Electronic Colloquium on Computational Complexity, Report No. 36 (2014)

On Unification of QBF Resolution-Based Calculi

Mikoláš Janota[1], Leroy Chew[2], and Olaf Beyersdorff[2]

[1]INESC-ID, Lisbon, Portugal
[2]School of Computing, University of Leeds, United Kingdom

Abstract. Several calculi for quantified Boolean formulas (QBFs) exist, but relations between them are not yet fully understood. This paper defines a novel calculus, which is resolution-based and enables unification of the principal existing resolution-based QBF calculi, namely Q-resolution, long-distance Q-resolution and the expansion-based calculus ∀Exp+Res. All these calculi play an important role in QBF solving. This paper shows simulation results for the new calculus and some of its variants. Further, we demonstrate how to obtain winning strategies for the universal player from proofs in the calculus. We believe that this new proof system opens new avenues for further research and provides a suitable formalism for certification of existing solvers.

1 Introduction

Proof complexity has been the subject of research for a number of reasons. The seminal paper of Cook and Reckhow showed an important connection between proof complexity and computational complexity [8]; similarly there are strong links to first-order logic, in particular to bounded arithmetic [7]. In areas like model checking, proofs have turned out to be important artifacts when solving certain types of problems [27]. Last but not least, for automated theorem provers, it is desirable that they provide a proof as a certificate that the given answer is indeed correct [19,33].

This paper is concerned with the proof systems for quantified Boolean formulas (QBF). Currently, a handful of systems exist. Krajíček and Pudlák define a Gentzen-style sequent calculus for QBF [25]. Kleine Büning et al. define a resolution-like calculus called *Q-resolution* [21]. There are several extensions of Q-resolution; notably *long-distance Q-resolution*, which is important as it enables tracing certain type of *DPLL-based* QBF solvers [36,2,24]. It has also been shown to be more powerful than plain Q-resolution [10].

Recently, a proof system ∀Exp+Res was introduced with the motivation to trace expansion-based QBF solvers [13]. ∀Exp+Res also uses resolution but is rather different from Q-resolution. At this point it is only known that ∀Exp+Res cannot p-simulate Q-resolution [16], but it is unknown whether Q-resolution p-simulates ∀Exp+Res. We conjecture that the two systems are incomparable as it has been shown that expansion-based solving can exponentially outperform DPLL-based solving. (An overview of these calculi is given in Section 2 and some other variants are mentioned.)

The disparity between the existing resolution-based calculi does not only represent a theoretical question. Indeed, it prohibits unified certification of QBF solvers or certification of solvers combining expansion and DPLL (expansion is also commonly used in preprocessing [5]). The quest for a unified calculus for the aforementioned solvers does not only have certification as motivation. If we can define a calculus that is able to trace both types of solvers, we should ask, whether we can develop QBF solvers based on this calculus rather than on the more limited ones.

ISSN 1433-8092

The objective of this paper is to define a calculus that is able to capture the existing QBF resolution-based calculi and yet remains amenable to machine manipulation. The contributions of the paper are as follows. (1) A novel calculus is defined (with several variants). (2) It is shown that this calculus is sound and complete. (3) It is shown that this calculus p-simulates ∀Exp+Res and long-distance Q-resolution (and therefore Q-resolution). (4) It is shown how to obtain a winning strategy for the universal player from proofs in this calculus. We note that to our best knowledge, constructions of strategies from expansion-based solvers were not known prior to this paper.

The rest of the paper is structured as follows. Section 2 introduces concepts and notation used throughout the paper. Section 3 introduces a novel calculus and shows that it p-simulates Q-resolution and the expansion-based calculus ∀Exp+Res. Section 4 extends the calculus from the previous section such that it also p-simulates long-distance Q-resolution; further we demonstrate how to obtain winning strategies from refutations in that calculus. This also serves as a soundness proof of the defined calculi. Finally, Section 5 concludes the paper and points to directions of future work.

2 Preliminaries

A *literal* is a Boolean variable or its negation; we say that the literal x is *complementary* to the literal $\neg x$ and vice versa. If l is a literal, $\neg l$ denotes the complementary literal, i.e. $\neg\neg x = x$. A *clause* is a disjunction of zero or more literals. If a clause contains no literals, it is denoted as $\perp$, which is semantically equivalent to false. A formula in *conjunctive normal form* (CNF) is a conjunction of clauses. Whenever convenient, a clause is treated as a set of literals and a CNF formula as a set of sets of literals. For a literal $l = x$ or $l = \neg x$, we write $\mathsf{var}(l)$ for x. For a clause C, we write $\mathsf{var}(C)$ to denote $\{\mathsf{var}(l) \mid l \in C\}$ and for a CNF ψ, $\mathsf{var}(C)$ denotes $\{l \mid l \in \mathsf{var}(C), C \in \psi\}$.

Substitutions are treated as functions from variables to formulas and are denoted as $\psi_1/x_1, \dots, \psi_n/x_n$, with $x_i \neq x_j$ for $i \neq j$. The set of variables $x_1, \dots, x_n$ is called the *domain* of the substitution. An application of a substitution is denoted as $\phi[\psi_1/x_1, \dots, \psi_n/x_n]$ meaning that the variables x_i are simultaneously substituted with the corresponding formulas ψ_i in the formula ϕ. A substitution is called an *assignment* iff each ψ_i is one of the constants 0, 1. An assignment is called *total*, or *complete*, for a set of variables X if each $x \in X$ is in the domain of the assignment. Applications of substitutions imply basic simplifications, e.g. $(x \vee y)[0/y] = x$.

A *proof system* (Cook, Reckhow [8]) for a language L over alphabet Γ is a polynomial-time computable partial function $f : \Gamma^* \rightharpoonup \Gamma^*$ with $rng(f) = L$ and where Γ^* is the set of strings over Γ. An f*-proof* of string y is a string x such that $f(x) = y$.

We say that a proof system f *p-simulates* g ($g \leq_p f$) if there exists a polynomial p such that for every g-proof π_g there is an f-proof π_f with $f(\pi_f) = g(\pi_g)$ and $|\pi_f| \leq p(|\pi_g|)$ (they have *polynomially similar size*) and π_f can be constructed from π_g in polynomial time.

Quantified Boolean Formulas (QBFs) [20] are an extension of propositional logic with quantifiers with the standard semantics that $\forall x.\, \Psi$ is satisfied by the same truth assignments as $\Psi[0/x] \wedge \Psi[1/x]$ and $\exists x.\, \Psi$ as $\Psi[0/x] \vee \Psi[1/x]$.

Unless specified otherwise, we assume that QBFs are in *closed prenex* form with a CNF *matrix*, i.e., we consider the form $\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k.\ \phi$, where X_i are pairwise disjoint sets of variables; $\mathcal{Q}_i \in \{\exists, \forall\}$ and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$. The formula ϕ is in CNF and is defined only on variables $X_1 \cup \dots \cup X_k$. The propositional part ϕ of a QBF is called the *matrix* and the rest the *prefix*. If a variable x is in the set X_i, we say that x is at *level* i and write $\mathsf{lv}(x) = i$; we write $\mathsf{lv}(l)$ for $\mathsf{lv}(\mathsf{var}(l))$. A closed QBF is *false* (resp. *true*), iff it is semantically equivalent to the constant 0 (resp. 1).

Often it is useful to think of a QBF $\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k.\ \phi$ as a *game* between a *universal and an existential player*. In the i-th step of the game, the player $\mathcal{Q}_i$ assigns value to the variables X_i. The existential player wins the game iff the matrix ϕ evaluates to 1 under the assignment constructed in the game. The universal player wins iff the matrix ϕ evaluates to 0 under the assignment. A QBF is false if and only if there exists a *winning strategy* for the universal player, i.e. the universal player can win any possible game.

2.1 Resolution-based Calculi for QBF

This section gives a brief overview of the main existing resolution-based calculi for QBF. Q-resolution (Q-Res), by Kleine Büning et al. [21], is a resolution-like calculus that operates on QBFs in prenex form where the matrix is a CNF. The rules are given in Figure 1.

$$\frac{}{C}\ \text{(Axiom)} \qquad\qquad \frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}\ \text{(Res)}$$

C is a clause in the matrix. Variable x is existential. If $z \in C_1$, then $\neg z \notin C_2$.

$$\frac{C \cup \{u\}}{C}\ (\forall\text{-Red})$$

Variable u is universal. If $x \in C$ is existential, then $\mathsf{lv}(x) < \mathsf{lv}(u)$.

Fig. 1. The rules of Q-Res [21]

Long-distance resolution (LD-Q-Res) appears originally in the work of Zhang and Malik [36] and was formalized into a calculus by Balabanov and Jiang [2]. It allows merging complementary literals of a universal variable u into the special literal u^*. These special literals prohibit certain resolution steps. In particular, different literals of a universal variable u may be merged only if $\mathsf{lv}(x) < \mathsf{lv}(u)$, where x is the resolution variable. The rules are given in Figure 2. Note that the rules do not prohibit resolving $w^* \vee x \vee C_1$ and $u^* \vee \neg x \vee C_2$ with $\mathsf{lv}(w) \leq \mathsf{lv}(u) < \mathsf{lv}(x)$ as long as $w \neq u$.

Janota et al. [17] formalized a proof system $\forall$Exp+Res, whose objective is to emulate the behavior of expansion-based solving, cf. [3,4,5,15,14]. Here we present an adapted version of this calculus so that it is congruent with the other resolution-based calculi (semantically it is the same as in [17]). The calculus is presented in Figure 3. The $\forall$Exp+Res calculus operates on clauses that comprise only existential variables from the original QBF; but additionally, these existential variables are annotated with the substitutions to universal variables. Any existential variable x is annotated with substitutions to those variables that precede it in the quantification order. For instance, the clause $x \vee b^{0/u}$ can be derived from the original clause $x \vee u \vee x$ under the prefix $\exists x \forall u \exists b$.

Besides the aforementioned resolution-based calculi, there is a system by Klieber et al. [24,23], which operates on pairs of sets of literals, rather than clauses; this system is in its workings akin to LD-Q-Res. Van Gelder defines an extension of Q-Res, called *QU-resolution*, which

$$\frac{}{C} \text{ (Axiom)} \qquad \frac{C \cup \{u\}}{C} \text{ (}\forall\text{-Red)} \qquad \frac{C \cup \{u^*\}}{C} \text{ (}\forall\text{-Red}^*\text{)}$$

Variable u is universal and $\mathsf{lv}(u) \geq \mathsf{lv}(l)$ for all $l \in C$.

$$\frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2} \text{ (Res)}$$

$$\frac{C_1 \cup \{x, u\} \qquad C_2 \cup \{\neg x, \neg u\}}{C_1 \cup C_2 \cup \{u^*\}} \text{ (LD0)}$$

$$\frac{C_1 \cup \{x, u\} \qquad C_2 \cup \{\neg x, u^*\}}{C_1 \cup C_2 \cup \{u^*\}} \text{ (LD1)}$$

$$\frac{C_1 \cup \{x, u^*\} \qquad C_2 \cup \{\neg x, u^*\}}{C_1 \cup C_2 \cup \{u^*\}} \text{ (LD2)}$$

Variable x is existential, variable u is universal, and $\mathsf{lv}(x) < \mathsf{lv}(u)$. There may be several universal literals u with $\mathsf{lv}(x) < \mathsf{lv}(u)$ that are merged in the resolution step.

Fig. 2. The rules of LD-Q-Res [2]

$$\frac{}{\{l^{\tau_l} \mid l \in C, l \text{ is existential }\} \cup \{\tau(l) \mid l \in C, l \text{ is universal}\}} \text{ (Axiom)}$$

C is a clause from the matrix. τ is an assignment to all universal variables, τ_l are assignments from τ to only variables u with $u < l$.

$$\frac{C_1 \cup \{x^\tau\} \qquad C_2 \cup \{\neg x^\tau\}}{C_1 \cup C_2} \text{ (Res)}$$

Fig. 3. The rules of $\forall$Exp+Res (adapted from [17])

additionally supports resolution over universal variables [35]. Another extension of Q-Res are *variable dependencies* [30,32,34] which enable more flexible $\forall$-reduction than traditional Q-Res; this leads to speedups in solving [26]. For proofs of true QBFs *term-resolution* was developed [11] or *models* in the form of Boolean functions [22] but those do not provide polynomially-verifiable proof system. Some limitation of term-resolution were shown by Janota et al. [13]. A comparison of sequent calculi [25] and Q-Res was done by Egly [9].

3 IRF-calc, an Instantiation Calculus for QBF

This section introduces a calculus called an instantiation calculus as it effectively enables instantiating a variable in a clause by a constant; such instantiation is recorded in the form of a superscript of appropriate existential or variables. The calculus operates on clauses where any existential variable is *annotated* with a cube on universal variables of the formula. Formally, a *cube* is a partial assignment to universal variables. Further, if there is a clause C containing the literal x^t, the cube t contains only universal variables with level smaller than the level of x, i.e. $l < x$ for any $l \in t$. A cube is *contradictory* if it contains $0/u$ and $1/u$ for any universal variable u. The calculus ensures that the cubes in annotations are non-contradictory.

Before we give the actual rules of the calculus, let us define an auxiliary function $\mathsf{inst}(C, c/u)$, which for a given clause C and the substitution c/u, where $c \in \{0, 1\}$ and u is a universal variable, returns a clause C' obtained from C by applying c/u and by adding c/u to the annotations of all annotated existential literals $l^t \in C$ with $\mathsf{lv}(u) > \mathsf{lv}(l)$. The function inst can be applied only if it does not lead to a contradictory cube in an annotation, and, if the clause does not contain a literal that is made true by the substitution. So for instance $\mathsf{inst}(x^{1/u}, 0/u)$ or $\mathsf{inst}(\neg u \vee b, 0/u)$ are *not* applicable. We extend the function to a set of non-contradicting substitutions, i.e. $\mathsf{inst}(C, \{c/u\} \cup \tau) = \mathsf{inst}(\mathsf{inst}(C, c/u), \tau)$. Intuitively, the function inst "moves" universal literals into annotations, e.g. $\mathsf{inst}(u \vee b, 0/u) = b^{0/u}$ if $\mathsf{lv}(u) < \mathsf{lv}(b)$.

For the calculus we consider the following rules. The side-condition of inst applies to the whole rule whenever it is invoked.

- (A) (*Axiom*) If C is in the matrix, derive the clause C' obtained from C by annotating each existential literal in C with the empty cube.
- (I) (*Instantiation*) If $u \in C$ derive $\mathsf{inst}(C, 0/u)$. If $\neg u \in C$ derive $\mathsf{inst}(C, 1/u)$.
- (R) (*Resolution*) From $C_1 \cup \{\neg x^{t_1}\}$ and $C_2 \cup \{x^{t_2}\}$ derive $\mathsf{inst}(C_1, t_2) \cup \mathsf{inst}(C_2, t_1)$, where x is an existential variable. (Note that the side-condition of inst requires that the cubes t_1 and t_2 must be non-contradictory and that t_2 does not contradict with an existing annotation of C_1 and vice versa.)
- (F) (*Factoring*) From $C \cup \{l^{t_1}, l^{t_2}\}$ where t_1 and t_2 are non-contradictory, derive $C \cup \{l^{t_1 \cup t_2}\}$.

The intuition of (R) is that the resolved clauses are brought into a form where the literals being resolved on have the same annotation in both clauses, e.g. resolving $b^{0/p} \vee c^{0/p}$ and $\neg b^{1/u} \vee d^{1/u}$ yields $c^{0/p,1/u} \vee d^{0/p,1/u}$ for the prefix $\forall up \exists bcd$.

We define calculi IR-calc and IRF-calc, which both contain the axioms (A) and rules (I) and (R); the calculus IRF-calc additionally contains the factoring (F) rule. First we show that the calculi can simulate Q-Res and $\forall$Exp+Res and therefore they are complete. Soundness is shown later in Section 4.1 along with the soundness of a more general calculus introduced in Section 4.

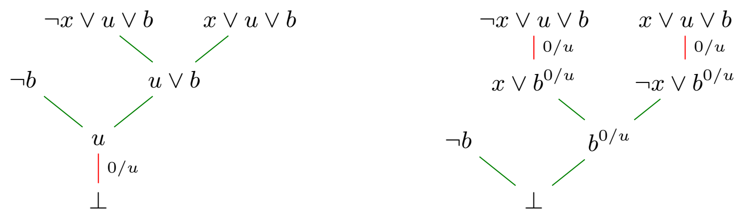

Fig. 4. Examples of Q-Res proofs in IR-calc over the prefix $\exists x \forall u \exists b$.

3.1 Simulations

Theorem 1. *IR-calc p-simulates Q-Res.*

Proof. Given a QBF Γ and its Q-Res refutation π, we find an IR-calc refutation π^* of Γ where the size of π^* is polynomial in the size of π. We proceed by induction.

Induction Hypothesis (on the number of clauses in π): Given a QBF Γ and a Q-Res derivation π of clause C we can find an IR-calc derivation π' of size polynomial in π, which is derived from π by annotating each literal with the empty cube.

Base Case (C is an axiom): We can find π' immediately by applying the axiom rule.

Inductive Step: For any resolution step in π we apply a resolution in IR-calc on the same clauses, since by the induction hypothesis all cubes are empty, they remain empty by resolution.

Whenever a universal literal l is $\forall$-reduced in the Q-Res proof, instantiation is applied on that literal. Since $\forall$-reduction is applied on the literal only if the literal is of the highest level in the clause, the instantiation does not lead to any new annotations in the clause, thus preserving the induction hypothesis. ◻

A simple proof corresponding to a Q-Res proof can be found in Figure 4(a). Alternatively, one could instantiate all universal variables at the beginning of the proof as in Figure 4(b).

Remark 2. The proof of Theorem 1 shows that IR-calc can be easily extended to support resolution on universal variables and thus p-simulate QU-resolution [35].

Theorem 3. *IR-calc p-simulates $\forall$Exp+Res.*

Proof. We observe that IR-calc and $\forall$Exp+Res operate similarly to one another as both annotate existential literals with assignments to universal variables. The difference between them is that $\forall$Exp+Res requires that *all* universal variables in the formula are given a value when an axiom is introduced, whereas in IR-calc this is done only when necessary, i.e. annotations in $\forall$Exp+Res are more restrictive than in IR-calc.

For any $\forall$Exp+Res proof we can construct a similar IR-calc proof by some simple steps. Remove all universal literals from axioms at the beginning, by instantiation. Perform resolution steps as in the $\forall$Exp+Res proof. This procedure guarantees that an IR-calc clause contains a literal l^t iff the corresponding $\forall$Exp+Res clause contains a literal $l^{t'}$ with $t \subseteq t'$. Consequently, IR-calc resolution steps are valid. ◻

Corollary 4. *The calculi IR-calc and IRF-calc are complete.*

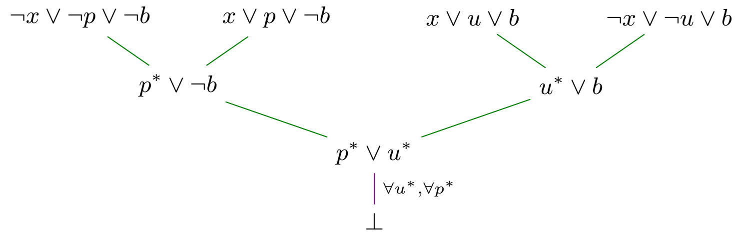

Fig. 5. An LD-Q-Res refutation problematic for IRF-calc under the prefix $\exists x \forall p \forall u \exists b$.

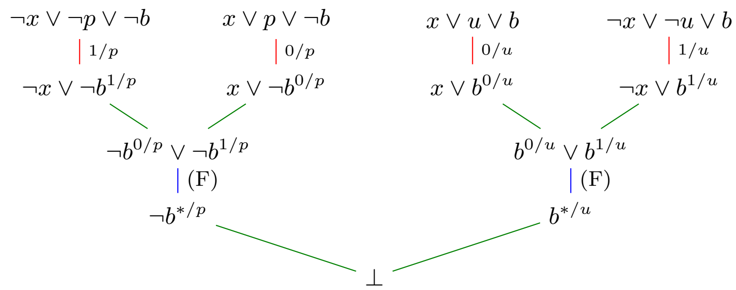

Fig. 6. Example of an IRF*-calc proof under the prefix $\exists x \forall p \forall u \exists b$.

As another consequence, we obtain a separation:

Corollary 5. *The calculi $\forall$Exp+Res and IR-calc are exponentially separated, i.e., there exist formulas that require exponential-size proofs in $\forall$Exp+Res but admit polynomial-size proofs in IR-calc.*

This follows by combining results from [16] ($\forall$Exp+Res does not p-simulate Q-res) and Theorem 1.

Observe that IRF-calc can immediately replicate certain LD-Q-Res steps by resolution and subsequent factoring. For instance resolution of $\neg x \vee \neg u \vee b$ and $x \vee u \vee c$ corresponds to $\neg x \vee b^{1/u}$ and $x \vee c^{0/u}$ resulting into $b^{1/u} \vee c^{0/u}$ (corresponding to $u^* \vee b \vee c$). Duplicate literals, such as $b^{0/p}$ and $b^{1/u}$, can be merged into one by factoring. Factoring, however, is not enabled for literals with complementary annotations, e.g. when resolving $\neg x \vee \neg u \vee b$ and $x \vee u \vee b$.

4 Extending IRF-calc

We observed a significant difference between LD-Q-Res and IR-calc. LD-Q-Res never creates different copies of the same literal whereas IR-calc may do so. This issue is illustrated by an LD-Q-Res proof in Figure 5. In IR-calc, instead of the clauses $u^* \vee b$ and $p^* \vee \neg b$, the clauses $b^{0/u} \vee b^{1/u}$ and $\neg b^{0/p} \vee \neg b^{1/p}$ would be produced. It is not clear how this LD-Q-Res resolution step should be replicated. This indicates an issue—one can easily construct such formula with more universal variables.

Motivated by this issue, we extend IRF-calc with an operation that lets the calculus "merge" two literals with contradictory annotations. We extend IRF-calc to IRF*-calc by the following modification. Extend annotations with the notation $*/u$. The intuition is that $*$ corresponds to both 0 and 1. Additionally to the existing side-condition of inst, in the extended version of the calculus, $\mathsf{inst}(C, */u)$ must not be applied when C contains a literal l^t with $c/u \in t$. Conversely, $\mathsf{inst}(C, c/u)$ is not applicable if C contains a literal annotated with $*/u$.

Table 1. Reduction of an IRF*-calc proof π by τ into π_τ. Clauses C_1, C_2 are part of π_τ; the clauses D_1, D_2 are the corresponding clauses of π; $c \in \{0,1\}$. The condition $l \in \top$ is considered true for any l.

(a) Axiom reduction

If	$\text{axiom}(D_1)$
$l^t \in D_1$ s.t. $\tau(l) = 1$	$\top$
otherwise	$\{l^t \in D_1 \mid \tau(l) \neq 0\}$

(b) Resolution reduction

If	$\mathsf{res}_x(D_1, D_2)$
$x^{t_1} \in C_1$ and $C_2 = \top$	$\top$
$C_1 = \top$ and $\neg x^{t_2} \in C_2$	$\top$
$x^{t_1} \notin C_1$ and $\neg x^{t_2} \in C_2$	C_1
$x^{t_1} \in C_1$ and $\neg x^{t_2} \notin C_2$	C_2
$x^{t_1} \notin C_1$ and $\neg x^{t_2} \notin C_2$	shorter of C_1, C_2
otherwise	$\mathsf{res}_x(C_1, C_2)$

(c) Instantiation reduction

If	$\mathsf{instant}(D_1, c/u)$
$C_1 = \top$	$\top$
$c = 0$ and $u \notin C_1$	C_1
$c = 1$ and $\neg u \notin C_1$	C_1
otherwise	$\mathsf{instant}(C_1, c/u)$

(d) Factoring reduction

If	$\mathsf{factr}(D_1, x^{t_1}, x^{t_2})$
$C_1 = \top$	$\top$
$x^{t_1} \notin C_1$ or $x^{t_2} \notin C_1$	C_1
otherwise	$\mathsf{factr}(C_1, x^{t_1}, x^{t_2})$

(A) (*Axiom*) If C is in the matrix, derive the clause C' obtained from C by annotating each existential literal in C with the empty cube. (Unchanged from the from previous version.)

(F) (*Factoring*) For a clause $C \cup \{l^{t_1}, l^{t_2}\}$ derive the clause $C \cup \{l^t\}$, where t is defined as follows. If $c/u \in t_1$ and $d/u \in t_2$ s.t. $c \neq d$, for $c, d \in \{0, 1, *\}$, then $*/u \in t$. Otherwise, for $i \in \{1, 2\}$ and $c \in \{0, 1, *\}$, if $c/u \in t_i$ then $c/u \in t$. (Substitutions are copied into the resulting one except for the contradicting ones, which are merged into the $*$ substitution.)

(I) (*Instantiation*) If $u \in C$ derive $\mathsf{inst}(C, 0/u)$. If $\neg u \in C$ derive $\mathsf{inst}(C, 1/u)$.

(R) (*Resolution*) From $C_0 \cup \{\neg x^{\tau_0}\}$ and $C_1 \cup \{x^{\tau_1}\}$ derive $\mathsf{inst}(C_0, \tau_1) \cup \mathsf{inst}(C_1, \tau_0)$. Additionally to the side-condition of inst, if $*/u \in \tau_i$ for some u, then $c/u \notin \tau_{1-i}$ for $i \in \{0, 1\}$ and $c \in \{0, 1, *\}$.

The refutation in Figure 6 is an example of an IRF*-calc proof replicating the LD-Q-Res proof from Figure 5.

4.1 Soundness and Strategies for IRF*-calc

The purpose of this section is twofold: show how to obtain a winning strategy for the universal player given an IRF*-calc proof, and, to show that IRF*-calc is sound (and therefore also IRF-calc). First we show how to obtain a winning strategy for the universal player from a proof. From this, the soundness of the calculus follows because a QBF is false if and only if such strategy exists.

The approach we follow is similar to the one used for Q-Res [12] or LD-Q-Res [10]. Given an IRF*-calc proof, we provide an algorithm that responds with the appropriate assignments to the universal variables after given an assignment to the existential variables. The order of the assignments follows the game-based point of view on QBF (see Section 2). (An alternative to this approach would be to construct a set of Boolean functions for the strategy as in [2].)

More precisely, consider a QBF $\Gamma = \exists E \forall U. \Phi$, where E and U are sets of variables and Φ is a QBF—potentially with further quantification. Let π be an IRF*-calc refutation of Γ, and let τ be a total assignment to E. Reduce π into a π_τ by τ so that π_τ is a refutation

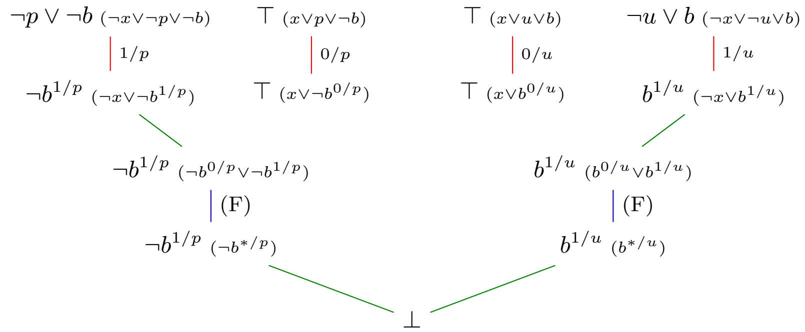

Fig. 7. A reduction of proof Figure 6 by $1/x$ with the original clauses in parentheses; under the prefix $\exists x \forall p \forall u \exists b$.

of $\forall U. \Phi[\tau]$ (recall that assignments are treated as substitutions). We will calculate a total assignment μ to U from π_τ. The assignment μ constitutes a *response* of the universal player to the assignment τ made by the existential player. Reduce π_τ by μ, obtaining a refutation $\pi_{\tau,\mu}$ of $\Phi[\tau, \mu]$. Repeat this procedure with the formula $\Phi[\tau, \mu]$ and the refutation $\pi_{\tau,\mu}$. The process stops when Φ does not contain any universal variables. We follow this notation for the rest of the section.

Reduction of π by τ is done similarly as for Q-Res [12] or LD-Q-Res [10]. The rules of the reduction are shown in Table 1. Each operation made in the refutation π is repeated in π_τ according to the rules given by Table 1. In some cases, the reduction produces the special clause $\top$. Intuitively, the $\top$ clause corresponds to deleting the clause from the original proof.

In the case of the axiom clauses (see Table 1(a)), the reduction introduces into π_τ the same axiom as π but performs the substitution τ (recall that the objective is to construct a refutation of $\forall U. \Phi[\tau]$). Observe that $\top$ is produced if the axiom contains a literal l for which $\tau(l) = 1$ since such clause is no longer available in the matrix of $\forall U. \Phi[\tau]$. In the case of the instantiation and factoring operation, the reduction performs them only if they are applicable. The reduction of resolution steps is such that the literal being resolved on does not appear in the resolvent unless this is impossible, then the resolution step results into $\top$ (see Table 1(b)).

Note that the reduction is defined so that the reduced proof π_τ contains some "dummy" steps, i.e. a clause C may be derived from clauses C and D. In such cases, D is not considered to be part of the proof of C. While in practice such steps would be removed, here they simplify further discussion as each clause in π_τ has its corresponding original clause in π.

Figure 7 illustrates a reduction of the example proof Figure 6. The original clauses are displayed in parentheses. Observe that there are resolution steps where one of the antecedents is ignored and the other one is simply copied.

Consider π_τ as a directed graph and delete all nodes that are not connected to the sink $\perp$.

Consider the proof π_τ and collect all substitutions on U that appear in this proof to make set μ_1. We also use a second ingredient, we consider π_τ as a sequence and consider the first clause in π_τ that contains only universal variables and take all substitutions (restricted to substitutions on U) needed to refute it as set μ_2. This is similar to the construction in [12].

To obtain an assignment μ to U, we take the union of μ_1 and μ_2 and make these into a total assignment by choosing arbitrary values for variables that do not appear in the collection. So for instance in Figure 7 collect the assignments $1/p$ and $1/u$. To obtain $\pi_{\tau,\mu}$, remove occurrences of U-variables from the proof of $\perp$ in π_τ.

To show that this procedure is correct, we need to show that the reduction returns a valid IRF*-calc refutation π_τ, and, that π_τ does not contain any annotations giving contradictory values to variables in U. We begin by a lemma showing that π_τ is a valid IRF*-calc proof (but not necessarily a refutation).

Lemma 6. *Consider π_τ constructed as above; π_τ is a valid IRF*-calc proof.*

Proof. By induction on derivation depth we prove that the derivation of any clause in π_τ is a valid IRF*-calc proof along with the property that annotations in π_τ are "subsets" of annotations in π. We give a precise version of this.

Induction Hypothesis (on the derivation depth): For any clause C in π_τ it holds that if $x^t \in C$ then there is an $x^{t'}$ in the original clause s.t. if $c/u \in t$ then $c/u \in t'$ or $*/u \in t'$ for $c \in \{0, 1, *\}$. (Intuitively, for the subset notion, we imagine that $*/u$ corresponds to $1/u, 0/u \in t$.) Furthermore π_τ up to the clause C is a valid proof.

Base Case: The induction hypothesis is satisfied by the axiom reduction as all annotations are empty.

Inductive Step: The hypothesis is preserved by the factoring rule. The application of instantiation is possible due to the induction hypothesis and the fact that the corresponding instantiation steps are valid in π. Instantiation preserves the induction hypothesis as it only may add annotations if they are present in the original proof π.

Let us focus now on the resolution step (Table 1(b)). The only problematic case is when the actual resolution is performed (when $x^{t_1} \in C_1$, $\neg x^{t_2} \in C_2$, and none of C_1, C_2 is $\top$ from Table 1(b)), because otherwise the resolvent is just a copy of one of its antecedents. Hence, consider a resolution step of clauses $x^{t_1} \vee C_1$ and $\neg x^{t_2} \vee C_2$ in π_τ. These correspond to some original clauses $x^{t'_1} \vee D_1$ and $\neg x^{t'_2} \vee D_2$ in π.

First we need to verify that t_1 and t_2 can be merged, i.e. there is no $c/u \in t_1$ and $d/u \in t_2$ s.t. $c \neq d$, and that there is no $*/u \in t_1$ and $c/u \in t_2$ (without loss of generality, which shall be assumed for the remainder of this proof).

Due to the induction hypothesis, it cannot be that $*/u \in t_1$ and $c/u \in t_2$ as these would appear in t'_1 and t'_2 as well. If $0/u \in t_1$ and $1/u \in t_2$ then due to induction hypothesis the following scenarios may appear in the original clauses: (1) $0/u \in t'_1$ and $1/u \in t'_2$ (2) $0/u \in t'_1$ and $*/u \in t'_2$ (3) $*/u \in t'_1$ and $1/u \in t'_2$ (4) $*/u \in t'_1$ and $*/u \in t'_2$. All these scenarios are prohibited by the side-condition of resolution. Using the same argument, we show that the operations $\mathsf{inst}(C_2, t_1)$ and $\mathsf{inst}(C_1, t_2)$ are valid. ☐

In the following lemma we show that π_τ is indeed an IRF*-calc refutation, i.e. that it derives $\perp$.

Lemma 7. *Consider the proof π_τ constructed as above; π_τ is an IRF*-calc refutation.*

Proof. The proof proceeds by induction on derivation depth.

Induction Hypothesis (on derivation depth): If a clause C in π_τ is $\top$, then the corresponding original clause in π contains a literal l^P s.t. $\tau(l) = 1$.

To see how the lemma follows from this induction hypothesis we also recall the induction hypothesis proven in the proof of Lemma 6, which says if a clause in π_τ contains some literal l^t, then the original clause π must contain a corresponding literal. This, together with the induction hypothesis we are about to prove, lets us conclude that the root clause C of π_τ must be $\perp$ because it corresponds to the original clause $\perp$ in π and therefore C cannot be $\top$ and it cannot contain any literals.

Base Case: The induction hypothesis is satisfied by the reduction of an axiom as a clause D is replaced by $\top$ if $l \in D$ s.t. $\tau(l) = 1$ (see Table 1(a)).

Inductive Step: The hypothesis is preserved by factoring and instantiation as these produce $\top$ iff given $\top$ as the argument.

Consider a resolution step of C_1 and C_2 over x^{t_1} and $\neg x^{t_2}$ with the resolvent R. These correspond to the original clauses D_1, D_2, and Z, respectively. For the resolvent R to be $\top$, either both C_1 and C_2 are $\top$ or one of them is $\top$ and the second contains the literal to be resolved on (Table 1(b)).

Hence, for R to be $\top$ at least one of C_1, C_2 must be $\top$. Without loss of generality, let $C_1 = \top$. From the induction hypothesis, D_1 contains a literal l^t s.t. $\tau(l) = 1$. This literal is copied into the original resolvent Z unless it is the literal x^{t_1}. If l^t is copied into Z, then the induction hypothesis is preserved. If $\tau(x) = 1$, then $\tau(\neg x) = 0$. Therefore, if $R = \top$, there must be a different literal in D_2 that evaluates to 1, which is copied into Z. $\square$

The following lemma shows that the refutation π_τ enables us to calculate an unambiguous assignment to U by joining μ_1 and μ_2.

Lemma 8. *Consider the IRF*-calc proof π_τ defined as above and the set μ which is the union of μ_1 and μ_2 as defined before Lemma 6. The set μ does not contain both $0/u$ and $1/u$ nor does it contain $*/u$ for any $u \in U$.*

Proof. By induction on derivation depth we show that the condition holds for any derived clause in π_τ. Consider the set μ_C comprising all annotations appearing in the proof of C. Let us consider an induction hypothesis comprising the following conditions on any derived clause C.

Induction Hypothesis (on the derivation depth):

(1) If $c/u \in \mu_C$ for $u \in U$ then $c/u \in t$ for all $l^t \in C$. (Any annotation on U that appears in the proof of C appears in annotations of all literals of C.)
(2) There is no $u \in U$ such that $0/u \in \mu_C$ and $1/u \in \mu_C$. (There are no complementary values for variables on U appearing in the proof of C.)
(3) There are no $*/u \in \mu_C$ for $u \in U$. (Note that this follows from (2) because Factoring cannot produce $*/u$.)
(4) If $u \in C$ for $u \in U$, then $1/u \notin \mu_C$, likewise if $\neg u \in C$ for $u \in U$, then $0/u \notin \mu_C$.

Base Case: The induction hypothesis is satisfied by axioms since all annotations are empty.

Inductive Step: If a literal $u \in C$ with $u \in U$ is instantiated, then $0/u$ must be added to annotations of *all* existential literals in C because the variables U have the lowest level in π_τ (recall that the first level $\exists E$ was reduced by τ). Thus (1) is preserved. Further, instantiation can only be performed if it does not lead to contradictory annotations, so (2) is preserved. It can also not be performed if a literal satisfied by the substitution appears in the clause, so (4) is preserved. The condition (3) is preserved as no new $*/u$ annotation can be produced by instantiation.

Factoring cannot produce an annotation $*/u$ for $u \in U$ due to the condition (2) of the induction hypothesis.

Let us focus now on the resolution step (Table 1(b)). As in Lemma 6, the only problematic case is when the actual resolution is performed because otherwise the resolvent is just a copy of one of its antecedents. Hence, consider a resolution step of clauses $x^{t_1} \vee C_1$ and $\neg x^{t_2} \vee C_2$ in π_τ.

Consider all substitutions t' to U that appear in both t_1 and t_2, i.e. let $t' = \{l \in t_1 \cup t_2 \mid \mathsf{var}(l) \in U\}$. Observe that from the definition of resolution, the set t' must be non-contradictory and all $c/u \in t'$ appear in s for all p^s in the resolvent $\mathsf{inst}(C_2, t_1) \cup \mathsf{inst}(C_1, t_2)$.

The set t' contains all literals on U for the proofs of either of the antecedents due to the condition (1) of the induction hypothesis and thus this condition is preserved. Consequently, the condition (2) is preserved since τ' is non-contradictory. Furthermore the condition (4) is preserved as the inst function would not be allowed if it added an annotation that satisfied any literal.

From the induction we have proved that conditions (2) and (3) are true for when C is the sink hence μ_1 is non-contradictory. The set μ_2 is easily seen as non-contradictory; the clause it refutes will not be tautological else we cannot instantiate and we cannot produce a $*/u$ in the set μ_2.

We will argue that μ_1 and μ_2 are mutually non-contradictory. Let C be the first purely universal clause, then condition (4) gives us this as no annotations appear later in the proof (only instantiation is performed with no existential literals).

Therefore μ must be non-contradictory. ☐

From Lemma 8, the strategy for the universal player is well-defined. We need to show that this is also a winning strategy.

Theorem 9. *The process above constructs a winning strategy for the universal player.*

Proof. For any QBF $\Gamma = \exists E \forall U. \Phi$, and τ, the construction provides a $\pi_{\tau,\mu}$ that is an IRF*-calc refutation of $\Phi[\tau, \mu]$. After this process is iterated until no universal variables are left in the formula, we are left with an IRF*-calc refutation of whatever was left from the matrix of Γ. Since an IRF*-calc refutation on a formula with no universal variables is in fact a classic propositional resolution refutation, we are left with an unsatisfiable formula, i.e. a formula with no winning move for the existential player. Hence, all the considered assignments correspond to a game won by the universal player. Since this process works for any assignment made by the existential player, this process provides a winning strategy for the universal player. ☐

The soundness of IRF*-calc follows directly from Theorem 9.

Corollary 10. *The calculi IR-calc, IRF-calc, and IRF*-calc are sound.*

4.2 Simulation of LD-Q-Res by IRF*-calc

Consider an LD-Q-Res refutation $C_1, \dots, C_n$. We construct clauses $D_1, \dots, D_n$, which will form the skeleton of the IRF*-calc proof. The construction proceeds as follows. If C_i is an axiom, D_i is constructed by the axiom rule from the same clause and instantiation of all universal literals. If C_i is a $\forall$-reduction of C_j with $j < i$, then we set D_i equal to D_j. If C_i is obtained by a resolution step from C_j and C_k with $j < k < i$, the clause D_i is obtained by a resolution step from D_j and D_k, and factoring of any literals l^{t_1} and l^{t_2} (using the (F) rule). In order to prove a p-simulation we need the following lemma.

Lemma 11. *The construction above yields a valid IRF*-calc refutation.*

Proof. The construction establishes the following invariant for all D_i, $i \in 1, \ldots, n$, which we show by induction on i.

Induction hypothesis (on i):

(1) For an existential literal l, it holds that $l \in C_i$ iff $l^t \in D_i$ for some t.
(2) The clause D_i has no literals l^{t_1} and l^{t_2} such that $t_1 \neq t_2$.
(3) If $l^t \in D_i$ with $*/u \in t$, then $u^* \in C_i$.
(4) If $l^t \in D_i$ with $0/u \in t$, then $u \in C_i$ or $u^* \in C_i$.
(5) If $l^t \in D_i$ with $1/u \in t$, then $\neg u \in C_i$ or $u^* \in C_i$.

Base Case: All conditions hold for the axioms and initial instantiations.

Inductive Step: We distinguish two cases on the nature of the rule that was applied to derive C_i in the LD-Q-Res proof.

If C_i is a $\forall$-reduction of C_j in the LD-Q-Res refutation, then C_i and D_i both retain the same existential literals. Since D_i is the same as D_j there are no literals with differing annotations. For conditions (3)–(5) we argue that if $l^t \in D_i$ and the annotation c/u appears in t then by condition (1) that literal l appears in C_j. That l blocks the literal of u from being $\forall$-reduced in that step hence the literal is retained for C_i.

If C_i is derived by a resolution step, (1) holds true. Condition (2) holds because we always perform factoring in the IRF*-calc refutation. Conditions (3)–(5) hold because universal variables are not lost in resolution, the only consideration is that the variable is merged in the LD-Q-Res proof, but conditions (3)–(5) allow that. This finishes the proof of the induction.

It remains to show that all IRF*-calc steps are valid. Consider resolving $D_j = x^{t_0} \vee D'_0$ and $D_k = \neg x^{t_1} \vee D'_1$ in the constructed IRF*-calc proof, which corresponds to a resolution step of C_j and C_k in the original LD-Q-Res proof. We need to be show that the side-condition of resolution is fulfilled. First, t_0 and t_1 must be mutually non-contradictory and if $*/u \in t_i$ then u is not in t_{1-i} for $i \in \{0, 1\}$. Second, the operations $\mathsf{inst}(D'_i, t_{i-1})$ must be permitted for $i \in \{0, 1\}$. Both conditions are argued for by contradiction. Without a loss of generality, let $1/u \in t_0$ and $0/u \in t_1$ or $*/u \in t_0$ and $c/u \in t_1$, for $c \in \{0, 1, *\}$. Then from conditions (3)–(5), there is a corresponding literal u, $\neg u$, or u^* with $u < x$ in C_j and C_k. These literals, however, would prohibit such step in LD-Q-Res (contradiction). Now let us assume that $c/u \in t_0$ and $d/u \in t$ for some $l^t \in D_1$ such that $c \neq d$ or one of them is $*$. Observe that $u < x$ since t_1 contains substitutions only to variables with level lower than x. As in the previous argument, this would lead to universal literals in C_j and C_k that are prohibited by LD-Q-Res. ☐

Theorem 12. *IRF*-calc p-simulates LD-Q-Res.*

Proof. We observe that the stages in the construction allow it to be constructed in polynomial time. Lemma 11 tells us that it is a valid refutation. Furthermore, if we derive $\perp$ in an LD-Q-Res proof, by (1) from Lemma 11 we derive a clause with no existential literals. It also contains no universal variables as these are all instantiated at the axioms. Therefore, this must also be the empty clause $\perp$. ☐

4.3 Instantiation and First Order Logic

Here we would like to point out an important relation between IRF-calc and first order logic (FOL). QBF can be easily defined as special case of FOL [31]. Then, Robinson's FOL-based resolution [29,1] can be used as a proof system for QBF. Let us briefly review the FOL formulation of QBF. The translation introduces a predicate P (representing truth) and replaces each occurrence of a variable x in the given QBF with $P(x)$. Next it replaces each existential variable with an appropriate Skolem function. Lastly, it introduces two constants $\top$ and $\bot$ along with the axioms $P(\top)$ and $\neg P(\bot)$; cf. [31].

Due to the structure of the FOL clauses obtained by this translation, the following operations may take place. If a clause contains a literal $P(u)$, where u is universal, the axiom $\neg P(\bot)$ can be used to remove $P(u)$ from the clause, which effectively replaces u with $\bot$ in that clause. Analogously, $\top$ replaces u if the clause contains the literal $\neg u$. This corresponds to the operation inst in IRF-calc. If two clauses are resolved over the literals $P(f(\dots))$ and $\neg P(f(\dots))$, the function arguments are *unified*, which effectively means that if some variable was replaced with $c \in \{\top, \bot\}$ in one of the atoms earlier, it must be done so also in the other atom now. This correspond to the resolution step in IRF-calc. Note, however, that in FOL resolution, also variables shared between the antecedents must be renamed. *Factoring* in FOL corresponds to factoring in IRF-calc. All these correspondences, are not exact. Further, application of FOL to QBF is not pragmatic—besides the syntactic overhead, unification and variable renaming is required for any resolution step. It is unclear how IRF*-calc corresponds to FOL resolution.

5 Conclusions and Future Work

This paper introduces a novel resolution-based calculus for QBF (with several variations). This calculus not only provides a formalism needed for unified certification of QBF solvers, namely DPLL and expansion-based solvers, but also, it provides us with a number of insights. Firstly, we observe that all the resolution-based calculi are in a way similar to Robinson's resolution in first-order logic. Hence, we can view the calculi as reasoning on Skolem functions. Indeed, the clause $u \vee b$ tells us that the whole QBF is false when the Skolem function $b(u)$ returns 0 whenever $u = 0$. In the introduced calculus this is recorded directly through the annotation $b^{0/u}$. Note that Q-resolution does not enable us to reason about two different values of u, i.e. u and $\neg u$ cannot appear in the same clause. In long-distance Q-resolution, this is enabled by the u^* notation, which represents that u may take either value. The introduced calculus, through annotations, gives us a more fine-grained approach because instead of $u^* \vee c \vee d$ we can infer clauses such as $c^{0/u} \vee d^{1/u}$, which records what happens for each of the values of u. The paper further indicates that this fine-grained approach can be sometimes harmful. For this, the calculus contains the operation of factoring, which lets us merge literals of the form x^t and $x^{t'}$ into a single literal.

The paper opens a number of avenues for further research. From theoretical perspective it is interesting to study the respective strengths of the introduced calculus and its restrictions, representing the previously existing ones. Extensions of Q-resolution, such as QU-resolution or variable dependencies, should be explored in the context of the instantiation calculus (QU-resolution was already mentioned in Remark 2).

From the practical perspective, the new calculus begs the question, whether we can develop QBF algorithms based on the presented calculus and thus obtain a more efficient approach to QBF solving.

Acknowledgments

The second author was supported by a Doctoral Training Grant from EPSRC. This work was partially supported by FCT grants ATTEST (CMU-PT-/ELE/0009/2009), POLARIS (PTDC/EIA-CCO/123051/2010), INESC-ID's multiannual PIDDAC funding PEst-OE/EEI/LA0021/2011, and a grant from the John Templeton Foundation.

References

1. Bachmair, L., Ganzinger, H.: Resolution theorem proving. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 19–99. Elsevier and MIT Press (2001)
2. Balabanov, V., Jiang, J.H.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012)
3. Benedetti, M.: Evaluating QBFs via symbolic Skolemization. In: Baader, F., Voronkov, A. (eds.) LPAR. vol. 3452, pp. 285–300. Springer (2004)
4. Biere, A.: Resolve and expand. In: SAT. pp. 238–246 (2004)
5. Bubeck, U.: Model-based transformations for quantified Boolean formulas. Ph.D. thesis, University of Paderborn (2010)
6. Cimatti, A., Sebastiani, R. (eds.): Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings, vol. 7317. Springer (2012)
7. Cook, S.A., Nguyen, P.: Logical Foundations of Proof Complexity. Cambridge University Press (2010)
8. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. J. Symb. Log. 44(1), 36–50 (1979)
9. Egly, U.: On sequent systems and resolution for QBFs. In: Cimatti and Sebastiani [6], pp. 100–113
10. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: McMillan et al. [28], pp. 291–308
11. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause/term resolution and learning in the evaluation of quantified Boolean formulas. Journal of Artificial Intelligence Research 26(1), 371–416 (2006)
12. Goultiaeva, A., Van Gelder, A., Bacchus, F.: A uniform approach for generating proofs and strategies for both true and false QBF formulas. In: Walsh, T. (ed.) IJCAI. pp. 546–553. IJCAI/AAAI (2011)
13. Janota, M., Grigore, R., Marques-Silva, J.: On QBF proofs and preprocessing. In: McMillan et al. [28], pp. 473–489
14. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. In: Cimatti and Sebastiani [6], pp. 114–128
15. Janota, M., Marques-Silva, J.: Abstraction-based algorithm for 2QBF. In: Sakallah, K.A., Simon, L. (eds.) SAT. pp. 230–244. Springer (2011)
16. Janota, M., Marques-Silva, J.: ∀Exp+Res does not P-Simulate Q-resolution. International Workshop on Quantified Boolean Formulas (2013)
17. Janota, M., Marques-Silva, J.: On propositional QBF expansions and Q-resolution. In: Järvisalo and Van Gelder [18], pp. 67–82
18. Järvisalo, M., Van Gelder, A. (eds.): Theory and Applications of Satisfiability Testing - SAT 2013, vol. 7962. Springer (2013)
19. Jussila, T., Biere, A., Sinz, C., Kröning, D., Wintersteiger, C.M.: A first step towards a unified proof checker for QBF. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT. vol. 4501, pp. 201–214. Springer (2007)
20. Kleine Büning, H., Bubeck, U.: Theory of quantified boolean formulas. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 735–760. IOS Press (2009)
21. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. 117(1), 12–18 (1995)
22. Kleine Büning, H., Subramani, K., Zhao, X.: Boolean functions as models for quantified boolean formulas. J. Autom. Reasoning 39(1), 49–75 (2007)
23. Klieber, W., Janota, M., Marques-Silva, J., Clarke, E.M.: Solving QBF with free variables. In: Schulte, C. (ed.) CP. Lecture Notes in Computer Science, vol. 8124, pp. 415–431. Springer (2013)
24. Klieber, W., Sapra, S., Gao, S., Clarke, E.M.: A non-prenex, non-clausal QBF solver with game-state learning. In: Strichman, O., Szeider, S. (eds.) SAT. vol. 6175, pp. 128–142. Springer (2010)
25. Krajíček, J., Pudlák, P.: Quantified propositional calculi and fragments of bounded arithmetic. Mathematical Logic Quarterly 36(1), 29–46 (1990)
26. Lonsing, F., Biere, A.: DepQBF: A dependency-aware QBF solver. JSAT 7(2-3), 71–76 (2010)
27. McMillan, K.L.: Interpolation and SAT-based model checking. In: Jr., W.A.H., Somenzi, F. (eds.) CAV. Lecture Notes in Computer Science, vol. 2725, pp. 1–13. Springer (2003)

28. McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.): Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings, Lecture Notes in Computer Science, vol. 8312. Springer (2013)
29. Robinson, J.A.: A machine-oriented logic based on the resolution principle. J. ACM 12(1), 23–41 (1965)
30. Samer, M., Szeider, S.: Backdoor sets of quantified Boolean formulas. J. Autom. Reasoning 42(1), 77–97 (2009)
31. Seidl, M., Lonsing, F., Biere, A.: qbf2epr: A Tool for Generating EPR Formulas from QBF. In: Fontaine, P., Schmidt, R.A., Schulz, S. (eds.) PAAR@IJCAR. EPiC Series, vol. 21, pp. 139–148. EasyChair (2012)
32. Slivovsky, F., Szeider, S.: Variable dependencies and Q-Resolution. International Workshop on Quantified Boolean Formulas (2013)
33. Van Gelder, A.: Decision procedures should be able to produce (easily) checkable proofs. In: Workshop on Constraints in Formal Verification (2002), (in conjunction with CP02)
34. Van Gelder, A.: Variable independence and resolution paths for quantified Boolean formulas. In: Lee, J.H.M. (ed.) CP. Lecture Notes in Computer Science, vol. 6876, pp. 789–803. Springer (2011)
35. Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Milano, M. (ed.) CP. vol. 7514, pp. 647–663. Springer (2012)
36. Zhang, L., Malik, S.: Conflict driven learning in a quantified Boolean satisfiability solver. In: ICCAD. pp. 442–449 (2002)

ECCC ISSN 1433-8092

http://eccc.hpi-web.de