Electronic Colloquium on Computational Complexity, Report No. 178 (2018)

Hardness and Optimality in QBF Proof Systems Modulo NP

Leroy Chew*

School of Computing, University of Leeds, UK

Abstract. Quantified Boolean Formulas (QBFs) extend propositional formulas with Boolean quantifiers. Working with QBF differs from propositional logic in its quantifier handling, but as propositional satisfiability (SAT) is a sub-problem of QBF, all SAT hardness in solving and proof complexity transfers to QBF. This makes it difficult to analyse efforts dealing with "QBF hardness" specifically. In one response to this, Beyersdorff et. al. [18] created a framework for proof systems that allows us to factor out genuine QBF hardness from propositional hardness using oracles from the polynomial hierarchy.
Our work specifically deals with the most important case- when we use an NP oracle, removing any hardness due to propositional satisfiability in our proof systems.
The first results we present are re-examinations of the proof complexity landscape for QBF. Looking at the collapses of the QBF proof complexity simulation structure when including NP oracles. Some new simulations occur due to the NP oracles, in particular both universal resolution and weak extended variables are subsumed by the NP derivations. Some separations remain intact even with the NP-derivation rule such as the gap between expansion and QCDCL systems.
The NP-derivation rule allows us to say even more about QBF proof complexity. We show that QBF systems with general extension variables are incredibly powerful, and when adding in the NP-derivation rule, we can say something about its optimality and its use in practice. In Jussila et. al [31] Extended Q-Res was put forward as reasonable calculus for a universal proof checking of solver. We here justify that partially through theory. We find that, with the NP derivation rule, general extended QU-Res achieves an optimality result simulating all QBF proof systems as long as they have strategy extraction.

1 Introduction

Quantified Boolean formulas (QBF) extend propositional logic by adding Boolean quantification. Boolean existential and universal quantifiers are short-hands for expressions involving disjunction and conjunction respectively. QBF cannot be any more expressive than propositional logic but may be more succinct.

The question over its succinctness can be seen through the relation to complexity classes. While satisfiability and tautology in propositional logic is NP-complete and coNP-complete respectively [22], QBF has been proven to be PSPACE-complete [36]. Whether QBF provides a meaningful shorthand for very large satisfiability problems comes down to the unsolved NP vs PSPACE problem. QBFs have practical applications, including verification and planning. QBFs are also good at coding games including "Connect4" [25] and some problems in checkers [2] as well as popular video games [1].

QBFs are a superset of propositional satisfiability (SAT) problems. This means that when considering QBF lower bounds in solving or proofs, we also have to factor in all the propositional lower bounds. An example of this is in QBF proof complexity, QBF resolution incorporates new QBF rules into the propositional resolution system [20, 38]. However, these rules do nothing on purely propositional problems. If they were to in a meaningful way, it would in fact give rise to a new propositional system. In this way, every QBF proof system has an underlying propositional system, the same is true for solving.

* Supported by a Postdoctoral Prize Fellowship from EPSRC.

ISSN 1433-8092

The fact that hardness can be lifted from propositional logic can hinder our interpretation of hardness in QBF. One way to tackle this are the purely comparative studies in [5, 10, 24], where hardness is assessed relative to the other QBF proof systems. Another approach is to study QBF systems whose underlying propositional proof system has no known lower bounds, indeed while $\mathsf{AC}^0[p]$-Frege has no known lower bounds, it was shown [8] that the QBF system $\mathsf{AC}^0[p]$-$\mathsf{Frege} + \forall\mathsf{red}$, which extends it in a natural way, has a known lower bound. The final way from [18,21] which we explore further in this paper is to relax the notion of a proof system to allow the use of an NP oracle, factoring out the propositional component.

In [18], they present a way to augment a QBF proof system modulo NP by adding a rule that can do all propositional inference. In this work, we re-examine the QBF proof complexity whilst using this framework. Automatically it means that we cannot use purely propositional lower bounds as lower bounds to our system. It unfortunately means that lower techniques lifted from propositional logic to QBF (as in [12,13,15]) cannot be reliably used. There are two techniques that work specifically for QBF, strategy extraction and cost-capacity. Fortunately these also work under NP oracles.

We first look at some QBF lower bounds, now under the spotlight of NP oracles. Most of these have either been investigated prior to this or are simply corollaries of the fact that some techniques still work with NP oracles. We do show that Clique-CoClique formulas that were originally shown as hard for $\mathsf{QU\text{-}Res}$ are still hard under NP oracles. Originally hardness was shown via monotone feasible interpolation, which is a propositional technique lifted to QBF. We show it here again using the cost-capacity technique.

Next we look at the pairwise comparisons between the different QBF proof systems now with NP oracles. A simulation that existed prior to using NP oracles should still hold, as NP oracle rules should in most cases directly simulate each other. Where simulations were not known or known to be impossible there is a possibility that they can be shown here, in fact we even show some systems become equivalent under an NP oracle.

The final result we show is an optimality result. We show that among all QBF proof systems with strategy extraction, that using NP oracles with the Extended Q-Resolution proof system is optimal. Because we are using NP oracles and are only looking at proof systems with the strategy extraction property we do not really say anything about the major open problems of optimality. This may be relevant to solving, however. Strategy extraction is usually seen as helpful in solvers. Sometimes you do not just require the existence of a winning strategy, but the strategy itself. Indeed in a QBF representing a two-player game, it would be natural to want to know the winning strategy. Furthermore, the use of NP-oracle is natural, as QBFs regularly use SAT-solvers as black boxes. It makes the study of QBF proof systems with strategy extraction using NP oracles relevant to the state of practice.

2 Preliminaries

2.1 Proof Complexity.

Formally, a *proof system* [23] for a language L over alphabet Γ is a polynomial-time computable partial function $f : \Gamma^\star \rightharpoonup \Gamma^\star$ with $rng(f) = L$. The partial function f actually gives a proof checking function. Soundness is given by $rng(f) \subseteq L$ and completeness is given by $rng(f) \supseteq L$. The polynomial-time computability is an indication of feasibility, relying on the complexity notion that if something is in polynomial time it is considered feasible.

From the definition of a proof system, we can start defining proof size. For a proof system f for language L and string $x \in L$ we define $s_f(x) = \min(|w| : f(w) = x)$. Thus the partial function s_f tells us the minimum proof size of a theorem. We can overload the notation by setting $s_f(n) = \max(s_f(x) : |x| \leq n)$ where $n \in \mathbb{N}$. We do not focus on the exact numerical proof size, but how proof size behaves asymptotically. For a function $t : \mathbb{N} \to \mathbb{N}$, a proof system f is called *t-bounded* if $\forall n \in \mathbb{N}, s_f(n) \leq t(n)$.

A proof system f is *polynomially bounded* if there is a polynomial $p(x)$ such that $s_f(n) \leq p(n)$. Cook and Reckhow [23] proved that $\mathsf{NP} = \mathsf{coNP}$ if and only if there is a polynomially bounded proof system of propositional tautologies, where coNP is the class of all languages whose complements are in NP. A *super-polynomial lower bound* is an infinite family of formulas Φ_n where there is no polynomial p such that the shortest proof of each formula is $\leq p(|\Phi_n|)$. An *exponential lower bound* is an infinite family of formulas Φ_n where there is an exponential function $f = 2^{n^{\Omega(1)}}$ such that the shortest proof of each formula is $\geq f(|\Phi_n|)$.

Proof systems are compared by simulations. We say that a proof system f *simulates* g ($g \leq f$) if there exists a polynomial p such that for every g-proof π_g there is an f-proof π_f with $f(\pi_f) = g(\pi_g)$ and $|\pi_f| \leq p(|\pi_g|)$. If π_f can even be constructed from π_g in polynomial time, then we say that f *p-simulates* g ($g \leq_p f$). Two proof systems f and g are *(p-)equivalent* ($g \equiv_{(p)} f$) if they mutually (p-)simulate each other.

Definition 1 (Messner, Toran [35]). *A proof system in language $\mathcal{L}$ is* optimal *if and only if it can simulate all other proof systems for $\mathcal{L}$.*

It is not known if optimal propositional proof systems exist. Note that an optimal proof system does not necessarily imply that it is polynomially bounded, as a lower bound may be in a family of formulas that are not polynomial time recognisable.

2.2 Quantified Boolean Formulas

Quantified Boolean Formulas extend propositional logic with quantifiers $\forall, \exists$ [32]. The standard semantics are that $\forall x. \Psi$ is satisfied by the same truth assignments as $\Psi[0/x] \wedge \Psi[1/x]$ and $\exists x. \Psi$ is satisfied by the same truth assignments as $\Psi[0/x] \vee \Psi[1/x]$.

A prenex QBF is a QBF where all quantification is done outside of the propositional connectives. A prenex QBF Φ therefore consists of a propositional part ϕ called the matrix and a prefix of quantifiers Π and can be written as $\Phi = \Pi \cdot \phi$. A literal is a variable or its negation. A clause is disjunction of literals and a conjunctive normal form is a conjunction of clauses. When the propositional matrix of a prenex QBF is a CNF, then we have a PCNF.

If we have that each variable is bound only once then this causes no issues. We then can transform the matrix into a CNF using Tseitin variables, these Tseitin variables need to be quantified and the quantifier of the new variable must occur to the right of all variables they depend on. A prenex QBF without any variables in the prefix is just a propositional formula.

A closed QBF is a QBF where all variables are bound in quantifiers. A closed QBF must be either true or false, since if we semantically expand all the quantifiers we have a Boolean connective structure on $0, 1$.

QBF Game Semantics. Often it is useful to think of a closed prenex QBF $\mathcal{Q}_1 X_1 \ldots \mathcal{Q}_k X_k. \phi$, where X_i are blocks of variables, as a *game* between $\forall$ and $\exists$. In the i-th step of the game, the player $\mathcal{Q}_i$ assigns values to all the variables X_i. The existential player wins the game if and only

if the matrix ϕ evaluates to 1 under the assignment constructed in the game. The universal player wins if and only if the matrix ϕ evaluates to 0. Given a universal variable u with index i, a *strategy for* u is a function, which maps assignments of 0/1 values to the variables of lower index than u to $\{0, 1\}$ (the intended response for u). Therefore a QBF is false if and only if there exists a *winning strategy* for the universal player, i.e. if the universal player has a strategy for all universal variables that wins any possible game [26] [3, Sec. 4.2.2] [36, Chap. 19].

2.3 Proof Systems for Quantified Boolean Formulas

Resolution Systems. $\forall Exp{+}Res$ by Janota and Marques de Silva is a resolution-like refutation system that operates on QBFs in prenex form where the matrix is a CNF. It uses the basic idea behind the semantics of the quantifiers, allowing you to expand out a universal variable to create two copies of the matrix. Doing such would lead to an expansion explosion, however careful refutation could potentially only use the clauses it needs to get the contradiction, rather than the fully expanded matrix.

The key here is that variables have to be duplicated and distinguished, to keep track of this we label variables with an annotation and only resolve variables that have matching annotations. The calculus can be defined as in Figure 1.

$$\frac{}{\bigvee\left(\left\{l^{\mathsf{restrict}_l(\tau)} \mid l \in C, l \text{ is existential}\right\} \cup \{\tau(l) \mid l \in C, l \text{ is universal}\}\right)} \text{(Axiom)}$$

C is a clause from the matrix and τ is an assignment to all universal variables.
$\mathsf{restrict}_l(\tau)$ are partial assignments obtained by restricting τ to universal variables u with $\mathrm{lv}(u) < \mathrm{lv}(l)$.

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

Fig. 1. The rules of $\forall$Exp+Res (adapted from [30])

Q-resolution by Kleine Büning, Karpinski, and Flögel [33] is also a QBF resolution system, but is more straightforward. It uses the propositional resolution rule on existential variables. In addition, Q-resolution has a universal reduction rule to remove universal variables (for Figure 2 recall that $\neg\neg z = z$ for literals).

Example 2. We wish to refute the following PCNF in Q-Res:

$$\exists x_1 \exists x_2 \forall y_1 \forall y_2 \exists x_3 (x_1 \vee y_1 \vee x_3) \wedge (\neg x_1 \vee \neg y_1 \vee x_3) \wedge (\neg x_2 \vee y_2 \vee \neg x_3) \wedge (x_2 \vee \neg y_2 \vee \neg x_3)$$

This QBF can be shown as false by a semantic argument, using the two-player game. As long as the universal player plays $y_1 \leftarrow x_1$ and $y_2 \leftarrow \neg x_2$, then whatever the existential player sets for x_3 one clause gets refuted.

A refutation is given in Figure 3. Note that we cannot perform universal reduction on any of the axioms. We first can resolve the x_3 variables in every possible combination. The resulting clauses unblock universal literals which can then be reduced. Finally we can use resolution to reach the empty clause.

$$\frac{}{C}\ (\text{Ax}) \qquad\qquad \frac{C \vee x \qquad D \vee \neg x}{C \vee D}\ (\text{Res})$$

Ax: C is a clause in the propositional matrix.
Res: variable x is existential.

$$\frac{C \vee l}{C}\ (\forall\text{-Red})$$

literal l has variable u, which is universal and all other existential variables $x \in C$ are left of u in the quantifier prefix. Literal $\neg l$ does not appear in C.

Fig. 2. The rules of Q-Res [33]

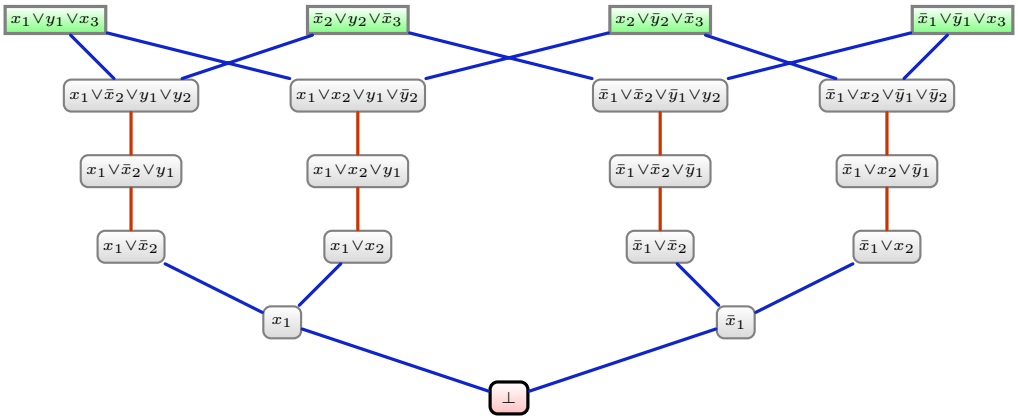


Fig. 3. An example of a Q-Res refutation

Q-Res has a number of augmentations. *Long-distance Q-resolution (LD-Q-Res)* appears originally in the work of Zhang and Malik [44] and was formalized into a calculus by Balabanov and Jiang [4]. It merges complementary literals of a universal variable u into the special literal u^*. These special literals behave like universal literals in that they can be reduced, but there are conditions on valid merging steps and further use of merged literals. We will not use this calculus in this paper, because it is unclear how to use propositional inference from the NP oracle when dealing with these merged literals.

QU-resolution (QU-Res) [43] removes the restriction from Q-Res that the resolved variable must be existential and instead allows resolution of universal variables.

Extended resolution for propositional resolution [41], enables adding clauses expressing the equality $v \Leftrightarrow (\neg x \vee \neg y)$, for a fresh variable v. We follow this idea in the context of Q-resolution. Here, we need to decide the position of the fresh variable in the prefix. Two versions are considered; a weak one and a general one. Both versions require extension variables to be existential, however they differ in their placement of the existential quantifier.

Figure 4 defines the two forms of the extension rule, which gives us two flavours of extended Q-resolution.

Weak extended Q-resolution [31] is the calculus of Q-Res enhanced with the extension rule in its weak form. Every extension variable appears at the end (innermost) of the prefix.

Extended Q-resolution is the calculus of Q-Res enhanced with the extension rule in general form. Each extension variable is quantified directly after the variables it is defined from.

$$\frac{}{(\neg x \vee \neg y \vee \neg v), (x \vee v), (y \vee v)}$$

where:
x, y are variables already in the formula
v is a fresh variable,
v is inserted into prefix as existentially quantified,
weak extension: insert v at the end of the prefix.
general extension: insert v after x and y in the prefix.

Fig. 4. Two versions of *extension rule*

Frege Systems. C-Frege + ∀red uses circuit lines we from the class C. It combines rules from Frege systems that operate on the circuit class C, with a generalised reduction. While Frege systems are inferential, because we are using reduction, C-Frege + ∀red is a refutational system.

$$\frac{}{C}\ (\text{Ax}) \qquad \frac{C_1, \dots \quad C_k}{D}\ (\text{C-Frege})$$

Ax: C is a circuit in the propositional matrix.
C-Frege: deriving circuit D from circuits $C_1, \dots C_k$ is compliant with an axiom or rule in C-Frege.

$$B \text{ is a C circuit in variables left of } u.\ \frac{C}{C[B/u]}\ (\forall\text{-Red})$$

Variable u is universal and all other variables $x \in C$ are left of u in the quantifier prefix.

Fig. 5. The rules of C-Frege + ∀red [8]

In practice we concentrate on a few special cases of C, particularly when C is AC^0 (bounded-depth), $\mathsf{AC}^0[p]$ (bounded depth with mod p gates), NC^1 (the standard Frege systems) or P/poly (circuit Frege, equivalent to eFrege).

NP Oracles. In the above QBF proof systems, we take a propositional proof system and augment it with some rules in order for it to deal with genuine QBFs. This approach is mostly unavoidable as every QBF proof system also is a propositional system. The drawback is that when observing lower bounds every propositional lower bound is inherited for QBFs. We would like to separate lower bounds from propositional logic with "genuine" QBF hardness.

Recent work [18, 21] has started to factor out the component of propositional hardness in QBF. Most work has been done on the QU-Res systems although this approach generalises to other systems as well.

Definition 3 (Σ_k^p Oracle derivations [18]). *A $P^{\Sigma_k^p}$ proof of a QBF Φ is a derivation of the empty clause by any of the P rules or the Σ_k^p-derivation rule.*

$$\frac{C_1, \dots \quad C_l}{D}\ (\Sigma_k^p\text{-derivation})$$

For any l, where there is some Σ_k^b-relaxation Π' of the prefix Π such that $\Pi' \bigwedge_{i=1}^{l} C_i \vDash \Pi' \bigwedge_{i=1}^{l} C_i \wedge D$. D and C_i have to be lines permitted in P (e.g. clauses, formulas).

We will not here define a Σ_k^b-relaxation for every k but for $k = 1$, we replace all universal quantifiers with existential ones. In other words, we can infer D from $\Pi \bigwedge_{i=1}^{l} C_i$ whenever $\bigwedge_{i=1}^{l} C_i \vDash D$. When we do add D we do not change the prefix Π. Hence P^{NP} augments QBF proof system P with all propositional inference.

Notice that P^{NP} is not a proof system unless we can check the NP-derivation in polynomial time. This cannot be done unless $\mathsf{P} = \mathsf{NP}$. However it gives us a framework for analysing QBF proof systems ignoring propositional hardness, which would otherwise be pervasive in QBF proof complexity. A similar approach was made previously in [21].

Definition 4. *Let P, Q be QBF proof systems, then we write $P \equiv^{\mathsf{NP}} Q$ whenever Q^{NP} and P^{NP} mutually p-simulate each other.*

3 Lower bounds for QBF proof systems modulo NP

The first thing we should note about QBF proof systems modulo NP is that any technique directly lifted from propositional logic no longer works in general. This comes from the fact that every propositional lower bound no longer exists when the NP derivation can be used. This means that feasible interpolation [12, 37], the Prover-Delayer game [15, 17] and the size-width relation [6, 13, 16] cannot be used in their current QBF forms. It may be possible that some sort of adaptation of these techniques can exist that deal only with QBF difficulty, indeed there are formulas that are genuinely QBF that make use of these techniques to show a lower bound. We will not focus on adapting these propositional techniques, instead will we focus on the highly successful techniques that are specific to QBF.

3.1 Strategy Extraction

Definition 5 (Strategy Extraction). *A refutational proof system P has* strategy extraction *if there is a polynomial time algorithm that takes P refutations π of QBF ϕ and outputs a circuit D_u for each universal variable u in prenex QBF ϕ, where the input variables of D_u are quantified to the left of u in ϕ and playing every u according to the output of D_u constitutes a winning strategy for the universal player.*

We look at the strategy extraction technique, using the circuit extracted from the proof. If that circuit is large then the proof must also be large. The technique depends on the proof systems having a strategy extraction property- that a circuit giving the winning strategy for the universal player can be efficiently extracted from the proof. For specific circuit class C, C-strategy extraction for a particular proof system P is the property that there is a polynomial time way to extract from a P-proof of a false QBF, a winning universal strategy in circuit class C for the relevant false QBF. For example, the QBF proof system $AC[p]$-$\mathsf{Frege} + \forall\mathsf{red}$ has $AC[p]$-strategy extraction [8]. Circuit lower bounds for $AC[p]$ can then be exploited to prove $AC[p]$-$\mathsf{Frege} + \forall\mathsf{red}$ proof-size lower bounds.

One circuit model that is very useful when dealing with strategy extraction is the decision list. Below we define the C-decision list for circuit class C.

Definition 6 (C-decision list). *A* C-decision list *is a program of the following form*

$$
\begin{array}{l}
\texttt{if } C_1(\boldsymbol{x}) \texttt{ then } u \leftarrow B_1(\boldsymbol{x}); \\
\quad \texttt{else if } C_2(\boldsymbol{x}) \texttt{ then } u \leftarrow B_2(\boldsymbol{x}); \\
\qquad \vdots \\
\quad \texttt{else if } C_{\ell-1}(\boldsymbol{x}) \texttt{ then } u \leftarrow B_{\ell-1}(\boldsymbol{x}); \\
\qquad \texttt{else } u \leftarrow B_\ell(\boldsymbol{x}),
\end{array}
$$

where $C_1, \ldots, C_{\ell-1}$ and $B_1, \ldots, B_\ell$ are circuits in the class C*. Hence a decision list as above computes a Boolean function $u = g(\boldsymbol{x})$.*

This comes from the original decision list where C_i is a term (conjunction of literals) and B_i is a Boolean constant. QU-Res has strategy extraction in these original depth-1 decision lists, while other QBF systems have strategy extraction in C-decision lists where C depends on the system. We find that the situation is very similar when we include NP derivations.

Theorem 7. *The following strategy extraction theorems hold for* NP *derivations:*

- *QU-Res* $^{\mathsf{NP}}$ *has depth-1 circuit decision list strategy extraction.*
- *CP+$\forall$red* $^{\mathsf{NP}}$ *has LTF-decision list strategy extraction.*
- *For any circuit class $\mathcal{C}$, $\mathcal{C}$-*Frege$+\forall$red $^{\mathsf{NP}}$ *has $\mathcal{C}$-decision list strategy extraction.*

Proof. We use a similar proof as from [8,10,14] except we account for the NP derivation rule exactly as how propositional rules were dealt with.

Let $\pi = (L_1, \ldots, L_s)$ be a refutation of the false QBF $\Pi\phi$ in one of these systems and let

$$\pi_i = \begin{cases} \emptyset & \text{if } i = s, \\ (L_{i+1}, \ldots, L_s) & \text{otherwise.} \end{cases}$$

We show, by downward induction on i, that from π_i it is possible to construct in linear time (w.r.t. $|\pi_i|$) a winning strategy σ^i for the universal player for the QBF formula $\Pi\phi_i$, where

$$\phi_i = \begin{cases} \phi & \text{if } i = 0, \\ \phi \wedge L_1 \wedge \cdots \wedge L_i & \text{otherwise,} \end{cases}$$

such that for each universal variable u in $\Pi\phi$, there exists an $\mathcal{C}$-decision list D^i_u computing σ^i_u as a function of the variables in $\mathcal{Q}$ left of u, having size $O(|\pi_i|)$. If we include every universal u, σ^i_u is part of a wider strategy σ^i.

The statement of this theorem corresponds to the case when $i = 0$. The base case of the induction is for $i = s$. In this case σ^s is trivial since ϕ_s contains the line $L_s = \bot$, and we can define all the D^s_u as $u \leftarrow 0$.

We show now how to construct σ^{i-1}_u and D^{i-1}_u from σ^i_u and D^i_u:

• If L_i is derived from an NP derivation rule, then for each universal variable u we set $\sigma^{i-1}_u = \sigma^i_u$ and $D^{i-1}_u = D^i_u$.

• If L_i is derived by some propositional rule, then for each universal variable u we set $\sigma^{i-1}_u = \sigma^i_u$ and $D^{i-1}_u = D^i_u$.

• If L_i is the result of an application of a $\forall$red rule, that is $\dfrac{L_j}{L_j[\alpha(u)]}$, where α is an assignment to the rightmost universal variable u in L_j. $L_j[\alpha(u)]$ is a circuit in $\mathcal{C}$ using only

variables on the left of u, and $L_j[\alpha(u)] = L_i$. Let $\boldsymbol{x}_v$ denote the variables with lower level than universal v in the quantifier prefix of $\mathcal{Q}\phi$. Then we define

$$\sigma_v^{i-1}(\boldsymbol{x}_v) = \begin{cases} \sigma_v^i(\boldsymbol{x}_v) & \text{if } v \neq u, \\ \alpha(v) & \text{if } v = u \text{ and } L_j[\alpha(u)](\boldsymbol{x}_u) = 0, \\ \sigma_v^i(\boldsymbol{x}_v) & \text{if } v = u \text{ and } L_j[\alpha(u)](\boldsymbol{x}_u) = 1. \end{cases}$$

Moreover we set $D_{u'}^{i-1} = D_{u'}^i$ and we set D_u^{i-1} as follows:

$$\begin{aligned} &\texttt{if } \neg L_j[\alpha](\boldsymbol{x}_u) \texttt{ then } u \leftarrow \alpha(u); \\ &\quad \texttt{else } D_i^u(\boldsymbol{x}_u). \end{aligned}$$

We now check that for each universal v, σ_v^{i-1} respects all the properties of the inductive claim.

By the argument in [8] σ_v^{i-1} and D_v^{i-1} are well defined and constructed in linear time w.r.t. $|\pi_{i-1}|$. D_v^{i-1} computes σ_v^{i-1}.

σ^{i-1} is a winning strategy for $\Pi\phi_{i-1}$. Fix an assignment ρ to the existential variables of ϕ. Let τ_i be the complete assignment to existential and universal variables, constructed in response to ρ under the strategy σ^i. By induction hypothesis τ_i falsifies ϕ_i. We need to show that τ_{i-1} falsifies ϕ_{i-1}. For propositional rules and $\forall$red it has already been argued in [8].

If L_i is derived by some NP derivation rule, then $\sigma^{i-1} = \sigma^i$ and $\tau_{i-1} = \tau_i$. Hence by induction hypothesis, τ_i falsifies a conjunct from ϕ_i. To argue that τ_{i-1} also falsifies a conjunct from ϕ_{i-1} we only need to look at the case when the falsified conjunct is L_i. As L_i is false under τ_i and L_i is derived by a sound NP derivation, one of the parent formulas of L_i in the application of the NP derivation rule is falsified as well. Hence τ_{i-1} falsifies ϕ_{i-1}.

We now have a strategy extracted from our proof. The theorem states that the circuit classes of these strategies depend on the proof system. This is because of the line structure of the proof systems, but this is not argued here (it is in [8] when dealing with reduction). In QU-Resolution the lines are depth-1 circuits, so the terms on the decision list are depth-1 circuits. Likewise in cutting planes the lines are all LTF circuits so the decision list has LTF terms. For $\mathcal{C}$-Frege systems the lines are circuits from the class $\mathcal{C}$, so the terms in the decision list are in $\mathcal{C}$. □

Example 8. Let $f(X)$ be a Boolean function in variables X which is computable in PSPACE. Due to the PSPACE completeness of QBF, $f(X)$ and $\neg f(X)$ can be written as a QBFs with free variables. The closed QBF Q-$f = \exists X \forall z(z \vee f(X)) \wedge (\neg z \vee f(X))$ is false and has the unique winning strategy of playing z as $f(X)$. The Q-f formulas where $f(X)$ is a Boolean function in PSPACE\C is a lower bound in any proof system with strategy extraction in class C. We can use any appropriate circuit lower bound.

1. Q-Parity is an exponential lower bound in QU-Res $^{\mathsf{NP}}$ and $\mathsf{AC}^0[p]$-Frege $\forall$red $^{\mathsf{NP}}$ (see [8]).
2. Q-IP is a superpolynomial lower bound in CP+$\forall$red $^{\mathsf{NP}}$ (see [14]).

3.2 Cost Capacity

A lower bound technique that is specific to QBF is the Cost-Capacity theorem from [7]. It works on $P + \forall$red systems. It shares some similarities to strategy extraction in the sense that it bounds the number of universal reductions from below and that the type of lower bounds

that work depend on the circuit class of the line being reduced. However it differs where strategy extraction cares about circuit hardness for the universal player, cost-capacity cares about the explicit number of varying responses a universal player has to make.

Definition 9 (Cost).
Let $\Phi = \forall U_1 \exists E_1 \ldots \forall U_k \exists E_k \phi$ be a QBF.
Let S_i be the strategy function that maps existential assignments from $\bigcup_{j=1}^{i-1} E_j$ to an assignment in block U_i for a winning strategy S.

$$\mathit{cost}(\Phi) = \min(\max(\mathit{rng}(S_i) : i \in [k]) : S \text{ is a winning strategy})$$

A response set is a set of all assignments to the universal variables in a line that is consistent with some winning strategy.

Definition 10 (Response Map). *Let L be a propositional proof line and Π be QBF prefix, and U defines the set of universal variables from Π that appear in L that are quantified rightmost amongst the variables in L (U variables are in the same block, meaning they have equal quantifier level). And let X denote the set of Π variables in L not in U. We use the notation 2^X and 2^U to denote the sets of total assignments on X and U respectively. A response map $R : 2^X \to 2^U$ on a line L under prefix Π satisfies the following, for $\alpha \in 2^X$*

$$L(\alpha) \text{ not a tautology} \Rightarrow L(\alpha)(R(\alpha)) \text{ is false.}$$

The range of a response map, measures the number of potential falsifying universal assignments. This gets formalised in the notion of capacity.

Definition 11 (Capacity). *Let π be a refutation of QBF with prefix Π and matrix ϕ in a proof system $\mathsf{P}+\forall\mathsf{red}$ with lines $L_1 \ldots L_m$.*

$$\mathit{capacity}(\pi) = \max(\min(\mathsf{rng}(R) : R \text{ is a response map of } L_i) : i \in [m])$$

The capacity for a $\mathsf{QU\text{-}Res}$ derivation turns out to be 1, because the number of useful universal assignments in each clause is 1, in a refutation you only ever wish to falsify a literal.

Theorem 12 (Beyersdorff, Blinkhorn, Hinde 18 [7]). *For a proof system $\mathsf{P}+\forall\mathsf{red}$ which has axiomatic equivalence, inferential equivalence and restrictive closure (see [7] for an explanation). Then for a $\mathsf{P}+\forall\mathsf{red}^{\mathsf{NP}}$ refutation π for QBF Φ we have that $|\pi| \geq \frac{\mathit{cost}(\Phi)}{\mathit{capacity}(\pi)}$.*

In previous work [12,13,15,16] lower bound techniques from propositional logic are lifted to QBF. These cannot translate when NP oracles are in place. In [12] a QBF version of Feasible Interpolation was shown to work on QBF resolution systems. The feasible interpolation property fails when the rules allow inferences that can subsume Frege rules (which does not have feasible interpolation). However the lower bound using the Clique-CoClique formulas would now be a conditional lower bound due to strategy extraction.

Definition 13 (Clique-CoClique family).
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 $\boldsymbol{p}$ be the set of variables $\{p_{uv} \mid 1 \leq u < v \leq n\}$. An assignment to $\boldsymbol{p}$ picks a set of edges, and thus an n-vertex graph that we denote $G_{\boldsymbol{p}}$.

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

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

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

$A_{n,k}(\boldsymbol{p},\boldsymbol{q})$ is the conjunction of these clauses.

Let $\boldsymbol{r}$ be the set of variables $\{r_{iu} \mid i \in [k], u \in [n]\}$. We use the following clauses Let $\boldsymbol{t}$ be the set of variables $\{t_K \mid K \in A_n, k(p,q)\}$

$$\begin{array}{rll} K^t_{C_i} & = \neg t_{C_i} \vee r_{i1} \vee \cdots \vee r_{in} & \text{for } i \in [k] \\ K^{r_{ij}}_{C_i} & = t_{C_i} \vee \neg r_{ij} & \text{for } i \in [k], j \in [n] \\ K^t_{D_{i,j,u}} & = \neg t_{D_{i,j,u}} \vee \neg r_{iu} \vee \neg r_{ju} & \text{for } i,j \in [k], i < j \text{ and } u \in [n] \\ K^{r_{iu}}_{D_{i,j,u}} & = t_{D_{i,j,u}} \vee r_{iu} & \text{for } i,j \in [k], i < j \text{ and } u \in [n] \\ K^{r_{ju}}_{D_{i,j,u}} & = t_{D_{i,j,u}} \vee r_{ju} & \text{for } i,j \in [k], i < j \text{ and } u \in [n] \\ K^t_{E_{i,u,v}} & = \neg t_{E_{i,u,v}} \vee \neg r_{iu} \vee \neg r_{iv} & \text{for } i \in [k] \text{ and } u,v \in [n], u < v \\ K^{r_{iu}}_{E_{i,u,v}} & = t_{E_{i,u,v}} \vee r_{iu} & \text{for } i \in [k] \text{ and } u,v \in [n], u < v \\ K^{r_{iv}}_{E_{i,u,v}} & = t_{E_{i,u,v}} \vee r_{iv} & \text{for } i \in [k] \text{ and } u,v \in [n], u < v \\ K^t_{F_{i,j,u,v}} & = \neg t_{F_{i,j,u,v}} \vee \neg r_{iu} \vee \neg r_{jv} \vee p_{uv} & \text{for } i,j \in [k], i < j \quad \text{and} \quad u \neq v \in [n]. \\ K^{r_{iu}}_{F_{i,j,u,v}} & = t_{F_{i,j,u,v}} \vee r_{iu} & \text{for } i,j \in [k], i < j \quad \text{and} \quad u \neq v \in [n]. \\ K^{r_{jv}}_{F_{i,j,u,v}} & = t_{F_{i,j,u,v}} \vee r_{jv} & \text{for } i,j \in [k], i < j \quad \text{and} \quad u \neq v \in [n]. \\ K^{p_{uv}}_{F_{i,j,u,v}} & = t_{F_{i,j,u,v}} \vee \neg p_{uv} & \text{for } i,j \in [k], i < j \quad \text{and} \quad u \neq v \in [n]. \\ K_{clauses} & = \bigvee_{i \in [k]} \neg t_{C_i} \vee \bigvee^{i<j}_{i,j \in [k], u \in [n]} \neg t_{D_{i,j,u}} \vee \bigvee^{u<v}_{i \in [k], u,v \in [n]} \neg t_{E_{i,u,v}} \vee \bigvee^{i<j, u \neq v}_{i,j \in [k], u,v \in [n]} \neg t_{F_{i,j,u,v}} & \end{array}$$

$B_{n,k}(\boldsymbol{p},\boldsymbol{r})$ is the conjunction of these clauses.

The QBF CliquecoClique(n,k) *is given by $\exists\boldsymbol{p}\exists\boldsymbol{q}\forall\boldsymbol{r}\exists\boldsymbol{t} A_{n,k}(\boldsymbol{p},\boldsymbol{q}) \wedge B_{n,k}(\boldsymbol{p},\boldsymbol{r})$*

As feasible interpolation can be used to show propositional lower bounds it cannot work in a QBF setting where we have NP derivations. In [12] it was discussed that feasible interpolation can be seen as a special case of strategy extraction though we have to add a universal variable. If we are concerned only with the original formula we might again think that strategy extraction might be a good approach, indeed universal R variables must compute a clique if one exists. It would seem that coupled with the fact that k-clique problems have large bounded depth circuits in the P variables [39] we would get a lower bound for Q-Res$^{\mathsf{NP}}$ via strategy extraction, except for the fact that the way we write the Clique-coClique formulas quantify the Q variables before the R variables. This means that one winning strategy for the universal player is fairly straightforward, play the R variables exactly the same as the corresponding Q variables. Strangely enough this strategy is only an artefact of the way we chose to order the two parts. Were we to quantify the other way round (let us call it the coClique-Clique formulas), we can get the lower bound via strategy extraction. This is peculiar as the order was never important for the feasible interpolation lower bound. This still leaves the question of whether the original Clique-coClique formulas are hard, this can be done using the other technique that works for NP derivation QBF systems- the cost-capacity theorem.

Theorem 14. CLIQUE-coCLIQUE *QBFs are exponentially hard for* QU-Res$^{\mathsf{NP}}$.

Proof. We use the cost-capacity theorem. The capacity of QU-Res and CP+$\forall$red is 1 and this does not change under an NP derivation [7]. We now have to argue for cost, we argue the cost is above a certain threshold by looking at a selection of important assignments. These are assignments to P that details the edges of a clique and where Q correctly identifies a clique. We define this formally.

$V(H)$ is a subset of $[n]$ of size k. H is a complete graph. $G_{\boldsymbol{p}}$ is the graph on n vertices such that $u\ v$ if and only if $u, v \in V(H)$. Hence

$$p_{uv} = \begin{cases} 1 & u, v \in V(H), \\ 0 & \text{otherwise.} \end{cases}$$

We define some bijection $\sigma_H : [k] \longleftrightarrow V(H)$.

$$q_{i,u} = \begin{cases} 1 & \sigma_H(i) = u, \\ 0 & \text{otherwise.} \end{cases}$$

This will satisfy all the clauses in $A_{n,k}$: $E_{i,u,v}$ will be satisfied because σ_H is a well-defined partial function. C_i will be satisfied because σ_H is a total function. $D_{i,j,u}$ will be satisfied because σ_H is an injection. $F_{i,j,u,v}$ will be satisfied by p_{uv} when $u, v \in V(H)$. And when , without loss of generality u is not in $V(H)$, σ_H is bijective with $V(H)$ so q_{iu} must be false, satisfying the clause.

Under this assignment the clauses that can be falsified appear in $B_{n,k}$. These also depend of the universal $\boldsymbol{r}$ variables and the existential $\boldsymbol{t}$ variables after it in the prefix. In general the existential player can make any response in $\boldsymbol{t}$, but since every clause in $B_{n,k}$ except K_{clauses} is part of a Tseitin definition, the existential player satisfies all of $B_{n,k} \backslash \{K_{\text{clauses}}\}$ if and only if they play all $\boldsymbol{t}$ variables according to their definitions. We will assume that the existential player will play according to this strategy.

We have to show that in order to falsify K_{clauses}, the universal player requires a particular kind of response in $\boldsymbol{r}$.

Suppose first that for some $i \in [k]$ the universal player plays $r_{iu} = r_{iv} = 1$ for some $u < v$. Then the existential player's response would be to set $t_{E_{i,u,v}}$ to 0 satisfying K_{clauses}.

Hence for a falsifying universal response we can define a **partial function** $\tau_H : [k] \rightharpoonup [n]$ such that

$$r_{iu} = \begin{cases} 1 & \tau_H(i) = u \\ 0 & \text{otherwise} \end{cases}$$

τ_H is a total function: Suppose there is some $i \in [k]$ such that $\tau_H(i) \neq 1 \ldots \tau_H(i) \neq n$. Then $\neg r_{i1} \ldots \neg r_{in}$ is played by the universal player. This means that t_{C_i} is false and thus satisfies K_{clauses}.

τ_H is an injection: Suppose that for some $u \in [n]$, $u = \tau_H(i) = \tau_H(j)$ for some $i < j$. That means the universal player plays $r_{iu} = r_{ju} = 1$. Then the existential players response would be to set $t_{D_{i,j,u}}$ to 0 satisfying K_{clauses}.

$\tau_H : [n] \longleftrightarrow V(H)$ is a bijection: Suppose that there is some $w \in V(H)$ such that no value in $[k]$ maps to w under τ_H, then by injectivity there is some $u \notin V(H)$ such that there is some $i \in [k]$ so that $\tau_H(i) = u$. Let us take any $j \in [k], j \neq i$. $\tau_H(j) = v$ by totality and since

$u \notin V(H)$ p_{uv} must be false. Without loss of generality we assume $i < j$. In that case r_{iu} and r_{jv} are true so $t_{F_{i,j,u,v}}$ is set to false by the existential player satisfying K_{clauses}.

We have shown each isolated clique H requires a response determined by some τ_H. $\tau_H = \tau_{H'}$ is impossible if $V(H) \neq V(H')$. This is because if $\tau_H = \tau_{H'}$ then $V(H) = \mathsf{rng}(\tau_H) = \mathsf{rng}(\tau_{H'}) = V(H')$. Hence the universal player needs $\binom{n}{k}$ responses indicating the cost and indeed the proof size is at least exponential when $k = \frac{n}{2}$. □

3.3 Semantic Lower Bounds

∀Exp+Res gives us another approach. We can show that the total number of instances of the axiom rule we need is exponential. Specifically this can be done in this system because axioms can be instantiated in exponentially-many different ways. This was originally used in [30]. This means, even with an NP derivation rule ∀Exp+Res still has the semantic lower bound from [30] which is easy for Q-Res. We can use this technique for Clique-coClique.

Theorem 15. *The* Clique-coClique *QBFs are exponentially hard for* ∀*Exp+Res* $^{\mathsf{NP}}$.

Proof. We expand our CNF in our universal variables. To argue for the lower bound we must argue that at least $\binom{n}{k}$ many clauses are required to even be unsatisfiable, hence a ∀Exp+Res $^{\mathsf{NP}}$ proof would still require exponentially many lines.

Like before we concentrate only on counting the $B(n, k)$ clauses. Similarly to Theorem 14 we only look at the models that assign p_{uv} according to some clique H and set q_{iu} to order the vertices of the clique.

For each of these cliques assignments to $\boldsymbol{p}$ and $\boldsymbol{q}$, we require an assignment α in $\boldsymbol{r}$ that makes the formula unsatisfiable in the remaining $\boldsymbol{t}^{\alpha}$ variables. This, as argued before in the proof of Theorem 14, must correspond to a bijection between our k clique slot values to the vertices of H. We require at least one set of clauses with some annotation corresponding to one of these bijection. For any two different cliques H, H' we need two different bijections and hence two different annotations. Therefore our unsatisfiable expanded CNF requires at least $\binom{n}{k}$ different annotations.

□

4 Simulations and equivalences

We can use the power of NP oracles to give new simulations. In these cases we are showing that any previously existing separations were contingent on propositional hardness. We look at Resolution-based systems, and systems with extension variables.

4.1 Resolution

Theorem 16. *Tree-like* ∀*Exp+Res* $\equiv^{\mathsf{NP}}$ ∀*Exp+ Ext. Res.*

Proof. ($\leq$) ∀Exp+ Ext. Res has all the rules of ∀Exp+Res.

($\geq$) Suppose FQBF ϕ has a ∀Exp+Ext. Res$^{\mathsf{NP}}$ refutation π. We take all axiom clauses that are used in π. Our refutation in ∀Exp+Res $^{\mathsf{NP}}$ uses these same axioms and a single instance of an NP derivation on all the axioms to derive the empty clause. □

Theorem 17. *Q-Res* $\equiv^{\mathsf{NP}}$ *QU-Res.*

Proof. ($\leq$) QU-Res $^{\mathsf{NP}}$ has every rule from Q-Res $^{\mathsf{NP}}$.

($\geq$) The NP derivation rule derives a propositional implicant from a finite number of clauses. This means it subsumes the resolution rule for universal variables. □

Though ∀Exp+Res $^{\mathsf{NP}}$ cannot simulate Q-Res, the reverse is not possible either.

Theorem 18. *For odd prime p, tree-like* $\forall Exp{+}Res \not\leq \mathsf{AC}^0[p]\mathsf{Frege} + \forall\mathsf{red}^{\mathsf{NP}}$.

Proof. We use the family of formulas QPARITY$_n$ from [10]. These false formulas each have a unique winning strategy for the universal player namely the parity function on n variables. Because the parity function is asymptotically an exponential lower bound [28, 40] in bounded depth circuits with odd mod p gates and we have strategy extraction from Theorem 7, we know this gives an exponential lower bound to $\mathsf{AC}^0[p]\mathsf{Frege} + \forall\mathsf{red}^{\mathsf{NP}}$. For the short proofs in tree-like ∀Exp+Res see Theorem 2 in [9]. □

4.2 Cutting planes

Theorem 19. *CP+∀red* $^{\mathsf{NP}}$ *does not simulate* $\mathsf{Frege} + \forall\mathsf{red}$.

Proof. We use the $\mathcal{Q}$-IP formulas from Corollary 10 in [14], these formulas express that the inner product of two bit-vectors cannot be both odd and even. In [42] 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+∀red proof for $\mathcal{Q}$-IP$_n$ must have length greater than $2^{n/2} - 1$ due to Theorem 7. □

4.3 Weak Extension Variables

We will find that between Q-Res, QU-Res, Weak Ext. Q-Res and Weak Ext QU-Res there are only the five trivial simulations and the rest are separations. However by using NP derivations we find that all these separations rely on propositional hardness.

The separation between Q-Res and QU-Res comes from the formulas from Kleine Büning, Karpinski and Flögel [33,43]. QU-Res cannot simulate weak extended Q-Res due to propositional lower bounds like the pigeonhole principle [27], we are only left to show one more separation and we get the complete picture.

Proposition 20. *Weak extended Q-Res does not simulate QU-Res.*

We can use the alternative proof from [7] to the lower bound of the formulas from Kleine Büning, Karpinski and Flögel [33] for Q-Res.

Lemma 21. *Weak extended Q-Res and weak extended QU-Res have capacity* 1.

Proof. All reduction steps in weak extended Q-Res and weak extended QU-Res are reduction steps that do not include extension variables, as these variables must appear existentially at the end of the prefix. The minimum response map in a clause is to refute the universal literals, no matter the input. Since our output is 1 assignment, the capacity here is 1. □

KBKF(t) has prefix $\exists d_1, e_1 \forall x_1 \exists d_2, e_2 \forall x_2 \dots \exists d_t, e_t \forall x_t \exists f_1 \dots f_t$ and matrix clauses

$$
\begin{array}{llll}
C_0 = \neg d_1 \vee \neg e_1 & & & \\
D_i = d_i \vee x_i \vee \neg d_{i+1} \vee \neg e_{i+1} & E_i = e_i \vee \neg x_i \vee \neg d_{i+1} \vee \neg e_{i+1} & & \text{for } i \in [t-1], \\
D_t = d_t \vee x_t \vee \neg f_1 \vee \dots \vee \neg f_t & E_t = d_t \vee \neg x_t \vee \neg f_1 \vee \dots \vee \neg f_t & & \\
F_i^0 = x_i \vee f_i & F_i^1 = \neg x_i \vee f_i & & \text{for } i \in [t].
\end{array}
$$

$\mathrm{KBKF}_{\mathrm{u}}(t)$ [5] has prefix $\exists d_1, e_1 \forall x_1, z_1 \exists d_2, e_2 \forall x_2, z_2 \ldots \exists d_t, e_t \forall x_t, z_t \exists f_1 \ldots f_t$ and matrix clauses

$$
\begin{array}{llll}
C_0 = \neg d_1 \vee \neg e_1 & & & \\
D_i = d_i \vee x_i \vee z_i \vee \neg d_{i+1} \vee \neg e_{i+1} & E_i = e_i \vee \neg x_i \vee \neg z_i \vee \neg d_{i+1} \vee \neg e_{i+1} & & \text{for } i \in [t-1] \\
D_t = d_t \vee x_t \vee z_t \vee \neg f_1 \vee \cdots \vee \neg f_t & E_t = d_t \vee \neg x_t \vee \neg z_t \vee \neg f_1 \vee \cdots \vee \neg f_t & & \\
F_i^0 = x_i \vee z_i \vee f_i & F_i^1 = \neg x_i \vee \neg z_i \vee f_i & & \text{for } i \in [t].
\end{array}
$$

Finally the QBF $\lambda'(t)$ [7] has the same clauses as $\mathrm{KBKF}_{\mathrm{u}}(t)$ but its prefix is

$$\exists d_1, e_1 \forall x_1 \exists d_2, e_2 \forall x_2 \ldots \exists d_t, e_t \forall x_t \forall z_1 \ldots z_t \exists f_1 \ldots f_t.$$

We first show that Weak extended QU-Res requires an exponential refutations for $\mathrm{KBKF}_{\mathrm{u}}$ using the cost-capacity method. We then argue that any proof in Weak extended Q-Res of KBKF can give a similar size proof in QU-Res. Thus Weak extended Q-Res requires exponential lower bounds for KBKFand thus cannot simulate QU-Res.

Lemma 22. *Weak extended QU-Res requires 2^t-size refutations for $\lambda'(t)$.*

Proof (Sketch Proof). The easiest way to make this argument is to recall that the extension variables are all quantified to the right of all the z_i variables and thus have not come into play in the two player game until after the z_i variables have been dealt with. In addition, the extension clauses need not to be considered by the universal player, the existential player can always satisfy them by playing the extension variables consistent to the functions they represent. This means the cost-capacity argument from [7] still works in much the same way.

Consider the two-player game on the $\lambda'(t)$. Suppose the existential player has played $d_j \neq e_j$ for every $j \leq k$. Now suppose the universal player sets $x_k \neq e_k$ then D_k, E_k are satisfied, the existential player can satisfy the remaining D_i, E_i, for $i > k$ by setting both d_i and e_i to 1 and satisfy F_i^c by setting all the f_i variables positive, all that remains is to set the extension variables accordingly.

We consider A the set of assignments α to existential variables $\{d_1, e_1 \ldots d_n, e_n\}$ such that $\alpha(d_k) \neq \alpha(e_k)$. We will show the only winning strategy is to play $z_k = x_k = e_k$.

We have already argued that the universal player will set $x_k = e_k$, we now have to argue that z_k must be set to x_k. If some $x_k \neq z_k$ then f_k can be set to 0 by the existential player, while satisfying both F_k^0 and F_k^1. This means both D_t and E_t can be satisfied by $\neg f_k$ and all other f_i, $i \neq k$ variables can be set to 1 satisfying all F_k^1 and F_k^0.

Thus we show the cost for this block of z_k variables is 2^t. By the Cost-Capacity theorem we show an exponential lower bound on the number of reduction steps. □

Lemma 23. *Weak extended QU-Res requires 2^t-size refutations for* $\mathrm{KBKF}_{\mathrm{u}}(t)$.

Proof. $\lambda'(t)$ is a relaxation of $\mathrm{KBKF}_{\mathrm{u}}(t)$. Therefore if weak extended QU-Res requires $2^{O(t)}$ size refutations of $\lambda'(t)$, then it also requires $2^{O(t)}$ size refutations of $\lambda'(t)$. □

Lemma 24. *Weak extended Q-Res refutations of* $\mathrm{KBKF}(t)$ *can be transformed into weak extended QU-Res refutations for* $\mathrm{KBKF}_{\mathrm{u}}(t)$ *in linear time.*

The clauses in our $\text{KBKF}_\text{u}(t)$ refutation will be the same except for an additional z_i literal. We keep an invariant in our constructed weak extended QU-Res proof that whenever variables x_i and z_i appear in the same clause they appear in the same polarity. This will allow us to prove that the resolution steps we perform in QU-Res to remain valid whenever they were valid in the Q-Res proofs.

Proof. Each extension step is replicated exactly, every extension clause in the original proof is in the new proof. Each axiom clause in $\text{KBKF}(t)$ is replaced by the corresponding clause in $\text{KBKF}_\text{u}(t)$ which adds z_i literals whenever x_i literals appear and $\neg z_i$ literals whenever $\neg x_i$ literals appear. Any reduction step that appears in $\text{KBKF}(t)$ that reduces x_i is copied exactly except it will also reduce z_i in the same polarity as x_i (if it exists). Any resolution step is copied with the same existential pivot. We actually will not need to use universal resolution steps. In order for these steps to be valid we need to ensure there are no conflicting universal literals. Inductively, we show that $z_i, \neg z_i$ literals only appear in clauses where $x_i, \neg x_i$ literals appear, respectively and all other literals are the same as in the original proof. This is true in our axioms steps, in our extension clauses $z_i, \neg z_i$ literals do not appear. In reduction steps we remove both x_i and z_i variables simultaneously. Finally in resolution steps, since we do not resolve on universal variables we keep the invariant. Since x_i and $\neg x_i$ literals were present in the original proof these would not cause a conflict and neither will $z_i, \neg z_i$. This means that resolution steps are valid in $\text{KBKF}_\text{u}(t)$ whenever they were before since all clauses will just be the same apart from an extra universal literal. This also means when we arrive at the empty clause, we at most have an extra z_i (or $\neg z_i$) literal, however this would have been reduced with x_i (or $\neg x_i$). Hence we refute $\text{KBKF}_\text{u}(t)$ in the same number of lines. □

The corollary of this is Proposition 20. Thus we prove the following complete simulation structure in Figure 6. The opposite, however, is true when using NP derivations

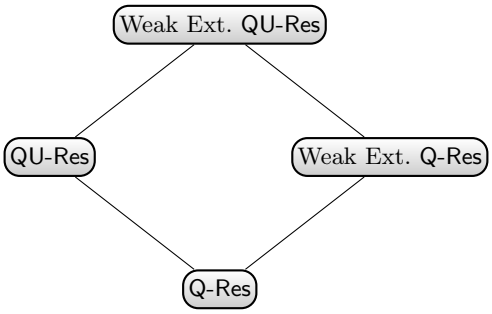


Fig. 6. The simulation structure of four variants of Q-Res, all pairwise simulations are given and are strictly one-way and other pairs do not yield a simulation

Theorem 25. *Q-Res* $\equiv^{\text{NP}}$ *QU-Res* $\equiv^{\text{NP}}$ *Weak Ext.Q-Res* $\equiv^{\text{NP}}$ *Weak Ext.QU-Res.*

Proof. Q-Res $\equiv^{\text{NP}}$ QU-Res and Weak Ext.Q-Res $\equiv^{\text{NP}}$ Weak Ext.QU-Res because NP derivations can be used to simulate universal resolution steps directly. We are left to show Q-Res $\equiv^{\text{NP}}$ Weak Ext.Q-Res, in other words, that we can simulate what we can do with weak extension using NP derivations.

The first observation is that in every universal reduction step in Weak Ext. Q-Res has no extension variables, since these would always be quantified to the right of every universal variable (and thus block their reduction). This means the first lines we perform universal reduction on are just propositional implications of axioms. Likewise any later lines we perform universal reduction on are propositional implications of the axioms plus the clauses that result from universal reduction (which are not inferred propositionally). So what we can do in Q-Res$^{\mathsf{NP}}$ to simulate Weak Ext. Q-Res$^{\mathsf{NP}}$ proofs is to use NP derivations to get to the lines that need universal reduction and then just do universal reduction on these clauses and continue to alternate between NP derivations steps and universal reduction steps. □

5 Optimality in QBF proof systems

In the previous sections we have examined the simulation structure of QBF proof systems under NP oracles, this complements previous work [5, 10, 11, 24, 30, 43] which undertook the same task without NP derivations. In this section we undertake a task that is more specific to the NP derivation situation- optimality.

Optimality has proven difficult to show in propositional proof complexity, so factoring out propositional difficulty via the NP derivation might help us gain optimality results. In this work we do not show an optimal QBF proof system, instead we look at all QBF proof systems with strategy extraction and find something optimal among them.

We first prove the following, which does not rely on NP oracles:

Theorem 26. *For every QBF Proof System P that has* P/poly*-strategy extraction there is a set of polynomial-time verifiable propositional tautologies $\|\Phi\|$ such that* eFrege $+\forall$red $+\|\Phi\|$ *p-simulates P.*

Proof. Let P be our TQBF proof system. Let $\Pi\phi$ be a closed QBF where Π is a quantifier prefix and ϕ is purely propositional. The strategy extraction means that from a proof π of QBF we can extract in polynomial-time circuits σ_e that are Skolem functions for each existential variable e. Let $\phi_{\sigma,\Pi}$ be the propositional formula that results from replacing every existential variable e with σ_e in ϕ. Since the strategy is correct, $(\phi)_{\sigma,\Pi}$ must be a propositional tautology.

We can use this observation to design a propositional proof system $Sko(P)$. Using the Cook-Reckhow definition of a proof system as a checking function we define it as follows:

$$Sko(P)(\pi) = \begin{cases} \phi_{\sigma,\Pi}, & \pi \text{ is a } P \text{ proof of } \Pi\phi \text{ and } \sigma \text{ is the strategy extracted from it,} \\ \mathsf{eFrege}(\pi), & \text{otherwise.} \end{cases}$$

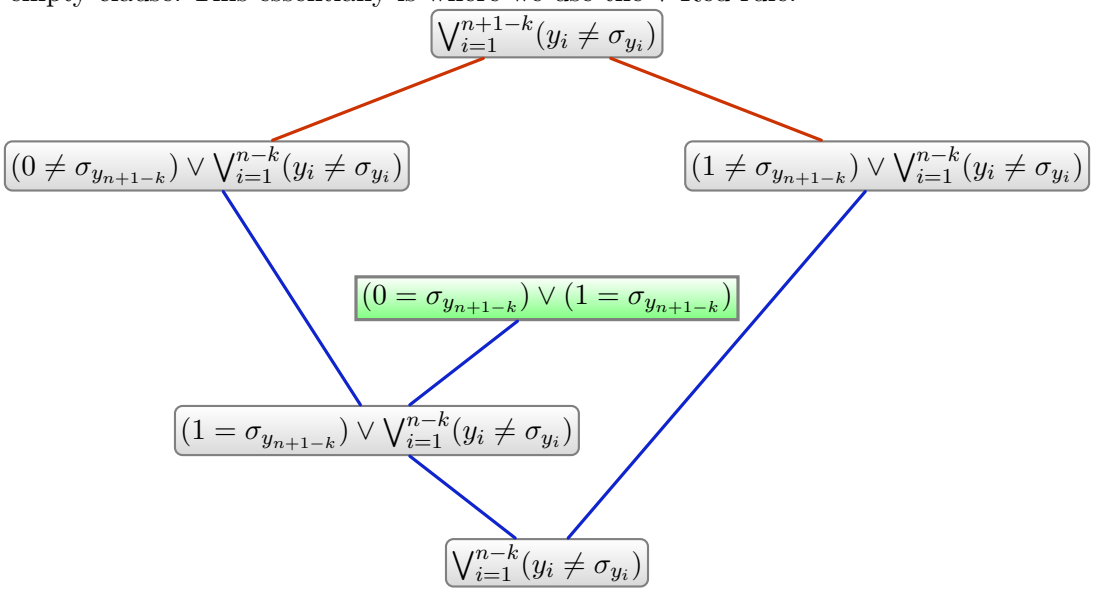
$Sko(P)$ is simulated by eFrege $+\|\text{refl}(Sko(P))\|$ [34], where $\|\text{refl}(Sko(P))\|$ is a polynomial-time recognition set of propositions that encode an arithmetic statement of the correctness of $Sko(P)$. We show that eFrege $+\forall$red $+\|\text{refl}(Sko(P))\|$ simulates P, so we let π be a proof of $\Pi\phi$ in P with strategy extracted σ. Note that π is also a $Sko(P)$ proof.

We let π'_1 be the eFrege $+\|\text{refl}(Sko(P))\|$ proof that simulates π in $Sko(P)$. We know this can be of polynomial size in π. Likewise as we know the σ_y are polynomial size, this means that the circuit $\neg\phi \wedge \phi_{\sigma,\Pi} \rightarrow \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ has a polynomial size proof π'_2, where y_i are the existential variables in Π in order (y_n being the innermost existential variable). π'_2 simply involves the eFrege proof that when the substitutions $\bigwedge_{i=1}^{n}(y_i = \sigma_{y_i})$ are assumed then ϕ and $\phi_{\sigma,\Pi}$ are interchangeable.

In order to show eFrege $+\forall$red $+\|\text{refl}(Sko(P))\|$ can prove $\Pi\phi$, we refute $\bar{\Pi}\neg\phi$ (where $\bar{\Pi}$ swaps $\exists$ and $\forall$ quantifiers) since eFrege $+\forall$red $+\|\text{refl}(Sko(P))\|$ is a refutational system.

$$\dfrac{\dfrac{\neg\phi \qquad \phi_{\sigma,\Pi}}{\neg\phi \wedge \phi_{\sigma,\Pi}} \qquad \neg\phi \wedge \phi_{\sigma,\Pi} \to \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})}{\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})}$$

We show an inductive proof of $\bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$ for increasing k eventually leaving us with the empty clause. This essentially is where we use the $\forall$-Red rule.



□

This result tells us how important eFrege + $\forall$red is, although it is conditional on adding the propositional tautologies of $\|\text{refl}(Sko(P))\|$ if they cannot already be derived in eFrege. We are now going to remove this condition and replace it with an NP-oracle rule, however first we will do something to make this result stronger, we show that eFrege (the circuit version) and Extended QU-Res are equivalent systems. This will mean our optimality result will hold for both Extended QU-Res and eFrege + $\forall$red.

Theorem 27. *Extended QU-Res and* eFrege + $\forall$red *are p-equivalent.*

Proof. First we show extended QU-Res p-simulates eFrege + $\forall$red. Let π be a refutation in eFrege + $\forall$red of $\Pi\phi$. Π is a prefix where every universal is y_i for some $1 \leq i \leq n$ and $\text{lv}(y_i) < \text{lv}(y_{i+1})$. We can change π into a normal form eFrege + $\forall$red proof π' as in [19]. This normal form contains an eFrege proof of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$,where y_i are the universal variables in index order and σ_{y_i} are the extracted strategies from π. Since extended resolution (with weakening) simulates eFrege. We can find short derivations of a CNF version of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ with extension variables involved. This can be written as:

$$\bigwedge_{i=1}^{n} Def(s_i = \sigma_{y_i}) \wedge t_n \wedge \neg t_0 \wedge \bigwedge_{i=1}^{n}(\neg t_i \vee y_i \vee s_i \vee t_{i-1}) \wedge \bigwedge_{i=1}^{n}(\neg t_i \vee \neg y_i \vee \neg s_i \vee t_{i-1})$$

s_i are extensions variables that are defined as σ_{y_i} in $Def(s_i = \sigma_{y_i})$ possibly using more extension variables for the gates. t_i are extra variables that allow us to split our large disjunction up, for $j \geq 0$, t_j can be seen as an extension variable defining $\bigvee_{i=1}^{j}(y_i \neq s_i)$.

Since the gates in σ_{y_i} and the s_i variables only depend on variables to the left of y_i we can place them in the quantifier prefix before y_i which will make it easier to $\forall$-reduce y_i, which is what we do inductively to get the refutation.

We next remove all weakening steps and end up with a Extended QU-Res proof.

We now show the converse, that eFrege + $\forall$red p-simulates Extended QU-Res. We take a proof π in extended QU-Res. In order to convert it into a proof of eFrege + $\forall$red we first have to convert between the clausal line in π to circuits without extension variables.

We replace every extension variable with the circuit it is describing (using the full circuit when an extension variable is based on other extension variables). The circuits introduced are only as large as π because they have to be defined using extension clauses. Hence the new proof is polynomial

A resolution rule can be easy copied by eFrege steps. The extension rules are now tautologies that can be easily inferred (or taken as axioms in fact). The reduction rules can be copied but we have to verify that the reduction rules are allowed. The new clauses now have circuits in place of extension variables, however the variables of the circuits are left of the extension variables, by definition of extended QU-Res. A clause $C \vee u$ in π where the variables in C are quantified before u is transformed into a circuit $D \vee u$ where the circuit D is in variables that are quantified before u. Hence reduction is valid. □

Theorem 28. *Extended Q-Res $^{\mathsf{NP}}$ is optimal among all QBF proof systems with strategy extraction*

By "optimal among all QBF proof systems with strategy extraction" we mean that it simulates all QBF proof systems with (P/poly-)strategy extraction and has strategy extraction itself. The caveat is that Extended QU-Res $^{\mathsf{NP}}$ is not a proof system due to the NP oracle.

Proof. Extended Q-Res $^{\mathsf{NP}}$ simulates Extended QU-Res $^{\mathsf{NP}}$ since universal resolution is subsumed by the NP-derivation rule. We know that Extended QU-Res $^{\mathsf{NP}}$ has strategy extraction by Theorem 7.

Suppose we have QBF proof system P, that has strategy extraction. We know from Theorem 26 we can simulate this by system eFrege $\forall$red $+\|\text{refl}(Sko(P))\|$, we can simulate this by Ext. QU-Res $^{\mathsf{NP}}$, because $\|\text{refl}(Sko(P))\|$ can be derived directly from the NP derivation and eFrege + $\forall$red rules can be simulated by Extended QU-Res rules. Note that it does not matter here if P uses an NP derivation rule as this can be directed simulated by the NP derivation rule in Extended QU-Res $^{\mathsf{NP}}$. □

5.1 Proof systems without strategy extraction

In the previous section we showed that Ext. Q-Res $^{\mathsf{NP}}$ is optimal among all QBF proof systems with strategy extraction and any proof system simulated by Ext. Q-Res $^{\mathsf{NP}}$ must also have strategy extraction. But what about proof systems without strategy extraction?

It is not easy to show that a proof system does not have strategy extraction as it is tied to unsolved problems in complexity theory. We prove the following:

Theorem 29. P=PSPACE *if and only if every QBF proof system has strategy extraction.*

Proof. We assume P=PSPACE first. Suppose we are using QBF proof system f. Let $\chi = \forall y_1 \exists x_1 \ldots \forall y_n \exists x_n \Phi$ where Φ is propositional and χ is a closed QBF that we want to refute

(any QBF can be converted to this form with dummy variables). Suppose we have f refutation π of χ. π is only relevant in so far that we can get χ from it in polynomial time (we can do so using f), and that it is valid.

Now we want to show a strategy for variable y_i, we will informally construct a program that allows us to get a response for y_i and show that it will produce a correct response. We do it inductively increasing i.

Induction Hypothesis: Suppose we are the universal player in our i-th round of the two-player semantic game on χ. We can find in polynomial time (in χ) a value of y_i, that still allows us to win the game later.

Base Case: Either $\exists x_1 \dots \forall y_n \exists x_n \phi_i[0/y_i]$ or $\exists x_1 \dots \forall y_n \exists x_n \phi_i[1/y_1]$ is false. Since $\mathsf{P} = \mathsf{PSPACE}$ there is a polynomial-time algorithm that tells us the truth value of QBFs. This we use to find a value of y_1 that allows us to continue with a falsifiable QBF.

Inductive Step: We assume we have a $\{0,1\}$ assignment $\alpha(Z_i)$ to all the variables $Z_i = \bigcup_{j=1}^{i-1} Y_j \cup X_j$.

Let $\phi_i = \phi[\alpha(Z_i)/Z_i]$, $\forall y_i \exists x_i \dots \forall y_n \exists x_n \phi_i$ is also a QBF which is false assuming the y_j variables have been played correctly for $j < i$. That means either $\exists x_i \dots \forall y_n \exists x_n \phi_i[0/y_i]$ or $\exists x_i \dots \forall y_n \exists x_n \phi_i[1/y_i]$ is false. We can check which one is false in polynomial time when we assume $\mathsf{P} = \mathsf{PSPACE}$ (since QBF is now in polynomial time) and we play y_i to that value. If we do this for increasing values of i we never exceed our polynomial time (in size of χ) for extracting this.

For the reverse we assume $\mathsf{P} \neq \mathsf{PSPACE}$, since QBF is PSPACE-complete problem, there will be no polynomial time algorithm that decides the truth of a QBF.

Suppose we have a closed QBF Δ. We can create a QBF $\forall z(z \leftrightarrow \Delta)$ which is always false because Δ is either equivalent to 0 or 1, and create a QBF proof system f that recognises this fact in a single line for every QBF Δ (this can be done in polynomial time). However, now we assume f has strategy extraction, then we can, in polynomial time, extract circuits from a vacuous proof that chooses the value of z, essentially providing a truth value for Δ. This gives us a polynomial time algorithm for QBF. □

6 Conclusion

The optimality result of Extended $\mathsf{QU\text{-}Res}$ has two caveats. Firstly, it only simulates systems with strategy extraction, and secondly it relies on an NP oracle. We can argue that these caveats are not incongruous with the state of QBF in practice. For some QBF applications, like chess, not only is the existence of a solution interesting but one would also want to know which chess moves ought to be played in order to reach that solution. Indeed many solvers do indeed allows strategies to be extracted from them.

In regards to the NP oracle, we see this as in line with QBF algorithms that use SAT algorithms as black boxes. Nonetheless, Theorem 26 allows us to remove the NP oracle and be in a proper proof system.

This result may have implications for other proof systems. QRAT [29] is a proof system that can simulate extended $\mathsf{QU\text{-}Res}$ and it is currently being proposed as a proof checking format for QBF solvers. If all these solvers have strategy extraction then $\mathrm{QRAT}^{\mathsf{NP}}$ would indeed be sufficient to check these solvers. Furthermore, we can remove the NP oracle and just assume that DRAT, the propositional version of QRAT, is sufficient for checking the propositional component of these solvers.

There is now reason to be cautious about QBF proof systems that simulate Ext. QU-Res or eFrege + ∀red. Either they are equivalent to Ext. QU-Res, do not have strategy extraction, or create a propositional system that is strictly stronger than eFrege (a long term open problem in propositional proof complexity). We suspect that QRAT may not have (Herbrand) strategy extraction or may be equivalent to Ext. QU-Res.

And finally we can assume (conditionally on P ≠ PSPACE) that we do have proof systems and solvers without strategy extraction. This creates a potential trade-off, strategy extraction is a nice property related to what we might want in practice, but it could inflate the sizes of our proofs and the running times of our algorithms.

Acknowledgements We are grateful to Olaf Beyersdorff and Luke Hinde for their discussions on NP derivations in QBF proof systems. Research supported by EPSRC.

References

1. Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta. Classic nintendo games are (computationally) hard. *Theoretical Computer Science*, 586:135 – 160, 2015. Fun with Algorithms.
2. Carlos Ansotegui, Carla P Gomes, and Bart Selman. The achilles' heel of QBF. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 2, pages 2–1, 2005.
3. Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.
4. Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012.
5. Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proc. 17th International Conference on Theory and Applications of Satisfiability Testing*, pages 154–169, 2014.
6. Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.
7. Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random qbfs. *CoRR*, abs/1712.03626, 2017.
8. 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.
9. Olaf Beyersdorff, Leroy Chew, Judith Clymo, and Meena Mahajan. Short proofs in QBF expansion. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:102, 2018.
10. Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proc. Symposium on Theoretical Aspects of Computer Science*, pages 76–89. LIPIcs series, 2015.
11. Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Extension variables in QBF resolution. In *Beyond NP, Papers from the 2016 AAAI Workshop*, 2016.
12. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. In *Proc. International Colloquium on Automata, Languages, and Programming (ICALP'15)*, pages 180–192. Springer, 2015.
13. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not simple. In *Proc. Symposium on Theoretical Aspects of Computer Science (STACS'16)*, 2016.
14. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for qbfs. *Information and Computation*, 262:141 – 161, 2018.
15. Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. In *LATA*, pages 486–498. Springer, 2015.
16. Olaf Beyersdorff and Judith Clymo. More on size and width in QBF resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:25, 2018.
17. Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games. *Information Processing Letters*, 110(23):1074–1077, 2010.
18. Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:44, 2017.
19. 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.
20. A. Blake. *Canonical expressions in boolean algebra*. PhD thesis, University of Chicago, 1937.
21. Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In *ICALP*, pages 94:1–94:14, 2016.
22. Stephen A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.
23. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.
24. Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *LPAR*, pages 291–308. Springer, 2013.
25. Ian P Gent and Andrew GD Rowley. Encoding connect-4 using quantified Boolean formulae. *2nd Intl. Work. Modelling and Reform. CSP*, pages 78–93, 2003.
26. Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In Toby Walsh, editor, *International Joint Conference on Artificial Intelligence IJCAI*, pages 546–553. IJCAI/AAAI, 2011.

27. A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.
28. Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proc. 18th ACM Symposium on Theory of Computing*, pages 6–20. ACM Press, 1986.
29. Marijn Heule, Martina Seidl, and Armin Biere. A unified proof system for QBF preprocessing. In *Automated Reasoning – 7th International Joint Conference, IJCAR*, volume 8562, pages 91–106. Springer, 2014.
30. Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.
31. Toni Jussila, Armin Biere, Carsten Sinz, Daniel Kröning, and Christoph M. Wintersteiger. A first step towards a unified proof checker for QBF. In João Marques-Silva and Karem A. Sakallah, editors, *SAT*, volume 4501, pages 201–214. Springer, 2007.
32. Hans Kleine Büning and Uwe Bubeck. Theory of quantified Boolean formulas. 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 735–760. IOS Press, 2009.
33. Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.
34. Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.
35. J. Messner and J. Torán. Optimal proof systems for propositional logic and complete sets. Technical Report TR97-026, Electronic Colloquium on Computational Complexity, 1997. A revised version appears at STACS'98.
36. Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
37. Pavel Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.
38. John Alan Robinson. Theorem-proving on the computer. *Journal of the ACM*, 10(2):163–174, 1963.
39. Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 721–730, New York, NY, USA, 2008. ACM.
40. 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.
41. Grigorii Samuilovich Tseitin. On the complexity of proof in prepositional calculus. *Zapiski Nauchnykh Seminarov POMI*, 8:234–259, 1968.
42. 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.
43. Allen Van Gelder. Contributions to the theory of practical quantified boolean formula solving. In *Principles and Practice of Constraint Programming*, pages 647–663. Springer, 2012.
44. Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *ICCAD*, pages 442–449, 2002.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il