l_{k-1}$ are from R or have been added through ALA extensions and further assume there is a clause $E = \{l_1, \ldots, l_{k-1}, l_k\} \in \psi$ which is used to add $\neg l_k$ next. Observe that $\sigma(E) = \sigma(\psi) = \top$ and by the induction hypothesis we have $\sigma(l_1) = \cdots = \sigma(l_{k-1}) = \bot$, which leads to $\sigma(l_k) = \top$. This concludes the induction proof resulting in $\sigma(\mathsf{ALA}(R)) = \bot$, which is impossible for the tautology $\mathsf{ALA}(R)$. The assumption is invalid and thus $\tau^x(O) = \tau^x(D) = \top$. □

Theorem 1. *Given a QBF $\phi = \Pi.\psi$ and a clause $C \in \psi$ with* QRAT *on an existential literal $l \in C$ with respect to QBF $\phi' = \Pi'.\psi'$ where $\psi' = \psi \setminus \{C\}$ and Π' is Π without the variables of C not occurring in ψ'. Then ϕ and ϕ' are satisfiability equivalent.*

Proof. If ϕ is satisfiable then ϕ' is also satisfiable, since all models of ϕ are also models of ϕ'. In the following, we show that if ϕ' is satisfiable then ϕ is also satisfiable. Let M be a model for ϕ', which is not a model for ϕ. Then for every assignment $\tau^x \bar{l} \tau_x$ in M which satisfies $\psi' = \psi \setminus \{C\}$ and falsifies C we replace in M' all assignments $\tau^x \bar{l} \rho_x$ by $\tau^x l \rho_x$. Now we need to show that all these $\tau^x l \rho_x$ still satisfy ψ'. Since $\tau^x \bar{l} \rho_x$ satisfies all clauses in ψ', the only clauses in ψ' that can be falsified by $\tau^x l \rho_x$ must contain literal $\bar{l}$. Lemma 2 shows, however, that all clauses $D \in \psi$ with $\bar{l} \in D$ are satisfied by τ^x and hence by $\tau^x l \rho_x$. Thus the resulting pre-model M' turns out to be a model of ϕ. □

In order to remove or to add a clause which has QRAT on literal l requires l to be existential. The following example illustrates that it would not be sound either to allow for universal variables or to ignore the variable dependency restrictions.

Example 2. Consider the false QBF $\exists x \forall y.(x \vee y) \wedge (\bar{x} \vee \bar{y})$. Clause $(x \vee y)$ has QRAT on y w.r.t. $(\bar{x} \vee \bar{y})$, but eliminating $(x \vee y)$ does not preserve unsatisfiability. Hence, one cannot remove clauses based on QRAT on a universal literal. If we would drop the variable dependency restriction, then $(x \vee y)$ would have QRAT on x w.r.t. $(\bar{x} \vee \bar{y})$. Again, removing $(x \vee y)$ does not preserve unsatisfiability.

The elimination of a clause which has QRAT or AT w.r.t. a QBF ϕ is called QRATE. We write QRATE also as $\Pi.\psi \cup \{C\} \xrightarrow{\mathsf{QRATE}} \Pi.\psi$. Analogously, QRAT allows the introduction of clauses. The addition of a clause which has QRAT or AT w.r.t. a QBF ϕ is called QRATA. We write QRATA also as $\Pi.\psi \xrightarrow{\mathsf{QRATA}} \Pi'.\psi \cup \{C\}$. Note that the added clause may contain variables which do not occur in the original QBF. Then the prefix has to be extended by these variables for getting a closed QBF again. These variables may be quantified arbitrarily and put at any position within the prefix.

Example 3. Consider the true QBF $\Pi.\psi = \forall a\,\exists b, c.(a \vee b) \wedge (\bar{a} \vee c) \wedge (b \vee \bar{c})$. Clause $(a \vee c)$ has QRAT on c w.r.t. $\Pi.\psi$: the only clause that contains literal $\bar{c}$ is $(b \vee \bar{c})$, which produces the outer resolvent $(a \vee b \vee c)$. $\mathsf{ALA}(\psi, (a \vee b \vee c)) = (a \vee \bar{a} \vee b \vee \bar{b} \vee c \vee \bar{c})$ is a tautology. Therefore, QRATA can add $(a \vee c)$ to ψ. Now consider a new existential variable d in the innermost quantifier block. The clause $(\bar{b} \vee c \vee d)$ has QRAT on c (and d) w.r.t. ψ. Adding $(\bar{b} \vee c \vee d)$ to ψ will result in the true QBF $\forall a\,\exists b, c, d.(a \vee \bar{b}) \wedge (\bar{a} \vee c) \wedge (b \vee \bar{c}) \wedge (\bar{b} \vee c \vee d)$.

However, as we will show below, one can remove universal literals if they have QRAT. This is similar to the pure literal elimination rule (see next section) which is a clause elimination technique if the pure literal is existentially quantified and which is a literal elimination technique if the pure literal is universally quantified.

For the proof of the following theorem we need the concept of "dual assignment". The dual of an assignment σ in a model for a universal literal is the unique τ in the same model obtained from flipping this literal in σ but keeping all literals before l and all universal literals after l untouched, or more formally:

Definition 6. *Given a model M of a QBF and $\sigma = \sigma^x l \sigma_x \in M$ and a literal l with $\sigma(l) = \top$ then $\tau = \sigma^x \bar{l} \tau_x \in M$ is the* dual *of σ w.r.t. l iff all universal literals in σ_x are the same in τ_x, e.g. for all universal literals k we have $\sigma_x(k) = \tau_x(k)$.*

Note, that existential literals in σ_x and τ_x might have opposite signs.

Theorem 2. *Given QBF $\phi_0 = \Pi.\psi$ and $\phi = \Pi.\psi \cup \{C\}$ where C has* QRAT *on a universal literal $l \in C$ with respect to ϕ_0. Further, let $\phi' = \Pi.\psi \cup \{C'\}$ with $C' = C \setminus \{l\}$. Then ϕ and ϕ' are satisfiability equivalent.*

Proof. We need to show that if ϕ is satisfiable, then ϕ' is satisfiable. The reverse is trivial. Let M be a model of ϕ. We are going to define a model M' for ϕ' from M as follows. All assignments σ in M with $\sigma(C') = \top$ are kept in M'. If $\sigma(C') = \bot$ and $\tau = \sigma^x \bar{l} \tau_x$ is the dual assignment of σ w.r.t. l in M, then we replace σ by $\sigma' = \sigma^x l \tau_x$, which is the same as the dual τ of σ w.r.t. l but with l flipped. It is apparent that the set of assignments M' defined this way actually forms a tree and thus a pre-model. Further, note, that C' is satisfied on all paths in M', since either $\sigma(C') = \top$ or otherwise (if $\sigma(C') = \bot$) we have $\sigma'(C') = \tau(C') = \tau(C) = \top$. As in the proof of Theorem 1 we assume that the pre-model M' defined above is not a model. Then we have $\sigma' \in M'$, a clause $D \in \psi$ with $\sigma'(D) = \bot$, and σ' was obtained from σ, by replacing σ by the dual τ of σ w.r.t. l with l flipped and $\sigma(C') = \bot$. Since $\tau(D) = \top$ and σ' differs from τ only for l, we know that $\bar{l}$ single satisfies D in σ', thus $\bar{l} \in D$. The outer clause $O = \mathcal{O}(\Pi, D, \bar{l}) \subset D$ has the property $\sigma'(O) = \bot$ and since $\sigma^x = \sigma'^x$ we derive that $\sigma(R) = \bot$. Observe that $\sigma(R) = \bot$ for the outer resolvent $R = \mathcal{R}(\Pi, C, D, l) = O \cup C'$ (note that l is removed, e.g., $l \notin C'$). Using similar arguments as in the proof of Lemma 2 we can show that all the literals added to R are false under σ. This is in contradiction to the assumption that $\mathsf{ALA}(\psi, R)$ is a tautology. As a consequence M' is a model of ϕ' and ϕ' is satisfiable too. □

In principle, a universal literal on which a clause C has QRAT w.r.t. to a QBF ϕ may be safely removed from C or vice versa added. In the following, we only need the elimination of universal literals. The elimination of a universal literal l from a clause C which has QRAT on l w.r.t. a QBF ϕ is called QRATU. We write QRATU also as $\Pi.\psi \cup \{C\} \xrightarrow{\text{QRATU}} \Pi.\psi \cup \{C \setminus \{l\}\}$.

The definition of outer resolvent depends on quantification. QRATU is not sound if we allow the existential variant of outer resolvent for a universal literal.

Example 4. Consider the true QBF $\forall x \exists y, z.(\bar{x} \vee \bar{y}) \wedge (\bar{x} \vee \bar{z}) \wedge (x \vee y) \wedge (x \vee y \vee z)$. Both resolution candidates for resolving $(x \vee y \vee z)$ on x lead to an empty outer clause. Incorrectly using the existential variant would keep x in the outer resolvent which is identical to the original clause $(x \vee y \vee z)$, which in turn is subsumed. However, removing x from the original clause $(x \vee y \vee z)$ makes the QBF false and thus it is incorrect to keep a universally quantified pivot in the outer resolvent before checking for asymmetric tautology.

4 Preprocessing for QBFs

For successfully solving quantified Boolean formulas (QBF), the introduction of an additional preprocessing step has been shown to be extremely beneficial to focus the search of many solvers. Frequently, preprocessing is crucial to solve a QBF formula. In general, the preprocessed formula is not logically equivalent, but satisfiability equivalent. Below, we introduce the most prominent techniques for preprocessing used in state-of-the-art tools.

We can distinguish three types of rules: (1) clause elimination rules; (2) clause modification rules; and (3) clause addition rules. Table 1 summarizes the preprocessing techniques and their necessary *preconditions*. We omit showing their soundness as this is extensively discussed in the referenced literature.

Clause Elimination Rules remove clauses while preserving unsatisfiability. *Tautology elimination* (E1) removes clauses containing a positive and negative occurrence of a variable. *Subsumption* (E2) removes clauses that are a superset of other clauses. *Existential pure literal elimination* (E3) removes all clauses with an existential literal that occurs only positive or only negative in the formula. *Quantified blocked clause elimination* (E4) removes clauses which contain a variable producing only tautological resolvents when used as pivot.

Clause Modification Rules add, remove, and rename literals. The *universal reduction rule* (M1) removes a universal literal if it is the innermost literal in a clause. The *strengthening rule* (M2) relies on clauses produced by resolution which subsume one of its antecedents. If an existential literal l occurs in a clause of size one, then *unit literal elimination* (M3) allows to remove clauses containing l and literal occurrences $\bar{l}$. *Universal pure literal elimination* (M4) removes a universal literal if it occurs only in one polarity in the whole formula. Covered literal addition (M5) extends a clause with literals that occur in all non-tautological resolvents. Finally, the *equivalence replacement rule* (M6) substitutes the occurrence of a literal l (and $\bar{l}$) by a literal k (and $\bar{k}$) if clauses of the form $(l \vee \bar{k})$ and $(\bar{l} \vee k)$ are in the formula. Literal l must be existentially quantified and $l \geq_\Pi k$.

Clause Addition Rules extend the formula with new clauses, while modifying and removing old ones. The *variable elimination rule* (A1), also known as DP resolution, replaces the clauses in which a certain existential variable occurs by all non-tautological resolvents on that variable. The *universal expansion rule* (A2) removes an innermost universal variable by duplicating and modifying all clauses that contain one or more innermost existential variables.

Table 1. Preprocessing Rules.

	name	rewriting rule	precondition	
E1.	*tautology elimination*	$\Pi.\psi, C \vee l \vee \bar{l} \overset{\mathsf{Taut}}{\Longrightarrow} \Pi.\psi$	none	clause elimination
E2.	*subsumption*	$\Pi.\psi, C, D \overset{\mathsf{Subs}}{\Longrightarrow} \Pi.\psi, C$	$C \subseteq D$	
E3.	*exist. pure literal elim.*	$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l \overset{\mathsf{Pure}_\exists}{\Longrightarrow} \Pi.\psi$	$\mathsf{Q}(\Pi, l) = \exists$, $\bar{l} \notin \psi \wedge C_1 \wedge \ldots \wedge C_n$	
E4.	*blocked clause elimination*	$\Pi.\psi, C \overset{\mathsf{QBCE}}{\Longrightarrow} \Pi.\psi$	$\exists y \in C$ with $\mathsf{Q}(\Pi, y) = \exists$, $\forall D \in \psi$ with $\bar{y} \in D$: $l, \bar{l} \in C \otimes_y D$ with $l \leq_\Pi y$	
M1.	*universal reduction*	$\Pi.\psi, C \vee l \overset{\mathsf{URed}}{\Longrightarrow} \Pi.\psi, C$	$\mathsf{Q}(\Pi, l) = \forall$, $\nexists k \in C$ with $l <_\Pi k$	clause modification
M2.	*strengthening*	$\Pi.\psi, l \vee C, \bar{l} \vee D \overset{\mathsf{Str}}{\Longrightarrow} \Pi.\psi, C, \bar{l} \vee D$	$C \subseteq D$	
M3.	*unit literal elimination*	$\Pi.\psi, l, C_1 \vee \bar{l}, \ldots, C_n \vee \bar{l}, D_1 \vee l, \ldots, D_m \vee l \overset{\mathsf{Unit}}{\Longrightarrow} \Pi.\psi, C_1, \ldots, C_n$	$\mathsf{Q}(\Pi, l) = \exists$	
M4.	*univ. pure literal elim.*	$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l \overset{\mathsf{Pure}_\forall}{\Longrightarrow} \Pi.\psi, C_1, \ldots, C_n$	$\mathsf{Q}(\Pi, l) = \forall$, $\bar{l} \notin \psi \wedge C_1 \wedge \ldots \wedge C_n$	
M5.	*covered literal addition*	$\Pi.\psi, C \overset{\mathsf{QCLA}}{\Longrightarrow} \Pi.\psi, C \vee l$	$\exists y \in C$ with $\mathsf{Q}(\Pi, y) = \exists$, $\forall D \in \psi$ with $\bar{y} \in D$: $l \in D$ or $k, \bar{k} \in C \otimes_y D$ with $k, l \leq_\Pi y$	
M6.	*equivalence replacement*	$\Pi.\psi, \bar{l} \vee k, l \vee \bar{k} \overset{\mathsf{Equiv}}{\Longrightarrow} \Pi.\psi[l/k]$	$\mathsf{Q}(\Pi, l) = \exists$, $k \leq_\Pi l$	
A1.	*variable elimination*	$\Pi\exists y.\psi, C_1 \vee \bar{y}, \ldots, C_n \vee \bar{y}, D_1 \vee y, \ldots, D_m \vee y \overset{\mathsf{VElim}}{\Longrightarrow} \Pi.\psi \bigwedge_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} (C_i \cup D_j)$	$\mathsf{Q}(\Pi, y) = \exists$, $y \notin \mathsf{vars}(\psi)$	clause addition
A2.	*universal expansion*	$\Pi\forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_n \vee \bar{x}, D_1 \vee x, \ldots, D_m \vee x, E_1, \ldots, E_p \overset{\mathsf{UExp}}{\Longrightarrow} \Pi\exists Y Y'.\psi, C_1, \ldots, C_n, D'_1, \ldots, D'_m, E_1, \ldots, E_p, E'_1, \ldots, E'_p$	$\mathsf{Q}(\Pi, x) = \forall$, $\exists y_i \in \mathsf{vars}(E_j), y_i \notin \mathsf{vars}(\psi)$ $x \notin \mathsf{vars}(\psi \wedge C_i \wedge D_j \wedge E_k)$, $D'_i = D_i[y_1/y'_1, \ldots, y_n/y'_n]$, $E'_i = E_i[y_1/y'_1, \ldots, y_n/y'_n]$	

5 Representing Preprocessing Techniques with QRAT

The QRAT proof system as presented above provides clause elimination and addition rules as in propositional logic when the pivot variable is existentially quantified. Further, QRAT allows for the removal/addition of variables in the case of universal pivots. This is almost sufficient to express the preprocessing rules introduced in the previous section. The only missing element is universal reduction, which also marks the difference between propositional resolution and resolution for QBF. To this end, we introduce the concept of *extended universal resolution* what is based on Theorem 4.9 of Van Gelder's work on resolution path dependency schemes [16]. In the following, we do not introduce the concept of resolution path dependencies, but we describe the universal literal elimination criterion according to the terminology used in the rest of the paper.

Definition 7 (Inner Clause). *Let C be a clause occurring in QBF $\Pi.\psi$. The* inner clause *of C on literal $l \in C$, denoted by $\mathcal{I}(\Pi, C, l)$, is given by the clause $\{k \mid k \in C, k = \bar{l} \text{ or } k >_\Pi l\}$.*

Lemma 3. *Given a QBF formula $\Pi.\psi$, let $\mathcal{E}(\Pi, C, l)$ be the unique clause obtained by repeatedly applying the extension rule*

$$C := C \cup \mathcal{I}(\Pi, D, l) \text{ if exists } k \in C, D \in \psi \text{ with } \bar{k} \in D, Q(k) = \exists, \text{ and } k >_\Pi l$$

until fixpoint. Given a QBF $\Pi.\psi \wedge \{E\}$ with a universal literal $l \in E$ such that $\bar{l} \notin \mathcal{E}(\Pi, E, l)$. Then, the removal of l from E is satisfiability preserving.

Lemma 3 is a generalization of the universal reduction rule which we call *extended universal reduction* in the following. For the application of extended universal reduction we write $\Pi.\psi \cup \{C\} \xrightarrow{\mathsf{EUR}} \Pi.\psi \cup \{C \setminus \{l\}\}$.

Now we are able to express the preprocessing techniques shown in Table 1 with only four rules: QRATE, QRATA, QRATU, and EUR. Table 2 shows the translations for the clause elimination techniques, Table 3 for the clause modification techniques, and Table 4 for the clause addition techniques. We refer to Table 1 for the preconditions for the application of the preprocessing rules.

Tautologies, subsumed clauses as well as blocked clauses have QRAT, so only one application of QRATE is necessary for their removal. If an existential literal is pure than all clauses in which it occurs are blocked w.r.t. this literal and therefore can be omitted by multiple applications of QRATE.

For strengthening a clause $C \vee l$, we first add the resolvent with $D \vee \bar{l}$ which is C. Now, $C \vee l$ is subsumed and can, as we have discussed before, be removed by QRATE. To express unit literal elimination, we first add clauses C_i, i.e., the resolvents of $C_i \vee \bar{l}$ and l. Then $C_i \vee \bar{l}$ become QRAT and can be removed. Now the literal l occurs only in one polarity and hence, the clauses containing l can be removed by QRATE (cf., existential pure literal elimination). Universal pure literal elimination simply maps to multiple applications of QRATU such that l does not occur in the formula anymore. If a universal literal l is removed from a clause C, this can naturally be expressed by extended universal resolution.

Table 2. Clause Elimination Rules.

preprocessing rule	rewriting
$\Pi.\psi, C \vee l \vee \bar{l} \overset{\mathsf{Taut}}{\Longrightarrow} \Pi.\psi$	$\Pi.\psi, C \vee l \vee \bar{l} \xrightarrow{\mathsf{QRATE}} \Pi.\psi$
$\Pi.\psi, C, D \overset{\mathsf{Subs}}{\Longrightarrow} \Pi.\psi, C$	$\Pi.\psi, C, D \xrightarrow{\mathsf{QRATE}} \Pi.\psi, C$
$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l \overset{\mathsf{Pure}_\exists}{\Longrightarrow} \Pi.\psi$	$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l \xrightarrow{\mathsf{QRATE}*} \Pi.\psi$
$\Pi.\psi, C \overset{\mathsf{QBCE}}{\Longrightarrow} \Pi.\psi$	$\Pi.\psi, C \xrightarrow{\mathsf{QRATE}} \Pi.\psi$

Table 3. Clause Modification Rules.

preprocessing rule	rewriting
$\Pi.\psi, C \vee l, D \vee \bar{l}$ $\overset{\mathsf{Str}}{\Longrightarrow} \Pi.\psi, C, D \vee \bar{l}$	$\Pi.\psi, C \vee l, D \vee \bar{l}$ $\xrightarrow{\mathsf{QRATA}} \Pi.\psi, C, C \vee l,\ D \vee \bar{l} \xrightarrow{\mathsf{QRATE}} \Pi.\psi, C, D \vee \bar{l}$
$\Pi.\psi, C_1 \vee \bar{l}, \ldots, C_n \vee \bar{l},$ $l, D_1 \vee l, \ldots, D_m \vee l$ $\overset{\mathsf{Unit}}{\Longrightarrow} \Pi.\psi, C_1, \ldots, C_n$	$\Pi.\psi, C_1 \vee \bar{l}, \ldots, C_n \vee \bar{l},\ l, D_1 \vee l, \ldots, D_m \vee l$ $\xrightarrow{\mathsf{QRATA}*} \Pi.\psi, C_1 \vee \bar{l}, \ldots, C_n \vee \bar{l},$ $l, D_1 \vee l, \ldots, D_m \vee l, C_1, \ldots, C_n$ $\xrightarrow{\mathsf{QRATE}*} \Pi.\psi,\ l, C_1, \ldots, C_n \xrightarrow{\mathsf{QRATE}} \Pi.\psi,\ C_1, \ldots, C_n$
$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l$ $\overset{\mathsf{Pure}_\forall}{\Longrightarrow} \Pi.\psi, C_1, \ldots, C_n$	$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l$ $\xrightarrow{\mathsf{QRATU}*} \Pi.\psi, C_1, \ldots, C_n$
$\Pi.\psi, C \vee l$ $\overset{\mathsf{URed}}{\Longrightarrow} \Pi.\psi, C$	$\Pi.\psi, C \vee l$ $\xrightarrow{\mathsf{EUR}} \Pi.\psi, C$
$\Pi.\psi, \bar{l} \vee k, l \vee \bar{k}$ $\overset{\mathsf{Equiv}}{\Longrightarrow} \Pi.\psi[l/k]$	$\Pi.\psi, C_1 \vee l, \ldots, C_n \vee l, D_1 \vee \bar{l}, \ldots, D_m \vee \bar{l}, \bar{l} \vee k, l \vee \bar{k}$ $\xrightarrow{\mathsf{QRATA}*} \Pi.\psi, C_1 \vee l, \ldots, C_n \vee l, D_1 \vee \bar{l}, \ldots, D_m \vee \bar{l},$ $\bar{l} \vee k, l \vee \bar{k}, C_1 \vee k, \ldots, C_n \vee k, D_1 \vee \bar{k}, \ldots, D_m \vee \bar{k}$ $\xrightarrow{\mathsf{QRATE}*}$ $\Pi.\psi, \bar{l} \vee k, l \vee \bar{k}, C_1 \vee k, \ldots, C_n \vee k, D_1 \vee \bar{k}, \ldots, D_m \vee \bar{k}$ $\xrightarrow{\mathsf{QRATE}*} \Pi.\psi, C_1 \vee k, \ldots, C_n \vee k, D_1 \vee \bar{k}, \ldots, D_m \vee \bar{k}$
$\Pi.\psi, C$ $\overset{\mathsf{QCLA}}{\Longrightarrow} \Pi.\psi, C \vee l$	$\Pi.\psi, C$ $\xrightarrow{\mathsf{QRATA}} \Pi.\psi, C, C \vee l \xrightarrow{\mathsf{QRATE}} \Pi.\psi, C \vee l$

If l is a covered literal for C w.r.t. $\Pi.\psi$, then $C \vee l$ has QRAT w.r.t. $\Pi.\psi$. After adding $C \vee l$ using QRATA, C gets QRAT and can be removed using QRATE. Quantified covered clause elimination [8] is a clause elimination procedure that extends clauses with covered literals until clauses become blocked. To represent this procedure, we add an intermediate clause for each covered literal addition. When the clause is blocked, it can be eliminated using QRATE.

If a literal l shall be substituted by a literal k due to equivalence replacement, the formula has to contain the binary clauses $(l \vee \bar{k})$ and $(\bar{l} \vee k)$. Then first the clauses $C_i \vee k$ and $D_j \vee \bar{k}$ are added by resolution, i.e., by QRATA. All clauses

containing l and $\bar{l}$ are asymmetric tautologies, because $k, \bar{k} \in \mathsf{ALA}(\psi, C_i \vee l)$ and can therefore be removed by QRATE.

Variable elimination is rewritten as follows. First all possible non-tautological resolvents on elimination variable y are added with QRATA. Then all clauses containing y or $\bar{y}$ become QRAT and can be eliminated by QRATE.

Finally, we describe universal expansion using redundancy elimination and addition rules. Consider the QBF $\Pi\forall x \exists Y.\psi$ from which we want to eliminate the innermost universal variable x. Let $E = \{E_i \mid E_i \in \psi_Y, x \notin E_i, \bar{x} \notin E_i\}$. In the first step, we add clauses $E_i \vee \bar{x}$ (which are subsumed by E_i) using QRATA. This is necessary, because we later need to eliminate E_i. We introduce conditional equivalences represented by the clauses $x \vee y_j \vee \bar{y}'_j$ and $x \vee \bar{y}_j \vee y'_j$ for all $y_j \in Y$ and append $\exists Y'$ to the prefix. Now we copy all original clauses with literal y_j, but without $\bar{x}$ and add literal x in case it is not already present. The conditional equivalences allow to treat original and primed copies of clauses with x as alternative. One version can be exchanged for the other as long the equivalence clauses are there. We add the primed copies and afterwards remove the original ones. Now all clauses E_i are asymmetric tautologies and can be removed. Next, we remove the conditional equivalences $x \vee y_j \vee \bar{y}'_j$ and $x \vee \bar{y}_j \vee y'_j$ which have QRAT on the y_j after removal of the E_i clauses. At this point, clauses containing variables from Y do not contain x and clauses with variables from Y' do not contain $\bar{x}$. So extended universal reduction can remove the literals x and $\bar{x}$.

Table 4. Clause Addition Rules. Clauses are added / removed in order of appearance.

preprocessing rule	**rewriting**
$\Pi\exists y.\psi, C_1 \vee \bar{y}, \ldots, C_n \vee \bar{y},$ $D_1 \vee y, \ldots, D_m \vee y$ $\overset{\mathsf{VElim}}{\Longrightarrow} \Pi.\psi \bigwedge_{\substack{1\leq i\leq n \\ 1\leq j\leq m}} (C_i \cup D_j)$	$\Pi\exists y.\psi, C_1 \vee y, \ldots, C_n \vee y, D_1 \vee \bar{y}, \ldots, D_m \vee \bar{y}$ $\xrightarrow{\mathsf{QRATA}*} \Pi\exists y.\psi, C_1 \vee y, \ldots, C_n \vee y,$ $D_1 \vee \bar{y}, \ldots, D_m \vee \bar{y}, C_1 \cup D_1, \ldots, C_n \cup D_m$ $\xrightarrow{\mathsf{QRATE}*} \Pi.\psi, C_1 \cup D_1, \ldots, C_n \cup D_m$
$\Pi\forall x \exists Y.\psi,$ $C_1 \vee \bar{x}, \ldots, C_n \vee \bar{x},$ $D_1 \vee x, \ldots, D_m \vee x,$ $E_1, \ldots, E_p$ $\overset{\mathsf{UExp}}{\Longrightarrow} \Pi\exists YY'.\psi,$ $C_1, \ldots, C_n, E_1, \ldots, E_p,$ $D'_1, \ldots, D'_m, E'_1, \ldots, E'_p$	$\Pi\forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_n \vee \bar{x},$ $D_1, \vee x, \ldots, D_m \vee x, E_1, \ldots, E_p$ $\xrightarrow{\mathsf{QRATA}*} \Pi\forall x \exists YY'.\psi, C_1 \vee \bar{x}, \ldots, C_n \vee \bar{x},$ $D_1 \vee x, \ldots, D_m \vee x, E_1, \ldots, E_p, E_1 \vee \bar{x}, \ldots, E_p \vee \bar{x},$ $x \vee y_1 \vee \bar{y}'_1, \ldots, x \vee y_{\lvert Y\rvert} \vee \bar{y}'_{\lvert Y\rvert},$ $x \vee \bar{y}_1 \vee y'_1, \ldots, x \vee \bar{y}_{\lvert Y\rvert} \vee y'_{\lvert Y\rvert},$ $D'_1 \vee x, \ldots, D'_m \vee x, E'_1 \vee x, \ldots, E'_p \vee x$ $\xrightarrow{\mathsf{QRATE}*} \Pi\forall x \exists YY'.\psi, C_1 \vee \bar{x}, \ldots, C_n \vee \bar{x},$ $E_1 \vee \bar{x}, \ldots, E_p \vee \bar{x}, D'_1 \vee x, \ldots, D'_m \vee x,$ $E'_1 \vee x, \ldots, E'_p \vee x$ $\xrightarrow{\mathsf{EUR}*} \Pi\exists YY'.\psi, C_1, \ldots, C_n, E_1, \ldots, E_p,$ $D'_1, \ldots, D'_m, E'_1, \ldots, E'_p$

6 QRAT Proofs

This section describes our new proof format for QBF formulas, how to check it and an experimental evaluation. The syntax of the proof format is very similar to the DRUP proof format [12] for CNF formulas. We extend the DRUP syntax to express elimination of universal literals. Furthermore, the redundancy check is different than proofs in DRUP because we deal with QBF formulas.

6.1 The QRAT Proof Format

Proofs are sequences of clause additions, deletions, and modifications. They are build using three kind of lines: addition (QRATA), deletion (QRATE), and universal elimination (QRATU and EUR). Addition lines have no prefix and are unconstrained in the sense that one can add any clause at any point in the proof. Clause deletion lines, with prefix "`d`", and universal elimination lines, with prefix "`u`", are restricted. The clause after a "`d`" or "`u`" prefix must be either present in the original formula or as a clause added earlier in the proof.

Let $\Pi.\psi$ be a QBF formula and P be a QRAT proof for $\Pi.\psi$. We denote the number of lines in a proof P by $|P|$. For each $i \in \{0, \ldots, |P|\}$, we define a CNF formula ψ_P^i below. C_i refers to the clause on line i of P and l_i refers to the first literal on line i of P.

$$\psi_P^i := \begin{cases} \psi & \text{if } i = 0; \\ \psi_P^{i-1} \setminus \{C_i\} & \text{if the prefix of } C_i \text{ is "d"}; \\ \psi_P^{i-1} \setminus \{C_i\} \cup \{C_i \setminus \{l_i\}\} & \text{if the prefix of } C_i \text{ is "u"}; \\ \psi_P^{i-1} \cup \{C_i\} & \text{otherwise.} \end{cases}$$

A proof P is called a *satisfaction proof* for QBF formula $\Pi.\psi$ if the following two properties hold. First, for all $i \in \{1, \ldots, |P|\}$, if clause C_i has prefix "`d`", then it must have QRAT on l_i with respect to ψ_P^i. In case l_i is universally quantified, we check whether $\mathsf{ALA}(\psi_P^i, C_i)$ is a tautology. Second, $\psi_P^{|P|}$ must be empty.

A proof P is called a *refutation proof* for QBF formula $\Pi.\psi$ if the following three properties hold. First, for all $i \in \{1, \ldots, |P|\}$, if clause C_i has no prefix, then it must have QRAT on l_i with respect to ψ_P^{i-1}. In case l_i is universally quantified, we check whether $\mathsf{ALA}(\psi_P^{i-1}, C_i)$ is a tautology. Second, for all $i \in \{1, \ldots, |P|\}$, if clause C_i has has prefix "`u`", then l_i must be universally quantified. Additionally, C_i must have either QRAT on l_i with respect to ψ_P^{i-1}, or l_i can be removed using EUR. Third, $C_{|P|}$ must be the empty clause (without a prefix). Fig. 1 shows a true and a false QBF and a QRAT proof for both.

A universal elimination line in satisfaction proofs can be replaced by a clause addition and deletion line to obtain another satisfaction proof. Simply add the clause without its first literal, and afterwards delete the subsumed clause. For example, consider the line "`u 1 2 3 0`" in a satisfaction proof. This line can be replaced by "`2 3 0`" followed by "`d 1 2 3 0`". Consequently, any satisfaction proof can be converted such that it contains only addition and deletion lines.

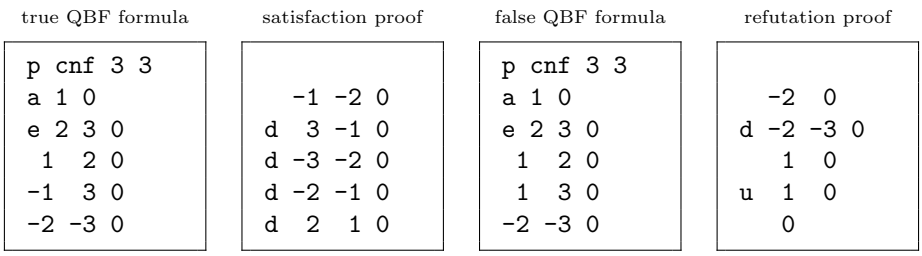

Fig. 1. Two QBFs formulas and QRAT proofs. On the left a true QBF with a satisfaction proof next to it. On the right a false QBF with a refutation next to it. The formulas and proofs are spaced to improve readability. Proofs consist of three kind of lines: addition (no prefix), deletion ("`d` " prefix) and universal elimination ("`u` " prefix).

Recall that QRATA can add clauses that contain new variables. The QRAT proof format does not support describing the quantifier block for new variables. For all known preprocessing techniques, newly introduced variables are placed in the innermost active existential quantifier block. Consequently, the QRAT format assumes this convention for all new variables.

6.2 Checking QRAT Proofs

Although the syntax for QRAT proof is identical for true and false QBFs, validating a proof is different. For true QBFs only the clause deletion lines (the ones with a "`d` " prefix) have to be checked, while for false QBFs, all the lines except the clause deletion lines have to be checked.

The easiest, but rather expensive, method to validate proofs checks the redundancy of each clause: for true QBFs all deletion lines and for false QBFs all addition and universal elimination lines. However, one can check proofs more efficiently my marking involved clauses during each redundancy check. That way the checker can be restricted to validate marked clauses only. The marking procedure is a bit tricky. In short, it marks all involved clauses that were required to compute the last unique implication point from each conflict.

Checking only marked clauses was proposed to check clausal proofs of CNF formulas efficiently [11]. For false QBFs, the checking is similar to the SAT case: during initialization the empty clause is marked. Refutation proofs should be validated in reverse order, starting with the marked empty clause. For true QBFs the procedure is different: initially all original clauses are marked and satisfaction proofs are checked in chronological order. When a clause is deleted that was not marked by any redundancy check, the clause can be skipped.

Example 5. Consider the true QBF $\Pi.\psi = \forall a \exists b, c.(a \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee \bar{c})$. This is the same QBF as in Fig. 1 (left). Fig. 1 also shows the satisfaction proof $P := (\bar{a} \vee \bar{b}), \mathtt{d}(c \vee \bar{a}), \mathtt{d}(\bar{c} \vee \bar{b}), \mathtt{d}(\bar{b} \vee \bar{a}), \mathtt{d}(b \vee a)$. Satisfaction proofs are checked in chronological order. So, first, $(\bar{a} \vee \bar{b})$ is added, afterwards $(c \vee \bar{a})$ is removed, until all original clauses and all added clauses have been deleted.

6.3 Implementation

We equipped our preprocessor bloqqer [8] with QRAT-based tracing as described in Section 5. In contrast to previous extensions of bloqqer [7,17] we hardly had to modify its internal behavior. Hence, with QRAT-based tracing, we have the first QBF preprocessor fully supporting proof generation for true and false formulas.

We implemented an efficient QRAT checker QRATtrim, which is based on DRUPtrim [12], a clausal proof checking tool for CNF. It uses the optimizations of Section 6.2, such as validating marked clauses only and checking satisfaction and refutation proofs in chronological and reverse order, respectively.

Evaluations on the benchmark sets of the QBF evaluations 2010 and 2012 indicate that the power of the preprocessor is hardly reduced by enabling QRAT-based tracing. The benchmark set of 2010 (resp. 2012) contains 64 (resp. 32) true instances and 86 (resp. 36) false instances which can be solved by using only bloqqer. These formulas turn out to be extremely hard for conflict/solution-driven clause/cube learning solvers like DepQBF [18], which can only solve 26 (resp. 2) true formulas and 57 (resp. 14) false formulas. For the other formulas DepQBF timed-out, given a time limit of 900 seconds. The resolution-proof producing version of bloqqer which was presented in [7] is able to evaluate 28 (resp. 22) true formulas and 57 (resp. 22) false formulas. Please note that the resolution-proof producing version of bloqqer did not time out for the unsolved formulas. Less formulas are solved because the techniques for which no translation to resolution is presented in [7] are simply turned off. Our new QRAT-based proof producing version of bloqqer solves 63 (resp. 32) true formulas and 86 (resp. 36) false formulas, i.e., only one formula less is solved. We could verify all but two QRAT proofs. For solving these formulas, miniscoping is necessary – which is not yet supported by our checker, but can be realized by taking dependencies into account while computing outer clauses. If we turn off miniscoping and increase the bounds for variable elimination, these formulas can be solved and checked as well. For satisfiable formulas, solving is twice as fast as checking on average, in particular we have 1.4s (2.2s) for solving and 3.2s (5.9s) for checking. For unsatisfiable formulas, checking is considerably faster: we have 18.7s (30.1s) for solving and 5.2s (8.6s) for checking. Our bloqqer extension, the proof checking tools as well as the details on our experiments are available at `http://fmv.jku.at/bloqqer`.

7 Conclusion

We presented a proof system which captures recent preprocessing and solving techniques for QBF in a uniform manner. Based on *asymmetric tautologies*, the proof system consists only of four simple rules. We showed how state-of-the-art preprocessing techniques can be represented within this proof system. Our rules QRATE, QRATU, and QRATA may be applied as preprocessing rules themselves similar as QBCE and we plan to integrate them in our preprocessor. We deal with all the challenges regarding certificates and preprocessing for QBF recently listed in [7], namely: can we (1) produce polynomially-verifiable certificates for true QBFs in the context of preprocessing, (2) narrow the performance gap between

solving with and without certificate generation; and (3) develop methods to deal with universal expansion and other techniques. First, the size of our certificates for true QBFs is polynomial in the solving time and certificate checking can be done in polynomial time. Second, the overhead of emitting certificates is small and all existing preprocessing techniques are supported. Third, our proof system can simulate universal expansion and other existing techniques. Future work will focus on rewriting search based QBF solver techniques [18] to the QRAT proof system and extracting Skolem functions [4] from QRAT proofs.

References

1. Benedetti, M.: Extracting Certificates from Quantified Boolean Formulas. In: IJCAI, Professional Book Center (2005) 47–53
2. Kleine Büning, H., Subramani, K., Zhao, X.: Boolean Functions as Models for Quantified Boolean Formulas. J. Autom. Reasoning **39**(1) (2007) 49–75
3. Jussila, T., Biere, A., Sinz, C., Kröning, D., Wintersteiger, C.: A first step towards a unified proof checker for QBF. In: SAT. Volume 4501 of LNCS. Springer (2007) 201–214
4. Niemetz, A., Preiner, M., Lonsing, F., Seidl, M., Biere, A.: Resolution-Based Certificate Extraction for QBF. In: SAT. Volume 7317 of LNCS. (2012) 430–435
5. Janota, M., Grigore, R., Marques-Silva, J.: On Checking of Skolem-based Models of QBF. In: RCRA 2012. (2012)
6. Van Gelder, A.: Certificate Extraction from Variable-Elimination QBF Preprocessors. In: QBF, http://fmv.jku.at/qbf2013/reportQBFWS13.pdf (2013) 35–39
7. Janota, M., Grigore, R., Marques-Silva, J.: On QBF Proofs and Preprocessing. In: LPAR. Volume 8312 of LNCS., Springer (2013) 473–489
8. Biere, A., Lonsing, F., Seidl, M.: Blocked clause elimination for QBF. In: CADE 2011. Volume 6803 of LNCS., Springer (2011) 101–115
9. Giunchiglia, E., Marin, P., Narizzano, M.: sQueezeBF: An Effective Preprocessor for QBFs Based on Equivalence Reasoning. In: SAT. Volume 6175 of LNCS., Springer (2010) 85–98
10. Biere, A.: Resolve and expand. In: SAT (Selected Papers). Volume 3542 of LNCS., Springer (2005) 59–70
11. Goldberg, E.I., Novikov, Y.: Verification of proofs of unsatisfiability for CNF formulas. In: DATE. (2003) 10886–10891
12. Heule, M.J.H., Hunt, Jr., W.A., Wetzler, N.: Trimming while checking clausal proofs. In: FMCAD, IEEE (2013) 181–188
13. Heule, M.J.H., Hunt, Jr., W.A., Wetzler, N.: Verifying refutations with extended resolution. In: CADE. Volume 7898 of LNAI., Springer (2013) 345–359
14. Heule, M.J.H., Järvisalo, M., Biere, A.: Clause elimination procedures for CNF formulas. In: LPAR-17. Volume 6397 of LNCS., Springer (2010) 357–371
15. Järvisalo, M., Heule, M.J.H., Biere, A.: Inprocessing rules. In: IJCAR. Volume 7364 of LNCS., Springer (2012) 355–370
16. Van Gelder, A.: Variable Independence and Resolution Paths for Quantified Boolean Formulas. In: CP. Volume 6876 of LNCS., Springer (2011) 789–803
17. Könighofer, R., Seidl, M.: Partial witnesses from preprocessed quantified boolean formulas. In: accepted for DATE 2014. (2014)
18. Lonsing, F., Biere, A.: DepQBF: A Dependency-Aware QBF Solver. JSAT **7**(2-3) (2010) 71–76