Understanding Cutting Planes for QBFs

Olaf Beyersdorff [1a], Leroy Chew [b], Meena Mahajan [c], Anil Shukla [2c]

[a] *Institute of Computer Science, Friedrich Schiller University Jena, Germany*
[b] *School of Computing, University of Leeds, United Kingdom*
[c] *The Institute of Mathematical Sciences, HBNI, Chennai, India*

Abstract

We study the cutting planes system CP+∀red for quantified Boolean formulas (QBF), obtained by augmenting propositional Cutting Planes with a universal reduction rule, 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. A similar picture holds for a semantic version semCP+∀red.

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

Keywords: proof complexity, quantified Boolean formulas, cutting, planes, resolution, Frege proofs

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: since its inception there has been a tight link to computational complexity [25], in particular as a way towards the separation of complexity classes [21], and to bounded arithmetic, where proof systems relate to the strength of weak arithmetic theories [24, 41, 8]. Conceptually, proof complexity made major contributions by calibrating the relative strength of different proof systems and by supplying general techniques for lower bounds for proof size in various calculi (cf. [50, 57, 5]).

Proof complexity also deeply connects to practical solving, and recently this connection has been a main driver for the field. Modern SAT solvers routinely solve huge industrial instances of the NP-hard SAT problem in 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

[1] Corresponding author. Postal address: Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany. email: olaf.beyersdorff@uni-jena.de

[2] Present address: Department of Computer Science and Engineering, Indian Institute of Technology Ropar, Rupnagar 140001, Punjab, India.

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 can express many problems far more succinctly than SAT and thus applies to further fields such as formal verification or planning [52, 6, 28].

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 [40, 10], universal resolution QU-Res [60], long-distance resolution [3], and their extensions [4]; and proof systems modelling *expansion solving*: ∀Exp+Res [38] and their extensions [10]. Proof complexity research of these systems resulted in a complete understanding of the relative complexity of QBF resolution systems [11, 4], and the transfer of propositional techniques to QBF systems was thoroughly assessed [14, 13, 15]. In addition, stronger QBF Frege and Gentzen systems were defined and investigated [27, 9, 17].

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. [57, 21]). 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 (cf. [30]). While Frege systems appear too strong and proof search is hindered by non-automatisability results [44, 19], a natural system of intermediate strength is Cutting Planes first defined in [26].

Using ideas from integer linear programming [33, 23, 56], 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 [35], while it is simulated by and strictly weaker than Frege [32, 49].

As a system of intermediate strength, Cutting Planes has become the basis for a number of different pseudo-Boolean solvers (cf. [55] for a survey). Modern pseudo-Boolean solvers include Sat4j [7], open-WBO [46] and cdcl-cuttingplanes, which combine CDCL-style reasoning with the rules of Cutting Planes. While the above mentioned complexity results suggest that pseudo-Boolean reasoning has clear potential to outperform SAT-solvers based on resolution, practical findings present a rather mixed picture (cf. the recent comprehensive investigation in [30]). With the current theoretical paper on QBF Cutting Planes we open this discussion for QBF as well, preparing the theoretical ground for potential subsequent developments in quantified pseudo-Boolean solving.

Our contributions

In contrast to SAT, a Cutting Planes system based on integer linear programming has been missing for QBFs. 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. As discussed in [9], any propositional line-based proof system can be augmented with a universal reduction rule to obtain a proof system sound and complete for QBFs. We do precisely this for Cutting Planes, and study the resulting system CP+∀red.

1. Cutting Planes for QBF. We introduce a complete and sound QBF proof system CP+∀red that works with quantified linear inequalities, where each variable is either quantified existentially or universally in a quantifier prefix. The system CP+∀red extends the propositional Cutting Planes system with one single ∀-reduction rule allowing manipulation of universally quantified variables. The definition of the system thus naturally aligns with the QBF resolution systems Q-Res [40] and QU-Res [60] and the stronger QBF Frege systems [9] that likewise add universal reduction to their propositional base systems.

Inspired by the recent work on semantic Cutting Planes [31] we also define a stronger system semCP+∀red where in addition to universal reduction all semantically valid inferences between inequalities are allowed (Section 7).

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

Strategy extraction as a lower bound technique was first devised for Q-Res [11] and subsequently extended to QBF Frege systems [9, 17]. 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+∀red admits strategy extraction computable by decision lists of threshold functions, thus establishing an appealing link between CP+∀red proofs (which can count) and a fragment of the counting circuit class TC^0 (Theorem 2) for which exponential lower bounds are known. Thus we obtain lower bounds in CP+∀red (Corollary 5) and even semCP+∀red (Corollary 9).

Feasible interpolation is another propositional technique transferring circuit lower bounds to proof size lower bounds; however, here we import lower bounds for monotone arithmetic circuits [49] 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 propositional resolution [43] and Cutting Planes [49], and indeed was shown to be effective for all QBF resolution systems [14]. Following the approach of [49] we establish this technique for CP+∀red (Theorem 3) and in fact for the stronger semCP+∀red (Theorem 10).

It is interesting to note that while feasible interpolation is the only technique known for propositional Cutting Planes, we have two conceptually different lower bound methods – and hence more hard formulas in QBF. This is in line with recent findings in [17] showing that lower bounds for QBF Frege either stem from circuit lower bounds (for NC^1) or from propositional Frege lower bounds. Our results here illustrate the same paradigm for CP+∀red: lower bounds arise either from lower bounds for a fragment of TC^0 (via strategy extraction) or via propositional 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 propositional 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 5), CP+∀red is incomparable to even the base system ∀Exp+Res of the expansion resolution systems (Theorem 7). 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, which is also exponentially more powerful than CP+∀red (Theorem 8). While this separation could be achieved by lifting the propositional separation [49] to QBF by considering purely existentially quantified formulas, we highlight that our separation also holds for classes of natural QBFs. The first of these are QBFs based on the integer product modulo 2, where we use the strategy extraction technique for the CP+∀red lower bound. The second class of formulas expresses the clique-co-clique principle, which is not known to have a succinct propositional representation. Here we employ the feasible interpolation technique for the CP+∀red lower bound.

It is worthwhile noting that on formulas with only existential quantifiers, CP+∀red degenerates to Cutting Planes (as also QU-Res and Q-Res and ∀Exp+Res to Res, Frege+∀red to Frege). That is, CP+∀red cannot speed up refutations of purely existential formulas. Thus basic separations between CP+∀red and QU-Res, or between Frege+∀red and CP+∀red, come for free from the propositional domain, as mentioned above. However, our lower bounds are "genuine QBF lower bounds"; they hold even in the presence of SAT oracles. (For a formalisation of SAT oracles in QBF proof systems and genuineness of QBF lower bounds, see [16].)

2. Notation and preliminaries

Circuit classes. We recall the definitions of some standard circuit classes (cf. [61]). The class AC^0 contains all languages recognisable by polynomial-size circuits using ¬, ∨, ∧ with constant depth and unbounded fan-in. If also counting gates modulo p are allowed for a prime p, we obtain the class $\mathsf{AC}^0[p]$. For the class TC^0 the circuits may use ¬, ∨, ∧, and threshold gates (the circuits still have 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 ¬, ∨, ∧ gates, and by P/poly circuits of polynomial size. We use non-uniform classes throughout.

The class LTF refers to functions computed by depth-1 TC^0 circuits; this is exactly the functions that can be expressed as the sign of a linear form. Interested readers are referred to the book [47, Chapter 5].

Decision lists [53]. 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 (a conjunction of literals) 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). The length of L is r. 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). A decision list in which every term contains at most k literals is called a k-decision list. It is known that functions computed by 1-decision lists are all in the class LTF. For example, the 1-decision list $(x_1, 1), (\neg x_2, 0), (x_3, 1), (1, 0)$ is represented as $2^3 x_1 - 2^2(1 - x_2) + 2x_3 + 0 > 0$.

In [45], decision lists have been generalised to neural decision lists (or linear decision lists [59]), where instead of terms one can use linear threshold functions. We refer to such lists as LTF-decision lists. In [9], this is further generalised to $\mathcal{C}$-decision lists (for any circuit class $\mathcal{C}$), where instead of terms or linear threshold functions, one can use circuits from $\mathcal{C}$. A $\mathcal{C}$-decision list yields the circuit $C(x) = \bigvee_{i=1}^{r} \left(v_i \wedge C_i(x) \wedge \bigwedge_{j<i} \neg C_j(x) \right)$. In particular, polynomial-length LTF-decision lists are in TC^0, and are even known to be in depth-2 TC^0 (see [59]).

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 assignments to 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 ([34],[1, Sec. 4.2.2],[48, Chap. 19]).

Proof systems. Following notation from [25], 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 $|g(w)| \le p(|w|)$ 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.

A refutational propositional proof system f is "refutationally complete": $f(\{0,1\}^*) \supseteq$ UNSAT. If, furthermore, whenever X entails A, it is possible to derive A from X in the proof system, we say that the proof system is implicationally complete. (We say that a set of formulas X entails a formula A if every Boolean assignment satisfying X also satisfies A.)

QBF resolution calculi. *Resolution* (Res), introduced by Blake [18] and Robinson [54], 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 $\Box$.

Q-resolution (Q-Res) [40] 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 a clause containing universal variable has an existential variable x with $\mathrm{lv}(x) > \mathrm{lv}(u)$, then u cannot be reduced from this clause; we say that u is "blocked" by x.

If resolution is also permitted with universal variable x (as long as tautologies are not created), then we get the calculus QU-Res [60].

Expansion-based calculi are another type of resolution systems significantly different from Q-Res. These calculi are based on *instantiation* of universal variables and operate on clauses that comprise only existential variables from the original QBF, which are additionally *annotated* by a substitution to some universal variables, e.g. $\neg x^{u/0,v/1}$. For any annotated literal l^σ, the substitution σ must not make assignments to variables right of l, i.e. if $u \in \mathsf{dom}(\sigma)$, then u is universal and $\mathrm{lv}(u) < \mathrm{lv}(l)$. To preserve this invariant, we use the *auxiliary notation* $l^{[\sigma]}$, which for an existential literal l and an assignment σ to the universal variables filters out all assignments that are not permitted, i.e. $l^{[\sigma]} = l^{\{u/c \in \sigma \mid \mathrm{lv}(u) < \mathrm{lv}(l)\}}$. We say that an assignment is complete if its domain is all universal variables. Likewise, we say that a literal x^τ is fully annotated if all universal variables u with $\mathrm{lv}(u) < \mathrm{lv}(x)$ in the QBF are in $\mathsf{dom}(\tau)$, and a clause is fully annotated if all its literals are fully annotated.

In this paper, we will briefly refer to one such calculus, the $\forall$Exp+Res from [38]. This calculus works with fully annotated clauses on which resolution is performed. For each clause C from the matrix and an assignment τ to all universal variables, $\forall$Exp+Res uses axiom $\left\{l^{[\tau]} \mid l \in C, l \text{ existential}\right\} \cup \{\tau(l) \mid l \in C, l \text{ universal}\}$. For example, consider a QBF formula with the quantifier prefix $\exists e_1 \forall u_1 \exists e_2 \forall u_2 \exists e_3 \forall u_3$ and containing the clause $C = (e_1 \vee \neg e_2 \vee u_1 \vee e_3 \vee \neg u_3)$. Let $\tau = u_1 \leftarrow 0, u_2 \leftarrow 1, u_3 \leftarrow 1$. Note that τ is an assignment to all universal variables, which falsifies all universal literals in C. Now, given the filtering of annotations, $e_1^{[\tau]}$ is just e_1, $e_2^{[\tau]}$ is $e_2^{u_1/0}$, and $e_3^{[\tau]}$ is $e_3^{u_1/0,u_2/1}$. Hence in $\forall$Exp+Res, downloading the clause C with respect to τ gives clause $(e_1 \vee \neg e_2^{u_1/0} \vee e_3^{u_1/0,u_2/1})$. Likewise, we could download C with respect to $\sigma = u_1 \leftarrow 0, u_2 \leftarrow 0, u_3 \leftarrow 1$; this gives the clause $(e_1 \vee \neg e_2^{u_1/0} \vee e_3^{u_1/0,u_2/0})$. However, downloading C with respect to $\eta = u_1 \leftarrow 1, u_2 \leftarrow 0, u_3 \leftarrow 1$ gives the clause $(e_1 \vee \neg e_2^{u_1/1} \vee 1 \vee e_3^{u_1/1,u_2/0})$ which is a tautology and hence useless in the proof.

As its only rule, $\forall$Exp+Res uses the resolution rule on annotated variables

$$\frac{C \vee x^\tau \qquad D \vee \neg x^\tau}{C \cup D} \text{ (Res)}.$$

Frege systems. Frege proof systems are the common 'textbook' proof systems for propositional logic based on axioms and rules [25]. The lines in a Frege proof are propositional formulas built from propositional variables x_i and Boolean connectives $\neg$, $\wedge$, and $\vee$. A Frege system comprises a finite set of axiom schemes and rules, e.g., $\phi \vee \neg\phi$ is a possible axiom scheme. A *Frege proof* is a sequence of formulas 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. The exact choice of the axiom schemes and rules does not matter as any two Frege systems are p-equivalent, even when changing the basis of Boolean connectives [25] and [42, Theorem 4.4.13].

Usually Frege systems are defined as proof systems where the last formula is the proven formula. We use here the equivalent setting of refutation Frege systems where we start with the negation of the formula that we want to prove and derive the contradiction $\bot$.

A refutation of a false QBF $\mathcal{Q} \,.\, \phi$ in the system Frege+$\forall$red [9] 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

$$\frac{L_j}{L_j[u/B]} \, (\forall \mathbf{Red}),$$

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. Note that the quantifier prefix is not changed at any stage; all the manipulation is with respected to the inner formulas.

There are many sub-systems of Frege studied in the literature (see e.g. [5]). In these subsystems, the lines are restricted to circuits from a class $\mathcal{C}$, yielding the system $\mathcal{C}$-Frege (cf. [39] for a general definition). The system NC^1-Frege coincides with Frege. In this paper we are primarily interested in one other restriction, namely TC^0-Frege, where the lines are all circuits from TC^0. Again, we can lift these systems to QBF, yielding in particular TC^0-Frege+$\forall$red (cf. [9]).

3. The CP+∀red proof system

In this section we define a QBF analogue of the propositional Cutting Planes proof system by augmenting it with a reduction rule for universal variables. We denote this system by CP+∀red. Consider a false quantified set of inequalities $\mathcal{F} \equiv \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ F$, 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 $\vec{a}$ to the variables, satisfying F, to take only $0,1$-values. We point out that propositional Cutting Planes proof systems (only existential variables) can refute any inconsistent set of linear inequalities over integers ([33, 23, 56, 26]). 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 3.1 (CP+∀red proofs for inequalities)**.** Consider a set of quantified inequalities $\mathcal{F} \equiv \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ F$, where F also contains the Boolean axioms. A CP+∀red refutation π of $\mathcal{F}$ is a quantified sequence of linear inequalities $\mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.[I_1, I_2, \dots, I_l]$ 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 $j \in \{1, \dots, l\}$, 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 $\displaystyle\sum_k c_k x_k \geq C$ and $\displaystyle\sum_k d_k x_k \geq D$, derive $\displaystyle\sum_k (c_k + d_k) x_k \geq C + D$.
2. **Multiplication**: From $\displaystyle\sum_k c_k x_k \geq C$, derive $\displaystyle\sum_k d c_k x_k \geq dC$, where $d \in \mathbb{Z}^+$.
3. **Division**: From $\displaystyle\sum_k c_k x_k \geq C$, derive $\displaystyle\sum_k \frac{c_k}{d} x_k \geq \left\lceil \frac{C}{d} \right\rceil$, where $d \in \mathbb{Z}^+$ divides each c_k.
4. **∀-red**: From
$$\sum_{k \in [n] \setminus \{i\}} c_k x_k + h x_i \geq C, \text{ derive } \begin{cases} \displaystyle\sum_{k \in [n] \setminus \{i\}} c_k x_k \geq C & \text{if } h > 0; \\ \displaystyle\sum_{k \in [n] \setminus \{i\}} c_k x_k \geq C - h & \text{if } h < 0. \end{cases}$$

This rule can be used provided variable x_i is universal, and provided all existential variables with non-zero 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.

If an inequality has $c_i \neq 0$, $c_j \neq 0$ for some $i < j$ where x_i is universal and x_j is existential, then x_i cannot be reduced from this inequality. We say that x_i is blocked (by x_j).

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 3.2 (Encoding QBFs as inequalities)**.** We first describe how to encode a CNF formula F over variables $x_1, \dots, x_n$ as a set of linear inequalities. Define $R(x) = x$ and $R(\neg x) = 1 - x$. 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 $F_\phi = \{R(C_1), R(C_2), \dots, R(C_m)\} \cup B$, 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 $\mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ \phi$ with a CNF matrix ϕ, the encoding is the quantified set of linear inequalities $\mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ F_\phi$.

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 1. *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 3.3 (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 3.1.

It is worth noting that a CP+∀red proof for inequalities, as in Definition 3.1, 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.

For example, $\exists x \exists y \forall z [x + y + z \geq 2][x + y \leq 1]$ along with Boolean inequalities is false. But it is not the standard translation of any QBF.

Observe that in the $\forall$-red step of CP+$\forall$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+$\forall$red proof system defined in [9], 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+$\forall$red does not affect the strength of the proof system. That is, if we call a proof where the $\forall$-red steps are applied only to the rightmost universal variables with non-zero coefficients a **normal-form** proof, then any CP+$\forall$red proof can be efficiently converted to one in normal form. In later sections we often assume this normal form.

Lemma 2. *Any CP+$\forall$red proof can be converted into normal form in polynomial time.*

Proof. The idea is simple: 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.

Let π be any CP+$\forall$red proof of a false QBF φ. We efficiently convert π into a normal-form proof π' using the Boolean axioms. Let inequality I' be derived in π from I by a $\forall$-reduction step on w. If w is the rightmost universal variable in I, then nothing needs to be done. Otherwise, in any case, no existential variable right of w can have non-zero coefficient in I. Let $(w =)w_0, w_1, \ldots, w_k$ be the universal variables right of (including) w with non-zero coefficients $h_0, h_1, \ldots, h_k$ in I. We obtain I' from I via the following $(3k + 1)$ steps:
For $j = k$ down to 0, reduce w_j.
For $j = 1$ up to k, if $h_j > 0$ then add $h_j(w_j \geq 0)$, else add $(-h_j)(-w_j \geq -1)$.
Note that the constant on the right-hand-side may change along the way but finally reverts to its original value. Observe that this proof fragment is in normal-form. □

Now we show that CP+$\forall$red is a complete and sound proof system for false quantified inequalities containing the Boolean axioms.

Theorem 1. *CP+$\forall$red is a complete and sound proof system for false quantified inequalities containing the Boolean axioms. That is, if $\mathcal{F} = \mathcal{Q}.\ F$ is a false set of inequalities containing the Boolean axioms, then there exists a CP+$\forall$red refutation of $\mathcal{F}$ (completeness), and if there exists a CP+$\forall$red refutation of $\mathcal{F}$, then $\mathcal{F}$ is false (soundness).*

Proof. Completeness: The key idea is to use the implicational completeness of propositional Cutting Planes [33, 23, 56, 26] and to argue inductively on the correctness of winning strategies, formalised as inequalities.

Let ϕ be a set of inequalities in the variables $x_1, y_1, \ldots x_n, y_n$, and let $\forall_b, \exists_b$ be Boolean quantifiers. (This is equivalent to allowing arbitrary quantifiers but including the Boolean axioms in ϕ.) Without loss of generality, consider the quantifier prefix $Q = \forall_b y_1 \exists_b x_1 \ldots \forall_b y_n \exists_b x_n$. Assume that $Q.\phi$ is false. Then, in the two player game for quantified Boolean semantics, the universal player has a winning strategy. In other words, for every y_i there is a Boolean formula $C_i(x_1, \ldots, x_{i-1}, y_1, \ldots, y_{i-1})$ such that for any Boolean assignment $\vec{x} = \vec{a}$, $\vec{y} = \vec{b}$, if for each $i \in [n]$, $b_i = C_i(a_1, \ldots, a_{i-1}, b_1, \ldots, b_{i-1})$, then $\phi(\vec{a}, \vec{b})$ is false. (Note that the important fact is the existence of such a strategy. The size of the formula computing it is irrelevant.) Equivalently, if a Boolean assignment $\vec{x} = \vec{a}$, $\vec{y} = \vec{b}$ satisfies ϕ, then for some $i \in [n]$, b_i differs from $C_i(a_1, \ldots, a_{i-1}, b_1, \ldots, b_{i-1})$.

For each $j \in [n]$, define the propositional formulas

$$F_j : \bigvee_{i=1}^{j} \left(y_i \neq C_i(x_1, \ldots, x_{i-1}, y_1, \ldots, y_{i-1})\right)$$

Note that $F_j \equiv [y_j \neq C_j] \vee F_{j-1}$, where F_0 is the empty clause.

By the discussion above, F_n is a semantic consequence of ϕ. Representing F_n as a set $\mathcal{I}_n$ of linear inequalities, we can use the implicational completeness of propositional $\mathsf{Cutting\ Planes}$ to derive all the inequalities in $\mathcal{I}_n$ from ϕ.

Note that the inequalities in $\mathcal{I}_n$ do not involve x_n, and so y_n is not blocked. We can thus perform universal reduction on y_n wherever it appears in $\mathcal{I}_n$, with both 0 and 1. This will give us (inequalities corresponding to) the following eventual semantic consequences of $\forall_b y_n F_n$:

$$[C_n(x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) \neq 0] \vee F_{n-1},$$

$$[C_n(x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) \neq 1] \vee F_{n-1}.$$

Taking these two together, the semantic consequence F_{n-1} can be derived.

We repeat this process until we arrive at the empty clause F_0 derived from $\forall_b y_1 F_1$.

Thus we have shown that CP+$\forall$$\mathsf{red}$ is refutationally complete.

Soundness: Let $\mathcal{F} = \mathcal{Q}.\ F$ be a set of quantified inequalities, where F also includes the Boolean axioms. Let $\pi = \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.[I_1, I_2, \ldots, I_l]$ be any CP+$\forall$$\mathsf{red}$ refutation (see Definition 3.1) of $\mathcal{F}$. We can assume (using Lemma 2) that π is in normal form.

To prove soundness, we need to show that $\mathcal{Q}\,.\,\phi$ is false. We do this by showing that the following holds for each $j \in [l]$:

$$\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1}] \models \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1} \wedge I_j].$$

Thus if $\mathcal{F}$ is true, then so is $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.[I_1, I_2, \ldots, I_l]$. However, I_l is not satisfied by any assignment, so this statement is false. Hence $\mathcal{F}$ is false.

Observe that the cases when I_j is derived via Addition, Multiplication, or Division rules are straightforward, since every Boolean assignment satisfying $F \wedge I_1 \wedge \cdots \wedge I_{j-1}$ also satisfies I_j. We now concentrate on the $\forall$-red step.

Say I_j is derived from I_k, $k < j$, via the $\forall$-red rule. Let $u = x_r$ be the universal variable reduced, and let I_k be $\sum_s c_s x_s \geq C$ for some integers $c_1, \ldots, c_n, C$. Since π is in normal form, for all $s > r$, $c_s = 0$.

Suppose the claim is invalid. That is, $\mathcal{F}_{j-1} = \mathcal{Q}\,.\,F \wedge I_1 \wedge \cdots \wedge I_{j-1}$ is true but $\mathcal{F}_j = \mathcal{Q}\,.\,F \wedge I_1 \wedge \cdots \wedge I_j$ is false. Then the existential player has a winning strategy $\sigma_\exists$ for $\mathcal{F}_{j-1}$, while the universal player has a winning strategy $\sigma_\forall$ for $\mathcal{F}_j$. Let α be the assignment constructed when the players use these strategies for their variables. Then α satisfies $F \wedge I_1 \wedge \cdots \wedge I_{j-1}$, and in particular, I_k, but does not satisfy I_j. Define a new strategy $\sigma'_\forall$ for the universal player; it uses the same strategy as $\sigma_\forall$ for variables other than x_r, but flips the strategy of $\sigma_\forall$ for variable x_r. Let β be the assignment constructed by strategies $\sigma_\exists$ and $\sigma'_\forall$. Then $\beta(x_s) = \alpha(x_s)$ for all $s < r$, and $\beta(x_r) \neq \alpha(x_r)$. These are the only values that matter for evaluating I_k. An examination of the $\forall$-red rule shows that it derives the tighter of the two inequalities $I_k|_{x_r=0}$ and $I_k|_{x_r=1}$ as I_j, and hence $I_k(\beta)$ equals $I_j(\alpha)$ and is false. Thus the existential player using strategy $\sigma_\exists$ does not win against the universal player using strategy $\sigma'_\forall$, and hence is not a winning strategy for $\mathcal{F}_{j-1}$, a contradiction. □

We remark that to just show the completeness of CP+∀red for sets of inequalities arising from encodings of QBFs (where, in particular, only $0/1/-1$ coefficients appear in the inequalities), we could alternatively use the following easy simulation of QU-Res by CP+∀red and then refer to the known completeness of QU-Res for QBFs. As we will need the simulation later anyway, we state it here as a lemma.

Lemma 3. *CP+∀red p-simulates QU-Res.*

Proof. Let π be a QU-Res proof. For each $C \in \pi$ we show how to derive $R(C)$ in CP+∀red.

We know that the rules of the propositional cutting planes system can p-simulate the resolution rule [26]. Observe that the same simulation works independent of the quantifier prefix or the nature of the pivot variable. Now we show how CP+∀red simulates the $\forall$-red rule of QU-Res proof system. Consider a $\forall$-red step in QU-Res of the form $\frac{C \vee u}{C}$, where u is universal and all existential variables in the clause C come before u in the prefix. By induction we have derived the inequality $R(C \vee u)$ for the clause $C \vee u$. Reducing u from this inequality is valid. Clearly, the coefficient of u in the inequality $R(C \vee u)$ is $+1$. Hence in the CP+∀red proof, using the $\forall$-red rule assigns $u = 0$ and hence derives $R(C)$. Similarly, for $\frac{C \vee \neg u}{C}$, the coefficient of u in the inequality $R(C \vee \neg u)$ is -1 (the variable u contributes $(1-u)$ to $R(C \vee \neg u)$), hence the $\forall$-red rule in CP+∀red sets $u = 1$ and again derives $R(C)$. □

4. Strategy extraction for CP+∀red

Strategy extraction is an important paradigm in QBF, which is also very desirable in practice to certify the solution of QBF solvers (cf. [34, 3, 29, 10]). 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 LTF-decision lists.

Theorem 2 (Strategy Extraction Theorem)**.** *Given a false QBF $\varphi = \mathcal{Q}.\ \phi$, with n variables, and a CP+∀red refutation π of φ of length l, 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 an* LTF*-decision list of length at most l.*

Furthermore, if π has n_u steps where the variable u is reduced, then the LTF*-decision list computing σ_u has length at most n_u.*

Proof. We adapt the technique from [9]. Let $\mathcal{Q}.\ F$ be the standard encoding of φ, and let $\pi = \mathcal{Q}.\ [I_1, \ldots, I_l]$ be a normal-form $\mathsf{CP}+\forall\mathsf{red}$ proof of $\mathcal{Q}.\ F$ of length l. For $j \in \{0, 1, \ldots, l\}$, define $\pi_j = \mathcal{Q}.\ [I_{j+1}, \ldots, I_l]$ and $F_j = F \cup \{I_1, \ldots, I_j\}$ (note that: $\pi_l = \emptyset$ and $F_0 = F$). 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 chooses values for each universal variable u according to σ_u^j, then finally some inequality in F_j is falsified. We describe the functions σ_u^j by decision lists of size $O(l - j)$, where each condition is an LTF. The functions σ_u^0 are the desired strategies σ_u. To be precise, we show the following:

Claim 4. *For every $j \in [l]$, from π_j, one can extract a winning strategy for the universal player $\sigma^j(\vec{x})$ in the two player game played on $\mathcal{Q}.\ F_j$, such that $\sigma^j(\vec{x})$ can be computed by an LTF-decision list of length $O(l - j)$.*

As already mentioned, we prove Claim 4 by downward induction on j. Since all axioms are included in F, we can skip the axiom download steps in the $\mathsf{CP}+\forall\mathsf{red}$ proof.

Base case: When $j = l$, define $\sigma_u^l = 0$ for all u. Indeed σ_u^l can take any Boolean value as F_l contains I_l which is the contradiction $0 \geq 1$.

Induction hypothesis: Assume that Claim 4 is true at the j^{th} step.

Induction step: For $j \leq l$, if I_j is obtained by a propositional rule, then $\sigma_u^{j-1} \equiv \sigma_u^j$ for every universal variable u. By induction, against any strategy of the existential player, the assignment constructed by playing according to σ_u^j falsifies some inequality in F_j. If it does not falsify I_j, then it must falsify an $I_k \in F_j$ with $k < j$, that is, an $I_k \in F_{j-1}$. Otherwise, since it falsifies I_j and since the inference rules are sound, it also falsifies at least one of the hypotheses I_k, $k < j$.

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 $u' \neq u$, $\sigma_{u'}^{j-1} \equiv \sigma_{u'}^j$. 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.)

By induction, against any strategy of the existential player, the assignment constructed by playing according to σ_u^j falsifies some inequality in F_j. If does not falsify I_j, then it must falsify an $I_{k'} \in F_j$ with $k' < j$, that is, an $I_{k'} \in F_{j-1}$. In this case, we have defined $\sigma_u^{j-1} \equiv \sigma_u^j$, so playing according to σ_u^{j-1} also falsifies $I_{k'} \in F_{j-1}$. Otherwise, since it falsifies $I_j = I_k|_{u=b_j}$ and since in this case we have defined $\sigma_u^{j-1}(\vec{a}) = b_j$, so playing according to σ_u^{j-1} also falsifies $I_k \in F_{j-1}$.

The decision list D_u^{j-1} for σ_u^{j-1} is constructed as follows: If I_j is obtained using a propositional rule, or by reducing a variable other than u, then $D_u^{j-1}(\vec{x}) = D_u^j(\vec{x})$. If u is reduced, then the decision list is $D_u^{j-1}(\vec{x})$ is the following:

$$D_u^{j-1}(\vec{x}) = \text{If } \neg(I_k|_{z=b_j}(\vec{x})) \text{ Then } b_j \text{ Else } D_u^j(\vec{x}).$$

Observe that $D_u^{j-1}(\vec{x})$ has at most one more condition than $D_u^j(\vec{x})$.

By construction, the decision lists D_u^0 have length $O(l)$ and each condition is an LTF. □

We point out that the computational model of LTF-decision lists is weak enough to allow for *unconditional* lower bounds [59]. This is in contrast to TC^0 circuits (of which LTF-decision lists form a strict sub-class), where no unconditional lower bounds are currently known. In fact, strategy extraction in TC^0 is needed for TC^0-Frege+$\forall$red [9].

We now use the mentioned unconditional lower bound for LTF-decision lists together with Theorem 2 to obtain an exponential lower bound for CP+$\forall$red proof size for a specific family of QBFs.

Corollary 5. *There exists a family of false QBFs Q-IP$_n$ requiring exponential-size proofs in CP+$\forall$red.*

Proof. We use the function IP$_n$ that computes the Inner Product (mod 2) of two Boolean vectors. That is,

$$\forall x, y \in \{0,1\}^n, \qquad \mathrm{IP}_n(x,y) = \begin{cases} 1 & \text{if } \sum_i x_i y_i \equiv 1 \pmod 2 \\ 0 & \text{otherwise.} \end{cases}$$

Consider the following false sentence based on IP$_n$:

$$\exists x_1 \dots x_n \exists y_1 \dots y_n \forall z. \big[\mathrm{IP}_n(\vec{x}, \vec{y}) \neq z\big].$$

This can be expressed as a QBF with CNF matrix by using auxiliary variables $t_1, \dots, t_n$, where t_i computes $\sum_{j \leq i} x_i y_i \pmod 2$. Thus we start with the false sentence

$$\exists x_1 \dots x_n \exists y_1 \dots y_n \forall z \exists t_1 \dots t_n. \quad \begin{array}{ll} (\neg t_0) & \\ t_i \leftrightarrow (t_{i-1} \oplus (x_i \wedge y_i)) & \text{for } i \in [n], \\ (t_n \leftrightarrow \neg z) & \end{array}$$

and replace each line by an equivalent CNF formulation. We call the resulting formula Q-IP$_n$ and remark that it is a false prenex QBF with CNF matrix, and is of size $\Theta(n)$.

In the two-player game on Q-IP$_n$ or on its standard encoding, the only winning strategy for the universal variable z is the function IP$_n(\vec{x})$ itself. If there exists a CP+$\forall$red proof for Q-IP$_n$ of length l, then from Theorem 2, IP$_n$ has an LTF-decision list of length l. In [59] it is shown that any LTF-decision list for IP$_n$ must have length greater than $2^{n/2} - 1$. It follows that any CP+$\forall$red proof for Q-IP$_n$ must have length greater than $2^{n/2} - 1$. □

We complement this lower bound with an upper bound for refuting the same formulas in TC^0-Frege+$\forall$red.

Proposition 6. *QBFs Q-IP$_n$ have polynomial-size proofs in TC^0-Frege+$\forall$red.*

Proof. By [37], iterated multiplication can be performed in TC^0. Thus we can compute IP$_n$ by TC^0 circuits. Now we can use Theorem 5.2 of [9], stating that for all functions $f \in \mathsf{TC}^0$ the QBFs Q-f_n, constructed as above in Q-IP$_n$, can be refuted in TC^0-Frege+$\forall$red. This proves the claim. □

Thus, together with the simulation of CP+$\forall$red by Frege+$\forall$red (shown later in Theorem 8) this yields an exponential separation of the two systems.

5. Feasible (monotone) interpolation for CP+∀red

In this section we show that CP+∀red admits feasible monotone interpolation. We adapt the technique first used by Pudlák [49] to re-prove and generalise the result of Krajíček [43].

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 [14], 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 propositional case in [49]. Generalising this to the case of QBFs, we have the following definitions.

Definition 5.1 (Pudlák [49])**.** 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 5.2. 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.

Theorem 3. *CP+∀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+∀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 $\vec{p}$,*

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

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

Proof. Let $\pi = \exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\ [I_1,\dots,I_l]$ be a CP+∀red refutation of $\mathcal{F}$. The idea, as in [49], 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+∀red.
- I_1 can be derived solely from the $\mathcal{Q}\vec{r}.B(\vec{a},\vec{r})$ part in CP+∀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 $\mathcal{F}$.

We first describe the computation of D_0 and D_1 at each inequality. These are computed by two circuits, both of which have exactly the structure of π.

Consider the case when all e_k in $B(\vec{p},\vec{r})$ are non-positive; the other case is analogous. All axioms are considered as either A-axioms or as B-axioms. The Boolean axioms concerning $\vec{p}$ variables are treated as A-axioms in this case.

The computation of D_0 and D_1 proceeds bottom-up as described below.

How inequality I is obtained	D_0	D_1
Axioms:		
$p_k \geq 0$	$-a_k$	0
$-p_k \geq -1$	$a_k - 1$	0
$-q_i \geq -1$	-1	0
$-r_j \geq -1$	0	-1
$q_j \geq 0$ or $r_j \geq 0$	0	0
$\sum_k e_k p_k + \sum f_i q_i \geq D$	$D - \sum e_k a_k$	0
$\sum_k e_k p_k + \sum g_j r_j \geq D$	0	$D - \sum e_k a_k$
Arithmetic:		
Addition $I = I' + I''$	$D_0' + D_0''$	$D_1' + D_1''$
Multiplication $I = hI'$, $h > 0$	$h \times D_0'$	$h \times D_1'$
Division $I = I'/c$, $c > 0$	$\left\lceil \frac{D_0'}{c} \right\rceil$	$\left\lceil \frac{D_1'}{c} \right\rceil$
Reduction: $I = I' \mid_{u=b}$;		
(coefficient of u in I' is h).		
$h > 0$	D_0'	D_1'
$h < 0$ and u is a $\vec{q}$ variable	$D_0' - h$	D_1'
$h < 0$ and u is an $\vec{r}$ variable	D_0'	$D_1' - h$

As in the proof argument from [49], a straightforward induction shows that with these computations, at each proof line I, the inequalities I_0 and I_1 together imply $I \mid_{\vec{a}}$, and that each I_0 can be derived from the A-axioms alone and each I_1 can be derived from the B-axioms alone.

All the operations required for the arithmetic and reduction steps compute non-decreasing functions. At the axioms, note that the dependence of the D_1 values on the assignment values $\vec{a}$ is always with non-negative coefficients $-e_k$; hence these functions are also non-decreasing. Thus we obtain a monotone real circuit for D_1, of size $O(nl)$. □

Monotone real feasible interpolation for QBFs in CP+∀red follows trivially from Theorem 3. (The extra step to note is that if $\vec{p}$ occurs only positively in the clauses of A', the Boolean axioms corresponding to these variables should be included in B, and otherwise in A.)

Using monotone interpolation (Theorem 3), we now prove another lower bound for the CP+∀red proof system, which is based on the false clique-co-clique formulas from [14].

Definition 5.3. 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 [14], a particular QBF $\varphi_n \in$ CliqueCoClique$_{n,n/2}$ of size polynomial in n

is described. It can be easily generalised to QBFs $\varphi_{n,k} \in \textsc{CliqueCoClique}_{n,k}$ of size polynomial in n.

Let $\Phi_{n,k}$ be any QBF in $\textsc{CliqueCoClique}$, and suppose that it has a CP+$\forall$red proof of length l. From Theorem 3, 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 [49], Pudlák showed the following exponential lower bound on the size of real monotone circuits interpolating the famous "clique-color" encodings.

Theorem 4 (Pudlák [49])**.** *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 $(k-1)$-colorable. A graph may neither have a k-clique nor be $(k-1)$-colorable, 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+$\forall$red proof of $\Phi_{n,k}$ also satisfies the premise of Theorem 4. Hence, C must have size exponential in n. But C's size is polynomially related to the length of the CP+$\forall$red proof of $\Phi_{n,k}$. We have thus obtained the following:

Corollary 7. *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 CP+$\forall$red proof system. In particular, the QBF $\varphi_{n,k}$ from Definition 5.3 requires proofs of length exponential in $|\varphi_{n,k}|$ in CP+$\forall$red.*

6. Relative power of CP+$\forall$red and other QBF proof systems

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

6.1. Comparison to weaker QBF proof systems

We start by comparing CP+$\forall$red to the main QBF resolution systems.

Theorem 5. *CP+$\forall$red is exponentially stronger than Q-Res and QU-Res.*

Proof. By Lemma 3, CP+$\forall$red p-simulates QU-Res (and hence Q-Res), and is thus at least as strong as them. From propositional proof complexity we know that false CNF formulas based on the pigeonhole principle are easy for Cutting Planes proof system [26] but hard for resolution [35]. Therefore CP+$\forall$red is exponentially more powerful than any QBF proof system based on resolution (Q-Res, QU-Res, etc.); these systems cannot simulate CP+$\forall$red. □

Note that the separating QBFs have only existential quantification, and this is not the effect one wants to study in QBF proof complexity (cf. also [22] for a discussion). However, there are also natural separating QBFs using universal quantifiers. We discuss two of them below.

In [11] it has been shown that the false QBFs KBKF(t), introduced in [40], are hard for Q-Res. However, they are known to have a polynomial-size proofs in QU-Res [60], and by Lemma 3 in CP+∀red as well; thus they separate Q-Res from CP+∀red.

Arguably, the more interesting separation is between QU-Res and CP+∀red. For this we define the family of false QBFs $\text{QM}\textsc{ajority}_n$ in a manner similar to the definition of $Q\text{-IP}_n$ in Corollary 5. The QBF expresses the false sentence

$$\exists x_1, \ldots, x_{2n+1} \, \forall z \, \textsc{Majority}(x_1, \ldots, x_{2n+1}) \neq z.$$

To express this compactly with a CNF matrix, we use auxiliary variables t_k^i for $i \in [2n+1]$ and $0 \leq k \leq i$, and inductively define $t_k^i = \textsc{Threshold}_k(x_1, \ldots, x_i)$ by using $t_k^i = t_k^{i-1} \vee (t_{k-1}^{i-1} \wedge x_i)$. Therefore $t_{n+1}^{2n+1} = \textsc{Majority}(x_1, \ldots, x_{2n+1})$.

We define $\text{QM}\textsc{ajority}_n$ as the QBF with the prefix

$$\exists x_1, \ldots, x_{2n+1} \, \forall z \, \exists t_0^1, t_1^1, \, \exists t_0^2, t_1^2, t_2^2, \, \ldots \exists t_0^{2n+1}, t_1^{2n+1}, \ldots, t_{2n+1}^{2n+1}$$

and the CNF matrix

$$\begin{array}{llll}
\{t_0^i\} & & & i \in [2n+1] \\
\{x_1, \neg t_1^1\} & \{\neg x_1, t_1^1\} & & \\
\{t_{i-1}^{i-1}, \neg t_i^i\} & \{x_i, \neg t_i^i\} & \{\neg x_i, \neg t_{i-1}^{i-1}, t_i^i\} & 2 \leq i \leq 2n+1 \\
\{x_i, \neg t_k^i, t_k^{i-1}\} & \{\neg t_k^i, t_k^{i-1}, t_{k-1}^{i-1}\} & & 2 \leq i \leq 2n+1, k \in [i-1] \\
\{\neg t_k^{i-1}, t_k^i, \} & \{\neg x_i, t_k^i, \neg t_{k-1}^{i-1}\} & & 2 \leq i \leq 2n+1, k \in [i-1] \\
\{z, t_{n+1}^{2n+1}\} & \{\neg z, \neg t_{n+1}^{2n+1}\}. & &
\end{array}$$

Note that this is a false prenex QBF with CNF matrix, and is of size $\Theta(n^2)$.

Theorem 6. 1. *Any QU-Res proof for* $\text{QM}\textsc{ajority}$ *has exponential size.*
2. $\text{QM}\textsc{ajority}$ *has polynomial-sized proofs in CP+∀red.*

Proof. For the QU-Res lower bound, note that the only winning strategy for the single universal variable z is the function $\textsc{Majority}(x_1, \ldots, x_{2n+1})$ itself. By the results of [36], constant-depth circuits for $\textsc{Parity}$, and hence for $\textsc{Majority}$, must be of size exponential in the number of variables. On the other hand, we know that from any QU-Res proof of size S, one can extract a winning strategy for the universal player as an AC^0-decision list of length S as shown in [11]. Therefore, if $\text{QM}\textsc{ajority}$ has a QU-Res proof of size S, then the winning strategy for the universal player, and hence $\textsc{Majority}$, can be computed by an AC^0-decision list of length S. It follows that S must be exponential in n.

We now describe a CP+∀red proof for $\text{QM}\textsc{ajority}_n$ of length $\Theta(n^2)$.

The proof is tediously long but not very complicated. Here is a roadmap of the steps involved. Associate the predicate $t_k^i \equiv \textsc{Threshold}_k(x_1, \ldots, x_i)$ with the integer point (k, i) in an array of integer points with $i \in [2n+1]$ and $0 \leq k \leq i$. We want to derive the predicate at point $(n+1, 2n+1)$. We reach this predicate by a rightward sweep starting from $i = 1$. At each value of i, the predicates at the extremes $k = 0$ and $k = i$ are derivable from the

Boolean axioms, the clause axioms, and in the case $k = i$, from the predicate at $(i-1, i-1)$. Each other predicate (k, i) is derived from the axioms (Boolean and clause) and the predicates at $(k-1, i-1)$ and $(k, i-1)$.

To implement this, we must express the predicates by appropriate inequalities. To simplify the expressions, we use the notation PSum_i to denote the partial sum $\sum_{j \le i} x_j$. We write the implications and inequalities as follows:

	forward direction	backward direction
Implication	$t^i_k \to \mathrm{PSum}_i \ge k$	$t^i_k \leftarrow \mathrm{PSum}_i \ge k$
Inequality	$-kt^i_k + \mathrm{PSum}_i \ge 0$	$(i-k+1)t^i_k - \mathrm{PSum}_i \ge 1-k$

Now for the details. The reader who is convinced by the outline above can skip the detailed steps of the induction.

We proceed by induction on i.

Base case: For $i = 1$, k is 0 or 1. At $k = 0$, the forward direction is the Boolean axiom $x_1 \ge 0$, and the backward direction inequality $2t^1_0 - x_1 \ge 1$ is obtained by adding the Boolean axiom $-x_1 \ge -1$ and twice the unit clause axiom $t^1_0 \ge 1$. For $k = 1$, both directions are the inequalities corresponding to the axioms $t^1_1 \leftrightarrow x_1$.

Inductive Step: Now assume that $i \ge 2$. The extreme values of k, namely $k = 0$ and $k = i$, are easy and we deal with them first.

$\mathbf{k = 0}$: For the forward direction, we simply add all Boolean axioms $x_j \ge 0$ for $j \le i$ together to get $\mathrm{PSum}_i \ge 0$. For the backward direction, similarly, we add all Boolean axioms $-x_j \ge -1$ for $j \le i$ together to get $-\mathrm{PSum}_i \ge -i$, and then add $(i+1)$ times the unit clause axiom $t^i_0 \ge 1$.

$\mathbf{k = i}$: The forward inequality is derived as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\{\neg t^i_i, t^{i-1}_{i-1}\}}{-t^i_i + t^{i-1}_{i-1} \ge 0}}{-(i-1)t^i_i + (i-1)t^{i-1}_{i-1} \ge 0} \qquad -(i-1)t^{i-1}_{i-1} + \mathrm{PSum}_{i-i} \ge 0}{-(i-1)t^i_i + \mathrm{PSum}_{i-1} \ge 0} \qquad \dfrac{\{\neg t^i_i, x_i\}}{-t^i_i + x_{i-1} \ge 0}}{-it^i_i + \mathrm{PSum}_i \ge 0}$$

The backward inequality is derived as follows:

$$\dfrac{\dfrac{\{t^i_i, \neg t^{i-1}_{i-1}, \neg x_i\}}{t^i_i - t^{i-1}_{i-1} - x_i \ge -1} \qquad t^{i-1}_{i-1} - \mathrm{PSum}_{i-1} \ge 2-i}{t^i_i - \mathrm{PSum}_i \ge 1-i}$$

$\mathbf{1 \le k < i}$: Now we consider the intermediate values. The backward direction is a bit easier and we do it first. It uses the inductively derived backward direction for $i-1$.

We first derive inequalities for $\mathrm{PSum}_{i-1} \ge k \to t^i_k$ and $\mathrm{PSum}_{i-1} \ge k-1 \wedge x_i \to t^i_k$ and then derive the inequality for $\mathrm{PSum}_i \ge k \to t^i_k$.

The derivation of an inequality for $\mathrm{PSum}_{i-1} \ge k \to t^i_k$ is as follows.

$$\dfrac{\dfrac{\dfrac{\{t^i_k, \neg t^{i-1}_k\}}{t^i_k - t^{i-1}_k \ge 0}}{(i-k)t^i_k - (i-k)t^{i-1}_k \ge 0} \qquad (i-k)t^{i-1}_k - \mathrm{PSum}_{i-1} \ge 1-k}{(i-k)t^i_k - \mathrm{PSum}_{i-1} \ge 1-k}$$

The derivation of the inequality for $\textsc{PSum}_{i-1} \geq k-1 \wedge x_i \to t^i_k$ is as follows.

$$\dfrac{\dfrac{\dfrac{\{t^i_k, \neg t^{i-1}_{k-1}, \neg x_i\}}{t^i_k - t^{i-1}_{k-1} - x_i \geq -1}}{(i-k+1)(t^i_k - t^{i-1}_{k-1} - x_i) \geq k-i-1} \qquad (i-k+1)t^{i-1}_{k-1} - \textsc{PSum}_{i-1} \geq 2-k}{(i-k+1)t^i_k - \textsc{PSum}_{i-1} - (i-k+1)x_i \geq 1-i}$$

We can conclude with the following derivations.

$$\dfrac{\dfrac{(i-k)t^i_k - \textsc{PSum}_{i-1} \geq 1-k}{(i-k)^2 t^i_k - (i-k)\textsc{PSum}_{i-1} \geq (i-k)(1-k)} \qquad (i-k+1)t^i_k - \textsc{PSum}_{i-1} - (i-k+1)x_i \geq 1-i}{((i-k+1)(i-k)+1)t^i_k - (i-k+1)\textsc{PSum}_i \geq 1-k(i+1-k)}$$

$$\dfrac{\dfrac{\dfrac{t^i_k \geq 0}{(i-k)t^i_k \geq 0} \qquad ((i-k+1)(i-k)+1)t^i_k - (i-k+1)\textsc{PSum}_i \geq 1-k(i+1-k)}{(i-k+1)^2 t^i_k - (i-k+1)\textsc{PSum}_i \geq 1-k(i+1-k)}}{(i-k+1)t^i_k - \textsc{PSum}_i \geq 1-k}$$

The forward direction uses both the directions of the inductively derived inequalities for $i-1$. Recall that $i \geq 2$ and $1 \leq k \leq i-1$. We need to derive $t^i_k \to \textsc{PSum}_i \geq k$. The key to this is to derive and then combine inequalities for $t^i_k \to \textsc{PSum}_{i-1} \geq k-1$ and $t^i_k \to x_i \vee \textsc{PSum}_{i-1} \geq k$.

In order to show $t^i_k \to \textsc{PSum}_{i-1} \geq k-1$ from our inductive hypothesis and the clause $\neg t^i_k \vee t^{i-1}_k \vee t^{i-1}_{k-1}$, we use the fact that $t^{i-1}_k \to t^{i-1}_{k-1}$ is true. But first we must derive this fact from the induction hypothesis. We start the derivation as follows:

$$\dfrac{(i+1-k)t^{i-1}_{k-1} - \textsc{PSum}_{i-1} \geq 2-k \qquad -kt^{i-1}_k + \textsc{PSum}_{i-1} \geq 0}{-kt^{i-1}_k + (i+1-k)t^{i-1}_{k-1} \geq 2-k}$$

Now we equalise the coefficients on the left-hand-side by adding a multiple of an appropriate Boolean axiom, and then a division rule yields $-t^{i-1}_k + t^{i-1}_{k-1} \geq 0$.

If $i+1-2k > 0$, then we proceed as follows:

$$\dfrac{\dfrac{\dfrac{-t^{i-1}_k \geq -1}{-(i+1-2k)t^{i-1}_k \geq -(i+1-2k)} \qquad -kt^{i-1}_k + (i+1-k)t^{i-1}_{k-1} \geq 2-k}{-(i+1-k)t^{i-1}_k + (i+1-k)t^{i-1}_{k-1} \geq 2-(i+1-k)}}{-t^{i-1}_k + t^{i-1}_{k-1} \geq 0}$$

Alternatively, if $i+1-2k \leq 0$,

$$\dfrac{\dfrac{\dfrac{t^{i-1}_{k-1} \geq 0}{-(i+1-2k)t^{i-1}_{k-1} \geq 0} \qquad -kt^{i-1}_k + (i+1-k)t^{i-1}_{k-1} \geq 2-k}{-kt^{i-1}_k + kt^{i-1}_{k-1} \geq 2-k}}{-t^{i-1}_k + t^{i-1}_{k-1} \geq 0}$$

(At the last step, we may obtain 1 on the right hand side if $k = 1$. In that case, we further add $0 \geq -1$, which may be considered an axiom or may be derived by adding the two Boolean axioms for any variable.)

Next we use the derived inequality $-t_k^{i-1} + t_{k-1}^{i-1} \geq 0$ to derive $-t_k^i + t_{k-1}^{i-1} \geq 0$.

$$\dfrac{\dfrac{\dfrac{-t_k^{i-1} + t_{k-1}^{i-1} \geq 0 \qquad \dfrac{\{-t_k^i, t_k^{i-1}, t_{k-1}^{i-1}\}}{-t_k^i + t_k^{i-1} + t_{k-1}^{i-1} \geq 0}}{-t_k^i + 2t_{k-1}^{i-1} \geq 0} \qquad -t_k^i \geq -1}{-2t_k^i + 2t_{k-1}^{i-1} \geq -1}}{-t_k^i + t_{k-1}^{i-1} \geq 0}$$

This, with the inductive hypothesis, lets us derive $t_k^i \rightarrow \mathrm{P}\textsc{Sum}_{i-1} \geq k-1$.

$$\dfrac{\dfrac{-t_k^i + t_{k-1}^{i-1} \geq 0}{-(k-1)t_k^i + (k-1)t_{k-1}^{i-1} \geq 0} \qquad -(k-1)t_{k-1}^{i-1} + \mathrm{P}\textsc{Sum}_{i-1} \geq 0}{-(k-1)t_k^i + \mathrm{P}\textsc{Sum}_{i-1} \geq 0}$$

As described earlier, we also need an inequality for $t_k^i \rightarrow x_i \vee \sum_{j<i} x_j \geq k$. Under Boolean conditions, the inequality $-kt_k^i + \sum_{j<i} x_j + kx_i \geq 0$ suffices. We use an axiom clause along with the inductive hypothesis to derive it.

$$\dfrac{\dfrac{\dfrac{\{\neg t_k^i, t_k^{i-1}, x_i\}}{-t_k^i + t_k^{i-1} + x_i \geq 0}}{-kt_k^i + kt_k^{i-1} + kx_i \geq 0} \qquad -kt_k^{i-1} + \mathrm{P}\textsc{Sum}_{i-1} \geq 0}{-kt_k^i + \mathrm{P}\textsc{Sum}_{i-1} + kx_i \geq 0}$$

Now we combine the derived inequalities to obtain the inequality for the forward direction.

$$\dfrac{\dfrac{\dfrac{-kt_k^i + \mathrm{P}\textsc{Sum}_{i-1} + kx_i \geq 0 \qquad \dfrac{-t_k^i \geq -1}{(1-k)t_k^i \geq 1-k}}{(1-2k)t_k^i + \mathrm{P}\textsc{Sum}_{i-1} + kx_i \geq 1-k} \qquad \dfrac{-(k-1)t_k^i + \mathrm{P}\textsc{Sum}_{i-1} \geq 0}{-(k-1)^2 t_k^i + (k-1)\mathrm{P}\textsc{Sum}_{i-1} \geq 0}}{-k^2 t_k^i + k\mathrm{P}\textsc{Sum}_i \geq 1-k}}{-kt_k^i + \mathrm{P}\textsc{Sum}_i \geq 0}$$

After Induction: With the induction part of the proof completed, we have shown that $(n+1)t_{n+1}^{2n+1} - \mathrm{P}\textsc{Sum}_{2n+1} \geq -n$ and $-(n+1)t_{n+1}^{2n+1} + \mathrm{P}\textsc{Sum}_{2n+1} \geq 0$ can be derived in a short proof of length $\Theta(n^2)$. (We use $\Theta(1)$ additional steps for both directions of each (i,k) pair.) We now complete the refutation via universal reduction, which can be applied after eliminating the t variables; the partial sums do not block the reduction. The first fragment below reduces z by setting $z = 0$, the second one sets $z = 1$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\{z, t^{2n+1}\}}{z + t^{2n+1}_{n+1} \geq 1}}{(n+1)z + (n+1)t^{2n+1}_{n+1} \geq n+1} \qquad -(n+1)t^{2n+1}_{n+1} + \textsc{PSum}_{2n+1} \geq 0}{(n+1)z + \textsc{PSum}_{2n+1} \geq n+1}}{\textsc{PSum}_{2n+1} \geq n+1}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\{\neg z, \neg t^{2n+1}\}}{-z - t^{2n+1}_{n+1} \geq -1}}{-(n+1)z - (n+1)t^{2n+1}_{n+1} \geq -(n+1)} \qquad (n+1)t^{2n+1}_{n+1} - \textsc{PSum}_{2n+1} \geq -n}{-(n+1)z - \textsc{PSum}_{2n+1} \geq -2n-1}}{-\textsc{PSum}_{2n+1} \geq -n}$$

$$\dfrac{-\textsc{PSum}_{2n+1} \geq -n \qquad \textsc{PSum}_{2n+1} \geq n+1}{0 \geq 1}$$

□

As a corollary we obtain that strategy extraction for CP+∀red, established in Theorem 2 to have LTF-decision lists, cannot be improved to AC^0 or even $\mathsf{AC}^0[p]$ for any prime p.

Corollary 8. *CP+∀red does not admit strategy extraction in* AC^0 *or in* $\mathsf{AC}^0[p]$ *for any prime* p*.*

Proof. By results of [51, 58], Majority requires exponential-size AC^0 circuits, (in fact even $\mathsf{AC}^0[p]$ for a prime p). By the previous theorem QMajority has short proofs in CP+∀red and Majority is the only winning strategy for the universal player on the formula. Therefore we cannot extract winning strategies from CP+∀red proofs in AC^0 (and neither in $\mathsf{AC}^0[p]$). □

6.2. Incomparability results

Theorems 5, 6 show that CP+∀red is stronger than the propositional CDCL proof systems. However, as we show next, it is incomparable with even the base system of expansion solving.

Theorem 7. *CP+∀red and ∀Exp+Res are incomparable. i.e.,*

- *∀Exp+Res cannot simulate CP+∀red.*
- *CP+∀red cannot simulate ∀Exp+Res.*

Proof. In [38], 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 3), we conclude that ∀Exp+Res cannot simulate CP+∀red.

For the second claim, recall the QBF Q-IP$_n$; Corollary 5 shows that it needs exponential refutation size in CP+∀red. On the other hand, from [11, Proposition 28], we know that it (and in fact any similar formula Q-f_n where f_n has polynomial-sized circuits) can be refuted in ∀Exp+Res in $O(n)$ steps. Briefly, the refutation proceeds as follows: 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 equivalent to $t_{i-1}^b \oplus (x_i \wedge y_i)$. 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. □

Another consequence of the short CP+∀red proofs for QMajority (Theorem 6) is the (partial) incomparability of AC^0-Frege+∀red with CP+∀red. Because of the hardness of QMajority for $\mathsf{AC}^0[p]$-Frege+∀red for an arbitrary prime p, shown in [9], we can conclude that $\mathsf{AC}^0[p]$-Frege+∀red does not simulate CP+∀red, but a stronger Frege system is needed for this simulation. This directly leads to our next topic.

6.3. Comparison to stronger QBF proof systems

We now proceed to compare CP+∀red with stronger QBF systems. A natural candidate is Frege+∀red, which we will show to be exponentially stronger than CP+∀red.

Theorem 8. *Frege+∀red is exponentially stronger than CP+∀red: Frege+∀red p-simulates CP+∀red, whereas CP+∀red does not simulate simulate Frege+∀red.*

Proof. **Frege+∀red p-simulates CP+∀red**: In the classical (propositional) setting, Cook, Coullard and Turán [26] first showed that Extended Frege p-simulates Cutting Planes. Then Goerdt [32] showed that even Frege p-simulates Cutting Planes. Using techniques from [20], [26], and [32], we show that the same simulation goes through with minor modifications for QBFs.

Let φ be the false formula $\mathcal{Q}x_1 \cdots \mathcal{Q}x_n.\ [C_1 \wedge \cdots \wedge C_m]$, and let $\mathcal{F}$ denote its standard encoding as described in Definition 3.2. Fix any CP+∀red proof $\pi = \mathcal{Q}x_1 \cdots \mathcal{Q}x_n.\ [I_1, I_2, \ldots, I_m]$ of $\mathcal{F}$. By Lemma 2, we can assume that π is in normal form. We need to represent each inequality I as a propositional formula $\mathrm{Rep}(I)$, such that on each assignment α to the Boolean variables, $\mathrm{Rep}(I)(\alpha)$ is 1 if and only if $I|_\alpha$ is 1. We do this almost exactly as in [32].

Integer arithmetic is in NC^1. Thus, for a string of $(n+1)L$ Boolean variables $\vec{y}$ representing the bits of $n+1$ signed integers $a_1, a_2, \ldots, a_n, b$ with bit length L each, and n Boolean variables $x_1, \ldots, x_n$, there is a formula $F(\vec{y}, \vec{x})$ of size polynomial in $n+L$ (and depth logarithmic in nL) with the following properties:

- For every assignments β to the $\vec{y}$ variables, $F(\beta, \vec{x})$ represents the inequality $\sum_i a_i x_i \geq b$.
- For every assignments α to the $\vec{x}$ variables, we have $F(\beta, \alpha)$ is true if and only if $\sum_i a_i \alpha_i \geq b$ is true.

To represent a specific inequality $I : \sum_i a_i x_i \geq b$, we append to the leaves of F labelled from $\vec{y}$ subformulas of the form $x \vee \bar{x}$ or $x \wedge \bar{x}$ depending on the bits of the a_i's and b. The resulting formula has the variables $x_1, \ldots, x_n$ and is the representation $\mathrm{Rep}(I)$. It will be more convenient to think of $\mathrm{Rep}(I)$ as a formula with multiple output gates. The main output gate is the one described above, taking truth value 1 if and only if the inequality is satisfied. Additionally, for each bit of each a_i and b, one of the gates of $\mathrm{Rep}(I)$ evaluates to exactly that bit.

Our simulating Frege+∀red proof will have the structure

$$\pi_1, \mathrm{Rep}(I_1), \pi_2, \mathrm{Rep}(I_2), \ldots, \pi_m, \mathrm{Rep}(I_m), \pi_{m+1},\ \text{false},$$

where each π_i is a sequence of propositional formulas. That is, the simulating Frege+∀red proof is a sequence of formulas containing the subsequence

$$\mathrm{Rep}(I_1), \mathrm{Rep}(I_2), \ldots, \mathrm{Rep}(I_m), \text{ false}.$$

For each axiom clause C, we derive the formula $\mathrm{Rep}(R(C))$ by a short (polynomial in n) Frege+∀red proof. (Note that the clause C fixes the values of a_i and b, and hence it fixes the values of the y variables to some β.) For each coefficient a_i, $i \in [n]$, and b, inside $\mathrm{Rep}(R(C))$ there are explicit subformulas representing their bits a_{ij} and b_j for $i \in [n]$, $j \in [L]$. (To handle carry overflows, we pad each coefficient with 0s to length $\Theta(L)$ as in [32].) Also included within $\mathrm{Rep}(R(C))$ are explicit subformulas for each $a_{ij} \wedge x_i$, since these are the values used in testing the inequality.

We now need to derive each $\mathrm{Rep}(I_t)$ from $\mathrm{Rep}(I_j)$, $j < t$, via short (polynomial in the size of proof π) Frege+∀red proofs.

The addition rule, multiplication rule, and the division rule can be simulated as in the propositional case [32]: since integer arithmetic is in NC^1, we have small formulas G expressing the coefficients of the resulting inequality I from the used inequalities I' and I''. A Frege-style proof can describe how values from the subformulas in $\mathrm{Rep}(I')$ and $\mathrm{Rep}(I'')$ propagate through G to bits equivalent to the corresponding input bits of $\mathrm{Rep}(I)$.

Now we show the ∀-red step simulation.

Suppose the inequality I_k is obtained from I_j for some $j < k$ by applying the ∀-red rule, reducing universal variable u. Clearly, u is the rightmost variable in I_j with non-zero coefficient h_u. Inductively, we have already derived $\mathrm{Rep}(I_j)$. Let $b_u = 0$ if $h_u > 0$, otherwise $b_u = 1$. We need to instantiate u in $\mathrm{Rep}(I_j)$ with b_u. But u is not the rightmost variable in $\mathrm{Rep}(I_j)$. However, for each variable v to the right of u, we know that the coefficient a_v of v in I_j is 0, and hence the subformulas evaluating to the bits a_{vj}, as well as the subformulas evaluating $a_{vj} \wedge v$, are all 0. In Frege+∀red, we can transform the pair of subformulas, $a_{vj} \wedge v$, and $a_{vj} \equiv 0$, to the subformula $a_{vj} \wedge 0$, and thus eliminate v (note that v does not figure anywhere else in the formula).

Once this is done for all variables right of u, we have a formula R in which the ∀-reduction step is valid in Frege+∀red. Performing this reduction gives the formula $R' = R \mid_{u=b_u}$. Now, a short Frege proof allows us to derive $\mathrm{Rep}(I_j \mid_{u=b_u}) = \mathrm{Rep}(I_k)$. To see why such a proof exists, consider the case $b_u = 0$. Inside R' we have subformulas for the bits h_{uj} of the coefficient h_u of u, and bits for $h_{uj} \wedge u$, and at u we have attached a simple subformula evaluating to 0. What we want is subformulas where u is still free, but the bits of the new coefficient of u are all 0. That is, from $h_{uj} \wedge u$ and $u \equiv 0$, we want to derive $0 \wedge u$ (the reverse of what we did before in the reduction for variables v right of u). This is easy in Frege+∀red. The case when $b_u = 1$ is similar, with the added task of subtracting h_u from the right-hand-side. This too can be tracked using a NC^1 formula for subtraction.

CP+∀red does not simulate Frege+∀red. While we could just refer to the separation of the propositional fragments of these proof systems[3] it is more

[3] Frege is exponentially more powerful than Cutting Planes as witnessed by the clique-colour formulas [49] (see also Section 5), this separation carries over to CP+∀red and Frege+∀red, because, on existentially quantified formulas, CP+∀red coincides with CP (likewise for Frege).

interesting to achieve this separation on 'genuine' QBFs. Such a separation is provided by the Q-IP$_n$ formulas, which by Proposition 6 have polynomial-size Frege+∀red proofs, but require exponential-size CP+∀red proofs by Corollary 5. □

We believe that this result can possibly be strengthened to an exponential separation between CP+∀red and TC^0-Frege+∀red. As this holds already for the separating example by Proposition 6, we would just need to tighten the simulation to a simulation of CP+∀red by TC^0-Frege+∀red.

There are further examples separating CP+∀red from Frege+∀red, with non-trivial universal quantifiers. In Section 5, we described a class of QBF formulas expressing the clique-co-clique principle. By Corollary 7, 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. (However, not all encodings have short proofs; see the discussion after the proof of the theorem.)

Theorem 9. *There is a sequence $\Phi_{n,k} \in \textsc{CliqueCoClique}_{n,k}$ of size polynomial in n, with polynomial-size Frege+∀red proofs.*

Proof. Fix positive integers n (indicating the number of vertices of the graph) and $k \leq n$ (indicating the size of the clique queried) and let $\vec{p}$ be 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 $G_{\vec{p}}$.

The formula $\mathcal{Q}\vec{q}.\ A_{n,k}(\vec{p},\vec{q})$ should express the property $\textsc{Clique}(n,k)$, that $G_{\vec{p}}$ has a clique of size k, and $\mathcal{Q}\vec{r}.\ B_{n,k}(\vec{p},\vec{r})$ should express the property co-$\textsc{Clique}(n,k)$, that $G_{\vec{p}}$ has no clique of size k.

Let $\vec{q}$ be the set of variables $\{q_{iu} \mid i \in [k], u \in [n]\}$. We use the following clauses

$$\begin{array}{lll} C_i & = q_{i1} \vee \cdots \vee q_{in} & \text{for } i \in [k] \\ D_{i,j,u} & = \neg q_{iu} \vee \neg q_{ju} & \text{for } i,j \in [k], i < j \text{ and } u \in [n] \\ E_{i,u,v} & = \neg q_{iu} \vee \neg q_{iv} & \text{for } i \in [k] \text{ and } u,v \in [n], u < v \\ F_{i,j,u,v} & = \neg q_{iu} \vee \neg q_{jv} \vee p_{uv} & \text{for } i,j \in [k], i < j \ \text{ and } \ u \neq v \in [n]. \end{array}$$

We can now express $\textsc{Clique}(n,k)$ as a polynomial-size QBF $\exists\vec{q}.A_{n,k}(\vec{p},\vec{q})$, where

$$A_{n,k}(\vec{p},\vec{q}) = \bigwedge_{i\in[k]} C_i \wedge \bigwedge_{i<j,u\in[n]} D_{i,j,u} \wedge \bigwedge_{i\in[k],u<v} E_{i,u,v} \wedge \bigwedge_{i<j,u\neq v} F_{i,j,u,v}.$$

Here the edge variables $\vec{p}$ appear only positively in $A_{n,k}(\vec{p},\vec{q})$.

Likewise co-$\textsc{Clique}(n,k)$ can be written as a QBF $\forall\vec{r}\exists\vec{t}.B_{n,k}(\vec{p},\vec{r},\vec{t})$ of polynomial size. In [14] one way of doing so is described. Another way is take a disjoint copy of $\exists\vec{r}A_{n,k}(\vec{p},\vec{r})$ with the same $\vec{p}$ variables but new variables instead of $\vec{q}$, use auxiliary variables to transform the matrix to 3-CNF, negate it to obtain $\forall\vec{r}A'_{n,k}(\vec{p},\vec{r})$, where A' is a 3-DNF, and then obtain a short CNF equivalent of A'. We describe here a somewhat different and more transparent encoding. This encoding can be used to obtain the results of [14] as well, and is more convenient for us here because it allows us to obtain a short Frege+∀red proof. As noted in the remark after this proof, there are very similar encodings of the same clique-co-clique principle that are unlikely to have short proofs, so the specific encoding chosen is important.

In $\vec{r}$, we have a variable r_{iu} for every variable q_{iu}, and in $\vec{t}$ we have a variable for each clause of $A_{n,k}$, and one new variable t; that is, $\{t_K \mid K \in A_{n,k}\} \cup \{t\}$. For each clause K in $A_{n,k}(\vec{p}, \vec{q})$, replace each occurrence of each q_{iu} by the corresponding r_{iu} to obtain clause K'. These clauses are not put into the matrix. Instead, we include in $B_{n,k}(\vec{p}, \vec{r}, \vec{t})$ the equivalences $t_K \leftrightarrow K'$, which we represent as a set of clauses. If all these equivalences are satisfied by an assignment to $\vec{r}$, then $\wedge_K t_K$ asserts that the $\vec{r}$ variables encode a clique. We introduce clauses for $t \leftrightarrow \bigwedge_{K \in A_{n,k}} t_K$, so t indicates whether the $\vec{r}$ variables encode a clique. Because we want to represent the co-clique formula we also include the unit clause $\neg t$. These clauses together give $B_{n,k}(\vec{p}, \vec{r}, \vec{t})$, which yields the CNF formula co-$\textsc{Clique}(n,k) = \forall \vec{r} \exists \vec{t}. B_{n,k}(\vec{p}, \vec{r}, \vec{t})$.

Our clique-co-clique formulas $\Phi_{n,k}$ are $\exists \vec{p} \exists \vec{q} \forall \vec{r} \exists \vec{t}. A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{r}, \vec{t})$. We now show that these formulas are easy in Frege+$\forall$red.

We use a result from [17, Theorem 8.1] which shows that a Frege+$\forall$red super-polynomial lower bound must either come from a circuit lower bound or a propositional Frege lower bound. More precisely, if false QBFs Φ_n do not admit polynomial-size Frege+$\forall$red proofs, then either the universal player does not have NC^1 winning strategies for the universal variables, or if small NC^1 winning strategies exist, then the propositional formulas obtained by substituting the NC^1 circuits for universal variables in Φ_n are hard for propositional Frege.

In the case of the clique co-clique formulas $\Phi_{n,k}$ there exist short winning strategies for the universal player, namely $\vec{r} = \vec{q}$. To see this, we just need to consider the case where the existential player chooses a graph $\vec{p}$ that contains a k-clique exhibited in the $\vec{q}$-variables, because otherwise the universal player immediately wins on $A_{n,k}(\vec{p}, \vec{q})$. In this case, choosing $\vec{r} = \vec{q}$ ensures that $B_{n,k}(\vec{p}, \vec{r}, \vec{t})$ fails as $\vec{r}$ indeed is a k-clique.

Substituting these winning strategies into $\Phi_{n,k}$, we obtain the false propositional formulas $A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{q}, \vec{t})$, which admit short Frege refutations.

Using this intuition we can refute $\Phi_{n,k}$ in Frege+$\forall$red with short proofs. For this we first derive the tautology $\neg(A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{q}, \vec{t}))$ by demonstrating a way to find a contradiction in $A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{q}, \vec{t})$. To do this we observe that for any clause $K \in A_{n,k}(\vec{p}, \vec{q})$, we have the equivalences $(t_K \leftrightarrow K) \in B_n(\vec{p}, \vec{q}, \vec{t})$, so we derive all t_K. Then, because $(t \leftrightarrow \bigwedge_{K \in A_{n,k}} t_K) \in B_{n,k}(\vec{p}, \vec{q}, \vec{t})$, we obtain t. This means that with $\neg t \in B_{n,k}(\vec{p}, \vec{q}, \vec{t})$ we have a contradiction, thus proving the negation $\neg(A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{q}, \vec{t}))$.

Moving forward to the next step, we derive in (polynomially many) Frege steps the implication $\bigwedge_{i \in [k], j \in [\binom{n}{2}]} (q_{i,j} \leftrightarrow r_{i,j}) \rightarrow \neg(A_{n,k}(\vec{p}, \vec{q}) \wedge B_n(\vec{p}, \vec{r}, \vec{t}))$, from which together with the axiom $A_n(\vec{p}, \vec{q}) \wedge B_n(\vec{p}, \vec{r}, \vec{t})$ we derive the disjunction $\bigvee_{i \in [k], j \in [\binom{n}{2}]} (r_{i,j} \neq q_{i,j})$.

Now we perform $\forall$-reduction, starting with the rightmost universal variable r_{i_1,j_1} and instantiating it with both 0 and 1. Thus we obtain two lines:

$$(0 \neq q_{i_1,j_1}) \vee \bigvee_{i \in [k], i \neq i_1, j \in [\binom{n}{2}], j \neq j_1} (r_{i,j} \neq q_{i,j})$$

$$(1 \neq q_{i_1,j_1}) \vee \bigvee_{i \in [k], i \neq i_1, j \in [\binom{n}{2}], j \neq j_1} (r_{i,j} \neq q_{i,j})$$

We then use the tautology $(q_{i_1,j_1} \leftrightarrow 0) \vee (q_{i_1,j_1} \leftrightarrow 1)$ and the two instantiations to remove the disjunct $(r_{i_1,j_1} \neq q_{i_1,j_1})$ from the disjunction. Continuing this iteratively, we remove all disjuncts and are left with the empty disjunct, hence refuting $\Phi_{n,k}$ in polynomial size. □

Note that if we changed the quantification and used formula $\exists\vec{p}\forall\vec{r}\exists\vec{t}\exists\vec{q}.A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{r},\vec{t})$ we would still be describing exactly the same contradiction between clique and co-clique. However this encoding is unlikely to have short proofs in Frege+∀red. This is because the strategy extraction theorem from [9] allows us to extract, from a polynomially-sized Frege+∀red proof, an NC^1 circuit for a winning strategy of the universal player. In the encoding used in the proof of Theorem 9, the universal player has an exceedingly simple strategy for choosing an assignment to the $\vec{r}$ variables – simply copy the values chosen by the existential player for $\vec{q}$. However, with this new encoding, where only the quantifier prefix has changed, the universal player does not have access to the values of $\vec{q}$ while choosing values for $\vec{r}$. Any winning strategy must, given only the graph $G_{\vec{p}}$, choose an assignment to $\vec{r}$ that picks out a clique in $G_{\vec{p}}$. This is an NP-hard problem, yet a short proof in Frege+∀red would imply that it is in NC^1.

7. Semantic cutting planes for QBFs

The propositional 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 [31], 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 2 is valid in semCP+∀red as well, i.e., 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 propositional proof complexity we known that semantic Cutting Planes is exponentially more powerful than Cutting Planes [31]. That is, in [31, 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 2, 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 the lower bound on CP+∀red (Corollary 5 and the separation Theorem 7) continues to hold:

Corollary 9. *The false QBFs Q-IP require exponential size proofs in semCP+$\forall$red. Hence semCP+$\forall$red cannot simulate $\forall$Exp+Res.*

For extending the lower bound from Corollary 7 we need an analogue of real monotone interpolation (Theorem 3). For this, we adapt the corresponding proof technique used in the propositional case from [31]. Using their technique for semantic inference, and handling axioms and $\forall$-reduction rules as in the proof of Theorem 3, everything goes through as desired.

Theorem 10. *SemCP+$\forall$red admits monotone real feasible interpolation for false QBFs.*

Proof. Let $\varphi = \exists\vec{p}Q\vec{q}Q\vec{r}(A'(\vec{p},\vec{q}) \wedge B'(\vec{p},\vec{r}))$ be a false QBF formula. Without loss of generality, the $\vec{p}$ variables appear only negatively in $B'(\vec{p},\vec{r})$. Consider the standard encoding $\mathcal{F} = \exists\vec{p}Q\vec{q}Q\vec{r}(A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r}))$ of φ (see Definition 3.2). Clearly the coefficient of $\vec{p}$ variables in B are non-positive. As discussed before it is sufficient to extract a monotone real feasible interpolation for $\mathcal{F}$. Let π be any semCP+$\forall$red proof of $\mathcal{F}$, and as in the proof of Theorem 3, we construct a real monotone interpolating C to detect whether $D_1 > 0$. Axioms and the $\forall$-reduction rule are handled exactly as in Theorem 3. Now suppose that the inequality $I \equiv \sum\limits_k e_k p_k + \sum\limits_i f_i q_i + \sum\limits_j g_j r_j \geq D$ is semantically inferred from I' and I''. We define I_0, I_1 by defining D_0 and D_1.

$$D_0 = \min\left\{\sum_i f_i q_i|_\gamma : \gamma \in \{0,1\}^{|\vec{q}|}, \gamma \text{ satisfies } I_0', I_0''\right\}$$

$$D_1 = \min\left\{\sum_j g_j r_j|_\tau : \tau \in \{0,1\}^{|\vec{r}|}, \tau \text{ satisfies } I_1', I_1''\right\}$$

It suffices to show that $D_0 + D_1 \geq D - \sum\limits_k e_k a_k$. For D_0, let the minimum be achieved at assignment γ_0, and for D_1, let the minimum be achieved at assignment τ_1. Let ρ be the assignment to the $\vec{q}$ and $\vec{r}$ variables setting $\vec{q}$ as in γ_0 and $\vec{r}$ as in τ_1. Then ρ satisfies I_0', I_0'', I_1', I_1'' (at $\vec{p} = \vec{a}$). Hence by induction, ρ satisfies I' and I''. Since I is inferred semantically from I' and I'', ρ satisfies I as well. Hence

$$D_0 + D_1 = \sum_i f_i q_i|_{\gamma_0} + \sum_j g_j r_j|_{\tau_1} = \left(\sum_i f_i q_i + \sum_j g_j r_j\right)|_\rho \geq D - \sum_k e_k a_k,$$

as required.

Since $\vec{p}$ appears only negatively in $B(\vec{p},\vec{r})$, D_1 is a non-decreasing function of D_1' and D_1''. (As the values of D_1' and D_1'' increase, the set of assignments τ over which we take the minimum shrinks, and so the minimum value can only increase or stay the same.) □

The proof of Theorem 10 goes through even if the quantified set of linear inequalities $\mathcal{F}$ are of the form defined in Theorem 3, not just those arising from false QBFs. Therefore similar to Theorem 3, semCP+$\forall$red also admits monotone real feasible interpolation for inequalities.

Using Theorem 10, we obtain another exponential lower bound for semCP+$\forall$red, analogous to Corollary 7.

Corollary 10. *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 5.3 require proofs of length exponential in $|\varphi_{n,k}|$ in* semCP+∀red.

Acknowledgements

The authors thank Rahul Santhanam and Nitin Saurabh for helpful discussions concerning decision lists. We also thank the anonymous referees for helpful comments.

This work was supported by the EU Marie Curie IRSES grant CORCON, grant no. 60842 from the John Templeton Foundation, EPSRC grant EP/L024233/1, and a Doctoral Prize Fellowship from EPSRC (2nd author).

A preliminary version including many of the results of this article has appeared in the proceedings of the conference FSTTCS'16 [12].

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.

[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 *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT'14)*, pages 154–169, 2014.

[5] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific Publishing, 2001.

[6] Marco Benedetti and Hratch Mangassarian. QBF-based formal verification: Experience and perspectives. *JSAT*, 5(1-4):133–191, 2008.

[7] Daniel Le Berre and Anne Parrain. The Sat4j library, release 2.2. *JSAT*, 7(2-3):59–6, 2010.

[8] Olaf Beyersdorff. On the correspondence between arithmetic theories and propositional proof systems – a survey. *Mathematical Logic Quarterly*, 55(2):116–137, 2009.

[9] 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.

[10] Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In *Proc. Symposium on Mathematical Foundations of Computer Science (MFCS'14)*, pages 81–93. Springer-Verlag, Berlin Heidelberg, 2014.

[11] 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.

[12] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for QBFs. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'16)*, pages 40:1–40:15, 2016.

[13] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not so simple. *ACM Transactions on Computational Logic*, 19(1):1:1–1:26, Dec 2017. (preliminary version in STACS 2016).

[14] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible Interpolation for QBF Resolution Calculi. *Logical Methods in Computer Science*, Volume 13, Issue 2, June 2017. (preliminary version in ICALP 2015).

[15] Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. *To appear in Journal of Computer and System Sciences*, 2018.

[16] Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017*, pages 14:1–14:15, 2017.

[17] Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In *Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS'16)*, 2016.

[18] A. Blake. *Canonical expressions in boolean algebra.* PhD thesis, University of Chicago, 1937.

[19] 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.

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

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

[22] Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, pages 94:1–94:14, 2016.

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

[24] Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity.* Cambridge University Press, 2010.

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

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

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

[28] Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. Conformant planning as a case study of incremental QBF solving. *Ann. Math. Artif. Intell.*, 80(1):21–45, 2017.

[29] Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'13)*, pages 291–308, 2013.

[30] Jan Elffers, Jess Girldez-Cru, Jakob Jakob Nordström, and Marc Vinyals. Using combinatorial benchmarks to probe the reasoning power of pseudoboolean solvers. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'18)*, pages 75–93, 2018.

[31] Yuval Filmus, Pavel Hrubeš, and Massimo Lauria. Semantic versus syntactic cutting planes. In *Symposium on Theoretical Aspects of Computer Science STACS'16*, pages 35:1–35:13, 2016.

[32] Andreas Goerdt. Cutting plane versus Frege proof systems. In *Proc. Workshop on Computer Science Logic (CSL'90)*, pages 174–194, 1990.

[33] R. E. Gomory. An algorithm for integer solutions to linear programs. In R. L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963.

[34] Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 546–553, 2011.

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

[36] J. Håstad. *Computational Limitations of Small Depth Circuits.* MIT Press, Cambridge, 1987.

[37] William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Comput. Syst. Sci.*, 65(4):695–716, 2002.

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

[39] Emil Jeřábek. *Weak pigeonhole principle, and randomized computation.* PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.

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

[41] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.

[42] Jan Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Cambridge University Press, 1995.

[43] 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.

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

[45] Mario Marchand and Mostefa Golea. On learning simple neural concepts: from halfspace intersections to neural decision lists. *Network: Computation in Neural Systems*, 4:67–85, 1993.

[46] Ruben Martins, Vasco M. Manquinho, and Inês Lynce. Open-WBO: A modular MaxSAT solver. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'14)*, pages 438–445, 2014.

[47] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[48] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

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

[50] Pavel Pudlák. The lengths of proofs. In Samuel R. Buss, editor, *Handbook of Proof Theory*, pages 547–637. Elsevier, Amsterdam, 1998.

[51] A. A. Razborov. Lower bounds on the size of bounded depth networks over a complete basis with logical addition. *Matematicheskie Zametki*, 41:598–607, 1987. In Russian. English translation in *Mathematical Notes of the Academy of Sciences of the USSR* 41:333–338, 1987.

[52] Jussi Rintanen. Asymptotically optimal encodings of conformant planning in QBF. In *Proc. AAAI Conference on Artificial Intelligence (AAAI'07)*, pages 1045–1050. AAAI Press, 2007.

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

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

[55] Olivier Roussel and Vasco M. Manquinho. Pseudo-boolean and cardinality constraints. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 695–733. IOS Press, 2009.

[56] A Schrijver. On cutting planes. *Annals of Discrete Mathematics*, 9:291–296, 1980.

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

[58] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proc. 19th ACM Symposium on Theory of Computing*, pages 77–82. ACM Press, 1987.

[59] György Turán and Farrokh Vatan. Linear decision lists and partitioning algorithms for the construction of neural networks. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, pages 414–423. Springer, 1997.

[60] Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Proc. Principles and Practice of Constraint Programming (CP'12)*, pages 647–663, 2012.

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