On Stronger Calculi for QBFs*

Uwe Egly

Institut für Informationssysteme 184/3, Technische Universität Wien,
Favoritenstrasse 9–11, A-1040 Vienna, Austria
email: uwe@kr.tuwien.ac.at

Abstract. Quantified Boolean formulas (QBFs) generalize propositional formulas by admitting quantifications over propositional variables. QBFs can be viewed as (restricted) formulas of first-order predicate logic and easy translations of QBFs into first-order formulas exist. We analyze different translations and show that first-order resolution combined with such translations can polynomially simulate well-known deduction concepts for QBFs. Furthermore, we extend QBF calculi by the possibility to instantiate a universal variable by an existential variable of smaller level. Combining such an enhanced calculus with the propositional extension rule results in a calculus with a universal quantifier rule which essentially introduces propositional formulas for universal variables. In this way, one can mimic a very general quantifier rule known from sequent systems.

1 Introduction

Quantified Boolean formulas (QBFs) generalize propositional formulas by admitting quantifications over propositional variables. QBFs can be viewed in two different ways, namely (i) as a generalization of propositional logic and (ii) as a restriction of first-order predicate logic (where we interpret over a two element domain). A number of calculi are available for QBFs: the ones based on variants of resolution for QBFs [13,11,2,3], the ones based on instantiating universal variables with truth constants combined with propositional resolution and an additional instantiation rule [4], and different sequent systems [7,14,10,9].

In all these calculi (except the latter ones from [7,14,9]), the possibility to instantiate a given formula is limited. In purely resolution-based calculi, formulas (or more precisely universal variables) are never instantiated. In instantiation-based calculi, instantiation is restricted to truth constants. In contrast, sequent systems possess flexible quantifier rules, and (existential) variables as well as (propositional) formulas can be used for instantiation with tremendous speed-ups in proof complexity. This motivates why we are interested in strengthening instantiation techniques for instantiation-based calculi.

We allow to replace (some) universal variables \(x\) not only by truth constants but by existential variables left of \(x\) in the quantifier prefix. This approach mimics the effect of quantifier rules introducing atoms in sequent calculi from [9].

* The work was supported by the Austrian Science Foundation (FWF) under grant S11409-N23. Partial results have been announced at the QBF Workshop 2014 [http://www.easychair.org/smart-program/VSL2014/QBF-program.html].
add a propositional extension principle (known from extended resolution [19]),
which enables the introduction of propositional formulas for universal variables
via extension variables (or names for the formula). Contrary to [9], where we
proposed propositional extensions of the form \( \exists q (q \leftrightarrow F) \) which can be elimi-
nated if the cut rule is available in the sequent calculus, such an elimination is
not possible here for which reason we have to use (classical) extensions.

**Contributions.**

1. We consider different translations from QBFs to first-order logic [17] and
   provide a proof-theoretical analysis of the translation in combination with
   first-order resolution (\( R_1 \)). We exponentially separate two variants of the
   translation in Theorem 4.

2. We show that such combinations can polynomially simulate Q-resolution
   with resolution over existential and universal variables (QU-res [11], The-
  orem 1), Q-resolution (Q-res [13], Corollary 1) and the instantiation-based
calculus IR-calc [1] (Theorem 2, Corollary 2). The latter simulation provides
   a soundness proof for IR-calc independent from strategy extraction.

3. We show in Theorem 3 that neither Q-res nor QU-res, the long-distance
   Q-resolution variants LDQ-res, LDQU-res, LDQU\(^+\)-res [20,2], different in-
   stantiation-based calculi [1] nor Q(D)-res [18] can polynomially simulate \( R_1 \)
   with one of the considered translations.

4. We generalize IR-calc by the possibility to instantiate universal variables not
   only with truth constants but also with existential variables (similar to the
   corresponding quantifier rule in [9]). We show in Proposition 12 that this
generalized calculus is actually stronger than the original one.

5. We combine generalized IR-calc by a propositional extension rule [19,6] es-
   sentially enabling the introduction of Boolean functions (instead of atoms
   and truth constants) for universal variables.

**Structure.** In Sect. 2 we introduce necessary definitions and notations. In Sect. 3
different translations from QBFs to (restrictions of) first-order logic [17] are
reconsidered. In Sect. 4 different calculi based on (variants of) the resolution
calculus are described. Here, we introduce our calculi generalized from IR-calc.
In Sect. 5 we present our results on polynomial simulations between considered
calculi and in Sect. 6 we provide exponential separations. In the last section we
conclude and discuss future research possibilities.

## 2 Preliminaries

We assume familiarity with the syntax and semantics of propositional logic,
QBFs and first-order logic (see, e.g., [15] for an introduction). We recapitulate
some notions and notations which are important for the rest of the paper.

We consider a propositional language based on a set \( PV \) of Boolean variables
and truth constants \( \top \) (true) and \( \bot \) (false), both of which are not in \( PV \). A
variable or a truth constant is called *atomic* and connectives are from \( \{ \neg, \land, \lor, \rightarrow, \leftrightarrow, \oplus \} \). A *literal* is a variable or its negation. A *clause* is a disjunction of literals,
but sometimes we consider it as a set of literals. Tautological clauses contain a variable and its negation and the empty clause is denoted by \(\Box\). Propositional formulas are denoted by capital Latin letters like \(A, B, C\) possibly annotated with subscripts, superscripts or primes.

We extend the propositional language by Boolean quantifiers. Universal \((\forall^b)\) and existential \((\exists^b)\) quantification is allowed within a QBF. The superscript \(b\) is used to distinguish Boolean quantifiers from first-order quantifiers introduced later. QBFs are denoted by Greek letters. Observe that we allow non-prenex formulas, i.e., quantifiers may occur deeply in a QBF. An example for a non-prenex QBF is \(\forall^b p (p \rightarrow \forall^b q \exists^b r (q \land r \land s))\), where \(p, q, r\) and \(s\) are variables. Moreover, free variables (like \(s\)) are allowed, i.e., there might be occurrences of variables in the formula for which we have no quantification. Formulas without free variables are called closed; otherwise they are called open. The universal (existential) closure of \(\varphi\) is \(\forall^b x_1 \ldots \forall^b x_n \varphi\) \((\exists^b x_1 \ldots \exists^b x_n \varphi)\), for which we often write \(\forall^b X \varphi\) \((\exists^b X \varphi)\) if \(X = \{x_1, \ldots, x_n\}\) is the set of all free variables in \(\varphi\). A formula in prenex conjunctive normal form (PCNF) has the form \(Q_1^b p_1 \ldots Q_n^b p_n M\), where \(Q_1^b \ldots Q_n^b p_n\) is the quantifier prefix, \(Q \in \{\forall, \exists\}\) and \(M\) is the (propositional) matrix which is in CNF. Often we write a QBF as \(Q_1^b X_1 \ldots Q_k^b X_k M\) \((Q_i \neq Q_{i+1}\) for all \(i = 1, \ldots, k-1\) and the elements of \(\{X_1, \ldots, X_k\}\) are pairwise disjoint). We define the level of a literal \(\ell\), \(l_c(\ell)\), as the index \(i\) such that the variable of \(\ell\) occurs in \(X_i\). The logical complexity of a formula \(\Phi\), \(l_c(\Phi)\), is the number of occurrences of connectives and quantifiers.

We use a first-order language consisting of (objects) variables, function symbols (FSs), predicate symbols (PSs), together with the truth constants and connectives mentioned above. Quantifiers \(\forall\) and \(\exists\) bind object variables. Terms and formulas are defined according to the usual formation rules. We identify 0-ary FSs with propositional atoms, and 0-ary FSs with constants. Clauses, tautological clauses and the empty clause are defined as in the propositional case.

Let \(V\) be the set of first-order variables and \(T\) be the set of terms. A substitution is a mapping \(\sigma\) of type \(V \rightarrow T\) such that \(\sigma(v) \neq v\) only for finitely many variables \(v \in V\). We represent \(\sigma\) by a finite set of the form \(\{v_1 \setminus t_1, \ldots, v_n \setminus t_n\}\). The domain of \(\sigma\), \(\text{dom}(\sigma)\), is the set \(\{v \mid v \in V, \sigma(v) \neq v\}\). The range of \(\sigma\), \(\text{rg}(\sigma)\), is the set \(\{\sigma(v) \mid v \in \text{dom}(\sigma)\}\). We call \(\sigma\) a variable substitution if \(\text{rg}(\sigma) \subseteq V\). The empty substitution \(\epsilon\) is denoted by \(\{\}\). We often write substitutions post-fix, e.g., we use \(\sigma x\) instead of \(\sigma(x)\). Algebraically, substitutions define a monoid with \(\varepsilon\) being the neutral element under the usual composition of substitutions.

Substitutions are extended to terms and formulas in the usual way, e.g., \(f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)\), \((-p)(t_1, \ldots, t_n)\sigma = (-p)(t_1\sigma, \ldots, t_n\sigma)\), and \((F \circ G)\sigma = F\sigma \circ G\sigma\), where \(f\) is an \(n\)-place FS, \(p\) is an \(n\)-place PS, \(t_1, \ldots, t_n\) are terms, \(F\) and \(G\) are (quantifier-free) formulas and \(\circ\) is a binary connective. For substitutions \(\sigma\) and \(\tau\), \(\sigma\) is more general than \(\tau\) if there is a substitution \(\mu\) such that \(\sigma\mu = \tau\). A substitution \(\sigma\) is called a permutation if \(\sigma\) is one-one and a variable substitution. A permutation \(\sigma\) is called a renaming (substitution) of an expression \(E\) (i.e., \(E\) is a term or a quantifier-free formula) if \(\text{var}(E) \cap \text{rg}(\sigma) = \{\}\),
Traditionally, \( \Phi \) is a prefix of \( \Phi \) if for every prefix in the source QBF; \( p \) is a unary predicate symbol and \( f \) is used to construct constant and function symbols by indices.

Let \( \Phi \) be a closed QBF in PCNF with matrix \( M \). We introduce different translations of (closed) QBFs to (closed) formulas in (restrictions of) first-order logic. We start with the basic translation from [17] in Fig. 1. Obviously, the QBF \( \Phi \) and the first-order formula \( [\Phi]_p \) enjoy a very similar structure. Especially the variable dependencies expressed by the quantifier prefix are exactly the same.

**Proposition 1** Let \( \Phi \) be a (closed) QBF and let \( [\Phi]_p \) be its (closed) first-order translation. Then \( \Phi \cong [\Phi]_p \), i.e., \( \Phi \) and \( [\Phi]_p \) are isomorphic.

The proof in the appendix is by induction on the logical complexity of \( \Phi \).

The basic translations from Fig. 1 can be extended to \( \text{Sk}[\Phi]_p \) generating a skolemized version of \( [\Phi]_p \). We restrict our attention here to QBFs in PCNF.

**Definition 1.** Let \( \Phi \) be a closed QBF in PCNF with matrix \( M \) and let \( [\Phi]_p \) be its closed first-order translation. For any existential variable \( a \) in the quantifier prefix of \( \Phi \), let \( \text{dep}(a) \) be the sequence of universal variables left of \( a \) (in exactly the same order in which they occur in the prefix). Let \( f_a \) be the Skolem function symbol associated to \( a \). We call \( [M]_p \sigma \) the skolemized form of \( [M]_p \) and denote it by \( \text{Sk}[M]_p \), where the substitution \( \sigma \) is as follows.

\[
\sigma = \{ a \backslash f_a(\text{dep}(a)) \mid \text{for all existential variables } a \text{ in } \Phi \}
\]

Traditionally, \( \text{Sk}[M]_p \) is denoted as a quantifier-free formula with the assumption that all free variables are (implicitly) universally quantified.
C is a non-tautological clause from the matrix. If \( y \in C_1 \) then \( \neg y \notin C_2 \). Variable \( x \) is existential (Q-res) and existential or universal (QU-res). \( \ell \) is a literal and \( m \) is a universal literal. If \( e \in D \) is existential, then \( lv(e) < lv(m) \) holds.

![Fig. 2. The rules of Q-res and QU-res](image)

The number of universal variables a Skolem function depends on can be optimized, e.g., by using miniscoping or dependency schemes [17]. As we will see later on, most of our results do not depend on such optimizations.

**Proposition 2.** Let \( \Phi \) be a closed QBF in PCNF with matrix \( M \) and let \( \llbracket \Phi \rrbracket_p \) be its closed first-order translation. Let \( Sk\llbracket M \rrbracket_p^f \) be the skolemized form of \( \llbracket M \rrbracket_p^f \). Then \( M \cong Sk\llbracket M \rrbracket_p^f \).

Due to propositions 1 and 2 we can relate each literal of each clause from \( M \) to its isomorphic counterpart in \( \llbracket M \rrbracket_p^f \).

Since we interpret over a two-element domain, proper Skolem function symbols (i.e., the arity is greater than 0) can be eliminated by introducing new predicate symbols. The resulting formula belongs to EPR (Effectively Propositional logic or more traditionally it belongs to the Bernays-Schoenfinkel class).

**Definition 2.** Let \( \Phi \) be a closed QBF in PCNF with matrix \( M \) and let \( \llbracket \Phi \rrbracket_p^f \) be its closed first-order translation. Let \( Sk\llbracket M \rrbracket_p^f \) the skolemized form of \( \llbracket M \rrbracket_p^f \). Replace any occurrence of a predicate of the form \( p(f_b(X)) \) by \( p_b(X) \) where \( f_b \) is a proper function symbol and \( X \) is a non-empty list of universal variables. The resulting formula after all possible replacements is the EPR formula \( EPR\llbracket M \rrbracket_p^f \).

We will see later that the first-order and the EPR translation have different proof-theoretical properties because some resolutions are blocked by different predicate symbols. Proposition 3 is Lemma 1 in [17] (stated without a proof).

**Proposition 3** Let \( \Phi \) be a closed QBF. Then

\[
\Phi \text{ is satisfiable } \iff \llbracket \Phi \rrbracket_p^f \land p(f_1) \land \neg p(f_0) \text{ is satisfiable.}
\]

A proof can be found in the appendix.

### 4 Different calculi based on resolution

We introduce different calculi used in this paper. We start with two resolution calculi, Q-res and QU-res, for QBFs in Fig. 2. Observe that the consequence of each rule is non-tautological. We continue with the calculus IR-calc\((P, M)\) in Fig. 3 where we use the same presentation as in [1]. \( P \) is the quantifier prefix
\(\text{Axiom}\)

\[
\{e^{[a]} \mid e \in C, e \text{ is existential}\}
\]

\(C\) is a non-tautological clause from the matrix \(M\), \(\sigma = \{u \backslash 0 \mid u \in C \text{ universal}\}\) where \(u \backslash 0\) is a shorthand for \(x \backslash 0\) if \(u = x\) and \(x \backslash 1\) if \(u = \neg x\).

\[
\frac{x^r \lor C_1 \quad \neg x^r \lor C_2}{C_1 \lor C_2} \text{ Res}
\]

\[
\frac{C \lor \ell^r \lor \ell^r}{C \lor \ell^r} \text{ Fac}
\]

\(\text{Inst}\)

\[
\text{inst}(\tau, C)
\]

\(\tau\) is an assignment to universal variables and \(\tau_g(\tau) \subseteq \{0, 1\}\).

**Fig. 3.** The rules of \(\text{IR-calc}(P, M)\) taken from [4].

\[
\{e^{[a]} \mid e \in C\sigma_u, e \text{ is existential}\}
\]

1. \(C\) is a non-tautological clause from the matrix \(M\).
2. \(\sigma_u = \{x \backslash e \mid x \text{ is universal, } e \text{ is existential, } lv(e) < lv(x)\}\).
3. \(C\sigma_u = \{e \mid e \in C \text{ existential}\} \cup \{x\sigma_u \mid x \in C \text{ universal, } x \in \text{ dom}(\sigma_u)\} \cup \{x \mid x \in C \text{ universal, } x \notin \text{ dom}(\sigma_u)\}\).
4. \(\sigma, \text{ Res, Fac and Inst}\) are the same as in \(\text{IR-calc}(\cdot, \cdot)\).

**Fig. 4.** The rules of \(\text{IR-calc}(P, M, \sigma_u)\)

and \(M\) is the quantifier-free matrix in CNF. In the following instantiation-based calculi, inference rules do not work on usual clauses but on \emph{annotated clauses} based on \emph{extended assignments}. An extended assignment is a partial mapping from the Boolean variables to \(\{0, 1\}\). An annotated clause consists of \emph{annotated literals} of the form \(\ell^{[r]}\), where \(\tau\) is an extended assignment to universal variables and \([\tau] = \{u \backslash c \mid (u \backslash c) \in \tau, lv(u) < lv(\ell)\}\) with \(e \in \{0, 1\}\). Composition of extended assignments is defined using \emph{completion}. The expression \(\mu \vDash \tau\) is called the completion of \(\mu\) by \(\tau\). Then \(\sigma\), the completion of \(\mu\) by \(\tau\), is defined as follows.

\[
\sigma(x) = \begin{cases} 
\mu(x) & \text{if } x \in \text{ dom}(\mu); \\
\tau(x) & \text{if } x \notin \text{ dom}(\mu) \text{ and } x \in \text{ dom}(\tau).
\end{cases}
\]  

(1)

The function \(\text{inst}(\tau, C)\) allows instantiations of clauses; it computes \(\{\ell^{[a \backslash r]} \mid \ell^u \in C\}\) for an extended assignment \(\tau\) and an annotated clause \(C\). Later on, we will clarify the relation between annotations and substitutions in first-order logic.

We extend \(\text{IR-calc}(\cdot, \cdot)\) by the possibility to instantiate universal variables by existential ones. Technically the instantiation is performed by a global substitution \(\sigma_u\). If a universal variable \(x\) is replaced by some existential variable \(e\), i.e., \((x \backslash e) \in \sigma_u\), then \(lv(e) < lv(x)\) must hold. We name the calculus equipped with the substitution \(\sigma_u\) \(\text{IR-calc}(P, M, \sigma_u)\) and depict the rules in Fig. 4.

It is immediately apparent that this calculus is sound and complete. We get completeness, when we use the empty substitution as \(\sigma_u\), because then, \(\text{IR-calc}(\cdot, \cdot, \cdot)\) reduces to \(\text{IR-calc}(\cdot, \cdot)\) which is sound and complete [4]. Soundness
1. $C$ is a non-tautological clause from the matrix $M$ or from $\Delta$.
2. $\sigma_v, C\sigma_v, \sigma, \text{Res}$ and $\text{Inst}$ are the same as in IR-calc($\cdot, \cdot$).
3. If $C \in \Delta$ then $\sigma = \epsilon$ and $C = C\sigma_v$ by construction.

Fig. 5. The rules of IR-calc($P, M, \Delta, \sigma_v$)

follows from the validity of QBFs of the form

$$Q_1 \exists e Q_2 \forall x Q_3 \varphi(e, x) \rightarrow Q_1 \exists e Q_2 Q_3 \varphi(e, e).$$

If the right formula has an IR-calc($\cdot, \cdot$) refutation, then it is false and therefore the left formula has to be false.

We further enhance IR-calc($\cdot, \cdot$) by the possibility to use propositional extensions \[19,6\]. This extension operation is a generalization of the well-known structure-preserving translation to (conjunctive) normal form in propositional logic. For presentational reasons, we require to have all extensions at the very beginning of the deduction in order to allow extension variables as replacements for universal variables. Figure 5 shows the inference rules of this calculus IR-calc($P, M, \Delta, \sigma_v$), where $\Delta$ is a sequence $\delta_1, \ldots, \delta_d$ of (clausal representations of) extensions of the form $\delta_i: q_i \leftrightarrow F$ with $F$ being of the form $\neg p$ or of the form $p \circ r$ ($\circ \in \{\land, \lor, \rightarrow, \leftrightarrow, \oplus\}$) and $q_i$ is a variable neither occurring in $M$ nor in $F$ nor in $\delta_1, \ldots, \delta_{i-1}$. The variables $q_i, p, r$ are existential. The quantification $\exists q_i$ extends the quantifier prefix $P$ such that $lv(v) \leq lv(q_i)$ for all variables $v$ occurring in $F$ and $lv(q_i)$ is minimal. Due to the requirements on the extension variables $q_i$ and the placement of $\exists q_i$, the resulting calculus is sound. Completeness is not an issue here, because we can use an empty $\Delta$.

Remark 1. The usual formalization of clauses and resolvents as sets of literals can be simulated in our formalizations by the factoring rule $\text{Fac}$. We assume in the following that $\text{Fac}$ is applied as soon as possible.

We finally introduce first-order resolution. Let $C$ be a clause and let $K$ and $L$ be two distinct literals in $C$ both of which are either negated or unnegated. If there is an mgu $\sigma$ of $K$ and $L$, then the clause $D = C\sigma = \{N\sigma \mid N \in C\}$ is called a factor of $C$. The clause $C$ is called the premise of the factoring operation.

Let $C$ and $D$ be two clauses and let $D'$ be a variant of $D$ which has no variable in common with $C$. A clause $E$ is a resolvent of the parent clauses $C$ and $D$ if the following conditions hold:

1. $K \in C$ and $L' \in D'$ are literals of opposite sign whose atoms are unifiable by an mgu $\sigma$.
2. $E = (C\sigma \setminus \{K\sigma\}) \cup (D'\sigma \setminus \{L'\sigma\})$.

Let $C$ be a set of clauses. A sequence $C_1, \ldots, C_n$ is called $R_1$ deduction (first-order resolution deduction) of a clause $C$ from $C$ if $C_n = C$ and for all $i = 1, \ldots, n$, one of the following conditions hold.
1. \( C_i \) is an input clause from \( C \).
2. \( C_i \) is a factor of a \( C_j \) for \( j < i \).
3. \( C_i \) is a resolvent of \( C_j \) and \( C_k \) for \( j, k < i \).

An \( R_1 \) refutation of \( C \) is an \( R_1 \) deduction of the empty clause \( \Box \) from \( C \). The size of a deduction is given by \( \sum_{i=1}^{n} \text{size}(C_i) \), where \( \text{size}(C_i) \) is the number of character occurrences in \( C_i \). An \( R_1 \) deduction has tree form if every occurrence of a clause is used at most once as a premise in a factoring operation or as a parent clause in a resolution operation.

Next we introduce the subsumption rule taken from Definition 2.3.4 in [8]. Contrary to the usual use of subsumption in automated deduction as a deletion rule, here we add clauses which are (factors of) instantiations of clauses.

Definition 3. If \( C \) and \( D \) are clauses, then \( C \) subsumes \( D \) or \( D \) is subsumed by \( C \), if there is a substitution \( \sigma \) such that \( C\sigma \subseteq D \). A set \( S' \) of clauses is obtained from a set \( S \) by subsumption if \( S' = S \cup \{D\} \) where \( D \) is subsumed by a clause of \( S \).

Resolution can be extended by the subsumption rule (Definition 3.2.3 in [8]).

Definition 4. By a derivation of a set of clauses \( S_2 \) from a set of clauses \( S_1 \) by \( R_1 \) plus subsumption, we mean a sequence \( C_1, \ldots, C_n \) of clause such that the following conditions are fulfilled.

1. \( S_2 \subseteq S_1 \cup \{C_1, \ldots, C_n\} \).
2. For all \( k = 1, \ldots, n \), there is a clause \( C \in S_1 \cup \{C_1, \ldots, C_{k-1}\} \) subsuming the clause \( C_k \) or there exist clauses \( C, D \in S_1 \cup \{C_1, \ldots, C_{k-1}\} \) such that \( C_k \) is subsumed by a resolvent of \( C \) and \( D \).

Factors are not needed in item 2 because the factor of \( C \) can be generated by subsumption. We need a simplified version of Proposition 3.2.1 from [8].

Proposition 4. \( R_1 \) polynomially simulates \( R_1 \) plus subsumption.

The subsumption rule is not necessary but makes proofs of polynomial simulation results much more convenient. It allows instantiated deductions for which eventually the lifting theorem provides a deduction “on the most general level”.

5 Polynomial simulations of calculi

In this section we show that \( R_1 \) together with a suitable translation \( T \) (denoted by \( R_1 + T \)) polynomially simulates \( QU\text{-res} \), \( Q\text{-res} \) and \( IR\text{-calc}(\cdot, \cdot) \).

Theorem 1. \( R_1 + Sk[H, h]_f \) polynomially simulates \( QU\text{-res} \).

The proof is by induction on the number of clauses in the \( QU\text{-res} \) deduction. It can be found in the appendix. It shows that first-order literals obtained from universal literals in the QBF and eliminated by \( \forall R \) are eliminated by resolutions with \( p(f_1) \) and \( \neg p(f_0) \) without instantiating the first-order resolvent.
Corollary 1. The following results are immediate consequences of Theorem 1.

1. $R_1 + EPR\|\|_p$ polynomially simulates QU-res.
2. $R_1 + Sk\|\|_p$ as well as $R_1 + EPR\|\|_p$ polynomially simulates Q-res.

We present a soundness proof of IR-calc($\cdot$, $\cdot$) independent from strategy extraction by a polynomial simulation of IR-calc($\cdot$, $\cdot$) by $R_1$.

Definition 5. Let $\tau = \{x_1|x_1, \ldots, x_k\}$ and $\mu = \{y_1|y_1, \ldots, y_l\}$ be two substitutions. The composition of $\tau$ and $\mu$, $\tau\mu$, is obtained from

$$\{x_1|x_1s_1, \ldots, x_k|s_1k, y_1|t_1, \ldots, y_l|t_l\}$$

by deleting all $y_i|t_i$ for which $y_i \in \{x_1, \ldots, x_k\}$ holds.

Lemma 1. Let $\tau$ and $\mu$ be two substitutions as defined in Definition 5 where $x_1, \ldots, x_k, y_1, \ldots, y_l$ are universal variables and $\{s_1, \ldots, s_k, t_1, \ldots, t_l\} \subseteq \{0, 1\}$. Then $\tau \vee \mu$ is the composition $\tau\mu$.

Proof. Let $\sigma$ be the completion of $\tau$ by $\mu$ defined in (1). Since $dom(\tau)$ as well as $dom(\mu)$ is a subset of the set of universal variables and $rg(\tau)$ as well as $rg(\mu)$ is a subset of $\{0, 1\}$, $rg(\tau) \cap dom(\mu) = \{\}$ and therefore $s_i\mu = s_i$ for all $i = 1, \ldots, k$. Hence, the completion $\sigma$ of the two substitutions $\tau$ and $\mu$ is exactly their composition $\tau\mu$. \qed

In the following, we deal with annotated clauses $C$ of the form $\{\llbracket l_1^{[\sigma_1]}, \ldots, l_k^{[\sigma_k]} \rrbracket \}$ where any $l_i$ is an existential literal and any $[\sigma_i]$ is the restriction of assignment $\sigma_i$ to exactly those universal variables $x \in dom(\sigma_i)$ for which $lv(x) < lv(l_i)$ holds. We denote the sequence of all universal variables $x$ with $lv(x) < lv(l_i)$ by $dep(l_i) = \overline{X}_l$ where we assume the same order as in the quantifier prefix. A first-order clause $D$ corresponding to $C$ is constructed as follows

$$\{(-)p(f_e(\overline{X}_e))\sigma \mid (-)e[\sigma] \in C \text{ and } p(f_e(\overline{X}_e)) \equiv e\},$$

where $p(f_e(\overline{X}_e))$ is the isomorphic counterpart of $e$ (cf. the remark after Proposition 2). Using $\overline{X}_e$ together with $\sigma$ mimics the effect of $[\sigma]$; the difference is the explicit notation of all universal variables $\overline{X}_e$ left of $e$ and not only the variables in $\overline{X}_e \cap dom(\sigma)$.

Theorem 2. $R_1 + Sk\|\|_p$ polynomially simulates IR-calc($\cdot$, $\cdot$).

In the proof, we construct by induction on the number of derived clauses in the IR-calc deduction stepwisely a deduction in $R_1$ plus subsumption. We consider the sequence of first-order clauses obtained from the original clauses as a skeleton for the final proof. Since the clauses in the skeleton do not follow by a single application of an inference rule, we have to provide a short deduction of the clauses.
Proof. We utilize Proposition 4 and allow subsumption in the simulation. The proof is by strong mathematical induction on the number of derived clauses in the IR-calc deduction. Let \( P(n) \) denote the statement “Given a IR-calc deduction \( C_1, \ldots, C_n \) from a QBF \( Q,M \) and a sequence of first-order clauses \( D_1, \ldots, D_n \), the clause \( D_n \) has a short deduction in \( R_1 \) plus subsumption from \( p(f_i), \neg p(f_0), Sk\|M\|_p, D_1, \ldots, D_{n-1} \)’’.

**Base:** \( n = 1 \). \( C_1 \) is a consequence of the axiom rule using clause \( C \) from the matrix \( M \). Let \( \sigma \) be the assignment induced by \( C \). Then we have a clause \( D \in Sk\|M\|_p \) from which we can derive \( D_1 \sigma \) by resolution steps using \( p(f_1) \) and \( \neg p(f_0) \). The number of these steps is equal to the number of universal variables in \( C \).

**IH:** Suppose \( P(1), \ldots, P(n) \) hold for some \( n \geq 1 \).

**Step:** We have to show \( P(n+1) \). Consider \( C_1, \ldots, C_{n+1} \) and \( D_1, \ldots, D_{n+1} \).

**Case 1:** \( C_{n+1} \) is derived by the axiom rule. Then proceed like in the base case.

**Case 2:** \( C_{n+1} \) is a consequence of the rule \( \text{Inst} \) with premise \( C_i \) (for some \( i \) with \( 1 \leq i \leq n \)) and assignment \( \tau \). By IH and Remark 4 we have a short \( R_1 \) plus subsumption deduction of \( D_i = \{ (-\neg)p(f_e(\overline{x})), \sigma \mid (\neg)e^{[\sigma]} \in C_i \} \). \( C_{n+1} \) is of the form \( \{ (\neg)e^{[\sigma, \tau]} \mid (\neg)e^{[\sigma]} \in C_i \} \). By Lemma 4, \( x(\sigma \cdot \tau) = x(\sigma) \cdot \tau \) for any universal variable \( x \) with \( lv(x) < lv(e) \). Therefore \( D_{n+1} \) is of the form \( \{ (\neg)p(f_e(\overline{x})), \sigma \cdot \tau \mid (\neg)e^{[\sigma, \tau]} \in C_{n+1} \} \). Now \( D_{n+1} = D_i \tau \) and \( D_{n+1} \) can be derived by subsumption.

**Case 3:** \( C_{n+1} \) is a consequence of the rule \( \text{Fac} \) with premise \( C_i : \widehat{C_i} \lor \ell^\tau \lor \ell^\tau \) (for some \( i \) with \( 1 \leq i \leq n \)). By IH, we have a short \( R_1 \) plus subsumption deduction of \( D_i : \widehat{C_i} \lor L \lor L \), where \( L \) is of the form \( (\neg)p(f_e(\overline{x})), \tau \). We generate a factor \( D_{n+1} \) of \( D_i \) simply by omitting one of the duplicates.

**Case 4:** \( C_{n+1} \) is a consequence of the resolution rule with parent clauses \( C_i, C_j \) (for some \( i, j \) with \( 1 \leq i, j \leq n \)). By IH, we have two clauses

\[ D_i = \{ p(f_e(\overline{x})), \sigma \} \cup D'_i \quad \text{and} \quad D_j = \{ \neg p(f_e(\overline{x})), \sigma \} \cup D'_j \]

We use \( \lambda \) of the form \( \{ x \cdot y \} \) as a renaming of the variables in \( D_j \) such that \( D_j \lambda \) does not share any variable with \( D_i \). The resolvent is \( D'_i \cup D'_j \lambda \mu \) where \( \mu \) is the mgu of the form \( \{ y \cdot x \mid x \notin \text{dom}(\sigma) \} \). We add \( D'_i \cup D'_j \lambda \mu \lambda' \) by subsumption, where \( \lambda' \) maps all remaining variables \( y \) to their \( x \) counterpart. \( \square \)

**Corollary 2.** \( R_1 + EPR[\cdot]_p \) polynomially simulates IR-calc(\( \cdot, \cdot \)).

When we inspect the translation of (axiom) clauses, we observe that a universal variable \( x \) is translated to an atom of the form \( p(x) \). With the subsumption rule we can instantiate the clause by a substitution of the form \( \{ x \mid t \} \) for a term \( t \). This observation was the trigger to introduce the stronger calculus IR-calc(\( \cdot, \cdot \)), where universal variables cannot be replaced only by 0 or 1 but also by any existential variable \( e \) with \( lv(e) < lv(x) \).
6 Exponential separation of resolution calculi

We constructed in [9] a family \( \Phi_n \) of short closed QBFs in PCNF for which any Q-res refutation of \( \Phi_n \) is superpolynomial. We recapitulate the construction here. The formula \( \Phi_n \) is

\[
\exists^b X_n \lor ^b Y_n \exists^b Z_n (\mathbf{TPHP}_{n,Z_n} \land \mathbf{CPHP}_{n,X_n}).
\]

(2)

\( \text{CPHP}_{n,X_n} \) is the pigeon hole formula for \( n \) holes and \( n + 1 \) pigeons in conjunctive normal form and denoted over the variables \( X_n = \{ x_{1,1}, \ldots, x_{n+1,n} \} \). Variable \( x_{i,j} \) is intended to denote that pigeon \( i \) is sitting in hole \( j \). \( \text{CPHP}_{n,X_n} \) is

\[
\left( \bigwedge_{i=1}^{n+1} \left( \bigvee_{j=1}^{n} x_{i,j} \right) \right) \land \left( \bigwedge_{j=1}^{n} \bigwedge_{1 \leq i_1 < i_2 \leq n+1} \left( \lnot x_{i_1,j} \lor \lnot x_{i_2,j} \right) \right).
\]

The number of clauses in \( \text{CPHP}_{n,X_n} \) is \( l_n = (n+1)^2(n+1)/2 \) and \( \text{size}(\text{CPHP}_{n,X_n}) \) is \( O(n^3) \). The formula \( \mathbf{TPHP}_{n,Z_n} \) is obtained from the pigeon hole formula in disjunctive normal form, \( \text{DPHP}_{n,Z_n} \), by a structure-preserving polarity-sensitive translation to clause form \( \Pi \). The formula \( \text{DPHP}_{n,Z_n} \) is simply the negation of \( \text{CPHP}_{n,X_n} \) where negation has been pushed in front of atoms and double-negation elimination has been applied.

We use new variables of the form \( z_{i_1,i_2,j} \) for disjuncts in \( \text{DPHP}_{n,Z_n} \). For the first \( n + 1 \) disjuncts of the form \( \bigvee_{j=1}^{n} \lnot y_{i,j} \) with \( 1 \leq i \leq n + 1 \), we use variables \( z_{1,0,0}, \ldots, z_{n,0,0} \). For the second part, for any \( 1 \leq j \leq n \) and the \( n(n+1)/2 \) disjuncts, we use

\[
z_{i_1,0,0}, \ldots, z_{i_1,n+1,0}, z_{i_2,0,0}, \ldots, z_{i_2,n+1,0}, \ldots, z_{n,n+1,0}.
\]

(3)

The set of these variables for \( \text{DPHP}_{n,Z_n} \) is denoted by \( Z_n \). Due to this construction, we can speak about the conjunction corresponding to the variable \( z_{i_1,i_2,j} \).

We construct the conjunctive normal form \( \mathbf{TPHP}_{n,Y_n,Z_n} \) of \( \text{DPHP}_{n,Y_n,Z_n} \) as follows. First, we take the clause \( D_{n,Z_n}^n = \bigvee_{z \in Z_n} \lnot z \) over all variables in \( Z_n \). The formula \( D_{n,Y_n,Z_n}^n \) for the first \( (n + 1) \) disjuncts of \( \text{DPHP}_{n,Y_n,Z_n} \) is of the form

\[
\bigwedge_{i=1}^{n+1} \bigwedge_{j=1}^{n} (z_{i,0,0} \lor \lnot y_{i,j}).
\]

For the remaining \( n^2(n+1)/2 \) disjuncts of \( \text{DPHP}_{n,Y_n,Z_n} \), we have the formula \( Q_{n,Y_n,Z_n} \)

\[
\bigwedge_{j=1}^{n} \bigwedge_{1 \leq i_1 < i_2 \leq n+1} \left( (z_{i_1,i_2,j} \lor \lnot y_{i_1,j}) \land (z_{i_1,i_2,j} \lor \lnot y_{i_2,j}) \right).
\]

Then \( \mathbf{TPHP}_{n,Y_n,Z_n} \) is \( D_{n,Z_n}^n \land P_{n,Y_n,Z_n} \land Q_{n,Y_n,Z_n} \) and \( \text{size}(\mathbf{TPHP}_{n,Y_n,Z_n}) = O(n^3) \). It is easy to check that \( \text{DPHP}_{n,Y_n,Z_n} \leftrightarrow \exists^b Z_n \mathbf{TPHP}_{n,Y_n,Z_n} \) is valid.
Let us modify the quantifier prefix of $\Phi_n$. By quantifier shifting rules we get, in an “antiprenexing” step, the equivalent formula $(\forall^b Y_n \exists^b Z_n \text{TPHP}^{Y_n, Z_n}_n) \land (\exists^b X_n \text{CPHP}^{X_n}_n)$. Prenexing yields the equivalent QBF $\Omega_n$

$$\forall^b Y_n \exists^b Z_n \exists^b X_n (\text{TPHP}^{Y_n, Z_n}_n \land \text{CPHP}^{X_n}_n)$$

which has only one quantifier alternation instead of two. In [9] we showed that $\Phi_n$ and $\Omega_n$ have short cut-free tree proofs in a sequent system $G_q^*$, where weak quantifiers introduce atoms. The following extends Proposition 3 in [9].

**Proposition 5.** Any Q-res refutation of $\Phi_n$ from (2) and $\Omega_n$ from (4) has superpolynomial size.

The proof is based on the fact that (i) the two conjuncts belong to languages with different alphabets and (ii) that the alphabets cannot be made identical by instantiation of quantifiers in Q-res. Therefore we have to refute either $\text{TPHP}^{Y_n, Z_n}_n$ or $\text{CPHP}^{X_n}_n$ under the given quantifier prefix. Since $\forall^b Y_n \exists^b Z_n \text{TPHP}^{Y_n, Z_n}_n$ is true, we essentially have to refute $\text{CPHP}^{X_n}_n$ with propositional resolution and consequently, by Haken’s famous result [12], any Q-res refutation of $\text{CPHP}^{X_n}_n$ is superpolynomial in $n$.

Since QU-res, LDQ-res, LDQU-res, LDQU+-res, and Q(D)-resolution (Q(D)-res) are based on the same quantifier-handling mechanism as Q-res, the following corollary is obvious.

**Corollary 3.** Any refutation of $\Phi_n$ from (2) and $\Omega_n$ from (4) in the QU-res, LDQ-res, LDQU-res, LDQU+-res, or Q(D)-res calculus has superpolynomial size.

For IR-calc($\cdot$, $\cdot$) the situation is not better. Since universal literals are only replaced by 0, no unification of the two alphabets can happen.

**Proposition 6.** Any refutation of $\Phi_n$ from (2) and $\Omega_n$ from (4) in IR-calc($\cdot$, $\cdot$) has size superpolynomial in $n$.

The quantifier prefix is unfortunate if one expects $\Omega_n$ being false. Actually, the initial universal quantifier block prevents any non-empty $\sigma_n$ and consequently, any IR-calc($\cdot$, $\cdot$, $\cdot$) refutation of $\Omega_n$ reduces to an IR-calc($\cdot$, $\cdot$) refutation of $\Omega_n$.

**Proposition 7.** Any refutation of $\Omega_n$ from (4) in IR-calc($\cdot$, $\cdot$, $\cdot$) has size superpolynomial in $n$.

In the following we show that $\text{Sk}[\Omega_n]$ has a short refutation in $R_1$. We use $f_{x_{i,j}}$ to denote the Skolem function symbol corresponding to $x_{i,j} \in X_n$ and $f_{z_{i,j,k}}$ to denote the Skolem function symbols corresponding to $z_{i,j,k} \in Z_n$. All the Skolem function symbols have arity $|Y_n| = n(n+1)$. Let $\mathcal{F}$ denote the formula $F$ under the first-order translation. We have

$$\text{CPHP}^{X_n}_n : \left( \bigwedge_{i=1}^{n+1} \left( \bigvee_{j=1}^{n} p(f_{x_{i,j}}(Y_n)) \right) \right) \land$$
refutation of $\Omega_n$ is constructed as follows.

1. We use $P_{n+1}^{x_0}$ together with the first $n + 1$ clauses from $\tilde{\text{CPHPP}}_n^{x_0}$ to derive $p(f_{z_{i,0},0}(Y_n))\mu_i$ (for all $i = 1, \ldots, n + 1$). The deduction consists of $O(n^3)$ clauses and applies resolution and factoring. The substitution $\mu_i$ is $\bigcup_{j=1}^{n+1} \{ y_{i,j} \mid f_{x_{i,j}}(Y_n)\sigma_{i,j} \}$, where $\sigma_{i,j}$ is a variable renaming from the variant generation in resolution.

2. We use $Q_{n+1}^{x_0}$ together with the binary clauses from $\tilde{\text{CPHPP}}_n^{x_0}$ to derive $p(f_{z_{i,1},2}(Y_n))\nu_{i_1,i_2,j}$ (for all $j = 1, \ldots, n$ and $i_1, i_2$ with $1 \leq i_1 < i_2 \leq n + 1$). The deduction consists of $O(n^3)$ clauses and applies resolution and factoring. Then $\nu_{i_1,i_2,j}$ is $\{ y_{i_1,j} \mid f_{x_{i,j}}(Y_n)\sigma_{i_1,i_2,j}, y_{i_2,j} \mid f_{x_{i,j}}(Y_n)\sigma_{i_1,i_2,j} \}$. Again $\sigma_{i_1,i_2,j}$ is a variable renaming like above.

3. We use $D_{n+1}^{x_0}$ together with the derived instance of $p(f_{x_{k,1},0}(Y_n))$ to derive \( \square \) by resolution. Since any variable $y_{i,j}$ is assigned to a variant of $f_{x_{1,j}}(Y_n)$ for all $i = 1, \ldots, n + 1$ and all $j = 1, \ldots, n$, all resolution steps are possible. The deduction consists of $O(n^3)$ clauses.

The formula $\tilde{\text{SK}}[\Phi_n]_p^{x_0}$ can be refuted in a similar fashion in $R_1$ by replacing variants of the form $f_{x_{1,j}}(Y_n)$ by Skolem constants $a_{i,j}$.

**Proposition 8.** Let $(\Phi_n)_{n \geq 1}$ and $(\Omega_n)_{n \geq 1}$ be the families of closed QBFs defined above. Then $\| \Phi_n \|_p$ and $\| \Omega_n \|_p$ have short tree refutations in $R_1$ consisting of $O(n^3)$ clauses. Moreover the size of the refutation is $O(n^8)$.

**Theorem 3.** The calculi $\text{QU-res}$, $\text{LDQ-res}$, $\text{LDQU-res}$, $\text{LDQU}^+\text{-res}$, $\text{Q(D)-res}$, $\text{IR-calc(\cdot, \cdot)}$, $\text{IR-calc(\cdot, \cdot, \cdot)}$ and $\text{IRM-calc}$ cannot polynomially simulate tree $R_1 + \tilde{\text{SK}}[\cdot]_p$ or $R_1 + \text{EPR}[\cdot]_p$.

We use $(\Omega_n)_{n \geq 1}$ to exponentially separate $R_1$ combined with the two translations, i.e., we compare $\tilde{\text{SK}}[\cdot]_p$ with $\text{EPR}[\cdot]_p$.

**Proposition 9.** Let $(\Omega_n)_{n \geq 1}$ be the family of closed QBFs defined above and let $\Omega'_n$ be the EPR formula $\text{EPR}[\Omega_n]_p \land p(f_1) \land \neg p(f_0)$. Then $\Omega'_n$ has only refutation in $R_1$ of size superpolynomial in $n$. 
Proof (Sketch). Similar arguments as in Proposition \ref{prop:QBFs} apply, because the EPR translations of TP\textsubscript{PH}^1_n,\ldots,\text{CPH}^1_n and CPH\textsubscript{1,n} are denoted in different languages and literals from the former cannot be resolved with literals from the latter. Again, the pigeonhole formula has to be refuted. Consequently, the (essentially propositional) resolution proof has size superpolynomial in \( n \). \qed

**Theorem 4.** \( R_1 + EPR[\cdot; \cdot] \) cannot polynomially simulate tree \( R_1 + Sk[\cdot; \cdot] \).

Let us reconsider the family \((\Psi)_t \geq 1\) of QBFs from \cite{GKR16}. Formula \( \Psi_t \) has the prefix \( P_t : \exists d_0 \delta t e_1 \forall x_1 \exists d_2 e_2 \forall x_2 \exists d_3 e_3 \ldots \forall x_{t-1} \exists d_t e_t \forall x_t \exists f_1 \ldots f_t \) and the matrix \( M_t \) consisting of the following clauses:

\[
\begin{align*}
C_0 & : = d_0 \\
C_{2j} & : = d_j \lor \bar{e}_j \lor d_{j+1} \lor \bar{e}_{j+1} \\
C_{2j+1} & : = e_j \lor x_j \lor d_{j+1} \lor \bar{e}_{j+1} \quad j = 1, \ldots, t-1 \\
C_{2t} & : = d_t \lor x_t \lor f_1 \lor \cdots \lor f_t \\
B_{2j} & : = \bar{e}_{j+1} \\
B_{2j+1} & : = x_{j+1} \lor f_{j+1} \quad j = 0, \ldots, t-1
\end{align*}
\]

By Theorem 3.2 in \cite{GKR16} and Theorem 6 in \cite{Kol96}, any Q\textsubscript{res} refutation and any IR-calc(\( \cdot, \cdot \)) refutation of \( \Psi_t \) is exponential in \( t \). The formula \( \Psi_t \) has a polynomial size Q-resolution refutation if universal pivot variables are allowed \cite{GKR16}.

Let us extract Herbrand functions from such a short QU-res refutation of \( \Psi_t \) with the method of \cite{Kol96} resulting in \( \bar{d}_t \land e_i \) for \( x_i \). We explain in the following how we can produce short IR-calc\((P_t, M_t, \Delta, \sigma_t)\) refutations using such functions.

Let \( \Delta : \delta_1, \ldots, \delta_t \) where \( \delta_i \) is \( q_i \lor d_i \lor \bar{e}_i, \bar{d}_i, \bar{e}_i, \bar{d}_i \lor \bar{e}_i, \ldots \) and \( \delta_i \) is the clausal representation of \( q_i \leftrightarrow \bar{d}_i \land e_i \). The quantifier \( \exists x_i \) is in the same quantifier block as \( d_i \) and \( e_i \) and thus \( lv(q_i) < lv(x_i) \). Consequently, \( \sigma_t \) can replace \( x_i \) by \( q_i \).

**Proposition 10.** Let \( \Delta : \delta_1, \ldots, \delta_t \) where \( \delta_i \) is \( q_i \lor d_i \lor \bar{e}_i, \bar{d}_i, \bar{e}_i, \bar{d}_i \lor \bar{e}_i, \ldots \) and \( \delta_i \) is the clausal representation of \( q_i \leftrightarrow \bar{d}_i \land e_i \). Let \( \sigma_{v,t} = \{ x_i \mid q_i \mid 1 \leq i \leq t \} \). There is a tree refutation of \( \Psi_t \) in IR-calc\((P_t, M_t, \Delta, \sigma_{v,t})\) of size polynomial in \( t \).

**Proof (sketch).** Derive \( \bar{d}_1 \land \bar{e}_1, \ldots, \bar{d}_t \land \bar{e}_t \). The first clause is derived by a resolution step between \( C_0 \) and \( C_1 \). Then we derive \( \bar{d}_{j+1} \land \bar{e}_{j+1} \) from \( \bar{d}_j \land \bar{e}_j, C_{2j+1} \sigma_{v,t}, C_{2j+1} \), and the clauses obtained from \( q_j \leftrightarrow \bar{d}_j \land e_j \) as follows. Resolve \( \bar{d}_j \lor \bar{d}_{j+1} \lor \bar{e}_{j+1} \lor \bar{e}_{j+1} \) with \( \bar{d}_j \lor \bar{e}_j \lor \bar{d}_{j+1} \lor \bar{e}_{j+1} \) and derive \( \bar{d}_j \lor \bar{e}_j \lor \bar{d}_{j+1} \lor \bar{e}_{j+1} \) by resolution and factoring. Then continue with \( \bar{d}_j \lor \bar{e}_j \lor \bar{d}_{j+1} \lor \bar{e}_{j+1} \) by resolution and factoring. Use \( \bar{e}_j \lor \bar{d}_j \lor \bar{d}_{j+1} \lor \bar{e}_{j+1} \), resolve it with \( \bar{d}_j \lor \bar{e}_j \) and factor the resolvent resulting in \( \bar{d}_{j+1} \lor \bar{e}_{j+1} \lor \bar{e}_{j+1} \). Resolve \( R \) with the latter clause, factor the resolvent and obtain \( \bar{d}_{j+1} \lor \bar{e}_{j+1} \lor \bar{e}_{j+1} \).

Each of the 15 clauses has at most 5 literals. For \( j + 1 = t \), we have a similar deduction but with at most 2t + 3 literals per clause. We obtain \( \bar{f}_1 \lor \cdots \lor \bar{f}_t \) which can be resolved by \( f_t \) obtained from \( \bar{f}_1 \lor \bar{f}_t \) and \( q_t \lor f_t \). Finally, it is easy to check that the refutation has tree structure and is of size polynomial in \( t \). \( \Box \)

The Herbrand functions obtained from Q\textsubscript{res} or QU-res refutations by the method in \cite{Kol96} are often (too) complex. It is easy to check that atomic Herbrand functions \( e_i \) for \( x_i \) are sufficient and therefore a short tree IR-calc(\( \cdot, \cdot \)) refutation of \( \Psi_t \) is possible. The proof of the following proposition can be found in the appendix.
Proposition 11. Let \( \sigma_{v,t} = \{x_i|e_i| 1 \leq i \leq t\} \). Then there is a tree refutation of \( \Psi_t \) in IR-calc\((P_t, M_t, \sigma_{v,t})\) of size polynomial in \( t \).

Proposition 12. IR-calc\((\cdot, \cdot, \cdot)\) cannot polynomially simulate IR-calc\((\cdot, \cdot, \cdot)\).

According to Proposition 11 there are not only short tree refutations of \( \Psi_t \), but also the search space is limited if a simple heuristic restricting the number of possible variable replacements \( \sigma_{v,t} \) is employed during proof search. The heuristic requires that for each \((x|\epsilon) \in \sigma_{v,t}\), there is at least one clause \( C_{\sigma_{v,t}} \) which contain duplicate literals.

7 Conclusion

We studied various calculi for QBFs with respect to their relative strength. We provided polynomial simulations using first-order translations in order to clarify the possibility to employ (non-trivial) instantiations in refutations. By a simulation of Q-res and QU-res by \( R_1 \), we have seen that the former ones avoid instantiations. The simulation of simple instantiation-based calculi by \( R_1 \) revealed that instantiation of universal variables is possible by resolutions with \( p(f_1) \) and \( \neg p(f_0) \) together with the usual propagation of substitutions, and clarified the purpose of the employed framework of assignments and annotated clauses. We showed that enabling instantiations with existential variables and formulas increase the strength of instantiation-based calculi. For presentational reasons, we have chosen a rather simple approach where \( \sigma_v \) and \( \Delta \) are initially given, but it is possible in the underlying framework to generate \( \sigma_v \) and \( \Delta \) dynamically.

Open problems and future research directions: In all our comparisons, we did not optimize the quantifier prefix by (advanced) dependency schemes. It is well known that less dependencies between variables can considerably shorten proofs, for which reason one would like to integrate these techniques into calculi. We have left open some proof-theoretical comparisons like sequent systems for prenex formulas with propositional cuts and IR-calculi\((\cdot, \cdot, \cdot)\) or IRM-calc\([4]\) with our new calculi or \( R_1 \). The problem here is that \( R_1 \) is probably not strong enough because inference rules for Skolem function manipulation\([8,1]\) are not available but seem to be necessary for a polynomial simulation. The ultimate goal is to make instantiation-based calculi ready for proof search. A first step has been accomplished by showing (in the simulation) that unrestricted instantiations in IR-calculi\((\cdot, \cdot)\) can be restricted to minimal ones by simply using unification and mgus like in the first-order case. Achieving the goal for strong calculi is not an easy exercise because some techniques like extensions are hard to control.

References

1. M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In J. A. Robinson and A. Voronkov, editors, Handbook of Automated Reasoning, pages 273–333. Elsevier and MIT Press, 2001.
2. V. Balabanov and J.-H. R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012.

3. V. Balabanov, M. Widl, and J.-H. R. Jiang. QBF resolution systems and their proof complexities. In *SAT*, 2014.

4. O. Beyersdorff, L. Chew, and M. Janota. On unification of QBF resolution-based calculi. In E. Csurj, M. Dietzfelbinger, and Z. Esik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 81–93. Springer, 2014.

5. O. Beyersdorff, L. Chew, and M. Janota. Proof complexity of resolution-based QBF calculi. In E. W. Mayr and N. Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 76–89. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

6. O. Beyersdorff, L. Chew, and M. Janota. Extension variables in QBF resolution. In *AAAI-16 workshop Beyond NP*, 2016.

7. S. A. Cook and T. Morioka. Quantified propositional calculus and a second-order theory for NC¹. *Arch. Math. Log.*, 44(6):711–749, 2005.

8. E. Eder. *Relative complexities of first order calculi*. Artificial intelligence = Künstliche Intelligenz. Vieweg, 1992.

9. U. Egly. On sequent systems and resolution for QBFs. In A. Cimatti and R. Sebastiani, editors, *SAT*, volume 7317 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2012.

10. U. Egly, M. Seidl, and S. Woltran. A solver for QBFs in negation normal form. *Constraints*, 14(1):38–79, 2009.

11. A. Van Gelder. Contributions to the theory of practical quantified boolean formula solving. In M. Milano, editor, *CP*, volume 7514 of *Lecture Notes in Computer Science*, pages 647–663. Springer, 2012.

12. A. Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

13. H. Kleine Buning, M. Karpinski, and A. Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

14. J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and its Application*. Cambridge University Press, 1995.

15. A. Leitsch. *The resolution calculus*. Texts in theoretical computer science. Springer, 1997.

16. D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *J. Symb. Comput.*, 2(3):293–304, 1986.

17. M. Seidl, F. Lonsing, and A. Biere. qb2epr: A tool for generating EPR formulas from QBF. In P. Fontaine, R. A. Schmidt, and S. Schulz, editors, *PAAR@IJCAR*, volume 21 of *EPiC Series*, pages 139–148. EasyChair, 2012.

18. F. Slivovsky and S. Szeider. Variable dependencies and Q-resolution. In C. Sinz and U. Egly, editors, *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8561 of *Lecture Notes in Computer Science*, pages 269–284. Springer, 2014.

19. G. S. Tseitin. On the Complexity of Derivation in Propositional Calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part II*, pages 234–259. Seminars in Mathematics, V.A. Steklov Mathematical Institute, vol. 8, Leningrad, 1968.
20. L. Zhang and S. Malik. Conflict driven learning in a quantified boolean satisfiability solver. In L. T. Pileggi and A. Kuehlmann, editors, Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design, ICCAD 2002, San Jose, California, USA, November 10-14, 2002, pages 442-449. ACM / IEEE Computer Society, 2002.
A Proof of some propositions and theorems

Proposition 1 Let \( \Phi \) be a (closed) QBF and let \( \llbracket \Phi \rrbracket_p \) be its (closed) first-order translation. Then \( \Phi \cong \llbracket \Phi \rrbracket_p \), i.e., \( \Phi \) and \( \llbracket \Phi \rrbracket_p \) are isomorphic.

Proof. The proof is by induction on the logical complexity, \( lc(\Phi) \), of \( \Phi \).

Base: \( lc(\Phi) = 0 \). Then \( \Phi \) is \( \perp \), \( \top \) or a Boolean variable \( q \) and \( \llbracket \Phi \rrbracket_p = p(f_0), p(f_1) \) or \( p(x_q) \). Then \( \Phi \cong \llbracket \Phi \rrbracket_p \).

IH: For all QBFs \( \Psi \) with \( lc(\Psi) < k \), \( \Psi \cong \llbracket \Psi \rrbracket_p \), i.e., \( \Psi \) and \( \llbracket \Psi \rrbracket_p \) are isomorphic.

Step: Consider QBF \( \Phi \) with \( lc(\Phi) = k \). In all cases below, \( \Phi_1 \cong \llbracket \Phi_1 \rrbracket_p \) holds (\( i = 1, 2 \)) by the induction hypothesis.

Case 1: \( \Phi = \neg \Phi_1 \). Since \( \Phi_1 \cong \llbracket \Phi_1 \rrbracket_p \), \( \neg \Phi_1 \cong \neg \llbracket \Phi_1 \rrbracket_p = \neg \llbracket \Phi \rrbracket_p \) and therefore \( \Phi \cong \llbracket \Phi \rrbracket_p \) holds.

Case 2: \( \Phi = \Phi_1 \circ \Phi_2 \). Since \( \Phi_1 \cong \llbracket \Phi_1 \rrbracket_p \) as well as \( \Phi_2 \cong \llbracket \Phi_2 \rrbracket_p \), \( \Phi_1 \circ \Phi_2 \cong \llbracket \Phi_1 \rrbracket_p \circ \llbracket \Phi_2 \rrbracket_p = \llbracket \Phi_1 \circ \Phi_2 \rrbracket_p \) and therefore \( \Phi \cong \llbracket \Phi \rrbracket_p \) holds.

Case 3: \( \Phi = Q^b q \Phi_1 \). Since \( \Phi_1 \cong \llbracket \Phi_1 \rrbracket_p \), \( Q^b q \Phi_1 \cong Qq \llbracket \Phi_1 \rrbracket_p = \llbracket Q^b q \Phi_1 \rrbracket_p \) and therefore \( \Phi \cong \llbracket \Phi \rrbracket_p \) holds.

Proposition 3 Let \( \Phi \) be a closed QBF. Then

\( \Phi \) is satisfiable \iff \( \llbracket \Phi \rrbracket_p \land p(f_1) \land \neg p(f_0) \) is satisfiable.

Proof (sketch). \( \rightarrow \): \( \Phi \) is satisfiable. We show that \( \llbracket \Phi \rrbracket_p \land p(f_1) \land \neg p(f_0) \) has a model with a two-element domain \( \mathcal{U} = \{f_1, f_0\} \) and constants are mapped to itself by the interpretation function. Moreover, \( p(f_1) \) has to be true and \( p(f_0) \) has to be false. If we evaluate \( \Phi \) according to the semantics, we can, in a parallel way, expand \( \llbracket \Phi \rrbracket_p \) over \( \mathcal{U} \) and obtain two isomorphic expanded formulas. Evaluating isomorphic leaves in the same way and propagating the truth values from the leaves to the root (in the corresponding formula trees) yields the same evaluation result for both formulas. Hence, \( \llbracket \Phi \rrbracket_p \land p(f_1) \land \neg p(f_0) \) is satisfiable.

\( \leftarrow \): \( \Phi \) is unsatisfiable. Then there is a logically equivalent PCNF \( \Phi' \) and a Q-res refutation of \( \Phi' \) (because Q-res is complete). Due to Proposition 1 and the preservation of the quantifiers and connectives by \( \llbracket \cdot \rrbracket_p \), there is an isomorphic PCNF \( \Phi'_1 \) of \( \llbracket \Phi \rrbracket_p \) where \( \Phi'_1 \) is logically equivalent to \( \llbracket \Phi \rrbracket_p \). Skolemization yields the sat-equivalent first-order clause form \( \Phi''_1 \) of \( \Phi'_1 \). In Corollary 1 we show that we can simulate each Q-res refutation of \( \Phi' \) by a first-order resolution refutation of \( \Phi''_1 \land p(f_1) \land \neg p(f_0) \). By soundness of first-order resolution, we conclude that \( \Phi''_1 \land p(f_1) \land \neg p(f_0) \) and therefore \( \llbracket \Phi \rrbracket_p \land p(f_1) \land \neg p(f_0) \) is unsatisfiable.

Theorem 1. \( R_1 + Sk \llbracket \cdot \rrbracket_p \) polynomially simulates QU-res.
Proof. Let $\Phi$: $Q^k M$ be a QBF in PCNF with quantifier prefix $Q^k$ and matrix $M$. Consider the first-order translation $[\Phi]^f_p$ of $\Phi$ and $Sk[\Phi]^f_p$ (the skolemized form of $[\Phi]^f_p$). By Proposition 2, every literal in $M$ has an isomorphic counterpart in $Sk[\Phi]^f_p$. We employ this isomorphism in the following.

Let $C_1, \ldots, C_n$ be a QU-res deduction of $C_n$. For any clause $C_i$ ($1 \leq i \leq n$) of the form $L_i, \ldots \lor L_i, m_i$ generate a first-order clause $D_i$ of the form $K_{i,1} \lor \cdots \lor K_{i,m_i}$ where $K_{i,j} \equiv L_i, j$ for $j = 1, \ldots, m_i$. We show by induction on $n$ that there exists an $R_1$ deduction $p(f_1), \neg p(f_0), E_1, \ldots, E_n$ of $E_n$ from $Sk[\Phi]^f_p \land p(f_1) \land \neg p(f_0)$ such that the following holds for all $i = 1, \ldots, n$.

1. $E_i$ is non-tautological.
2. $D_i = E_i \sigma$ for some variable substitution $\sigma$.

Condition 2 implies that all $E_i$ are not instantiated with non-variable terms.

Base: $n = 1$. Then $C_1$ is an input clause from $M$, $C_1$ in non-tautological by assumption (of QU-res), and $D_1$ is a first-order input clause with $C_1 \equiv D_1$. Take $E_1 = D_1$ and $D_1 = E_1 \sigma$ where $\sigma = \epsilon$.

IH: Suppose $n \geq 1$ and for all $k \leq n$, we have based on $C_1, \ldots, C_k$ and $D_1, \ldots, D_k$ an $R_1$ deduction $p(f_1), \neg p(f_0), E_1, \ldots, E_k$ of $E_k$ from $Sk[\Phi]^f_p \land p(f_1) \land \neg p(f_0)$ such that conditions 1 and 2 hold.

Step: Consider $C_1, \ldots, C_{n+1}$ and $D_1, \ldots, D_{n+1}$.

Case 1: $C_{n+1}$ is an input clause. Then proceed as in the base case.

Case 2: $C_{n+1}$ is the consequence of a $\forall$ reduction applied to $C_i$ ($i \leq n$). Let $\ell$ be the universal literal removed. Without loss of generality, let $\ell$ be positive and of the form $x$. Then there is a clause $D_i$: $\bar{D_i} \lor p(x)$. Observe that the variable $x$ does not occur in $\bar{D_i}$, because we assume by Remark 1 applications of Fac as early as possible. By IH, we have a non-tautological clause $E_i$: $E_i \lor p(y)$ and a variable substitution $\sigma$ with $D_i = E_i \sigma$. $E_{n+1}$ is obtained from $E_i$ and $\neg p(f_0)$ by resolution resulting in $\bar{E_i}$. Then $D_{n+1} = E_{n+1} \sigma$ and $E_{n+1}$ is non-tautological because $E_i$ is non-tautological.

Case 3: $C_{n+1}$ is a factor of $C_i$ ($i \leq n$). Then there is a clause $D_i$: $\bar{D_i} \lor \ell(t) \lor \ell(t)$ where $\ell(t)$ is a literal with predicate symbol $p$ with a term $t$ as argument. By IH, we have a non-tautological clause $E_i$ and a variable substitution $\sigma$ with $D_i = E_i \sigma$. If $t$ is a constant, then $E_{n+1}$ is $E_i$ with one occurrence of $\ell(t)$ removed, $E_i$ is non-tautological and so is $E_{n+1}$ and $D_{n+1} = E_{k+1} \sigma$.

Let the term $t$ be of the form $f(X)$. Then $E_i$ is $\bar{E_i} \lor \ell(f(Y)) \lor \ell(f(Z))$ and $\sigma(u_r) = x_r$ for all $u_r \in Y \cup Z$ is the unifier of $\{ \ell(f(Y)), \ell(f(Z)) \}$ of the form $y_i \backslash z_i$ for all $y_i \in Y$. The factor $E_{n+1}$ is then $(\bar{E_i} \lor \ell(f(Z)))$ and $D_{n+1} = E_{k+1} \sigma$ holds.

We argue in the following that $E_{n+1}$ is non-tautological. Suppose $E_{n+1}$ is tautological. Then, since $D_{n+1} = E_{n+1} \sigma$, $D_{n+1}$ is tautological which in turn implies that $C_{n+1}$ is tautological. But this is impossible by the definition of Q-res and QU-res.
Let $t$ be a variable $x$. Then this case is similar to the case $t = f(X)$.

**Case 4:** $C_{n+1}$ is a Q-resolvent of $C_i$ and $C_j$ ($i, j \leq n$) upon the existential variable $e$. Then there are two clause $D_i : D_i \lor p(t_e)$ and $D_j : D_j \lor \neg p(t_e)$. By IH, we have non-tautological clauses $E_i$ with $D_i = E_i \sigma_1$ and $E_j$ with $D_j = E_j \sigma_2$ where $\sigma_1$ as well as $\sigma_2$ are variable substitutions.

**Subcase 4.1:** $t_e$ is a functional term $f_e(X)$. Then $E_i : \overline{E_i} \lor p(f_e(Y))$ and $\sigma_1(y_r) = x_r$ for all $y_r \in Y$;

$E_j : \overline{E_j} \lor \neg p(f_e(Z))$ and $\sigma_2(z_r) = x_r$ for all $z_r \in Z$.

Let $\mu$ be a renaming substitution such that $E_i \mu$ and $E_j$ are variable-disjoint. In order to construct the resolvent, we need the mgu $\pi$ of $\{p(f_e(Y))\mu, p(f_e(Z))\}$, which is $\{\mu(y_r) \lor z_r \mid \text{for all } y_r \in Y\}$. The unifier $\pi$ is a matcher; it affects only variables from $E_i \mu$. The resolvent $E_{n+1}$ is then $\overline{E_i \mu \pi} \lor E_j$.

We show that there exists a variable substitution $\sigma$ such that $D_{n+1} = E_{n+1} \sigma$. First observe that $D_i = E_i \mu \sigma'_1$ with $\sigma'_1 = \{\mu(u) \mid \sigma_1(u) \mid \text{for all } u \in \text{var}(E_i)\} \setminus \{u' \mid \text{u is a variable}\}$. Then with $\sigma_1'(u) = \pi(\mu(u)) \mid \sigma_1(u) \mid \text{for all } u \in \text{var}(E_i)\} \setminus \{u' \mid \text{u is a variable}\}$, we have $D_i = E_i \mu \pi \sigma_1'$. For all $y_i \in Y$, we have $\pi(\mu(y_i)) = z_i$, $\sigma_1(y_i) = x_i$ and $\sigma_2(z_i) = x_i$. Then

$$\overline{D_i} \lor \overline{D_j} = E_i \mu \pi \sigma_1'' \lor E_j \sigma_2 = (\overline{E_i \mu \pi} \lor \overline{E_j}) \sigma$$

where $\sigma$ is obtained from

$$\{\mu(u) \mid \sigma_1(u) \mid \text{for all } u \in \text{var}(E_i) \setminus Y\} \cup \{v \mid \sigma_2(v) \mid \text{for all } v \in \text{var}(E_j)\}$$

by deleting all elements of the form $u' \setminus u$. Observe that $rg(\pi) = \{Z\} \subseteq \text{var}(E_j)$ and $rg(\pi) \subseteq \text{dom}(\sigma_2)$. Therefore $D_{n+1} = E_{n+1} \sigma$.

**Subcase 4.2:** $t_e$ is a constant. Similar to **Subcase 4.1** but with an empty mgu $\pi$.

The clause $E_{n+1}$ from both subcases is non-tautological by the same reason as in **Case 3**.

**Case 5:** $C_{n+1}$ is a Q-resolvent of $C_i$ and $C_j$ ($i, j \leq k$) upon the universal variable $u$. Similar to **Subcase 4.1**.

**Proposition 11** Let $\sigma_{e,t} = \{x_i \mid e_i \mid 1 \leq i \leq t\}$. Then there is a tree refutation of $\Psi_1$ in IR-cal($P_t, M_s, \sigma_{e,t}$) of size polynomial in $t$.

**Proof (sketch).** Take $\sigma_{e,t} = \{x_i \mid e_i \mid 1 \leq i \leq t\}$ and derive $\overline{d_1} \lor \overline{d_t} \lor \overline{t}$.

The first clause is derived by a resolution step between $C_0$ and $C_1$. Then we derive $\overline{d_j+1} \lor \overline{\tau_j+1}$ from $\overline{d_j} \lor \overline{\tau_j}$ and $C_{2j} \sigma_{e,t}$ and $C_{2j+1} \sigma_{e,t}$ as follows. Resolve $\overline{d_j} \lor \overline{\tau_j}$ and $d_j \lor \tau_j \lor \overline{d_{j+1}} \lor \overline{\tau_{j+1}}$, obtain $\tau_j \lor \tau_j \lor \overline{d_{j+1}} \lor \overline{\tau_{j+1}}$ and factor it to get $R: \tau_j \lor \overline{d_{j+1}} \lor \overline{\tau_{j+1}}$. Next factor $e_j \lor e_j \lor d_j \lor \tau_j \lor \overline{e_j}$ and get $e_j \lor \overline{d_{j+1}} \lor \overline{\tau_{j+1}}$. Resolve the latter with $R$ and factor the resolvent. We get $\overline{d_{j+1}} \lor \overline{\tau_{j+1}}$ Each of the 8 clauses has at most 4 literals. For $j + 1 = t$, we have a similar deduction but with at most $2t + 2$ literals per clause. We obtain $\overline{f_1} \lor \cdots \lor \overline{f_t}$ which can be resolved by the $f_i$ obtained from $\overline{f_i} \lor \overline{f_i} \lor e_i$. Finally, it is easy to check that the refutation has tree structure and is of size polynomial in $t$. □