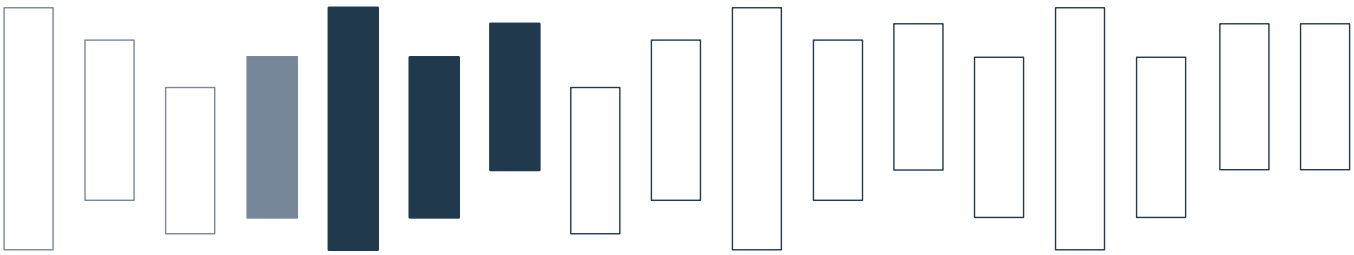
Technical Report AC-TR-21-013

July 2021

Hardness and Optimality in QBF Proof Systems Modulo NP

Leroy Chew



This is the authors' copy of a paper that appeared in the proceedings of SAT'21.

www.ac.tuwien.ac.at/tr

Hardness and Optimality in QBF Proof Systems Modulo NP

Leroy Chew( )

Technische Universität Wien, Vienna, Austria
lchew@ac.tuwien.ac.at
http://leroychew.wordpress.com/

Abstract. In this paper we show that extended Q-resolution is optimal among all QBF proof systems that allow strategy extraction modulo an NP oracle. In other words, for any QBF refutation system f where circuits witnessing the Herbrand functions can be extracted in polynomial time from f-refutations, f can be simulated by extended Q-resolution augmented with an NP oracle as described by Beyersdorff et al. We argue that using NP oracles and strategy extraction gives a natural framework to study QBF systems as they have relations to SAT calls and game instances, respectively, in QBF solving.

A weaker version of QBF extension variables also put forward by Jussila et al. does not have this optimality result, and we show that under an NP oracle there is no improvement of weak extended Q-Resolution compared to ordinary Q-Resolution.

Keywords: QBF · Proof complexity · Simulation · Resolution · Extended Frege · NP oracles · Optimal proof systems · Strategy extraction

Technical Report AC-TR-21-013

1 Introduction

Quantified Boolean formulas (QBF) are an extension of propositional logic and extend the SAT problem from NP-complete to PSPACE-complete [31]. In the last decade the SAT community has developed a strong interest in QBF solving as a successor to SAT and the number of QBF solvers, benchmarks and proof systems has multiplied considerably. QBF solving employs a variety of new reasoning techniques not found in SAT in order to deal with quantification. However, universally verifying the results of these different solvers over incomparable techniques remains a difficult problem. Proof systems such as extended Q-Res [22] or the even stronger QRAT [18] have been put forward as candidates for universal checking formats but have not yet been put to significant use.

We show that extended Q-Res has theoretical properties that make it a good candidate for a QBF checking format under a reasonable set of assumptions, and through these results, we can conjecture that it can simulate the proof systems that underpin the most commonly used QBF solving techniques.

© Springer Nature Switzerland AG 2021
C.-M. Li and F. Manyà (Eds.): SAT 2021, LNCS 12831, pp. 98–115, 2021.
https://doi.org/10.1007/978-3-030-80223-3_8

Just as in the SAT case, proof complexity is the main theoretical framework for analysing the relative strengths of QBF solvers. To use proof complexity, solvers are classified by their underlying proof systems, which express the limits of that solver. Because there are a variety of QBF solving methods, there are also many different QBF proof systems. Expansion-based solvers such as RAReQS [19] use the definition of QBF and expand into potentially exponential size propositional formulas. Expansion solvers are captured by proof systems such as ∀Exp+Res [20]. Conflict-driven clause-learning (CDCL), from SAT solving, is another technique that can be adapted for a QBF setting. This deals with quantification via a reduction rule. Combining existential resolution and universal reduction in proof theory gives the refutationally complete system Q-resolution (Q-Res) [26]. More general CDCL solvers that can perform stronger unit propagations are better described by so-called long-distance Q-Resolution (LD-Q-Res), an exponentially more powerful system[2]. An example of a QBF solver using CDCL is Dep-QBF [29]. The "Dep" part of Dep-QBF actually indicates another quantification technique which uses the awareness of spurious dependencies in the ordered quantifier prefix. The theory of dependency is also hugely important for QBF solving and theory [9,28,32] and has given rise to other stronger variants of Q-Res that utilise the dependency schemes, such as the reflexive resolution scheme [32].

Solvers can be modified to output certificates that are used to verify their results. It is natural for these certificates to be valid proofs for the corresponding proof systems. As well as being able to output proofs of truth or falsity, solvers are often asked to provide the *strategies* that witness how each variable must be set. In some applications, the strategy is the whole point of using a QBF solver. In Feldman et al. [13], circuit design algorithms explicitly used the strategy circuits output by QBF solvers rather than the true/false results. If these strategies are circuits that are easy to compute from the proofs, that proof system is said to have *strategy extraction*, an often desirable property for proof systems corresponding to solvers. All proof systems in the previous paragraph have this property.

While the above proof systems are meant to correspond to particular solvers, there is no agreed upon universal checking format for certification for every known type of QBF solver. One approach is to incorporate as many techniques as possible into the proof system. The proof system IRM-calc [6] combines the main concepts from the expansion-based ∀Exp+Res and the CDCL-based LD-Q-Res into one sound system. While this is interesting in understanding how expansion and CDCL systems can interact, IRM-calc is somewhat ad hoc, and a new technique could easily emerge which IRM-calc is not designed to deal with. For example, IRM-calc can not deal with the resolution of universal variables [6].

Another approach is to pick one strong system and prove that each solving technique can be simulated. This approach can be seen in the QRAT system, where it was first shown that a number of QBF preprocessing techniques were simulated by it [18]. Later it was shown to simulate LD-Q-Res [23] and ∀Exp+Res

Technical Report AC-TR-21-013

[24]. From these results, one could estimate that QRAT is indeed strong, but we would prefer a stronger theoretical reason for this.

If we want a QBF proof system suitable for universal certification, then the absolute ideal situation would be that it simulates every other QBF proof system. This is probably too ambitious as the existence of a theoretical optimal proof system remains a contested and open problem in propositional proof complexity, and it is an even stronger claim to suggest one exists for QBF. However, we can restrict our search to just proof systems with strategy extraction, and the problem becomes more manageable.

We find that with some extra help, extended Q-resolution is optimal among the proof systems with strategy extraction. Firstly we show in Theorem 1 that extended QU-Resolution (the 'U' in QU allows resolution on universal variables) is equivalent to the system eFrege + ∀red.

Theorem 1. *Extended* QU-Res *and* eFrege + ∀red *are p-equivalent.*

eFrege + ∀red has an important result where it can only have a lower bound if eFrege has a lower bound or PSPACE $\not\subseteq$ P/poly [5]. While this does not give us a simulation of another QBF proof system, it already indicates the strength of the system. With additional propositional power we show the next theorem.

Theorem 2. *For every refutational QBF Proof System S that has* P/poly*-strategy extraction, there is a set of polynomial-time verifiable propositional tautologies $\|\Psi\|$ such that* eFrege + ∀red + $\|\Psi\|$ *simulates S.*

For reasons that we discuss in Sect. 3, the extra propositional tautologies will not play a large role. Our main conjecture is that for the most interesting systems, the simulation requires no additional help.

Conjecture 1. ∀Exp+Res, IR-calc, LD-Q-Res, IRM-calc, QRAT(UR) and Q(D^{rrs})-Res are all simulated by eFrege + ∀red.

We saw that extra help needed for simulations can come in the form of propositional tautologies, but there is a second setting which achieves the same result- the use of NP oracles in a proof system.

This idea was first proposed by Chen [10] and refined by Beyersdorff, Hinde and Pich [8]. The Beyersdorff et al. NP derivation rule roughly allows one to make any propositional derivation in addition to the normal rules of whatever system we are adding the rule to. The motivation was to provide a theoretical framework that differentiated out genuine QBF hardness for QBF proof systems.

NP oracles model what happens in practice, as QBF solving algorithms often make black-box calls to SAT solvers. This usually does not affect strategy extraction as we see in Theorem 3.

Theorem 3. *The following strategy extraction theorems hold:*

- QU-Res ^{NP} *has depth-1 circuit decision list strategy extraction.*
- *For circuit class* C, C-Frege + ∀red ^{NP} *has* C*-decision list strategy extraction.*

Technical Report AC-TR-21-013

NP oracles remove the need for the families of propositional tautologies, and we can express our simulation results in terms of optimality.

Theorem 4. *Extended* Q-Res$^{\mathsf{NP}}$ *is optimal among all QBF proof systems with strategy extraction.*

The final three theorems examine a weaker form of extension in Q-Res and QU-Res under the lens of NP oracles.

Theorem 5. *Weak extended* QU-Res $^{\mathsf{NP}}$ *does not simulate extended* Q-Res*.*

Theorem 6. *Weak extended* Q-Res *does not simulate* QU-Res*.*

Theorem 7. Q-Res $\equiv^{\mathsf{NP}}$ QU-Res $\equiv^{\mathsf{NP}}$ *Weak Ext.*Q-Res $\equiv^{\mathsf{NP}}$ *Weak Ext.*QU-Res*.*

1.1 Organisation

In Sect. 2 we recap some essential definitions on QBF. In Sect. 3 we show Theorem 1 and 2 and discuss why this leads to Conjecture 1. Section 4 begins an analysis of proof systems under NP oracles with Theorems 3 and 4. This is finished in Sect. 5 where we prove Theorems 5, 6 and 7.

Technical Report AC-TR-21-013

2 Preliminaries

2.1 Proof Complexity

Formally, a *proof system* [12] for a language $\mathcal{L}$ over alphabet Γ is a polynomial-time computable partial function $f : \Gamma^\star \rightharpoonup \Gamma^\star$ with $rng(f) = \mathcal{L}$, where *rng* denotes the range. A proof system maps *proofs* to *theorems*. A *refutation* is a proof system where the language $\mathcal{L}$ is of contradictions. The partial function f gives a proof checking function. Soundness and completeness are given by $rng(f) \subseteq \mathcal{L}$ and $rng(f) \supseteq \mathcal{L}$, respectively. The polynomial-time computability is an indication of feasibility.

Proof size is given by the number of characters appearing in a proof. 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 [30]). *A proof system in language* $\mathcal{L}$ *is* (p-)optimal *if and only if it can (p-)simulate all other proof systems for* $\mathcal{L}$*.*

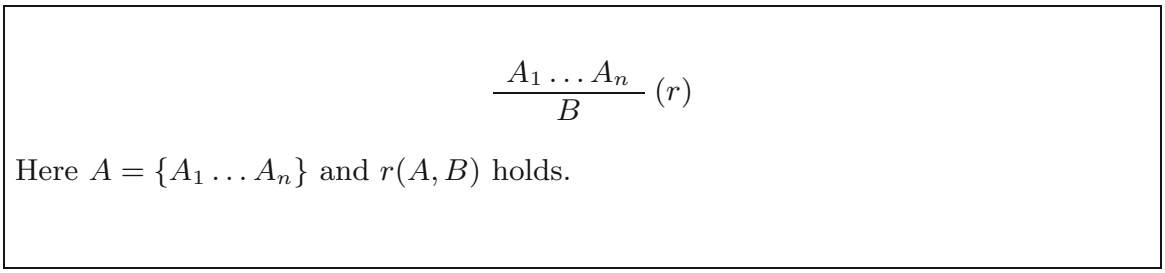


Fig. 1. Example of rule r in a line-based proof system

Line-Based Proofs. A proof system is *line-based* if every proof consists of a sequence $L_1 \dots L_n$ of lines L_i. The data types of lines are dependent on the proof systems. A line-based system is verified by a set of rules R. Each rule is a relation between a set of lines, which are known as the premises, and a single conclusion line. Correct proofs have that for each line L_i, there is some rule r in R and a subset A of $\{L_j | 0 \leq j < i\}$ such that $r(A, L_i)$ holds (see Fig. 1).

Given a line based proof system P with a set of rules R_P and a rule r, we can write $P + r$ to mean the proof system that consists of the rules of $R_P \cup \{r\}$ under the lines acceptable in P. If S is a set of propositional formulas, the proof system $P + S$ is the system $P + r$, where r is a rule that allows a conclusion s (with empty premises) if and only if $s \in S$. Note that rules and sets of lines have to be polynomial-time verifiable in order for the resulting system to be a proof system. While adding a rule r to a complete system P preserves completeness, soundness is not guaranteed and has to be reasoned for separately.

Technical Report AC-TR-21-013

2.2 Propositional Logic

Propositional logic involves Boolean variables under operations $\neg, \wedge, \vee, 0, 1$. A literal is a variable or its negation, a clause is a disjunction of literals and a conjunctive normal form (CNF) is a conjunction of clauses. A formula is satisfiable if there is a $0, 1$ assignment to variables so that the formula evaluates to 1. Deciding whether a propositional formula is satisfiable is NP-complete.

Propositional Proof Systems. *Resolution* (Res) is a propositional refutation system that works on formulas in conjunctive normal form. Resolution is line-based, where every line is a clause. The axiom rule allows us to download any clause in our original CNF. The inference rule takes two premise clauses $C \vee x$ and $D \vee \neg x$ and outputs conclusion $C \vee D$.

Extended resolution (Ext. Res) for propositional logic [33], enables adding clauses expressing the equality $v \Leftrightarrow (\neg x \vee \neg y)$, for a fresh variable v. As NAND gates can be defined by new variables, subsequent new variables can represent more complicated functions.

Frege systems are line-based systems that work on propositional formulas. Frege systems consist of an implicationally complete finite set of sound rules, each of which is represented by a single example, which can be generalised by

substitution. All Frege systems are known to be p-equivalent. While the lines of Frege systems are required to be formulas, a generalised version of Frege, denoted here by C-Frege, allows/restricts the lines to belong in circuit class C. For example, AC^0-Frege [3] is the Frege system where the lines are circuits with unbounded fan-in but have bounded-depth. NC^1-Frege is the Frege system where the lines have bounded fan-in and logarithmic depth, this is equivalent to the original Frege system [12] where lines are formulas. P/poly-Frege (defined as Circuit Frege by Jeřábek [21]) is the Frege system where general circuits have unbounded fan-in and depth. Extended Frege is known to be p-equivalent to P/poly-Frege, so we often use the notation eFrege to denote P/poly-Frege.

2.3 Quantified Boolean Formulas

Quantified Boolean Formulas extend propositional logic with quantifiers $\forall, \exists$ that work on propositional variables [25]. For formula (or circuit) A, we define $A[x/y]$ so that we replace all instances of y in A with x. The standard QBF 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 Ψ therefore consists of a propositional part ϕ called the matrix and a prefix of quantifiers Π and can be written as $\Psi = \Pi\phi$. Starting from left to right we give each bound variable a numerical level (lv) starting from 1 and increasing by one each time the quantifier changes (it stays the same whenever the quantifier is not changed). When the propositional matrix of a prenex QBF is a CNF, then we have a PCNF. We can feasibly transform any QBF into prenex form. 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 $0, 1$. TQBF and FQBF are used to denote the languages of true and false closed QBF, respectively.

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 the variables of lower index than u to $\{0, 1\}$ (the intended response for u). A *strategy* for the universal player for QBF $\Pi\phi$ is a set which contains exactly one strategy for each universal variable in Π. 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 [15][1, Sec. 4.2.2][31, Chap. 19]. Note that we differentiate between a universal strategy and what is known in the literature as a Herbrand function.

Technical Report AC-TR-21-013

Strategies are allowed to depend on previous universal variables, whereas the input to Herbrand functions must be purely existential (this allows us to get Theorem 3 to work). Since strategies for each universal variable are Boolean functions, they can be expressed as circuits. In many QBF solvers, as well as evaluating the truth of a QBF, solvers output circuits expressing the strategies for each universal (existential) variable whenever the QBF is false (true).

QBF Proof Systems. QBFs extend propositional formulas, therefore it is natural that many QBF proof systems use rules from propositional inference. In addition, QBF systems have to include rules that keep quantification in mind.

Q-resolution (Q-Res) by Kleine Büning, Karpinski, and Flögel [26] is a QBF resolution system. It uses the propositional resolution rule on existential variables. In addition, Q-resolution has a universal reduction rule to locally assign universal variables in clauses (for Fig. 2 recall that $\neg\neg z = z$ for literals). *QU-resolution (*QU-Res*)* [34] removes the restriction from Q-Res that the resolved variable must be existential and also allows resolution of universal variables.

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

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 [26]

Extended resolution for propositional resolution, 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. *Weak extended Q-resolution* [22] 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 (ext. Q-Res). Each extension variable is quantified after the variables it is defined from. Just as QU-Resolution introduces universal resolution to Q-Res, we can also get *extended QU-resolution* (ext. QU-Res) which adds

Technical Report AC-TR-21-013

universal resolution to extended Q-Res, the same can be done for *weak extended QU-resolution*.

C-Frege $+\forall$red uses circuit lines from the class C. It combines rules from Frege systems that operate on the circuit class C, with the reduction rule (See Fig. 3). While Frege systems are inferential, because we are using reduction, which is mainly used for refutation, C-Frege $+\forall$red is a refutational system.

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

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

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

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

Fig. 3. The rules of C-Frege + $\forall$red [5]

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).

Definition 2 (Strategy Extraction). *A refutational proof system P has (circuit)* strategy extraction *if there is a polynomial-time algorithm that takes P refutations π of QBF Ψ and outputs a circuit D_u for each universal variable u in prenex QBF Ψ, where the input variables of D_u are quantified to the left of u in Ψ 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 lower-bound technique, using the circuit extracted from the proof. 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. If that circuit is large then the proof must also be large. 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 $\mathsf{AC}^0[p]$-Frege + $\forall$red has $\mathsf{AC}^0[p]$-strategy extraction [5]. Circuit lower bounds for $\mathsf{AC}^0[p]$ can then be exploited to prove $\mathsf{AC}^0[p]$-Frege + $\forall$red proof-size lower bounds.

Technical Report AC-TR-21-013

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

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

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 circuit decision lists, while other QBF systems have strategy extraction in C-decision lists where C depends on the system. Extended Q-Res and extended QU-Res have strategy extraction [7] in P/poly since they use the bounded-depth strategy extraction of Q-Res and QU-Res, but the extension variables disguise arbitrary circuits.

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 from "genuine" QBF hardness.

Recent work [8,10] has started to factor out the component of propositional hardness in QBF. Most work has been done on the QU-Res systems but generalise to other systems as well.

Definition 4 (NP Oracle derivations[8]). *For QBF proof system* S*, a* S^{NP} *proof of a QBF* Ψ *is a derivation of the empty clause by any of the* S *rules or the* NP*-derivation rule.*

$$\frac{C_1, \ldots \qquad C_l}{D}\ \textit{(NP-derivation)}$$

For any l*, where there is some* Σ_1^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* S *(e.g. clauses, formulas).*

We will not here define a Σ_k^b-relaxation for every k we will just define for $k = 1$. We replace all universal quantifiers with existential ones. In other words, we can infer $\Pi D \wedge \bigwedge_{i=1}^{l} C_i$ from $\Pi \bigwedge_{i=1}^{l} C_i$ whenever $\bigwedge_{i=1}^{l} C_i \vDash D$ holds. When

Technical Report AC-TR-21-013

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 by Chen [10].

Definition 5. *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 Simulations with Extension Variables

In this section, we study the proof complexity of Ext QU-Resolution without NP oracles. NP oracles will be used in the next section. One may notice that in the definition of Beyersdorff et al. [5] $\mathsf{eFrege} + \forall\mathsf{red}$ is actually $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$, and despite its name, it does not use extension variables in its definition. The fact that $\mathsf{P/poly\text{-}Frege}$ and Frege with extension variables are equivalent propositionally requires the proof of Jeřábek [21], and this has to be proven again for QBF versions. In fact, we prove an even stronger equivalence by using only resolution instead of Frege.

Theorem 1. *Extended* $\mathsf{QU\text{-}Res}$ *(with general extension variables) and* $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$ *are p-equivalent.*

Proof. First, we show $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$ p-simulates extended $\mathsf{QU\text{-}Res}$. We take a proof π in extended $\mathsf{QU\text{-}Res}$ and convert it to a proof in $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$ with the same structure. In order to do this we must convert the clausal lines 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 others). The circuits introduced are only as large as π because they have to be defined using extension clauses. Hence the new proof is polynomial.

The resolution rule can be easily copied by $\mathsf{P/poly\text{-}Frege}$ steps. The extension rules are now tautologies that can be easily inferred (or taken as axioms). The reduction rule can be copied, but we have to verify that the new reduction instances are valid. The new clauses now have circuits in place of extension variables. Fortunately, the variables of the circuits are left of the extension variables, by definition. 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.

We now show the converse- that extended $\mathsf{QU\text{-}Res}$ p-simulates $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$. Let π be a refutation in $\mathsf{eFrege} + \forall\mathsf{red}$ of $\Pi\phi$. Π is a prefix where every universal is y_i for some $1 \leq i \leq n$ and $\mathrm{lv}(y_i) \leq \mathrm{lv}(y_{i+1})$. We can (in polynomial time) change π into a normal form $\mathsf{P/poly\text{-}Frege} + \forall\mathsf{red}$ proof π', which consists of two parts [5]. The first part contains a $\mathsf{P/poly\text{-}Frege}$ proof of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$,

Technical Report AC-TR-21-013

where σ_{y_i} are the extracted strategies from π. The second part is the QBF refutation of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ where reduction rules are used.

Consider a CNF version of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ with extension variables involved:

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

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 logic gates used in the circuits of σ_{y_i}. t_i are extra variables that allow us to split our large disjunction up, for $j \geq 0$, t_j is an extension variable defining $\bigvee_{i=1}^{j}(y_i \neq s_i)$. Since the gate variables in σ_{y_i} the s_i and t_{i-1} variables only depend on variables to the left of y_i we can place them in the quantifier prefix before y_i. As the CNF is a straightforward logical consequence from $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ it also has a short proof.

Induction Hypothesis: We can find short proofs of t_{n-k} using extended QU-Res with weakening (adding an extra literal to a clause) on $\Pi\phi$.

Base Case: The singleton clause (t_n) is a simple restatement of $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$. We can derive (t_n) in extended resolution with weakening (adding an extra literal), as extended resolution with weakening simulates P/poly-Frege in propositional logic. Note that when we incorporate this into QBF, we have to use Ext. QU-Res, not Ext. Q-Res as Ext. Res. does not distinguish between $\exists$ and $\forall$. (Whether Ext. QU-Res and Ext. Q-Res are equivalent is still an open problem.)

Inductive Step: Suppose we have clause (t_i) with $i = n - k$, we can resolve it with both $(\neg t_i \vee y_i \vee s_i \vee t_{i-1})$ and $(\neg t_i \vee \neg y_i \vee \neg s_i \vee t_{i-1})$ to get $(y_i \vee s_i \vee t_{i-1})$ and $(\neg y_i \vee \neg s_i \vee t_{i-1})$. Since s_i and t_{i-1} variables occur before y_i in the prefix we can reduce y_i in both cases to get $(s_i \vee t_{i-1})$ and $(\neg s_i \vee t_{i-1})$ which we can resolve to get clause (t_{i-1}).

Once we derive t_0, we get a contradiction. In order to derive (t_n), we added extra literals to the clauses with weakening. These literals are not needed in a refutation. Therefore, we remove all of these clause weakening steps and end up with an extended QU-Res refutation. □

Theorem 1 gives us that our next results will hold for both extended QU-Res and P/poly-Frege which we will now refer to as eFrege + ∀red.

But the proof itself also tells us something important- it uses *strategy extraction for simulation*. Contrast this with how strategy extraction has been used previously for QBF lower bounds [5,6]. This idea has the potential to be used for other proof systems or even solvers. Say we have proof system f that has P/poly strategy extraction. If we have an f refutation of QBF $\Pi\phi$, we can use strategy extraction to gain circuits σ_{y_i} for each of the universal variables y_i and substitute each y_i for σ_{y_i} in ϕ, giving us a propositional contradiction. If we can confirm this contradiction in eFrege, we would be able to prove $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$, and we can continue an eFrege + ∀red proof to get a refutation. This is *almost* a simulation of f by eFrege + ∀red. The thing that could go wrong is there is

Technical Report AC-TR-21-013

no guarantee that the substituted propositional matrix has a short eFrege proof. Nonetheless, eFrege is powerful enough for this problem not to occur very often. Theorems 2 and 4 give two different ways of clarifying what is meant by almost a simulation, but we need some technical lemmas on eFrege proofs.

Lemma 1. *For propositional circuits A, B and $\phi(X)$ any propositional tautology of the form $(A \leftrightarrow B) \rightarrow (\phi(A) \leftrightarrow \phi(B))$ has a polynomial-size proof in* eFrege*.*

Lemma 2. *Let Π be a QBF prefix where each $\forall$ variable is given as y_i for $1 \leq i \leq n$. Let ϕ and σ_{y_i} for $1 \leq i \leq n$ be propositional circuits. Now define $\phi_{\sigma,\Pi}$ to be the propositional circuit that replaces all occurrences of y_i with σ_{y_i}. The tautology $\phi \wedge \neg\phi_{\sigma,\Pi} \rightarrow \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ has polynomial-size proofs in* eFrege*, (in the sizes of ϕ and σ_{y_i}).*

We can now talk about simulation by eFrege $+\forall$red. In the next theorem, we have the additional condition that we may need an infinite family of polynomially-recognisable tautologies added to eFrege$+\forall$red. Bear in mind these are only *propositional* tautologies, not QBF.

Theorem 2. *For every refutational QBF Proof System S that has* P/poly*-strategy extraction, there is a set of polynomial-time verifiable propositional tautologies $\|\Psi\|$ such that* eFrege $+\forall$red $+\|\Psi\|$ *simulates S.*

It is known [27] that any propositional proof system P is simulated by eFrege $+\|\text{refl}(P)\|$ where $\|\text{refl}(P)\|$ is a set of propositional tautologies that code arithmetic statements of P's correctness (the name "reflection principle" comes from the challenge of a system proving its own soundness). The idea is to use these propositional tautologies in a QBF setting, but we also need reduction and essentially strategy extraction.

Proof. Let S be our FQBF proof system which allows polynomial-time strategy extraction in circuits. 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 refutation π of QBF $\Pi\phi$ we can extract in polynomial-time circuits σ_y that are strategies for each universal variable y. Let $\phi_{\sigma,\Pi}$ be the propositional formula that results from replacing every universal variable y with σ_y in ϕ. Since the strategy is correct, $\phi_{\sigma,\Pi}$ must be a propositional contradiction.

We can use this observation to design a propositional proof system $Strat(S)$. The idea is that this proof system verifies the proposition $(\neg\phi)_{\sigma,\Pi}$ instead of refuting the QBF $\Pi\phi$. Using the Cook-Reckhow definition of a proof system as a checking function (see Sect. 2.1) we define it as follows:

$$Strat(S)(\pi) = \begin{cases} \neg\phi_{\sigma,\Pi}, & \pi \text{ is an } S \text{ refutation of } \Pi\phi \\ & \text{and } \sigma \text{ is the strategy extracted from it,} \\ \text{eFrege}(\pi), & \text{otherwise.} \end{cases}$$

Using information from [27] we know $Strat(S)$ is simulated by eFrege $+$ $\|\text{refl}(Strat(S))\|$, where $\|\text{refl}(Strat(S))\|$ is a polynomial-time recognisable set of

Technical Report AC-TR-21-013

propositions that encode an arithmetic statement of the correctness of $\mathit{Strat}(S)$. We will show that $\mathsf{eFrege}+\forall\mathsf{red}+\|\mathrm{refl}(\mathit{Strat}(S))\|$ simulates S, so we let π be a proof of $\Pi\phi$ in S with strategy extracted σ. Note that π is also a $\mathit{Strat}(S)$ proof.

We let π'_1 be the $\mathsf{eFrege}+\|\mathrm{refl}(\mathit{Strat}(S))\|$ proof that simulates π in $\mathit{Strat}(S)$. We know this is of polynomial-size in π. Likewise as we know the σ_y are polynomial-size, this means that by using Lemma 2 the circuit $\phi \wedge \neg\phi_{\sigma,\Pi} \rightarrow \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ has a polynomial-size eFrege proof π'_2, where y_i are the universal variables in Π in order (y_n being the innermost universal variable).

We show that $\mathsf{eFrege}+\forall\mathsf{red}+\|\mathrm{refl}(\mathit{Strat}(S))\|$ can refute $\Pi\phi$ in a short proof.

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

Similarly to Theorem 1, 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. Since we already have $\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$ we have the base case and we only need to show the inductive step.

We derive from $\bigvee_{i=1}^{n+1-k}(y_i \neq \sigma_{y_i})$ both $(0 \neq \sigma_{y_{n+1-k}}) \vee \bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$ and $(1 \neq \sigma_{y_{n+1-k}}) \vee \bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$ from reduction. We can resolve both with the easily proved tautology $(0 = \sigma_{y_{n+1-k}}) \vee (1 = \sigma_{y_{n+1-k}})$ which allows us to derive $\bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$. We continue this until we reach the empty disjunction. □

Conjecture 1. $\forall$Exp+Res, IR-calc, LD-Q-Res, IRM-calc, QRAT(UR) and Q($\mathsf{D}^{\mathsf{rrs}}$)-Res are all simulated by $\mathsf{eFrege}+\forall\mathsf{red}$.

Let us take one example, e.g. $\forall$Exp+Res and suppose it is not true. Then $\|\mathrm{refl}(\mathit{Strat}(\forall\mathsf{Exp}+\mathsf{Res}))\|$ would have to be an eFrege lower bound, an answer to a major open problem. Put another way, $\mathit{Strat}(\forall\mathsf{Exp}+\mathsf{Res})$ would be a propositional proof system more powerful than eFrege on certain families. This would seem very unlikely. More likely would be that the steps of an $\forall$Exp+Res refutation of $\Pi\phi$ combined with formalised knowledge about the strategy extraction for $\forall$Exp+Res could help guide a short refutation of $\phi_{\sigma,\Pi}$ using extension variables and Frege. If so then we would get a simulation.

4 Extended Q-Res Modulo NP

We now analyse QBF proof systems with the NP oracle included. As it allows new derivations to occur immediately, this can change a system considerably. It is necessary to prove, where applicable, when strategy extraction remains.

Theorem 3. *The following strategy extraction theorems hold:*

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

Technical Report AC-TR-21-013

Proof. The proof follows the line-based strategy extraction used by Balabanov et al. [2] and later generalised by Beyersdorff et al. [5]. Purely propositional rules make no changes to the extraction, and NP-derivations are purely propositional. □

This is not an automatic result for any QBF proof system with strategy extraction; recent results [11] on strategy extraction indicate that expansion based systems may lose strategy extraction when equipped with NP oracles. It is also unclear whether variants of (Ext) Q(U)-Resolution that allow long-distance resolution steps have strategy extraction when NP oracles are allowed. Extended Q-Res$^{\mathsf{NP}}$ and extended QU-Res$^{\mathsf{NP}}$ are among the systems with strategy extraction. NP oracles allow us to remove $\|\text{refl}(\mathit{Strat}(S))\|$ used in Theorem 2, but also collapses Q-Res and QU-Res into the same system.

Theorem 4. *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 (circuit-)strategy extraction and has strategy extraction itself. The caveat is that neither extended Q-Res $^{\mathsf{NP}}$ nor extended QU-Res $^{\mathsf{NP}}$ are proof systems due to the NP oracle.

Proof. Ext. Q-Res $^{\mathsf{NP}}$ simulates ext. QU-Res $^{\mathsf{NP}}$ since universal resolution is subsumed by the NP-derivation rule. We know that ext. QU-Res $^{\mathsf{NP}}$ has strategy extraction by the equivalence of extended QU-Res and P/Poly-Frege + ∀red, which when augmented with an NP-derivation rule has strategy extraction by Theorem 3.

Suppose we have QBF proof system S that has strategy extraction. We know from Theorem 2 we can simulate this by system eFrege + ∀red + $\|\text{refl}(\mathit{Strat}(S))\|$, we can simulate this by ext. QU-Res $^{\mathsf{NP}}$, because $\|\text{refl}(\mathit{Strat}(S))\|$ can be derived directly from the NP derivation and eFrege + ∀red rules can be simulated by extended QU-Res rules. Note that it does not matter here if S uses an NP derivation rule as this can be simulated by the NP derivation rule. □

5 Weaker QBF Systems

So far we have only studied *extended* QU-Res and Q-Res with *general extensions*. There remains four weaker systems, *extended* QU-Res and Q-Res with *weak extensions* and standard QU-Res and Q-Res. We will analyse these four for the remainder of this paper, both with and without the NP oracle.

Theorem 5. *Weak extended QU-Resolution*$^{\mathsf{NP}}$ *does not simulate extended Q-Resolution.*

Proof. We take the QParity formulas [6] which are known to have short proofs in general extended Q-Res [7]. We will show that these are hard for weak extended

Technical Report AC-TR-21-013

QU Resolution$^{\mathsf{NP}}$. In fact, because of Theorem 7, we will only need to show these are hard for Q-Res$^{\mathsf{NP}}$.

Let $\mathsf{xor}(o_1, o_2, o)$ be the CNF $(\neg o_1 \vee \neg o_2 \vee \neg o) \wedge (o_1 \vee o_2 \vee \neg o) \wedge (\neg o_1 \vee o_2 \vee o) \wedge (o_1 \vee \neg o_2 \vee o)$, which defines o to be equal to $o_1 \oplus o_2$. Define QPARITY$_n$ as

$$\exists x_1 \dots x_n \, \forall z \exists t_2 \dots t_n \, \mathsf{xor}(x_1, x_2, t_2) \wedge \bigwedge_{i=3}^{N} \mathsf{xor}(t_{i-1}, x_i, t_i) \wedge (z \vee t_n) \wedge (\neg z \vee \neg t_n).$$

While QPARITY is false, the only winning strategy of the universal player on the QPARITY formulas is to actually compute the PARITY function. However, PARITY is the classic example of a function hard for bounded-depth circuits and AC^0-decision lists [14,17]. Q-Res$^{\mathsf{NP}}$ has strategy extraction in AC^0-decision lists, but these must be exponential size, which means the proofs themselves are required to be of exponential size. □

The separation between Q-Res and QU-Res comes from the formulas from Kleine Büning, Karpinski and Flögel [26,34]. QU-Res cannot simulate weak extended Q-Res due to propositional lower bounds like the pigeonhole principle [16]. We are only left to show one more separation, and we get the complete picture. Adapting the cost-capacity technique from [4], we can show that the KBKF formulas are also hard for weak ext. Q-Res, giving Theorem 6.

Theorem 6. *Weak extended* Q-Res *does not simulate* QU-Res.

Once we have that final lower bound, we prove the following complete simulation structure in Fig. 4. We then show in Theorem 7, that the opposite is true when using NP derivations.

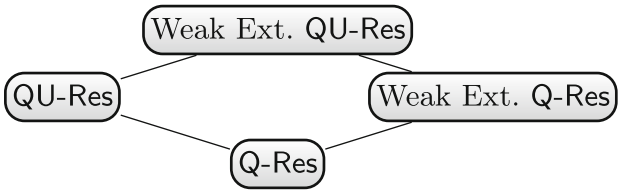


Fig. 4. The simulation structure of four variants of Q-Res, all pairwise simulations are given and are strictly one-way, and other pairs do not yield a simulation.

Theorem 7. Q-Res $\equiv^{\mathsf{NP}}$ QU-Res $\equiv^{\mathsf{NP}}$ *Weak Ext.*Q-Res $\equiv^{\mathsf{NP}}$ *Weak Ext.*QU-Res.

Proof. Q-Res $\equiv^{\mathsf{NP}}$ QU-Res and Weak Ext.Q-Res $\equiv^{\mathsf{NP}}$ Weak Ext.QU-Res because NP derivations can be used to simulate universal resolution steps directly. We are left to show Q-Res $\equiv^{\mathsf{NP}}$ Weak Ext.Q-Res.

The first observation is that 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

Technical Report AC-TR-21-013

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 ∀red these clauses and continue to alternate between NP derivations steps and universal reduction steps. □

6 Conclusion

We have shown that extended QU-Res and eFrege+∀red are equivalent as long as the extension variables are defined generally. eFrege+∀red has an important place among QBF proof systems, particularly among those with strategy extraction. This can be qualified with or without an NP oracle. This position allows us to conjecture that eFrege+∀red will simulate the known QBF systems with strategy extraction and will be able to certify solvers that have strategy extraction.

These properties do not hold for weak extension variables even with the NP oracles. In fact, under the NP oracle, weak extended QU-Res has no more strength than regular Q-Res.

Technical Report AC-TR-21-013

References

1. Arora, S., Barak, B.: Computational Complexity - A Modern Approach. Cambridge University Press, Cambridge (2009)
2. Balabanov, V., Jiang, J.H.R.: Unified QBF certification and its applications. Formal Methods Syst. Des. **41**(1), 45–65 (2012)
3. Bellatoni, S., Pitassi, T., Urquhart, A.: Approximation of small-depth Frege proofs. SIAM J. Comput. **21**, 1161–1179 (1992)
4. Beyersdorff, O., Blinkhorn, J., Hinde, L.: Size, cost, and capacity: a semantic technique for hard random QBFs. CoRR abs/1712.03626 (2017). http://arxiv.org/abs/1712.03626
5. Beyersdorff, O., Bonacina, I., Chew, L., Pich, J.: Frege systems for quantified boolean logic. J. ACM **67**(2), (2020). https://doi.org/10.1145/3381881
6. Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. ACM Trans. Comput. Theory **11**(4), 26:1–26:42 (2019). https://doi.org/10.1145/3352155
7. Beyersdorff, O., Chew, L., Janota, M.: Extension variables in QBF resolution. In: Beyond, N.P.: Papers from the 2016 AAAI Workshop (2016). http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12612
8. Beyersdorff, O., Hinde, L., Pich, J.: Reasons for hardness in QBF proof systems. Electron. Colloquium Comput. Complexity (ECCC) **24**, 44 (2017). https://eccc.weizmann.ac.il/report/2017/044
9. Blinkhorn, J.L.: Quantified Boolean Formulas: Proof Complexity and Models of Solving. Ph.D. thesis, University of Leeds (2019)
10. Chen, H.: Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In: ICALP, pp. 94:1–94:14 (2016)

11. Chew, L., Clymo, J.: How QBF expansion makes strategy extraction hard. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) IJCAR 2020. LNCS (LNAI), vol. 12166, pp. 66–82. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51074-9_5
12. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. J. Symbolic Logic **44**(1), 36–50 (1979)
13. Feldman, A., Kleer, J., Matei, I.: Design space exploration as quantified satisfaction (05 2019)
14. Furst, M.L., Saxe, J.B., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. Math. Syst. Theory **17**(1), 13–27 (1984)
15. Goultiaeva, A., Van Gelder, A., Bacchus, F.: A uniform approach for generating proofs and strategies for both true and false QBF formulas. In: Walsh, T. (ed.) International Joint Conference on Artificial Intelligence IJCAI, pp. 546–553. IJCAI/AAAI (2011)
16. Haken, A.: The intractability of resolution. Theor. Comput. Sci. **39**, 297–308 (1985)
17. Håstad, J.: One-way permutations in NC^0. Inf. Process. Lett. **26**(3), 153–155 (1987)
18. Heule, M., Seidl, M., Biere, A.: A unified proof system for QBF preprocessing. In: 7th International Joint Conference on Automated Reasoning (IJCAR), pp. 91–106 (2014)
19. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.: Solving QBF with counterexample guided refinement. In: Cimatti, A., Sebastiani, R. (eds.) SAT 2012. LNCS, vol. 7317, pp. 114–128. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31612-8_10
20. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. Theor. Comput. Sci. **577**, 25–42 (2015)
21. Jeřábek, E.: Dual weak pigeonhole principle, Boolean complexity, and derandomization. Ann. Pure Appl. Logic **129**, 1–37 (2004)
22. Jussila, T., Biere, A., Sinz, C., Kröning, D., Wintersteiger, C.M.: A first step towards a unified proof checker for QBF. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT 2007. LNCS, vol. 4501, pp. 201–214. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72788-0_21
23. Kiesl, B., Heule, M.J.H., Seidl, M.: A little blocked literal goes a long way. In: Gaspers, S., Walsh, T. (eds.) SAT 2017. LNCS, vol. 10491, pp. 281–297. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66263-3_18
24. Kiesl, B., Seidl, M.: QRAT polynomially simulates ∀-Exp+Res. In: Janota, M., Lynce, I. (eds.) SAT 2019. LNCS, vol. 11628, pp. 193–202. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24258-9_13
25. Kleine Büning, H., Bubeck, U.: Theory of quantified Boolean formulas. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 735–760. IOS Press (2009)
26. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. **117**(1), 12–18 (1995)
27. Krajíček, J.: Bounded Arithmetic, Propositional Logic, and Complexity Theory, Encyclopedia of Mathematics and Its Applications, vol. 60. Cambridge University Press, Cambridge (1995)
28. Lonsing, F.: Dependency Schemes and Search-Based QBF Solving: Theory and Practice. Ph.D. thesis, Informatik, Johannes Kepler University Linz (2012)
29. Lonsing, F., Biere, A.: DepQBF: a dependency-aware QBF solver. JSAT **7**(2–3), 71–76 (2010)

Technical Report AC-TR-21-013

30. Messner, J., Torán, J.: Optimal proof systems for propositional logic and complete sets. Technical Report, TR97-026, Electronic Colloquium on Computational Complexity, a revised version appears at STACS'98 (1997)
31. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Boston (1994)
32. Slivovsky, F.: Structure in# SAT and QBF. Ph.D. thesis (2015)
33. Tseitin, G.S.: On the complexity of proof in prepositional calculus. Zapiski Nauchnykh Seminarov POMI **8**, 234–259 (1968)
34. Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Milano, M. (ed.) CP 2012. LNCS, pp. 647–663. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33558-7_47

Technical Report AC-TR-21-013