Understanding Cutting Planes for QBFs

Olaf Beyersdorff[1], Leroy Chew[1], Meena Mahajan[2], and Anil Shukla[2]

1 **School of Computing, University of Leeds, United Kingdom**
2 **The Institute of Mathematical Sciences, HBNI, Chennai, India**

Abstract

We define a cutting planes system CP+∀red for quantified Boolean formulas (QBF) and analyse the proof-theoretic strength of this new calculus. While in the propositional case, Cutting Planes is of intermediate strength between resolution and Frege, our findings here show that the situation in QBF is slightly more complex: while CP+∀red is again weaker than QBF Frege and stronger than the CDCL-based QBF resolution systems Q-Res and QU-Res, it turns out to be incomparable to even the weakest expansion-based QBF resolution system ∀Exp+Res.

Technically, our results establish the effectiveness of two lower bound techniques for CP+∀red: via strategy extraction and via monotone feasible interpolation.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems: Complexity of proof procedures

Keywords and phrases proof complexity, QBF, cutting planes, resolution, simulations

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.

1 Introduction

The main problem of *proof complexity* is to understand the minimal size of proofs for natural classes of formulas in important proof systems. Proof complexity deeply connects to a number of other areas, most notably computational complexity, circuit complexity, first-order logic, and practical solving. Recently the connection to practical solving has been a main driver for the field. Modern SAT solvers routinely solve huge industrial instances of the NP-hard SAT problem with even millions of variables. Because runs of the solver on unsatisfiable formulas can be interpreted as proofs for unsatisfiability in a system corresponding to the solver, proof complexity provides the main theoretical tool for an understanding of the power and limitations of these algorithms.

During the last decade there has been great interest and research activity to extend the success of SAT solvers to the more expressive *quantified Boolean formulas (QBF)*. Due to its PSPACE completeness (even for restricted versions [2]), QBF is far more expressive than SAT and thus applies to further fields such as formal verification or planning [5, 21, 34].

Triggered by this exciting development in QBF solving, *QBF proof complexity* has seen a stormy development in past years. A number of resolution-based systems have been designed with the aim to capture ideas in QBF solving. Broadly, these systems can be classified into two types corresponding to two principal approaches in QBF solving: proof systems modelling *conflict driven clause learning (CDCL):* Q-resolution Q-Res [7, 29], universal resolution QU-Res [38], long-distance resolution [3], and their extensions [4]; and proof systems modelling *expansion solving*: ∀Exp+Res [28] and their extensions [7]. Proof complexity research of these systems resulted in a complete understanding of the relative complexity of QBF resolution systems [4, 8], and the transfer of classical techniques to QBF systems was thoroughly

assessed [9–11]. In addition, stronger QBF Frege and Gentzen systems were defined and investigated [6, 12, 20].

Most SAT and QBF solvers use resolution as their underlying proof system. Resolution is a weak proof systems for which a wealth of lower bounds and in fact lower bound techniques are known (cf. [16, 37]). This raises the question – often controversially discussed within the proof complexity and solving communities – whether it would be advantageous to build solvers on top of more powerful proof systems. While Frege systems appear too strong and proof search is hindered by non-automatisability results [14, 31], a natural system of intermediate strength is Cutting Planes first defined in [19].

Using ideas from integer linear programming [17, 25], Cutting Planes works with linear inequalities, allowing addition of inequalities as well as multiplication and division by positive integers as rules. Translating propositional clauses into inequalities, Cutting Planes derives the contradiction $0 \geq 1$, thereby demonstrating that the original set of inequalities (and hence the corresponding clause set) has no solution. As mentioned, Cutting Planes is a proof system of intermediate strength: it simulates resolution, but allows short proofs for the famous pigeonhole formulas hard for resolution [27], while it is simulated by and strictly weaker than Frege [24, 33].

Our contributions

For QBFs a similar Cutting Planes system based on integer linear programming has been missing. It is the aim of this paper to define a natural Cutting Planes system for QBF and give a comprehensive analysis of its proof complexity.

1. Cutting Planes for QBF. We introduce a complete and sound QBF proof system CP+$\forall$red that works with quantified linear inequalities, where each variable is either quantified existentially or universally in a quantifier prefix. The system CP+$\forall$red extends the classical Cutting Planes system with one single $\forall$-reduction rule allowing manipulation of universally quantified variables. The definition of the system thus naturally aligns with the QBF resolution systems Q-Res [29] and QU-Res [38] and the stronger QBF Frege systems [6] that likewise add universal reduction to their classical base systems.

Inspired by the recent work on semantic Cutting Planes [23] we also define a stronger system semCP+$\forall$red where in addition to universal reduction all semantically valid inferences between inequalities are allowed (Section 7).

2. Lower bound techniques for CP+$\forall$red. We establish two lower bound methods for CP+$\forall$red: strategy extraction (Section 4) and feasible interpolation (Section 5).

Strategy extraction as a lower bound technique was first devised for Q-Res [8] and subsequently extended to QBF Frege systems [6, 12]. The technique applies to calculi that allow to efficiently extract winning strategies for the universal player from a refutation (or alternatively Skolem functions for the existential variables from a proof of a true QBF). Here we show that CP+$\forall$red admits strategy extraction in TC^0, thus establishing an appealing link between CP+$\forall$red proofs (which can count) and the counting circuit class TC^0 (Theorem 8). For each function $f \in \mathsf{PSPACE/poly}$ we construct false QBFs $Q_{\text{qbf}}\text{-}f_n$ where each winning strategy forces the universal player to compute f. Thus assuming the existence of $f \in \mathsf{PSPACE/poly} \setminus \mathsf{TC}^0$ we obtain lower bounds for $Q_{\text{qbf}}\text{-}f_n$ in CP+$\forall$red (Corollary 9) and even semCP+$\forall$red (Corollary 21).

Feasible interpolation is another classical technique transferring circuit lower bounds to proof size lower bounds; however, here we import lower bounds for monotone arithmetic circuits [33] and hence the connection between the circuits and the lines in the proof system is

less direct than in strategy extraction. Feasible interpolation holds for classical resolution [30] and Cutting Planes [33], and indeed was shown to be effective for all QBF resolution systems [9]. Following the approach of [33] we establish this technique for CP+∀red (Theorem 12) and in fact for the stronger semCP+∀red (Theorem 22).

It is interesting to note that while feasible interpolation is the only technique known for classical Cutting Planes, we have two conceptually different lower bound methods – and hence more (conditionally) hard formulas in QBF. This is in line with recent findings in [12] showing that lower bounds for QBF Frege either stem from circuit lower bounds (for NC^1) or from classical Frege lower bounds. Our results here illustrate the same paradigm for CP+∀red: lower bounds arise either from TC^0 lower bounds (via strategy extraction) or via classical lower bound methods for Cutting Planes (feasible interpolation).

3. Relations to other QBF proof systems. We compare our new system CP+∀red with previous QBF resolution and Frege systems. In contrast to the classical setting, the emerging picture is somewhat more complex: while CP+∀red is strong enough to simulate the core CDCL QBF resolution systems Q-Res and QU-Res and indeed is exponentially stronger than these systems (Theorem 17), CP+∀red is incomparable (under a natural circuit complexity assumption) to even the base system ∀Exp+Res of the expansion resolution systems (Theorem 18). Conceptually, this means that in contrast to the SAT case, QBF solvers based on linear programming and corresponding to CP+∀red will not encompass the full strength of current resolution-based QBF solving techniques.

On the other hand, CP+∀red turns out to be simulated by Frege+∀red, and Frege+∀red is exponentially more powerful than CP+∀red (Theorem 19). While this separation could be achieved by lifting the classical separation [33] to QBF by considering purely existentially quantified formulas, we highlight that our separation also holds for natural QBFs expressing the clique-co-clique principle, which is not known to have a succinct propositional representation.

2 Notation and preliminaries

Quantified Boolean Formulas. A literal is a Boolean variable or its negation. We say a literal x is complementary to the literal $\neg x$ and vice versa. A *clause* is a disjunction of literals and a *term* is a conjunction of literals. The empty clause is denoted by $\Box$, and is semantically equivalent to false, denoted $\bot$. A formula in *conjunctive normal form* (CNF) is a conjunction of clauses. For a literal $l = x$ or $l = \neg x$, we write $\mathrm{var}(l)$ for x and extend this notation to $\mathrm{var}(C)$ for a clause C. Let α be any partial assignment. For a clause C, we write $C|_\alpha$ for the clause obtained after applying the partial assignment α to C.

Quantified Boolean Formulas (QBFs) extend propositional logic with Boolean quantifiers with the standard semantics that $\forall x.F$ is satisfied by the same truth assignments as $F|_{x=0} \wedge F|_{x=1}$ and $\exists x.F$ as $F|_{x=0} \vee F|_{x=1}$. We assume that QBFs are in *closed prenex form* with a CNF matrix, i.e., we consider the form $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ where each $\mathcal{Q}_i$ is either $\exists$ or $\forall$, and ϕ is a quantifier-free CNF formula, called the matrix, in the variables $x_1, \ldots, x_n$. Any QBF can be efficiently (in polynomial time) converted to an equivalent QBF in this form (using PSPACE-completeness of such QBFs). We denote such formulas succinctly as $\mathcal{Q} \,.\, \phi$. The *index* $\mathrm{ind}(y)$ of a variable y is its position in the prefix $\mathcal{Q}$; for each $i \in [n]$, $\mathrm{ind}(x_i) = i$. If $\mathrm{ind}(x) < \mathrm{ind}(y)$, we say that x occurs *before* y, or *to the left of* y. The *quantification level* $\mathrm{lv}(y)$ of a variable y in $\mathcal{Q} \,.\, \phi$ is the number of alternations of quantifiers to the left of y in the quantifier prefix of $\mathcal{Q} \,.\, \phi$. For instance, in the QBF $\exists x_1 \forall x_2 \forall x_3 \exists x_4 \phi$, $\mathrm{lv}(x_1) = 1$, $\mathrm{lv}(x_2) = \mathrm{lv}(x_3) = 2$, and $\mathrm{lv}(x_4) = 3$.

Often it is useful to think of a QBF $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ as a game between two players: *universal* ($\forall$) and *existential* ($\exists$). In the i-th step of the game, the player $\mathcal{Q}_i$ assigns a value to the variable x_i. The existential player wins if ϕ evaluates to 1 under the assignment constructed in the game. The universal player wins if ϕ evaluates to 0. A *strategy for* x_i is a function from all variables of index $< i$ to $\{0,1\}$. A *strategy* for the universal player is a collection of strategies, one for each universally quantified variable. Similarly, a *strategy* for the existential player is a collection of strategies, one for each existentially quantified variable. A strategy for the universal player is a winning strategy if using this strategy to assign values to variables, the universal player wins any possible game, irrespective of the strategy used by the existential player. Winning strategies for the existential player are similarly defined. For any QBF, exactly one of the two players has a winning strategy. A QBF is false if and only if there exists a *winning strategy* for the universal player ([26], [1, Sec. 4.2.2], [32, Chap. 19]).

Proof systems. Following notation from [18], a *proof system* for a language $\mathcal{L}$ is a polynomial-time onto function $f : \{0,1\}^* \to \mathcal{L}$. Each string $\phi \in \mathcal{L}$ is a *theorem*, and if $f(\pi) = \phi$, then π is a *proof* of ϕ in f. Given a polynomial-time function $f : \{0,1\}^* \to \{0,1\}^*$ the fact that $f(\{0,1\}^*) \subseteq \mathcal{L}$ is the *soundness property* for f and the fact that $f(\{0,1\}^*) \supseteq \mathcal{L}$ is the *completeness property* for f.

Proof systems for the language of propositional unsatisfiable formulas (UNSAT) are called *propositional proof systems* and proof systems for the language of false QBFs are called *QBF proof systems*. These are *refutational* proof systems. Equivalently, propositional proof systems and QBF proof systems can be defined respectively for the languages of true propositional formulas (TAUT) and of true QBFs. Since any QBF $\mathcal{Q} \,.\, \phi$ can be converted in polynomial time to another QBF $\mathcal{Q}' \,.\, \phi'$ such that exactly one of $\mathcal{Q} \,.\, \phi$ and $\mathcal{Q}' \,.\, \phi'$ is true, it suffices to consider only refutational QBF proof systems.

Given two proof systems f_1 and f_2 for the same language L, we say that f_1 simulates f_2, if there exists a function g and a polynomial p such that $f_1(g(w)) = f_2(w)$ and $|g(w)| \leq p(|w|)$ for all w. Thus g translates a proof w of $x \in L$ in the system f_2 into a proof $g(w)$ of $x \in L$ in the system f_1, with at most polynomial blow-up in proof-size. If there is such a g that is also polynomial-time computable, then we say that f_1 p-simulates f_2.

QBF resolution calculi. *Resolution* (Res), introduced by Blake [13] and Robinson [36], is a refutational proof system for formulas in CNF form. The lines in the Res proofs are clauses. The only inference (resolution) rule is $\dfrac{C \vee x \qquad D \vee \neg x}{C \cup D}$ where C, D denote clauses and x is a variable. A Res refutation derives the empty clause $\square$.

Q-resolution (Q-Res) [29] is a resolution-like calculus operating on QBFs in prenex form with a CNF matrix. The lines in the Q-Res proofs are clauses. It uses the propositional resolution rule above with the side conditions that variable x is existential, and if $z \in C$, then $\neg z \notin D$. (Unlike in the propositional case, dropping this latter condition that $C \cup D$ is not a tautology can lead to unsoundness.) In addition Q-Res has the universal reduction rule $\dfrac{C \vee u}{C}$ and $\dfrac{C \vee \neg u}{C}$ ($\forall$-Red), where variable u is universal and every existential variable $x \in C$ has $\mathrm{lv}(x) < \mathrm{lv}(u)$. If resolution is also permitted with universal variable x (as long as tautologies are not created), then we get the calculus QU-Res [38].

Expansion-based calculi are another type of resolution systems significantly different from Q-Res. In this paper, we will briefly refer to one such calculus, the $\forall$Exp+Res from [28]. In $\forall$Exp+Res, one expands the formula on universal variables, creating multiple annotated copies of existential variables, and then uses classical resolution. For details, see [28].

Frege systems. Frege proof systems are the 'textbook' proof systems for propositional logic based on axioms and rules [18]. A Frege system comprises a finite set of axiom schemes and rules. A *Frege proof* is a sequence of formulas (using $\wedge$, $\vee$, $\neg$) where each formula is

either a substitution instance of an axiom, or can be inferred from previous formulas by a valid inference rule. Frege systems are required to be sound and implicationally complete.

A refutation of a false QBF $\mathcal{Q}.\phi$ in the system Frege+$\forall$red [6] is sequence of lines $L_1, \ldots, L_\ell$ where each line is a formula, $L_1 = \phi$, $L_\ell = \bot$ and each L_i is inferred from previous lines L_j, $j < i$, using the inference rules of Frege or using the universal reduction rule $\dfrac{L_j}{L_j[u/B]}$ ($\forall$**Red**), where u is a universal variable and is the rightmost (highest index) variable among the variables of L_j, B is a formula containing only variables left of u, and $L_j[u/B]$ is the formula obtained from L_j by replacing each occurrence of u in L_j by B.

Circuit classes. We recall the definitions of some standard circuit classes (cf. [39]). The class TC^0 contains all languages recognisable by polynomial-size circuits using $\neg$, $\vee$, $\wedge$ and threshold gates with constant depth and unbounded fan-in. Stronger classes are obtained by using NC^1 circuits of polynomial size and logarithmic depth with bounded fan-in $\neg$, $\vee$, $\wedge$ gates, and by $\mathsf{P/poly}$ circuits of polynomial size. We use non-uniform classes throughout.

Decision lists [35]. A *decision list* is a list L of pairs $(t_1, v_1), \ldots, (t_r, v_r)$, where each t_i is a term and v_i is a value in $\{0, 1\}$, and the last term t_r is the constant term **true** (i.e., the empty term). A decision list L defines a Boolean function as follows: for any assignment α, $L(\alpha)$ is defined to be equal to v_j where j is the least index such that $t_j|_\alpha = 1$. (Such an item always exists, since the last term always evaluates to 1). In [6], this definition has been generalised to $\mathcal{C}$-decision lists (for some circuit class $\mathcal{C}$), where instead of terms one can use circuits from $\mathcal{C}$. A $\mathcal{C}$-decision list yields the circuit $f(x) = \vee_{i=1}^{r} (v_i \wedge C_i(x) \wedge \wedge_{j<i} \neg C_j(x))$. Thus a polynomial-sized TC^0-decision list yields a TC^0 circuit.

3 The CP+$\forall$red proof system

In this section we define a QBF analogue of the classical Cutting Planes proof system by augmenting it with a reduction rule for universal variables. We denote this system by CP+$\forall$red. Consider a false quantified set of inequalities $\mathcal{F} \equiv \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F$, where F is a set of linear inequalities of the form $\sum x_i a_i \geq A$ for integers a_i and A, and F includes the set of inequalities $B = \{x_i \geq 0, -x_i \geq -1 \mid i \in [n]\}$. The inequalities in B are called the Boolean axioms, because they force any integer-valued assignment $\bar{a}$ to the variables, satisfying F, to take only $0, 1$-values. We point out that classical Cutting Planes proof systems (only existential variables) can refute any inconsistent set of linear inequalities over integers. However, once universal quantification is allowed, dealing with an unbounded domain is more messy. Since our primary goal in defining this proof system is to refute false QBFs, and since QBFs have only Boolean variables, we only consider sets of inequalities that contain B.

▶ **Definition 1** (CP+$\forall$red proofs for inequalities)**.** Consider a set of quantified inequalities $\mathcal{F} \equiv \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F$, where F also contains the Boolean axioms. A CP+$\forall$red refutation π of $\mathcal{F}$ is a quantified sequence of linear inequalities $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.[I_1, I_2, \ldots, I_l]$ where the quantifier prefix is the same as in $\mathcal{F}$, I_l is an inequality of the form $0 \geq C$ for some positive integer C, and for every $j \in \{1, \ldots, l\}$, either $I_j \in F$, or I_j is derived from earlier inequalities in the sequence via one of the following inference rules:

1. **Addition**: From $\sum_k c_k x_k \geq C$ and $\sum_k d_k x_k \geq D$, derive $\sum_k (c_k + d_k) x_k \geq C + D$.
2. **Multiplication**: From $\sum_k c_k x_k \geq C$, derive $\sum_k d c_k x_k \geq dC$, where $d \in \mathbb{Z}^+$.
3. **Division**: From $\sum_k c_k x_k \geq C$, derive $\sum_k \frac{c_k}{d} x_k \geq \left\lceil \frac{C}{d} \right\rceil$, where $d \in \mathbb{Z}^+$ divides each c_k.

4. **∀-red**: From $\sum_{k\in[n]\setminus\{i\}} c_k x_k + h x_i \geq C$, derive $\begin{cases} \sum_{k\in[n]\setminus\{i\}} c_k x_k \geq C & \text{if } h > 0; \\ \sum_{k\in[n]\setminus\{i\}} c_k x_k \geq C - h & \text{if } h < 0. \end{cases}$

   This rule can be used provided variable x_i is universal, and provided all existential variables with nonzero coefficients in the hypothesis are to the left of x_i in the quantification prefix. (That is, if x_j is existential, then $j > i \Rightarrow c_j = 0$.) Observe that when $h > 0$, we are replacing x_i by 0, and when $h < 0$, we are replacing x_i by 1. We say that the universal variable x_i has been reduced.

Each inequality I_j is a line in the proof π. Note that proof lines are always of the form $\sum_k c_k x_k \geq C$ for integer-valued c_k, C. The length of π (denoted $|\pi|$) is the number of lines in it, and the size of π (denoted size(π)) is the bit-size of a representation of the proof (this depends on the number of lines and the binary length of the numbers in the proof).

In order to use CP+∀red as a refutational system for QBFs in prenex form with CNF matrix, we must translate QBFs into quantified sets of inequalities.

▶ **Definition 2** (Encoding QBFs as inequalities). We first describe how to encode a CNF formula F over variables $x_1, \ldots, x_n$ as a set of linear inequalities. Define $R(x) = x$, $R(\bar{x}) = 1 - x$. A clause $C \equiv (l_1 \vee \cdots \vee l_k)$ is translated into the inequality $R(C) \equiv \sum_{i=1}^{k} R(l_i) \geq 1$. A CNF formula $\phi = C_1 \wedge \cdots \wedge C_m$ is represented as the set of inequalities $F_\phi = \{R(C_1), R(C_2), \ldots, R(C_m)\} \cup B$, where B is the set of Boolean axioms $x \geq 0, -x \geq -1$ for each variable x. We call this the standard encoding. For a QBF $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ \phi$ with a CNF matrix ϕ, the encoding is the quantified set of linear inequalities $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F_\phi$.

We say that a 0,1-assignment α satisfies the inequality $I \equiv \sum_{i=1}^{n} a_i x_i \geq b$ (i.e., $I|_\alpha = 1$), if $\sum_{i=1}^{n} a_i \alpha_i \geq b$. For any clause C, an assignment satisfies C if and only if it satisfies $R(C)$. Since the standard encoding includes all Boolean axioms, we obtain the following:

▶ Proposition 3. Let $\mathcal{Q}\,.\,\phi$ be a QBF in closed prenex CNF, and let $\mathcal{F} = \mathcal{Q}.\ F_\phi$ be its encoding as a quantified set of linear inequalities. Then $\mathcal{Q}\,.\,\phi$ is false if and only if $\mathcal{F}$ is false.

As for QBFs, we can play the 2-player game on the encoding $\mathcal{F}$ of a QBF. Players choose 0-1 values for their variables in the order defined in the prefix. The ∀ player wins if the assignment so constructed violates some inequality in F. As before, when $\mathcal{F}$ is false, the universal player has a winning strategy; otherwise the existential player has a winning strategy.

▶ **Definition 4** (CP+∀red proofs for QBFs). Let $\mathcal{Q}\,.\,\phi = \mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n\,.\,\phi$ be a false QBF in prenex CNF, and let $\mathcal{F}$ be its encoding as a quantified set of linear inequalities. A CP+∀red (refutation) proof of $\mathcal{Q}\,.\,\phi$ is a CP+∀red proof of $\mathcal{F}$ as defined in Definition 1.

It is worth noting that a CP+∀red proof for inequalities, as in Definition 1, can start with encodings of QBFs, but can also start with quantified sets of inequalities that contain the Boolean axioms but do not correspond to any QBF, since the initial non-Boolean inequalities can have arbitrary integer coefficients.

Observe that in the ∀-red step of CP+∀red, if u is the universal variable being reduced, then u need not be the rightmost variable with a non-zero coefficient. There may be universal variables to the right of u with non-zero coefficients. This is analogous to the conditions in QU-Res, where we require only that every existential variable x in C has $\mathrm{lv}(x) < \mathrm{lv}(u)$. However, in the Frege+∀red proof system defined in [6], the variable being reduced from a formula is required to be the rightmost in the formula; that is, $\mathrm{ind}(x) < \mathrm{ind}(u)$ for every variable other than x in C. We show below that imposing such a condition in CP+∀red

does not affect the strength of the proof system. That is, if we call a proof where the ∀-red steps are applied only to the rightmost universal variables with non-zero coefficients a **normal-form** proof, then any CP+∀red proof can be efficiently converted to one in normal form. In later sections we often assume this normal form.

▶ **Lemma 5.** *Any CP+∀red proof can be converted into normal form in polynomial time.*

Proof. (Sketch.) To reduce a variable u, first reduce all universal variables to the right of u, then reduce u, then re-introduce the previously reduced variables using Boolean axioms. The constant on the right-hand-side may change along the way but finally reverts to its original value. ◀

Now we show that CP+∀red is a complete and sound proof system for false QBFs.

▶ **Theorem 6.** *CP+∀red is a complete and sound proof system for false QBFs. That is, if φ is a false QBF, then there exists a CP+∀red refutation of φ (completeness), and if there exists a CP+∀red refutation of φ, then φ is false (soundness).*

Proof. (Sketch.) Completeness: We show that CP+∀red p-simulates QU-Res; given a QU-Res proof π, for each $C \in \pi$ we can derive $R(C)$ in CP+∀red. (The resolution rule is simulated by the CP part as in the classicial case, and the ∀-Red rule of QU-Res is also present in CP+∀red.) Since QU-Res is known to be complete, it follows that CP+∀red is complete.

Soundness: Let $\mathcal{F} = \mathcal{Q}.\ F$ be the standard encoding of φ, and let $\pi = \mathcal{Q}.[I_1, I_2, \ldots, I_l]$ be a normal form CP+∀red refutation of $\mathcal{F}$. We show that the following is valid for each $j \in [l]$: $\mathcal{Q}.\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1}] \implies \mathcal{Q}.\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1} \wedge I_j]$. Thus if $\mathcal{F} = \mathcal{Q}.F$ is true, then so is $\mathcal{Q}.\ [F \wedge I_1 \wedge \cdots \wedge I_{l-1} \wedge I_l]$. However, I_l is not satisfied by any assignment, so this statement is false. Hence $\mathcal{F}$ is false, and by Proposition 3, φ is also false. ◀

Note that for false quantified inequalities, the soundness of CP+∀red follows from the same proof, but completeness will require an additional argument.

Since we will refer to the p-simulation of QU-Res by CP+∀red later, we state it as a separate lemma; the proof is in the completeness part of the proof of Theorem 6.

▶ **Lemma 7.** *CP+∀red p-simulates QU-Res.*

4 Strategy extraction for CP+∀red

Strategy extraction is an important paradigm in QBF, also very desirable in practice (cf. [3, 7, 22, 26]). Winning strategies for the universal player can be very complex. But a QBF proof system has the strategy extraction property for a particular class of circuits $\mathcal{C}$ whenever we can efficiently extract, from every refutation π of a false QBF φ, a winning strategy for the universal player where the strategies for individual universal variables are computable in circuit class $\mathcal{C}$.

In this section we show how to extract, from a refutation in CP+∀red, winning strategies computable by bounded depth circuits with threshold gates.

▶ **Theorem 8** (Strategy Extraction Theorem)**.** *Given a false QBF $\varphi = \mathcal{Q}.\ \phi$, with n variables, and a CP+∀red refutation π of φ of size m, it is possible to extract from π a winning strategy where for each universal variable $u \in \varphi$, the strategy σ_u can be computed by Boolean circuits of $(m+n)^{O(1)}$ size, constant depth, with unbounded fanin AND, OR, NOT gates as well as threshold gates. In particular, if φ can be refuted in CP+∀red in $n^{O(1)}$ size, then the winning strategies can be computed in TC^0.*

Proof. (Sketch.) We adapt the technique from [6]. Let $\mathcal{Q}.\ F$ be the standard encoding of φ, and let $\pi = \mathcal{Q}.\ [I_1, \ldots, I_l]$ be a normal-form CP+$\forall$red proof of $\mathcal{Q}.\ F$ of length l and size $m \geq l$. For $j \in \{0, 1, \ldots, l\}$, define $\pi_j = \mathcal{Q}.\ [I_{j+1}, \ldots, I_l]$ and $F_j = F \cup \{I_1, \ldots, I_j\}$. By downward induction on j, from π_j we show how to compute, for each universal variable u, a Boolean function σ_u^j that maps each assignment to the variables quantified before u to a bit $\{0, 1\}$. These functions satisfy the property that in a 2-player game played on the formula $\mathcal{Q}.\ F_j$, if the universal player uses strategy σ_u^j for each universal variable u, then finally some inequality in F_j is falsified. We describe the functions σ_u^j by decision lists of size $O(l)$, where each condition is checkable by a constant-depth polynomial-in-m sized threshold circuit.

Since all axioms are included in F, we can skip the axiom steps in the proof.

The strategy is as follows: $\sigma_u^l = 0$ for all u. For $j \leq l$, if I_j is obtained by a classical rule, then $\sigma_u^{j-1} = \sigma_u^j$ for every universal variable u. If I_j is derived using a $\forall$-red rule; that is $I_j = I_k|_{u=b_j}$ for some $k < j$, then for all $u' \neq u$, $\sigma_{u'}^{j-1} = \sigma_{u'}^j$. For u, if $I_k|_{u=b_j}(\vec{a}) = 0$, then $\sigma_u^{j-1}(\vec{a}) = b_j$, else $\sigma_u^{j-1}(\vec{a}) = \sigma_u^j(\vec{a})$. (The value $I_k|_{u=b_j}(\vec{a})$ can be determined since variables to the right of u have zero coefficient in I_k.) It is easy to see that these functions so defined have the desired property. ◀

Theorem 8 yields the following conditional lower bound for CP+$\forall$red proof size.

▶ **Corollary 9.** *If* PSPACE/poly $\not\subseteq$ TC^0*, then there exists a family of false QBFs* Q_{qbf}-f_n *that requires super-polynomial size proofs in CP+*$\forall$*red.*

Proof. Let $f_n \in$ PSPACE/poly $\setminus$ TC^0. Consider the following false sentence based on f_n:

$$\exists x_1 \ldots x_n \forall z. \big[f(\vec{x}) \neq z \big].$$

Since f_n is in PSPACE/poly and QBF is PSPACE-complete, the value of f_n can be compactly expressed by a QBF. That is, $f_n(\vec{x}) \equiv \mathcal{Q}_1 y_1 \ldots \mathcal{Q}_r y_r . \psi_n(\vec{x}, \vec{y})$ where r is polynomial in n and $\psi_n(\vec{x}, \vec{y})$ is in P/poly. Thus we have the false sentence

$$\exists x_1 \ldots x_n \forall z. \big[(\overbrace{\mathcal{Q}_1 y_1 \ldots \mathcal{Q}_r y_r . \psi_n(\vec{x}, \vec{y})}^{f_n(\vec{x})}) \leftrightarrow \neg z \big].$$

We now choose circuits C_n computing ψ_n and use additional variables $\vec{s}$ and $\vec{t}$ to represent the gate values in the P/poly circuits C_n and $\neg C_n$, respectively. We obtain the QBF

$$\exists x_1 \ldots x_n \forall z \mathcal{Q}_1 y_1 \ldots \mathcal{Q}_r y_r \bar{\mathcal{Q}}_1 w_1 \ldots \bar{\mathcal{Q}}_r w_r \exists \vec{s}, \vec{t}. \left[(C_n(\vec{x}, \vec{y}, \vec{s}) \vee z) \wedge \left(\neg C_n(\vec{x}, \vec{w}, \vec{t}) \vee \neg z \right) \right]$$

where $\bar{\mathcal{Q}} = \exists$ if $\mathcal{Q} = \forall$ and vice versa. We call this formula Q_{qbf}-f_n and remark that it is a false prenex QBF with CNF matrix. (C_n can be expressed as a CNF; then adding the literal z to each clause expresses $C_n \vee z$. Similarly for $\neg C_n \vee \neg z$.)

In the two-player game on Q_{qbf}-f_n or on its standard encoding, the only winning strategy for the universal variable z is the function $f_n(\vec{x})$ itself. Therefore if there exists a polynomial size CP+$\forall$red proof for Q_{qbf}-f_n, then from Theorem 8, $f_n \in$ TC^0, a contradiction. ◀

5 Feasible (monotone) interpolation for CP+$\forall$red

In this section we show that CP+$\forall$red admits feasible monotone interpolation. We adapt the technique first used by Pudlák [33] to re-prove and generalise the result of Krajíček [30].

Consider a false QBF of the form

$$\varphi = \exists \vec{p} \mathcal{Q} \vec{q} \mathcal{Q} \vec{r}. \big[A'(\vec{p}, \vec{q}) \wedge B'(\vec{p}, \vec{r}) \big]$$

where $\vec{p}$, $\vec{q}$, and $\vec{r}$ are mutually disjoint sets of propositional variables, $A'(\vec{p},\vec{q})$ is a set of clauses using only the $\vec{p}$ and $\vec{q}$ variables, and $B'(\vec{p},\vec{r})$ is a set of clauses using only the $\vec{p}$ and $\vec{r}$ variables. Thus $\vec{p}$ are the common variables between them. The $\vec{q}$ and $\vec{r}$ variables can be quantified arbitrarily, with any number of quantification levels. Since φ is false, on any assignment $\vec{a}$ to the variables in $\vec{p}$, either $\varphi_{\vec{a},0} = \mathcal{Q}\vec{q}.\ A'(\vec{a},\vec{q})$ or $\varphi_{\vec{a},1} = \mathcal{Q}\vec{r}.\ B'(\vec{a},\vec{r})$ (or both) must be false. An interpolant for φ is a Boolean function that, given $\vec{a}$, indicates which of $\varphi_{\vec{a},0}$, $\varphi_{\vec{a},1}$ is false. As defined in [9], a QBF proof system S admits feasible interpolation if from an S-proof π of such a QBF φ, we can extract a Boolean circuit C_π computing an interpolant for φ, such that, the size of C_π is polynomially related to the size of π. If, whenever the $\vec{p}$ variables occur only positively in A' or only negatively in B', the polynomial sized (with respect to the size of π) interpolating circuit for φ is monotone, then we say that S admits monotone feasible interpolation.

Cutting Planes naturally gives rise to arithmetic rather than Boolean circuits, as in the classical case in [33]. Generalising this to the case of QBFs, we have the following definitions.

▶ **Definition 10.** [33] A monotone real circuit is a circuit which computes with real numbers and uses arbitrary non-decreasing real unary and binary functions as gates.

We say that a monotone real circuit computes a Boolean function (uniquely determined by the circuit), if for all inputs of 0's and 1's the circuit outputs 0 or 1.

▶ **Definition 11.** A QBF proof system S admits *monotone real feasible interpolation* if for any false QBF φ of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A'(\vec{p},\vec{q}) \wedge B'(\vec{p},\vec{r})\big]$ where the $\vec{p}$ variables occur only positively in A' or only negatively in B', and for any S-proof π of φ, we can extract from π a monotone real circuit C of size polynomial in the length of π and the number n of $\vec{p}$ variables, such that C computes a Boolean function, and on every $0,1$ assignment $\vec{a}$ for $\vec{p}$,

$$C(\vec{a}) = 0 \implies \mathcal{Q}\vec{q}.A'(\vec{a},\vec{q}) \text{ is false, and}$$
$$C(\vec{a}) = 1 \implies \mathcal{Q}\vec{r}.B'(\vec{a},\vec{r}) \text{ is false.}$$

Such a C is called a monotone real interpolating circuit for φ.

We prove that the CP+$\forall$red proof system for false QBFs has this property:

▶ **Theorem 12.** *CP+$\forall$red for false QBFs admits monotone real feasible interpolation.*

To prove this, we will actually prove a stronger theorem, about interpolants for all false quantified sets of inequalities (not just those arising from false QBFs).

▶ **Theorem 13.** *CP+$\forall$red for inequalities admits monotone real feasible interpolation. That is, let $\mathcal{F}$ be any false quantified set of inequalities of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})\big]$ where $A \cup B$ includes all Boolean axioms, and where the coefficients of $\vec{p}$ are either all non-negative in A or are all non-positive in B. If $\mathcal{F}$ has a CP+$\forall$red-proof π, of length l, then we can extract a monotone real circuit C of size polynomial in l and the number n of $\vec{p}$ variables in $\mathcal{F}$, such that C computes a Boolean function, and on any $0,1$ assignment $\vec{a}$ to $\vec{p}$,*

$$C(\vec{a}) = 0 \implies \mathcal{Q}\vec{q}.A(\vec{a},\vec{q}) \text{ is false, and}$$
$$C(\vec{a}) = 1 \implies \mathcal{Q}\vec{r}.B(\vec{a},\vec{r}) \text{ is false.}$$

Such a C is called a monotone real interpolating circuit for $\mathcal{F}$.

Proof. (Sketch.) Let $\pi = \exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\ [I_1,\ldots,I_l]$ be a CP+$\forall$red refutation of $\mathcal{F}$. The idea, as in [33], is to associate with each inequality

$$I \equiv \sum_k e_k p_k + \sum_i f_i q_i + \sum_j g_j r_j \geq D$$

in π, two inequalities

$$I_0 \equiv \sum_i f_i q_i \geq D_0, \quad I_1 \equiv \sum_j g_j r_j \geq D_1$$

depending on the Boolean assignment $\vec{a}$ to the $\vec{p}$ variables, in such a way that

- I_0 and I_1 together imply $I|_{\vec{a}}$. (It suffices to ensure $D_0 + D_1 \geq D - \sum_k e_k a_k$.)
- I_0 can be derived solely from the $\mathcal{Q}\vec{q}.A(\vec{a},\vec{q})$ part in CP+$\forall$red.
- I_1 can be derived solely from the $\mathcal{Q}\vec{r}.B(\vec{a},\vec{r})$ part in CP+$\forall$red.

Then the inequalities corresponding to the last step of the proof, I_l, are $0 \geq D_0$ and $0 \geq D_1$, with $D_0 + D_1 \geq 1$. Hence $D_0 > 0 \implies \vec{Q}\vec{q}.A(\vec{a},\vec{q})$ is false, and $D_1 > 0 \implies \vec{Q}\vec{r}.B(\vec{a},\vec{r})$ is false. Note that we only need to compute one of the values D_0, D_1 to identify a false part of $\mathcal{F}$. Furthermore, we will show that if all the coefficients e_k in $B(\vec{p},\vec{r})$ are non-positive, then D_1 can be computed by a real monotone circuit of size $O(nl)$. If all the coefficients e_k in $A(\vec{p},\vec{q})$ are non-negative, then we will show that $-D_0$ can be computed by a real monotone circuit of size $O(nl)$. (The inputs to the circuit are an assignment $\vec{a}$ to the $\vec{p}$ variables.) Applying the unary non-decreasing threshold function $D_1 > 0?$ or $-D_0 \geq 0?$ to its output will then give a monotone real interpolating circuit for $\mathcal{F}$. ◀

Using monotone interpolation (Theorem 12), we now prove an unconditional lower bound for the CP+$\forall$red proof system, which is based on the false clique-co-clique formulas from [9].

▶ **Definition 14.** Fix positive integers k, n with $k \leq n$. CliqueCoClique$_{n,k}$ is the class of QBFs of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\ [A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{r})]$ where

- $\vec{p}$ is the set of variables $\{p_{uv} \mid 1 \leq u < v \leq n\}$. An assignment to $\vec{p}$ picks a set of edges, and thus an n-vertex graph that we denote $G_{\vec{p}}$.
- $\mathcal{Q}\vec{q}.\ A_{n,k}(\vec{p},\vec{q})$ is a QBF expressing the property that $G_{\vec{p}}$ has a clique of size k.
- $\mathcal{Q}\vec{r}.\ B_{n,k}(\vec{p},\vec{r})$ is a QBF expressing the property that $G_{\vec{p}}$ has no clique of size k.

Any QBF in CliqueCoClique$_{n,k}$ expresses the clique-co-clique principle (there is a graph both containing and not containing a k-clique) and is obviously false. In [9], a particular QBF $\varphi_n \in$ CliqueCoClique$_{n,n/2}$ of size polynomial in n is described. It can be easily generalised to QBFs $\varphi_{n,k} \in$ CliqueCoClique$_{n,k}$ of size polynomial in n.

Let $\Phi_{n,k}$ be any QBF in CliqueCoClique, and suppose that it has a CP+$\forall$red proof of length l. From Theorem 12, we obtain a monotone real circuit C of size $O(l + n^2)$ computing a Boolean function, such that for every 0, 1 input vector $\vec{a}$ of length $\binom{n}{2}$ encoding a graph G, $C(\vec{a}) = 1 \iff G$ has a k clique.

In [33], Pudlák showed the following exponential lower bound on the size of real monotone circuits interpolating the famous "clique-color" encodings.

▶ **Theorem 15** ([33]). *Suppose that the inputs for a monotone real circuit C are 0, 1 vectors of length $\binom{n}{2}$ encoding in the natural way graphs on an n-element set. Suppose that C outputs 1 on all cliques of size k and outputs 0 on all complete $(k-1)$-partite graphs, where $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$. Then the size of the circuit is at least $2^{\Omega((n/\log n)^{1/3})}$.*

(In some earlier literature, clique-color has been referred to as clique-co-clique. However, this is misleading because the clique-color encoding is weaker than $\Phi_{n,k}$ in the following sense. The clique-color encoding says that there exists a graph which has a k-clique and is complete $(k-1)$-partite (maximal $(k-1)$-colorable). A graph may neither have a k-clique nor be complete $(k-1)$-partite, so both parts of the clique-color formula may be false. Our clique-co-clique formulas, on the other hand, always have exactly one true part.)

Since complete $(k-1)$-partite graphs have no k-clique, the real monotone interpolating circuit C we obtain from a CP+∀red proof of $\Phi_{n,k}$ also satisfies the premise of Theorem 15. Hence, C must have size exponential in n. But C's size is polynomially related to the length of the CP+∀red proof of $\Phi_{n,k}$. We have thus obtained the following:

▶ **Corollary 16.** *For $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$, any false QBF $\Phi_{n,k} \in$ CliqueCoClique$_{n,k}$ requires proofs of length exponential in n in the CP+∀red proof system. In particular, the QBF $\varphi_{n,k}$ from Definition 14 requires proofs of length exponential in $|\varphi_{n,k}|$ in CP+∀red.*

6 Relative power of CP+∀red and other QBF proof systems

In this section we relate the power of CP+∀red with other well known QBF proof systems.

▶ **Theorem 17.** *CP+∀red is exponentially stronger than Q-Res and QU-Res.*

Proof. By Lemma 7, CP+∀red p-simulates QU-Res (and hence Q-Res), and is thus at least as strong as them. From classical proof complexity we know that false CNF formulas based on the pigeonhole principle are easy for Cutting Planes proof system [19] but hard for resolution [27]. Therefore CP+∀red is exponentially more powerful than any QBF proof system based on resolution (Q-Res, QU-Res, etc.); these systems cannot simulate CP+∀red. ◀

▶ Remark. Note that the separating QBFs have only existential quantification. However, there are also separating QBFs using universal quantifiers.

This means that CP+∀red is stronger than the classical CDCL proof systems. However, as we show next, it is weaker than even the base system of expansion solving.

▶ **Theorem 18.** *CP+∀red and ∀Exp+Res are incomparable unless* P/poly = TC^0*, i.e.,*

- *∀Exp+Res cannot simulate CP+∀red.*
- *If* P/poly $\not\subseteq \mathsf{TC}^0$ *then CP+∀red cannot simulate ∀Exp+Res.*

Proof. In [28], Janota and Marques-Silva show that there exists a family of false QBFs which are hard for ∀Exp+Res but easy to refute in Q-Res. As CP+∀red p-simulates Q-Res (Lemma 7), we conclude that ∀Exp+Res cannot simulate CP+∀red.

For the second claim, let $f_n \in \mathsf{P/poly} \setminus \mathsf{TC}^0$ be computed by circuit family C_n of size $l(n) \in n^{O(1)}$. We use C_n to express the obviously false sentence $\exists x_1 \cdots x_n \forall z. f(\vec{x}) \neq z$. Associate a variable t_i with each gate g_i in C_n, and consider the QBF

$$Q\text{-}f_n \equiv \exists x_1 \cdots x_n \forall z \exists t_1 \cdots t_l . (t_l \neq z) \wedge \bigwedge_{i=1}^{l} (t_i \text{ is consistent with the inputs to gate } i).$$

The inner formula can be written as an $O(l)$-sized CNF, so $Q\text{-}f_n$ has size $n^{O(1)}$. Note that $Q\text{-}f_n$ has a single universal variable z, and the (only) winning strategy for the universal player is $z = f(\vec{x})$. If $Q\text{-}f_n$ has a proof of size polynomial in n, then by Theorem 8, this strategy, and hence f_n, are in TC^0, a contradiction. On the other hand, from [8, Proposition 28], we know that the formula $Q\text{-}f_n$ can be refuted in ∀Exp+Res in $O(n+l)$ steps. (Here, expand on both polarities of the single universal variable z, creating two copies t_i^0 and t_i^1 of each variable t_i. Inductively derive that for each $b \in \{0,1\}$, t_i^b is consistent with the inputs to gate i with the same polarity b, and with the circuit inputs x_j which do not have any polarity. Hence derive $t_l^0 = t_l^1$. Since the clauses expressing $t_l \neq z$ on expansion give the unit clauses $\neg t_l^1$ and t_l^0, we obtain a contradiction.) ◀

▶ **Theorem 19.** *Frege+∀red is exponentially stronger than CP+∀red: Frege+∀red p-simulates CP+∀red, whereas CP+∀red does not simulate Frege+∀red.*

Proof. (Sketch.) In the classical (propositional) setting, Cook, Coullard and Turán [19] first showed that Extended Frege p-simulates Cutting Planes. Then Goerdt [24] showed that even Frege p-simulates Cutting Planes. Using techniques from [15], [19], and [24], we show that the same simulation goes through with minor modifications for QBFs.

Since Frege is exponentially more powerful than Cutting Planes over propositional formulas (as witnessed by the clique-colour formulas [33], see also Section 5), the converse simulation fails, and CP+∀red and Frege+∀red are exponentially separated. ◀

There are also separating examples with non-trivial universal quantifiers. In Section 5, we described a class of QBF formulas expressing the clique-co-clique principle. By Corollary 16, none of them have short proofs in CP+∀red. We show that a particular member of this class (i.e., a particular way of encoding clique-co-clique) has short proofs in Frege+∀red.

▶ **Theorem 20.** *There is a $\Phi_{n,k} \in \text{CliqueCoClique}_{n,k}$ of size polynomial in n, with a Frege+∀red proof of size polynomial in n.*

7 Semantic cutting planes for QBFs

The classical Cutting Planes proof system can be extended to the semantic Cutting Planes proof system by allowing the following semantic inference rule: from inequalities I', I'', we can infer I in one step if every Boolean assignment satisfying both I' and I'' also satisfies I. In [23], it is shown that semantic Cutting Planes is exponentially more powerful than Cutting Planes. We now augment the system semantic Cutting Planes with the ∀-reduction rule as defined for CP+∀red, to obtain a QBF version denoted semCP+∀red. In fact, in this system we need only two rules, semantic inference and ∀-reduction, since the addition, multiplication and division rules of Cutting Planes are also semantic inferences, and the Boolean axioms can be semantically inferred from any inequality.

It is clear that semCP+∀red is sound and complete. However it is not possible to verify the semantic rule efficiently (unless P= NP).

As in CP+∀red, we call a semCP+∀red proof π a normal-form proof if ∀-red is applied only to the rightmost universal variable. Since one can use Boolean axioms in semCP+∀red; Lemma 5 is valid in semCP+∀red as well. That is one can convert any semCP+∀red proof π into a normal form in polynomial time.

Clearly, SemCP+∀red is at least as powerful as CP+∀red. From classical proof complexity we known that semantic Cutting Planes is exponentially more powerful than Cutting Planes [23]. That is, in [23, Theorem 2], it has been shown that for every n, there exists a CNF formula F_n which has a short semantic Cutting Planes refutation but needs $2^{n^{\Omega(1)}}$ lines to refute in Cutting Planes. Thus semCP+∀red is also exponentially more powerful than CP+∀red, as witnessed by these purely existentially quantified formulas.

In Theorem 8, we established strategy extraction from CP+∀red proofs. These results hold for semCP+∀red proofs as well; if I_j is obtained by semantic inference, we do not change the strategy functions and let $\sigma_u^{j-1} = \sigma_u^j$ for every universal variable u. Thus all the conditional lower bounds on CP+∀red (Corollary 9, Theorem 18) continue to hold:

▶ **Corollary 21.** *1. If $\mathsf{PSPACE} \not\subseteq \mathsf{TC}^0$, then for any $f_n \in \mathsf{PSPACE} \setminus \mathsf{TC}^0$, the false QBFs Q_{qbf}-f_n require super-polynomial size proofs in semCP+∀red.*

2. If $\mathsf{P/poly} \not\subseteq \mathsf{TC}^0$, then semCP+∀red cannot simulate ∀Exp+Res. For any $f_n \in \mathsf{P/poly} \setminus \mathsf{TC}^0$, the false QBFs Q-$f_n$ require super-polynomial size proofs in semCP+∀red.

For obtaining unconditional lower bounds, we need an analogue of real monotone interpolation (Theorems 12, 13). For this, we adapt the corresponding proof technique used in the classical case from [23]. Using their technique for semantic inference, and handling axioms and ∀-reduction rules as in the proof of Theorem 13, everything goes through as desired.

▶ **Theorem 22.** *SemCP+∀red admits monotone real feasible interpolation for false QBFs.*

Using Theorem 22, we obtain an unconditional exponential lower bound for semCP+∀red, analogous to Corollary 16.

▶ **Corollary 23.** *For $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$, any false QBF $\Phi_{n,k} \in \textsc{CliqueCoClique}_{n,k}$ requires proofs of length exponential in n in the semCP+∀red proof system. In particular, the QBFs $\varphi_{n,k}$ from Definition 14 require proofs of length exponential in $|\varphi_{n,k}|$ in semCP+∀red.*

Acknowledgements. This work was supported by the EU Marie Curie IRSES grant CORCON, grant no. 48138 from the John Templeton Foundation, EPSRC grant EP/L024233/1, and a Doctoral Training Grant from EPSRC (2nd author).

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.
2 Albert Atserias and Sergi Oliva. Bounded-width QBF is PSPACE-complete. *J. Comput. Syst. Sci.*, 80(7):1415–1429, 2014. URL: http://dx.doi.org/10.1016/j.jcss.2014.04.014, doi:10.1016/j.jcss.2014.04.014.
3 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012.
4 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *SAT'14*, pages 154–169, 2014.
5 Marco Benedetti and Hratch Mangassarian. QBF-based formal verification: Experience and perspectives. *JSAT*, 5(1-4):133–191, 2008.
6 Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In *Proc. ACM Conference on Innovations in Theoretical Computer Science (ITCS'16)*, pages 249–260. ACM, 2016.
7 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In *MFCS, II*, pages 81–93, 2014.
8 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proc. Symposium on Theoretical Aspects of Computer Science (STACS'15)*, pages 76–89. LIPIcs, 2015.
9 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. In *Proc. International Colloquium on Automata, Languages, and Programming (ICALP'15)*, pages 180–192. Springer, 2015.
10 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not simple. In *Proc. Symposium on Theoretical Aspects of Computer Science (STACS'16)*, 2016.
11 Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. In *LATA*, pages 486–498. Springer, 2015.
12 Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In *Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2016.
13 A. Blake. *Canonical expressions in boolean algebra*. PhD thesis, University of Chicago, 1937.

14 Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth Frege proofs. *Computational Complexity*, 13(1–2):47–68, 2004.

15 Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *J. Symb. Log.*, 52(4):916–927, 1987.

16 Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.

17 Václav Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Math. Program.*, 5(1):29–40, 1973.

18 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

19 William J. Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.

20 Uwe Egly. On sequent systems and resolution for QBFs. In *Theory and Applications of Satisfiability Testing (SAT'12)*, pages 100–113, 2012.

21 Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. Conformant planning as a case study of incremental QBF solving. In *Artificial Intelligence and Symbolic Computation (AISC'14)*, pages 120–131, 2014.

22 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference (LPAR)*, pages 291–308, 2013.

23 Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS , Orléans, France*, pages 35:1–35:13, 2016.

24 Andreas Goerdt. Cutting plane versus Frege proof systems. In *Computer Science Logic, 4th Workshop, CSL '90, Heidelberg, Germany, October 1-5, Proceedings*, pages 174–194, 1990.

25 Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.

26 Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In *IJCAI*, pages 546–553, 2011.

27 Amin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

28 Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.

29 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

30 Jan Krajíček. Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

31 Jan Krajíček and Pavel Pudlák. Some consequences of cryptographical conjectures for S_2^1 and EF. *Information and Computation*, 140(1):82–94, 1998.

32 Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

33 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

34 Jussi Rintanen. Asymptotically optimal encodings of conformant planning in QBF. In *AAAI*, pages 1045–1050. AAAI Press, 2007.

35 Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

36 John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

37 Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

38 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *CP*, pages 647–663, 2012.

39 Heribert Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.