Electronic Colloquium on Computational Complexity, Report No. 120 (2014)

Proof Complexity of Resolution-based QBF Calculi

Olaf Beyersdorff[1], Leroy Chew[1], and Mikoláš Janota[2]

[1]School of Computing, University of Leeds, United Kingdom
[2]INESC-ID, Lisbon, Portugal

Abstract. Proof systems for quantified Boolean formulas (QBFs) provide a theoretical underpinning for the performance of important QBF solvers. However, the proof complexity of these proof systems is currently not well understood and in particular lower bound techniques are missing. In this paper we exhibit a new and elegant proof technique for showing lower bounds in QBF proof systems based on strategy extraction. This technique provides a direct transfer of circuit lower bounds to lengths of proofs lower bounds. We use our method to show the hardness of a natural class of parity formulas for Q-resolution. Variants of the formulas are hard for even stronger systems as long-distance and universal Q-resolution. With a completely different lower bound argument we show the hardness of the prominent formulas of Kleine Büning et al. [23] for the strong expansion-based calculus IR-calc, thus also confirming the hardness of the formulas for Q-resolution. Our lower bounds imply new exponential separations between two different types of resolution-based QBF calculi: proof systems for DPLL-based solvers (Q-resolution, long-distance Q-resolution) and proof systems for expansion-based solvers (∀Exp+Res and its generalizations IR-calc and IRM-calc). The relations between proof systems from the two different classes were not known before.

1 Introduction

Proof complexity studies the complexity of theorem proving in various formal systems, providing both sharp lower and upper bounds for the size of proofs of important combinatorial statements. One motivation for this research comes from its close connection to fundamental questions in computational complexity, and this connection has been present since the very beginnings of the field [13]. Another motivation is the tremendous success of SAT solvers, which today solve huge industrial instances of the NP-hard SAT problem with even millions of variables. Proof complexity provides the main theoretical tool for an understanding of the power and limitations of these algorithms. As most modern SAT solvers are based on resolution, this proof system has received key attention; and many ingenious techniques have been devised to understand the complexity of resolution proofs (cf. [29,10] for surveys).

During the last decade there has been great interest and research activity to extend the success of SAT solvers to the more expressive *quantified boolean formulas (QBF)*. Due to its PSPACE completeness, QBF is far more expressive than SAT and thus applies to further fields such as formal verification or planning [27,5]. As for SAT solvers, runs of QBF solvers produce witnesses respectively proofs of unsatisfiability, and there has been great interest in trying to understand which formal system would correspond to the solvers.

In particular, Kleine Büning et al. [23] define a resolution-like calculus called *Q-resolution* (Q-Res). There are several extensions of Q-Res; notably *long-distance Q-resolution* (LD-Q-Res) [2], which is more powerful than the standard Q-Res [14]. Q-Res and its extensions are important as they model QBF solving based on CDCL [16]. While Q-Res can only resolve on existential variables, the proof system *QU-Res*, introduced by Van Gelder [30], also allows to resolve on universal variables. Combining universal and long-distance resolution, Balabanov et al. [3] recently considered the system LQU^{+}-Res.

ISSN 1433-8092

Apart from CDCL, another main approach to QBF-solving is through *expansion of quantifiers* [8,4,18]. Recently, a proof system ∀Exp+Res was introduced with the motivation to trace expansion-based QBF solvers [17]. ∀Exp+Res also uses resolution, but is conceptually very different from Q-Res.

In the recent work [6] two further proof systems *IR-calc* and *IRM-calc* are introduced, which unify the CDCL and expansion based approaches in the sense that IR-calc simulates both Q-Res and ∀Exp+Res. The system IRM-calc enhances IR-calc and additionally simulates long-distance resolution. While IR-calc and IRM-calc are quite powerful, they still preserve the property of strategy extraction, which is important for verifying runs of QBF solvers.

In general, it is fair to say that the complexity and relations between QBF proof systems are not well understood. In particular, in sharp contrast to propositional proof complexity, we currently lack any lower bound techniques for QBF proof systems.

1.1 Our contributions

In this paper we aim towards a significantly better understanding of proof complexity of QBF proof systems. Our main contributions are the following:

1. A new lower bound method based on strategy extraction. We exhibit a new method to obtain lower bounds to the proof size in QBF proof systems, which directly allows to transfer circuit lower bounds to size of proof lower bounds. This method is based on the property of *strategy extraction*, which is known to hold for many resolution-based QBF proof systems. A QBF proof system has strategy extraction if given a refutation of a false QBF φ it is possible to efficiently compute a winning strategy for the universal player for φ.

The basic idea of our method is both conceptually simple and elegant: If we know that a family φ_n of false QBFs requires large winning strategies, then proofs of φ_n must be large in all proof systems with feasible strategy extraction. Now we need suitable formulas φ_n. Starting with a language L — for which we know (or conjecture) circuit lower bounds — we construct a family of false QBFs φ_n such that every winning strategy of the universal player for φ_n will have to compute L for inputs of length n. Consequently, a circuit lower bound for L directly translates into a lower bound for the winning strategy and therefore the proof size.

This immediately implies conditional lower bounds. However, if carefully implemented, our method also yields *unconditional lower bounds*. For Q-Res it is known that strategy extraction is computationally easy [2]; it is in fact possible in AC^0 as we verify here. Using the hardness of parity for AC^0 we can therefore construct formulas $\text{QP}\textsc{arity}_n$ that require exponential-size proofs in Q-Res.

Conceptually, our lower bound method via strategy extraction is similar to the feasible interpolation technique [24], which is one of the most successful techniques in classical proof complexity. In feasible interpolation, circuit lower bounds are also translated into proof size lower bounds. However, feasible interpolation only works for formulas of a special syntactic form, while our technique directly applies to arbitrary languages. It is a long-standing belief in the proof complexity community that there exists a direct connection between progress for showing lower bounds in circuit complexity and for proof systems (cf. [12]). For QBF proof systems our technique makes such a connection very explicit.

2. Lower bounds for QBF proof systems. Our new lower bound method directly gives a new lower bound for Q-Res for the parity formulas. In addition, we transfer this lower bound to the stronger systems of long-distance Q-resolution and QU-resolution by arguing that neither long-distance nor universal Q-resolution gives any advantage on a suitable modification of the parity formulas.

For the strong system IR-calc from [6] we show that the strategy extraction method is not directly applicable (at least for unconditional bounds in the way we use it here). However, we use a completely different lower bound argument to obtain an exponential lower bound for the well-known formulas KBKF(t) of Kleine Büning, Karpinski and Flögel [23] in IR-calc. In the same work [23], where Q-Res was introduced, these formulas were suggested as hard formulas for Q-Res. In fact, a number of further separations of QBF proof systems builds on this [14,3], even though the hardness of KBKF(t) has never been formally verified in the literature so far. Here we show in a technically involved counting argument that the formulas are even hard for IR-calc. As IR-calc simulates Q-Res [6] we obtain as a by-product a formal proof of the hardness of KBKF(t) in Q-Res.

3. Separations between QBF proof systems. Our lower bounds imply a number of new separations and incomparability results for QBF resolution systems. The two main new results are

1. IR-calc does not simulate LD-Q-Res.
2. LQU$^+$-Res does not simulate $\forall$Exp+Res.

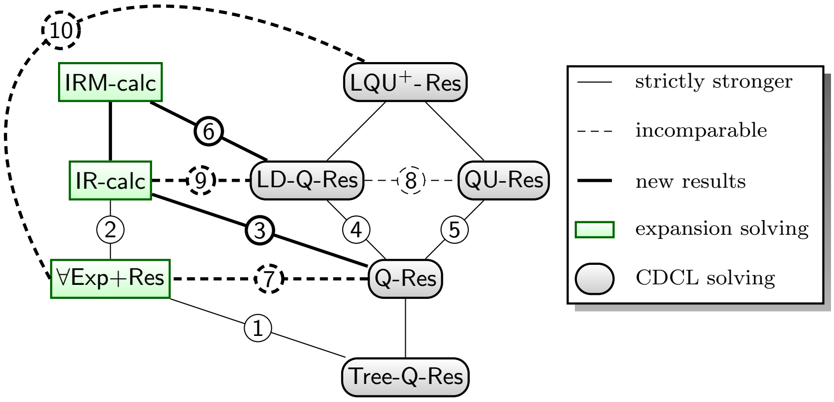
Both are in fact exponential separations. Item 1 is obtained from the lower bound for KBKF(t), while item 2 follows from the lower bound on a variant of the parity formulas. Together with previous simulation results these imply many further separations. Figure 1 depicts the simulation order of QBF resolution systems together with the separations. By a strict simulation we mean that one calculus simulates the other but the reverse simulation does not hold. Two calculi are incomparable if neither calculus simulates the other. New separation results proven here are drawn as bold lines. Together with previous simulations and separations (cf. the table accompanying Figure 1) this provides an almost complete understanding of the simulation order of resolution-based QBF proof systems.

1.2 Organisation of the paper

The rest of the paper is organized as follows. Section 2 overviews relevant QBF proof systems and introduces notation and concepts used throughout the paper. In Section 3 we show the lower bound for the formulas KBKF(t), implying an exponential separation between IR-calc (and Q-Res) and LD-Q-Res. In Section 4 we demonstrate our new lower bound method via strategy extraction by proving a lower bound for the parity formulas. This implies that Q-Res does not simulate $\forall$Exp+Res (and therefore IR-calc and IRM-calc). In Section 5 we push the lower bound further to LD-Q-Res and even LQU$^+$-Res, thereby improving the separation to LQU$^+$-Res vs. $\forall$Exp+Res. Section 6 shows that the proof technique used in Section 4 can be applied in a wider context and thus provide more general results. Section 7 concludes the paper and presents directions for future work.

2 Preliminaries

A *literal* is a Boolean variable or its negation; we say that the literal x is *complementary* to the literal $\neg x$ and vice versa. If l is a literal, $\neg l$ denotes the complementary literal, i.e.



	Simulations	/Separations		Incomparabilities
1	[20]	[20]	7	[19], Thm. 22
2	[6]	[19], [6]	8	[3]
3	[6]	Thm. 22	9	Cor. 14, Thm. 29
4	[14]	[14]	10	Cor. 14, Thm. 34
5	[30]	[30]		
6	[6]	Thm. 29		

Fig. 1. The simulation order of QBF resolution systems

$\neg\neg x = x$. A *clause* is a disjunction of zero or more literals and a *term* is a conjunction of literals. The empty clause is denoted by $\bot$, which is semantically equivalent to false. A formula in *conjunctive normal form* (CNF) is a conjunction of clauses. Whenever convenient, a clause is treated as a set of literals and a CNF formula as a set of clauses. For a literal $l = x$ or $l = \neg x$, we write $\text{var}(l)$ for x and extend this notation to $\text{var}(C)$ for a clause C and $\text{var}(\psi)$ for a CNF ψ.

Quantified Boolean Formulas (QBFs) [22] extend propositional logic with quantifiers with the standard semantics that $\forall x. \Psi$ is satisfied by the same truth assignments as $\Psi[0/x] \wedge \Psi[1/x]$ and $\exists x. \Psi$ as $\Psi[0/x] \vee \Psi[1/x]$. Unless specified otherwise, we assume that QBFs are in *closed prenex* form with a CNF *matrix*, i.e., we consider the form $\mathcal{Q}_1 X_1 \ldots \mathcal{Q}_k X_k.\ \phi$, where X_i are pairwise disjoint (ordered) sets of variables; $\mathcal{Q}_i \in \{\exists, \forall\}$ and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$. The formula ϕ is in CNF and is defined only on variables $X_1 \cup \ldots \cup X_k$. The propositional part ϕ of a QBF is called the *matrix* and the rest the *prefix*. If a variable x is in the set X_i, we say that x is at *level* i and write $\text{lv}(x) = i$; we write $\text{lv}(l)$ for $\text{lv}(\text{var}(l))$. In contrast to the level, the *index* of a variable x ($\text{ind}(x)$) provides the more detailed information on the actual position of x in the prefix, i.e. all quantifiers are indexed by $1, \ldots, n$ from left to right. A closed QBF is *false* (resp. *true*), iff it is semantically equivalent to the constant 0 (resp. 1).

Often it is useful to think of a QBF $\mathcal{Q}_1 X_1 \ldots \mathcal{Q}_k X_k. \phi$ as a *game* between the *universal* and the *existential player*. 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 iff the matrix ϕ evaluates to 1 under the assignment constructed in the game. The universal player wins iff the matrix ϕ evaluates to 0. A QBF is false iff there exists a *winning strategy* for the universal player, i.e. if the universal player can win any possible game [1, Sec. 4.2.2][26, Chap. 19]. Note that a winning strategy always exists for one and only one of the players.

A *proof system* (Cook, Reckhow [13]) for a language L over alphabet Γ is a polynomial-time computable partial function $f : \Gamma^\star \rightharpoonup \Gamma^\star$ with $rng(f) = L$. An *f-proof* of string y is a string x such that $f(x) = y$. If L consists of all propositional tautologies, then f is called a *propositional proof system*. For $L = \text{QBF}$ we speak of a *QBF proof system*. In the systems that we consider here, proofs are sequences of clauses; a *refutation* is a proof deriving $\bot$. A proof system S for L *simulates* a proof system P for L if there exists a polynomial p such that for all P-proofs π of x there is an S-proof π' of x with $|\pi'| \leq p\,(|\pi|)$. If such a proof π' can even be computed from π in polynomial time we say that S *p-simulates* P.

2.1 Resolution-based calculi for QBF

This section gives a brief overview of the main existing resolution-based calculi for QBF. *Q-resolution (Q-Res)*, by Kleine Büning et al. [23], is a resolution-like calculus that operates on QBFs in prenex form where the matrix is a CNF. The rules are given in Figure 2.

$$\frac{}{C}\ (\text{Axiom}) \qquad\qquad \frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}\ (\text{Res})$$

C is a clause in the matrix. Variable x is existential. If $z \in C_1$, then $\neg z \notin C_2$.

$$\frac{C \cup \{u\}}{C}\ (\forall\text{-Red})$$

Variable u is universal. If $x \in C$ is existential, then $\text{lv}(x) < \text{lv}(u)$.

Fig. 2. The rules of Q-Res [23]

Long-distance resolution (LD-Q-Res) appears originally in the work of Zhang and Malik [31] and was formalized into a calculus by Balabanov and Jiang [2]. It merges complementary literals of a universal variable u into the special literal u^*. These special literals prohibit certain resolution steps. In particular, different literals of a universal variable u may be merged only if $\text{lv}(x) < \text{lv}(u)$, where x is the resolution variable. The rules are given in Figure 3. Note that the rules do not prohibit resolving $w^* \vee x \vee C_1$ and $u^* \vee \neg x \vee C_2$ with $\text{lv}(w) \leq \text{lv}(u) < \text{lv}(x)$ as long as $w \neq u$.

$$\frac{}{C}\ (\text{Axiom}) \qquad\qquad \frac{D \cup \{u\}}{D}\ (\forall\text{-Red}) \qquad\qquad \frac{D \cup \{u^*\}}{D}\ (\forall\text{-Red}^*)$$

C is a clause in the matrix. Literal u is universal and $\text{lv}(u) \geq \text{lv}(l)$ for all $l \in D$.

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}\ (\text{Res})$$

Variable x is existential. If for $l_1 \in C_1, l_2 \in C_2$, $\text{var}(l_1) = \text{var}(l_2) = z$ then $l_1 = l_2 \neq z^*$. U_1, U_2 contain only universal literals with $\text{var}(U_1) = \text{var}(U_2)$. For each $u \in \text{var}(U_1)$ we require $\text{lv}(x) < \text{lv}(u)$. If for $w_1 \in U_1, w_2 \in U_2$, $\text{var}(w_1) = \text{var}(w_2) = u$ then $w_1 = \neg w_2$, $w_1 = u^*$ or $w_2 = u^*$. U is defined as $\{u^* \mid u \in \text{var}(U_1)\}$.

Fig. 3. The rules of LD-Q-Res [2]

QU-resolution (QU-Res) [30] removes the restriction from Q-Res that the resolved variable must be an existential variable and allows resolution of universal variables. The rules are given in Figure 4.

$$\frac{}{C}\ \text{(Axiom)} \qquad\qquad \frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}\ \text{(Res)}$$

C is a clause in the matrix. If $z \in C_1$, then $\neg z \notin C_2$.

$$\frac{C \cup \{u\}}{C}\ (\forall\text{-Red})$$

Variable u is universal. If $x \in C$ is existential, then $\mathrm{lv}(x) < \mathrm{lv}(u)$.

Fig. 4. The rules of QU-Res [30]

LQU$^+$-Res [3] extends LD-Q-Res by allowing short and long distance resolution pivots to be universal. Two important clarifications need to be made, firstly the pivot is never a merged literal z^*, and the level restriction now must become an index restriction, to differentiate between universal variables on the same level. The rules are given in Figure 5.

$$\frac{}{C}\ \text{(Axiom)} \qquad\qquad \frac{D \cup \{u\}}{D}\ (\forall\text{-Red}) \qquad\qquad \frac{D \cup \{u^*\}}{D}\ (\forall\text{-Red}^*)$$

C is a clause in the matrix. Literal u is universal and $\mathrm{lv}(u) \geq \mathrm{lv}(l)$ for all $l \in D$.

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}\ \text{(Res)}$$

If for $l_1 \in C_1, l_2 \in C_2$, $\mathrm{var}(l_1) = \mathrm{var}(l_2) = z$ then $l_1 = l_2 \neq z^*$. U_1, U_2 contain only universal literals with $\mathrm{var}(U_1) = \mathrm{var}(U_2)$. For each $u \in \mathrm{var}(U_1)$ we require $\mathrm{ind}(x) < \mathrm{ind}(u)$, as variable x could be at the same level as u. If for $w_1 \in U_1, w_2 \in U_2$, $\mathrm{var}(w_1) = \mathrm{var}(w_2) = u$ then $w_1 = \neg w_2$, $w_1 = u^*$ or $w_2 = u^*$. U is defined as $\{u^* \mid u \in \mathrm{var}(U_1)\}$.

Fig. 5. The rules of LQU$^+$-Res [3]

More recently, several calculi based on *instantiation* of universal variables were introduced: $\forall$Exp+Res [20], IR-calc, and IRM-calc [6]. All these calculi operate on clauses that comprise only existential variables from the original QBF, which are additionally *annotated* by a substitution to some universal variables, e.g. $\neg x^{0/u_1 1/u_2}$. For any annotated literal l^σ, the substitution σ does not make assignments to variables at a higher quantification level than l, i.e. if $u \in \mathsf{dom}(\sigma)$, then u is universal and $\mathrm{lv}(u) < \mathrm{lv}(l)$. To preserve this invariant, we use the auxiliary notation $l^{[\sigma]}$, which for an existential literal l and an assignment σ to the universal variables filters out all assignments that are not permitted, i.e. $l^{[\sigma]} = l^{\{c/u \in \sigma \mid \mathrm{lv}(u) < \mathrm{lv}(l)\}}$.

The simplest instantiation-based calculus we consider is the calculus $\forall$Exp+Res, whose rules are presented in Figure 6. Any axiom in a proof is taken from the matrix by choosing a complete assignment to the universal variables. Resolution is defined as in the propositional case where annotated literals are considered as distinct variables.

$$\frac{}{\left\{l^{[\tau]} \mid l \in C, l \text{ is existential}\right\} \cup \{\tau(l) \mid l \in C, l \text{ is universal}\}}\ (\text{Axiom})$$

C is a clause from the matrix and τ is an assignment to all universal variables.

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

Fig. 6. The rules of $\forall$Exp+Res [20]

$$\frac{}{\left\{x^{[\tau]} \mid x \in C, x \text{ is existential}\right\}}\ (\text{Axiom})$$

C is a non-tautological clause from the matrix. $\tau = \{0/u \mid u \text{ is universal in } C\}$, where the notation $0/u$ for literals u is shorthand for $0/x$ if $u = x$ and $1/x$ if $u = \neg x$.

$$\frac{x^{\tau} \vee C_1 \qquad \neg x^{\tau} \vee C_2}{C_1 \cup C_2}\ (\text{Resolution}) \qquad\qquad \frac{C}{\mathsf{inst}(\tau, C)}\ (\text{Instantiation})$$

τ is an assignment to universal variables with $\mathsf{rng}(\tau) \subseteq \{0,1\}$.

Fig. 7. The rules of IR-calc [6]

The calculus IR-calc extends $\forall$Exp+Res by enabling partial assignments in annotations. To do so, we utilize the auxiliary operations of *completion* and *instantiation*. For assignments τ and μ, we write $\tau \veebar \mu$ for the assignment σ defined as follows: $\sigma(x) = \tau(x)$ if $x \in \mathsf{dom}(\tau)$, otherwise $\sigma(x) = \mu(x)$ if $x \in \mathsf{dom}(\mu)$. The operation $\tau \veebar \mu$ is referred to as *completion* because μ provides values for variables that are not defined in τ. The operation is associative and therefore we can omit parentheses. For an assignment τ and an annotated clause C the function $\mathsf{inst}(\tau, C)$ returns the annotated clause $\left\{l^{[\sigma \veebar \tau]} \mid l^{\sigma} \in C\right\}$. Then, the calculus IR-calc is defined in Figure 7. Axioms are taken from the matrix by assigning only those universal variables that appear in that matrix clause. Any clause can be instantiated by giving values to some universal variables (this is similar to *specialization* in first-order logic). Resolution is defined as in $\forall$Exp+Res.

Axiom and instantiation rules as in IR-calc in Figure 7.

$$\frac{x^{\tau \cup \xi} \vee C_1 \qquad \neg x^{\tau \cup \sigma} \vee C_2}{\mathsf{inst}(\sigma, C_1) \cup \mathsf{inst}(\xi, C_2)}\ (\text{Resolution})$$

$\mathsf{dom}(\tau)$, $\mathsf{dom}(\xi)$ and $\mathsf{dom}(\sigma)$ are mutually disjoint. $\mathsf{rng}(\tau) = \{0,1\}$

$$\frac{C \vee b^{\mu} \vee b^{\sigma}}{C \vee b^{\xi}}\ (\text{Merging})$$

$\mathsf{dom}(\mu) = \mathsf{dom}(\sigma)$. $\xi = \{c/u \mid c/u \in \mu, c/u \in \sigma\} \cup \{*/u \mid c/u \in \mu, d/u \in \sigma, c \neq d\}$

Fig. 8. The rules of IRM-calc [6]

The calculus IRM-calc further extends IR-calc by enabling annotations containing an assignment to the special symbol $*$. We call such assignments *extended assignments*. The rules of the calculus IRM-calc are presented in Figure 8. The symbol $*$ may be introduced by the merge rule, e.g. by collapsing $x^{0/u} \vee x^{1/u}$ into $x^{*/u}$. Once the symbol $*$ appears in an annotation, certain resolution steps are not permitted. For instance, $x^{*/u}$ and $\neg x^{c/u}$ cannot be resolved.

The calculus IR-calc p-simulates $\forall$Exp+Res as well as Q-resolution. The calculus IRM-calc p-simulates IR-calc as well as long-distance Q-resolution [6].

3 A lower bound in IR-calc for the formulas of Kleine Büning et al.

Our first main result is a proof complexity analysis of a well-known family of formulas KBKF(t) first defined by Kleine Büning, Karpinski and Flögel [23]. The formulas are claimed to be hard for Q-Res in [23]. However, a formal proof of their hardness has never been given, even though further hardness results build on this [14,3]. Here we prove that the KBKF(t) formulas are even hard for IR-calc, which is stronger than Q-Res (Theorem 22). This provides the first non-trivial lower bound for IR-calc, and further even separates the system from LD-Q-Res. As a by-product we also formally prove the hardness of KBKF(t) for Q-Res as Q-Res is simulated by IR-calc [6].

Definition 1 (Kleine Büning, Karpinski and Flögel [23]). *Consider the clauses*

$$\begin{array}{llll} C_- &= \{\neg y_0\} & & \\ C_0 &= \{y_0, \neg y_{1,0}, \neg y_{1,1}\} & & \\ C_i^0 &= \{y_{i,0}, x_i, \neg y_{i+1,0}, \neg y_{i+1,1}\} & C_i^1 = \{y_{i,1}, \neg x_i, \neg y_{i+1,0}, \neg y_{i+1,1}\} & \textit{for } i \in [t-1] \\ C_t^0 &= \{y_{t,0}, x_t, \neg y_{t+1}, \dots, \neg y_{t+t}\} & C_t^1 = \{y_{t,1}, \neg x_t, \neg y_{t+1}, \dots, \neg y_{t+t}\} & \\ C_{t+i}^0 &= \{x_i, y_{t+i}\} & C_{t+i}^1 = \{\neg x_i, y_{t+i}\} & \textit{for } i \in [t] \end{array}$$

The KBKF(t) *formulas are defined as the union of these clauses under the quantifier prefix* $\exists y_0, y_{1,0}, y_{1,1} \forall x_1 \exists y_{2,0}, y_{2,1} \forall x_2 \dots \forall x_{t-1} \exists y_{t,0}, y_{t,1} \forall x_t \exists y_{t+1} \dots y_{t+t}$.

Let us verify that the KBKF(t) formulas are indeed false QBF and — at the same time — provide some intuition about them. The existential player starts by playing $y_0 = 0$ because of clause C_-. Clause C_0 forces the existential player to set one of $y_{1,0}, y_{1,1}$ to 0. Assume the existential chooses $y_{1,0} = 0$ and $y_{1,1} = 1$. If the universal player tries to win, he will correspond with $x_1 = 0$, thus forcing the existential player again to set one of $y_{2,0}, y_{2,1}$ to 0. This continues for t rounds, leaving in each round a choice of $y_{i,0} = 0$ or $y_{i,1} = 0$ to the existential player, to which the universal corresponds by setting x_i accordingly. Finally, the existential player is forced to set one of $y_{t+1}, \dots, y_{2t}$ to 0. This will contradict one of the clauses $C_{t+1}^0, C_{t+1}^1, \dots, C_{2t}^0, C_{2t}^1$, and the universal player wins.

It is clear from this explanation, that the existential player has exponentially many choices and the universal player likewise needs to uniquely correspond to all these choices to win. The aim of this section is to show that IR-calc and therefore Q-Res in some sense need to go through all these exponentially many options in order to refute the formula, thus forcing IR-calc and Q-Res proofs of exponential size.

Syntactically, KBKF(t) are *existential Horn formulas*, i.e., they contain at most one positive existential literal per clause. In fact, they even have a stronger property: C_- is the only clause without a head (a positive existential literal). We will strengthen this in the next lemma by a simple modification such that now all clauses have a head.

Lemma 2. *We can transform every IR-calc refutation π of* $\mathrm{KBKF}(t)$ *into a IR-calc proof π' of y_0 from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$*. We perform this by:*

1. *deleting every instance of the axiom $\{\neg y_0\}$;*
2. *for every clause without a positive existential literal we add the literal y_0 to the clause with the empty annotation.*

After this transformation, which preserves proof length, we can focus on proofs of y_0 from $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$. Exploiting that all axioms now contain exactly one positive literal we show a number of invariants, which hold for all clauses in all IR-calc proofs of the formulas.

Lemma 3. *Let C be an annotated clause in an IR-calc proof of y_0 from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$*. Then the following invariants hold for C:*

1. *C has exactly one positive literal $y^A_{h,a}$ for $h \leq t$ or y^A_h for $h > t$ (or y_0 with no annotation). We call this unique literal the* head *of C and use the indices h and a also in the following invariants to denote its position as well as A for its annotation.*
2. *If, for some $j \in [2t], b \in \{0,1\}$ and B some annotation, $\neg y^B_{j,b} \in C$ (or $\neg y^B_j \in C$), then $j > h$. i.e. the literals in the body are always at a higher quantification level than the head.*
3. *If $\neg y^B_{j,b} \in C$ (or $\neg y^B_j \in C$), then $A \cup \{a/x_h\} \subseteq B$, where all extra annotations in B are of the form c_k/x_k for $k > h$. This invariant acts vacuously for $h > t$ where the clauses contain no negative literals.*
4. *If $\neg y^B_{j,b} \in C$ (or $\neg y^B_j \in C$) then for all k, $h \leq k < j$ (or $h \leq k \leq t$, when $j > t$) there is $c_k \in \{0,1\}$ such that $c_k/x_k \in B$, i.e. all universal variables between h and j are instantiated.*
5. *If $\neg y^B_{j,b} \in C$ with $j \leq t$, then for $k \in [t], d \in \{0,1\}$ and D some annotation, there is no $\neg y^D_{k,d} \in C$ nor $\neg y^D_{t+k} \in C$ such that $B \cup \{b/x_j\} \subseteq D$.*

Proof. We will simultaneously prove all the invariants by one induction on the number of lines in the proof before C.

We start with the base case which requires to just look at the axioms. Invariant 1 holds as we no longer have $\{\neg y_0\}$ present in the axioms. Invariant 2 holds in all our axioms. For Invariant 3 we only need to consider the axioms C^c_i with negative existential literals. The head of C^c_i is $y_{i,c}$ and there are no universal variables of lower level than the head, hence its annotation $A = \emptyset$. The axiom is instantiated so that c/x_i is added to both negative literals, hence we satisfy the invariant. Invariant 4 holds by the same reasons. Invariant 5 holds, because C_0 and all C^c_j with $j \leq t$ are the only clauses with negative existential literals, these are all of the same level.

For the inductive step there are two possibilities: either C is derived from instantiation or by resolution.

Suppose first that C is derived from instantiation of clause D. Invariant 1 holds as instantiation does not change the polarities of the literals. Invariant 2 holds as we do not change the indices of the literals. For Invariant 3 we use Invariant 2 to know that $y^A_{h,a}$ is the lowest level literal in the clause D. Any annotation involving x_l with $l < h$ is therefore both added to A and to the annotations of all other literals. Invariant 4 holds as these annotations are not removed. For Invariant 5 we know from Invariants 2 and 4 that for any $\neg y^B_{j,b} \in D$ and $\neg y^E_{k,e} \in D$ (or $\neg y^E_k \in D$) with $k > j$ there is some $c/x_l \in B \cup \{b/x_j\}$ such that $(1-c)/x_l \in E$ and this conflict does not change by instantiation.

Now suppose C is derived from resolving D_1 and D_2. Without loss of generality we let the resolved variable be $y^E_{k,e}$ in D_2 (if y^E_{t+k} is the resolved variable we only remove a negative literal from D_1, which does not affect the invariants). Invariant 1 holds as there are two positive literals between D_1 and D_2, but the head $y^E_{k,e}$ of D_2 gets removed by resolution.

Invariant 2 holds as the head $y^A_{h,a}$ of C is the head of D_1. If we have a negative literal $\neg y^B_{j,b} \in C$, then $\neg y^B_{j,b} \in D_1$ or $\neg y^B_{j,b} \in D_2$. If in D_1 then $j > h$ by Invariant 2 for D_1. If in D_2 then $j > k$ by Invariant 2 for D_2. As $\neg y^E_{k,e} \in D_1$ we get $k > h$ again by Invariant 2 for D_1. Therefore $j > h$ and Invariant 2 holds for C. This works similarly if the negative literal is $\neg y^B_j$.

For Invariant 3 we again use the fact that the head $y^A_{h,a}$ of C is also the head of D_1. If negative literal $\neg y^B_{j,b} \in C$ comes from D_1 then $A \cup \{a/x_h\} \subseteq B$ by Invariant 3 for D_1. If, on the other hand, negative literal $\neg y^B_{j,b} \in C$ comes from D_2 then $E \cup \{e/x_k\} \subseteq B$ by Invariant 3 for D_2. However, as $\neg y^E_{k,e} \in D_1$ then $A \cup \{a/x_h\} \subseteq E \subseteq E \cup \{e/x_k\} \subseteq B$. Likewise for an annotation in B of level lower than $\mathrm{lv}(y_{h,a})$, it must be in E and hence in A. This works similarly if the negative literal is $\neg y^B_j$.

For Invariant 4, all literals in C that come from D_1 already fulfil the condition. For the literals coming from D_2 we use Invariant 3. Since $\neg y^E_{k,e} \in D_1$ the annotation E already contains all the necessary assignments for head $y^A_{h,a}$. But since $E \cup \{e/x_k\} \subseteq B$ for any $\neg y^B_{j,b} \in D_2$ (or $\neg y^B_j$), then together with Invariant 4 for D_2, these literals have the required annotations for the new head $y^A_{h,a}$.

For Invariant 5 we need to check that if $\neg y^B_{j,b} \in D_1$ and $\neg y^F_{l,f} \in D_2$ (or $\neg y^F_l \in D_2$), then there is some conflict in the annotations (by using Invariants 2 and 4 for C). By Invariant 5 for D_1 we know that there is some $c/x_l \in B \cup \{b/x_j\}$ such that $(1-c)/x_l \in E$. By Invariant 3 we have $E \cup \{e/x_k\} \subseteq F$, hence $(1-c)/x_l \in F$. □

We will now start our lower bound argument. The overall idea is as follows. Consider the clauses $C^0_{t+1}, C^1_{t+1}, \ldots, C^0_{t+t}, C^1_{t+t}$. These contain only a single positive existential literal. We will show in Lemma 9 that the negative versions of these literals appear together and get the same annotations (unless we make progress in the proof). One may be tempted to resolve the clauses with the corresponding negative literals (initially in C^0_t, C^1_t). However, one must choose between C^0_{t+i}, C^1_{t+i} for each $\neg y_{t+i}$ and do this for each i.

We will collect these choices in a set Σ associated with each clause, such that we can infer from Σ which choices were made by directly looking at the clause. We define $\Sigma(C)$ for a clause C in Definition 4 and show that we are counting the right thing in Lemmas 7, 10 and 11. More precisely, we show in Lemma 7 that axioms have empty Σ and that instantiation steps do not change Σ at all (Lemma 10). In a resolution step $\frac{D_1 \quad D_2}{D}$, the set $\Sigma(D)$ either equals $\Sigma(D_1) \cup \Sigma(D_2)$ or grows by exactly one new element (Lemma 11). In some sense, we only make progress in the proof in the latter case, and we need exponentially many resolution steps of this kind. Putting everything together we find that by the end of the proof we must have collected all the exponentially many choices in $\Sigma(y_0)$, implying an exponential lower bound to the proof length (Theorem 12).

We now give the formal arguments starting with the definition of Σ.

Definition 4. *Let C be a clause in an IR-calc proof of y_0 from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$*. We define the set $\Sigma(C)$ of complete annotations (to all x_i) by the following rules.*

1. *$\Sigma(C) = \emptyset$ when $C = \{y^B_{t+j}\}$* (type-1 clause).

Assume now that C is not type-1 and has the head $y^{A}_{h,a}$.

2. *$\Sigma(C) = \emptyset$ when there exists $j < h$ such that x_j is not given a value in the annotation A* (type-2 clause)*.*
3. *Otherwise* (type-3 clause)*, $\Sigma(C)$ is defined by the following process of adding and removing assignments according to C, which now has complete annotations for each literal by Invariant 4.*
   We start by initialising $\Sigma(C)$ as all complete annotations X to $x_1, \dots, x_t$ such that $A \cup \{a/x_h\} \subseteq X$ (if y_0 is the head we add the complete set of annotations).
   For each $\neg y^{B}_{j,b} \in C$ with $j \le t$ we remove from $\Sigma(C)$ all complete annotations X such that $B \cup \{b/x_j\} \subseteq X$. Finally, we remove annotations B for $y^{B}_{j} \in C$ with $j > t$ (note that B is necessarily complete by invariants 3 and 4).

Obviously, any clause C with head y_{t+j} for $j > 0$ is a type-1 clause. Further, C is an instantiation of C^{0}_{t+j} or C^{1}_{t+j} whose annotated versions are in fact unit clauses.

Example 5. Let $t = 5$ and assume that the clause

$$C = \{y^{\bar{x}_1}_{2,1}, \neg y^{\bar{x}_1 x_2}_{3,1}, \neg y^{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5}_{6}, \neg y^{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5}_{7}, \neg y^{\bar{x}_1 x_2 \bar{x}_3 x_4 x_5}_{7}\}$$

is derived in an IR-calc proof of KBKF(5). Because the annotation of the head $y^{\bar{x}_1}_{2,1}$ is as complete as possible, the clause C is a type-3 clause, so $\Sigma(C)$ is defined by the adding/removing process. This happens in the following steps:

- $y^{\bar{x}_1}_{2,1}$ means we add the $8 = 2^{(5-2)}$ complete annotations with prefix $\bar{x}_1 x_2$.
- $\neg y^{\bar{x}_1 x_2}_{3,1}$ means we then remove the 4 complete annotations with prefix $\bar{x}_1 x_2 x_3$.
- $\neg y^{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5}_{6}$ and $\neg y^{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5}_{7}$ means we remove the annotation $\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$.
- $\neg y^{\bar{x}_1 x_2 \bar{x}_3 x_4 x_5}_{7}$ means we remove the annotation $\bar{x}_1 x_2 \bar{x}_3 x_4 x_5$.

Therefore we obtain the set of annotations $\Sigma(C) = \{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 x_5, \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{x}_5\}$ of size 2.

Remark 6. For a type-3 clause C we have the following properties of $\Sigma(C)$:

- We only remove an annotation when it was originally added from the presence of the head $y^{A}_{h,a}$. This is true by Invariant 3.
- Unless $y^{X}_{j} \in C$ for $j > t$, we only remove an annotation X at most once from $\Sigma(C)$. This holds by Invariant 5. If $\neg y^{X}_{j} \in C$ for $j > t$, then X can be removed from Σ up to t times by other $\neg y^{X}_{l} \in C$. In this case, however, X is not removed by any $\neg y^{B}_{j,b} \in C$ with $j \le t$ by Invariant 5.

An important fact is that for axioms we get empty Σ as we verify in the next lemma.

Lemma 7. *For each clause $C \in \mathrm{KBKF}(t) \setminus \{\neg y_0\}$, instantiated as an IR-calc axiom C^{B}, we get $\Sigma(C^{B}) = \emptyset$.*

Proof. For axiom C_0 we first add all annotations to Σ, but then due to the presence of $\neg y_{1,0}$ and $\neg y_{1,0}$ remove all annotations starting with $0/x_1$ and $1/x_1$, respectively. This results in $\Sigma(C_0) = \emptyset$.

Using $C^{0}_{1} = \{y_{1,0}, x_1, \neg y_{2,0}, \neg y_{2,1}\}$ as an IR-calc axiom results in $\{y_{1,0}, \neg y^{0/x_1}_{2,0}, \neg y^{0/x_1}_{2,1}\}$. Computing Σ of this clause, we first add all annotations starting with $0/x_1$, but then remove all annotations starting with $(0/x_1, 0/x_2)$ and $(0/x_1, 1/x_2)$, yielding again empty Σ. Analogous reasoning applies to C^{1}_{1}.

When using clauses C_i^0, C_i^1 with $2 \leq i \leq t$ as IR-calc axioms, we obtain type-2 clauses, which have empty Σ by definition. The remaining clauses C_{t+i}^0, C_{t+i}^1 give rise to type-1 clauses, again with empty Σ. □

It will be crucial for our lower bound argument to understand how $\Sigma(C)$ changes when we go through the clauses C in the proof. For this we first need two technical lemmas on the structure of IR-calc proofs of $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$.

Lemma 8. *In an IR-calc proof from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$*, a type-2 clause cannot be resolved with a type-3 clause.*

Proof. Suppose we have a resolution step between a type-2 and a type-3 clause. The resolved variable has a complete annotation as all variables in type-3 clauses have them. However, because of Invariants 3 and 4 the head of the type-2 clause must have a complete annotation, contradicting our assumption. □

Lemma 9. *Let C be a clause in an IR-calc proof from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$*. If $\neg y_{t+j}^B \in C$ and $\neg y_{t+l}^B \notin C$ with $j, l > 0$, then there is an annotation of x_l in B.*

Proof. We proceed by induction on the number of lines to derive C. The base case is vacuously true as all $\neg y_{t+j}$ literals get introduced in the same axioms.

For the inductive step we distinguish whether C is derived by instantiation or by resolution. In the case that C is derived by instantiation of D we do not lose any annotation, so the hypothesis remains true.

If C is derived by resolving D_1 and D_2, the claim still holds if the resolved variable is different from any y_{t+l}^B. In the case that $\neg y_{t+l}^B \in C$ and we resolve on y_{t+l}^B, the clause C must be resolved with some instantiation of axiom C_{t+l}^0 or C_{t+l}^1. This instantiation introduces an annotation of x_l, hence the same annotation of x_l appears in B. □

The next two lemmas are the key to our lower bound. We show first that the set Σ is not affected by instantiation steps. In Lemma 11 we will then analyse how Σ changes in a resolution step.

Lemma 10. *Suppose in an IR-calc proof from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$ *we instantiate clause D to get clause D'. Then $\Sigma(D) = \Sigma(D')$.*

Proof. If D is a type-1 clause it remains a type-1 clause under instantiation. Hence $\Sigma(D')$ remains empty. If D is a type-3 clause it has complete annotations; hence under instantiation it does not change. If D is a type-2 clause then the only problem arises when instantiation makes the annotations complete, i.e., the clause turns into a type-3 clause. We will use induction on the number of lines until D.

Induction Hypothesis: Let D be a type-2 clause that can be instantiated by σ to get a type-3 clause D'. Then $\Sigma(D')$ is empty.

For the base case, we observe that instantiating any type-2 axiom always gives empty Σ. For the inductive step, we can derive D by instantiation or resolution. For the inductive argument we can ignore the case where D is derived by instantiation as that instantiation could have been incorporated into σ.

Let D now be derived by resolving two clauses D_1 and D_2. Assume without loss of generality that D_1 is a type-2 clause. By Lemma 8 the other clause D_2 must be type 1 or type 2.

Suppose first that D_2 is a type-1 clause $\{y^B_{t+j}\}$. Let D'_1 be the instantiation of D_1 by σ, since σ gives a complete assignment to the annotations in D it must also do so in D_1 by Invariant 3. By inductive hypothesis we have $\Sigma(D'_1) = \emptyset$. The only result of the resolution step with D_2 is the removal the literal $\neg y^B_{t+j}$. In order for this to have any effect on $\Sigma(D)$, there can be no other $\neg y^B_{t+l}$ (for $1 \leq l \leq t$, $l \neq j$) in D_1. Hence by Lemma 9 B contains annotations for all x_l with $1 \leq l \leq t$, $l \neq j$. Further B contains an annotation of x_j as $D_2 = \{y^B_{t+j}\}$ must contain such an annotation. Therefore B is a complete assignment to $x_1, \ldots, x_n$. By Invariant 3 this can only happen when D_1 is a type-3 clause rather than a type-2 clause.

Suppose instead, D_2 is a type-2 clause, and without loss of generality the resolved variable is positive in D_2, i.e., it is the head of D_2. Let D'_2 be the instantiation of D_2 by σ. Since σ gives a complete assignment to the annotations in D it must also do so in D_2, i.e., D'_2 is type 3. This holds since D_2 is type 2 and therefore has a negative literal; with Invariant 3 this implies that D'_2 is type 3. By inductive hypothesis we therefore have $\Sigma(D'_2) = \emptyset$. When computing $\Sigma(D')$, the lack of the negative literal on the resolved variable means we may have additional elements in $\Sigma(D')$. However, these were exactly the assignments that were added by the head in D'_2 and so we know — as $\Sigma(D'_2)$ is empty — that we have the sufficient literals in D' to remove all elements in $\Sigma(D')$. □

Lemma 11. *Let $\sqcup$ denote disjoint union. Suppose in an IR-calc proof from* $\mathrm{KBKF}(t) \setminus \{\neg y_0\}$ *we resolve D_1 with D_2 to get clause D, where the resolved variable is positive in D_2.*

If D_1 is a type-3 clause that is resolved with the type-1 clause $D_2 = \{y^B_{t+j}\}$ for $j > 0$ and there is no $k > 0$, $k \neq j$ such that $\neg y^B_{t+k} \in D_1$, then $\Sigma(D) = \Sigma(D_1) \sqcup \Sigma(D_2) \sqcup \{B\} = \Sigma(D_1) \sqcup \{B\}$. Otherwise $\Sigma(D) = \Sigma(D_1) \sqcup \Sigma(D_2)$.

Proof. In the first case, D_1 is a type-3 clause and D_2 is a type-1 clause. Then the resolution step is the same as removing $\neg y^B_{t+j}$ from the clause D_1. The resolvent D must be a type-3 clause as all annotations remain. Because there is no $k > 0$, $k \neq j$ such that $\neg y^B_{t+k} \in D_1$, we can use Lemma 9 to know B is complete and Remark 6 to infer that $B \in \Sigma(D)$. If we otherwise resolve a type-3 clause D_1 with a type-1 clause D_2, but there is another $\neg y^B_{t+k} \in D_1$, then the same assignments are added and deleted in $\Sigma(D)$ as in $\Sigma(D_1)$, hence $\Sigma(D) = \Sigma(D_1) = \Sigma(D_1) \sqcup \Sigma(D_2)$.

Consider now the remaining cases. If we resolve a type-1 clause with a type-2 clause, then we obtain a type-2 clause, hence Σ remains empty. Likewise, resolving two type-2 clauses results in a type-2 clause and therefore again empty Σ. By Lemma 8, we cannot resolve type-2 with type-3 clauses.

Therefore the last case is when two type-3 clauses are resolved. Let D_2 provide the positive resolved literal $y^E_{k,e}$. Because $y^E_{k,e}$ is the head of D_2, every annotation $X \in \Sigma(D_2)$ has $E \cup \{e/x_k\} \subseteq X$. As $\neg y^E_{k,e} \in D_1$, the sets of assignments $\Sigma(D_1)$ and $\Sigma(D_2)$ are disjoint. But also there is no annotation $Y \in \Sigma(D_1)$ with $E \cup \{e/x_k\} \subseteq Y$ because of the presence of $\neg y^E_{k,e}$ in D_1 and Invariant 5. Therefore $\Sigma(D)$ is the union of $\Sigma(D_1)$ and $\Sigma(D_2)$ because in our adding/removing process we only get rid of instructions from $y^E_{k,e}$ and $\neg y^E_{k,e}$ which cancel out anyway, and keep all other instructions.

□

We can now deduce that all proofs of $\mathrm{KBKF}(t)$ in IR-calc are of at least exponential size.

Theorem 12. *All proofs of* $\mathrm{KBKF}(t)$ *in IR-calc have length at least 2^t.*

Proof. We will show that all IR-calc proofs of y_0 from $\mathrm{KBKF}(t)\setminus\{\neg y_0\}$ are of exponential size. By Lemma 2 each refutation of $\mathrm{KBKF}(t)$ can be transformed into one of these in polynomial time. Hence each refutation of $\mathrm{KBKF}(t)$ must be of exponential size.

Consider now an IR-calc proof $\pi = (C_1, C_2, \ldots, C_m)$ of y_0 from $\mathrm{KBKF}(t)\setminus\{\neg y_0\}$ and define $s_i = |\bigcup_{j=1}^{i} \Sigma(C_j)|$. By Lemma 7, the axioms all have empty Σ, hence $s_1 = 0$. By Definition 4, the set $\Sigma(y_0)$ contains all 2^t complete annotations, therefore $s_m = 2^t$. Progressing in the proof from the axioms to y_0, we therefore build up the set Σ from an empty to an exponential-size set. If the clause C_{i+1} is an axiom or derived by instantiation, then $s_i = s_{i+1}$ by Lemmas 7 and 10. For a resolution step, we have $s_{i+1} \le s_i + 1$ by Lemma 11. Therefore the proof π contains at least 2^t resolution steps. □

Since IR-calc simulates Q-Res [6], we get as a corollary the hardness of $\mathrm{KBKF}(t)$ for Q-Res as already stated in [23].

Corollary 13. *All proofs of* $\mathrm{KBKF}(t)$ *in Q-Res are of at least exponential size.*

As the formulas $\mathrm{KBKF}(t)$ are easy for long-distance and universal resolution [14,30] we obtain the following exponential separations.

Corollary 14. *IR-calc does not simulate LD-Q-Res, QU-Res, LQU-Res, LQU$^+$-Res, or IRM-calc.*

Proof. The formulas $\mathrm{KBKF}(t)$ are known to have polynomial-size proofs in LD-Q-Res [14] and QU-Res [30], and therefore by the known simulations also in LQU-Res [3,14], LQU$^+$-Res [3,14] and IRM-calc [6,14]. □

4 A lower bound for Q-resolution via strategy extraction

In this section we show a new and conceptually very different lower bound for Q-Res. This lower bound constitutes in fact a new lower bound technique that is widely applicable (cf. Section 6). We illustrate this technique here with an exponential lower bound for parity formulas in Q-Res. This provides a separation between Q-Res and ∀Exp+Res.

The lower bound argument rests on strategy extraction, which is a widely used paradigm in QBF solving and proof systems. We recall that Q-Res admits strategy extraction via a computationally very restricted model, namely decision lists.

Definition 15 (decision list [28]). *A* decision list *is a finite sequence of pairs* (t_i, c_i) *where* t_i *is a term and* $c_i \in \{1, 0\}$ *is a Boolean constant. Additionally, the last term is the empty term, semantically equivalent to true.*

For an assignment μ*, a decision list* $D = (t_1, c_1), \ldots, (t_n, c_n)$ *evaluates to* c_i *if* i *is the least index such that* $\mu \models t_i$*. We say that* (t_i, c_i) triggers *under* μ *if this condition is satisfied. Observe that a decision list unequivocally defines a Boolean function since* $t_n = 1$*.*

Winning strategies in form of decision lists can now be efficiently extracted from Q-Res proofs:

Theorem 16 (Balabanov and Jiang [2]). *Given a Q-Res refutation* π *of QBF* ϕ*, there exists a winning strategy for the universal player for* ϕ *such that for each universal variable* u *of* ϕ *the winning strategy can be represented as a Boolean function* f_u *that is expressible as a decision list whose size is polynomial in* $|\pi|$*.*

Balabanov and Jiang use a different form than decision lists but it is semantically equivalent. We believe that decision lists are more intuitive for our purposes.

The general idea behind our lower bound technique is as follows. First, we observe that we can define a family of QBFs ϕ_f, such that every winning strategy of the universal player must compute a unique boolean function f (Lemma 18). If we then know that strategy extraction is possible by a weak computational model, say AC^0, we can carefully choose the boolean formula ϕ_f such that the unique winning strategy f cannot be computed by AC^0 circuits. As the extracted strategy is polynomial in the proof, this implies a lower bound on the proof size. Thus we immediately turn circuit lower bounds to lower bounds for the proof size.

We will now implement this idea for the *parity function* $\textsc{Parity}(x_1, \dots, x_n) = x_1 \oplus \cdots \oplus x_n$, which is the classical example of a function not computable in AC^0.

Theorem 17 (Furst, Saxe, Sipser [15]). $\textsc{Parity} \notin \mathsf{AC}^0$. *In fact, every non-uniform family of bounded-depth circuits computing* $\textsc{Parity}$ *is of exponential size.*

We first observe how to construct a QBF that forces a unique winning strategy.

Lemma 18. *Consider the QBF* $\exists x_1, \dots, x_n \forall z. (z \vee \phi_f) \wedge (\neg z \vee \neg \phi_f)$*, where* ϕ_f *is a propositional formula depending only on the variables* $x_1, \dots, x_n$*. Let* $f : 2^n \to \{0, 1\}$ *be a Boolean function that returns 1 iff* ϕ_f *evaluates to true. Then there is a unique strategy for the universal player for* z*, which is* $z \leftarrow f$*.*

Proof. The strategy for z may only depend on the variables $x_1, \dots, x_n$ and it must be so that the matrix evaluates to false under the given assignment μ to the x_i variables. By inspecting the matrix, z must be set to 0 whenever ϕ_f evaluates to 0 and the other way around. □

We will now use this idea specifically for the parity function.

Parity formulas. Consider the QBF $\Phi = \exists X \forall z \exists T.\ (F^+ \wedge F^-)$ where F^+ is a CNF encoding of $(z \vee \textsc{Parity}(X))$ and F^- encodes $(\neg z \vee \neg \textsc{Parity}(X))$. Both F^+ and F^- use additional variables in T. More precisely, for $N > 1$ define $\text{Q}\textsc{Parity}_N$ as follows. Let $\text{xor}(o_1, o_2, o)$ be the set of clauses

$$\{\neg o_1 \vee \neg o_2 \vee \neg o,\ o_1 \vee o_2 \vee \neg o,\ \neg o_1 \vee o_2 \vee o,\ o_1 \vee \neg o_2 \vee o\},$$

which defines o to be equal to $o_1 \oplus o_2$. Define $\text{Q}\textsc{Parity}_N$ as

$$\exists x_1, \dots, x_N \,\forall z \,\exists t_2, \dots, t_N.\ \text{xor}(x_1, x_2, t_2) \cup \bigcup_{i=3}^{N} \text{xor}(t_{i-1}, x_i, t_i) \cup \{z \vee t_N, \neg z \vee \neg t_N\}.$$

Note that since we want to encode parity as a CNF, i.e. a bounded-depth formula, and $\textsc{Parity} \notin \mathsf{AC}^0$, we need to use further existential variables (recall that existential AC^0 characterises all of NP). Choosing existential variables t_i to encode the prefix sums $x_1 \oplus \cdots \oplus x_i$ of the parity $x_1 \oplus \cdots \oplus x_N$ provides the canonical CNF formulation of parity.

To use the lower bound of Theorem 17 we need to verify that $\mathsf{Q\text{-}Res}$ allows strategy extraction in AC^0. This holds as every decision list can be turned into a bounded-depth circuit.

Lemma 19. *A function* f_D *that can be represented as a decision list* D *of polynomial size is in* AC^0.

Proof. Let $S = \{i \mid (t_i, 1) \in D\}$ be the indices of all pairs in D with 1 as the second component. Observe that f_D evaluates to 1 under μ iff one of the t_i with $i \in S$ triggers under μ. For each t_i with $i \in S$ construct a function $f_i = t_i \wedge \bigwedge_{l=1}^{i-1} \neg t_l$. Construct a circuit for the function $f_S = \bigvee_{i \in S} f_i$. The function f_S is equal to f_D and can be computed in AC^0 as all t_i are just terms. □

We can now put everything together and turn the circuit lower bound of Theorem 17 into a lower bound for proof size in Q-Res.

Theorem 20. *Any Q-Res refutation of* QPARITY_N *is of exponential size in* N*.*

Proof. By Lemma 18 there is a unique strategy for the variable z in QPARITY_N, which is the PARITY function on N variables. From Theorem 16, there is a polynomial-time algorithm for constructing a decision list D_N from any Q-Res refutation of QPARITY_N. Such decision list can be converted in polynomial time into a circuit with bounded depth by Lemma 19. Hence, the decision list must be of exponential size in N due to Theorem 17. □

In contrast to this lower bound, the parity formulas are easy in ∀Exp+Res.

Lemma 21. *The formulas* QPARITY_N *have polynomial-size ∀Exp+Res refutations.*

Proof (sketch). Expand z in both polarities, which generates the clauses $\text{xor}(x_1, x_2, t_2^{0/z}) \cup \bigcup_{i=3}^{N} \text{xor}(t_{i-1}^{0/z}, x_i, t_i^{0/z}) \cup \{t_N^{0/z}\}$ and $\text{xor}(x_1, x_2, t_2^{1/z}) \cup \bigcup_{i=3}^{N} \text{xor}(t_{i-1}^{1/z}, x_i, t_i^{1/z}) \cup \{\neg t_N^{1/z}\}$.

Inductively, for $i = 2, \ldots, N$ derive clauses representing $t_i^{0/z} = t_i^{1/z}$. This lets us derive a contradiction using the clauses $t_N^{0/z}$ and $\neg t_N^{1/z}$. □

Theorem 20 together with Lemma 21 immediately give the following separations.

Theorem 22. *Q-Res does not simulate ∀Exp+Res, IR-calc, IRM-calc.*

This also has consequences for the complexity of strategy extraction in ∀Exp+Res.

Corollary 23. *Winning strategies for ∀Exp+Res cannot be computed in* AC^0*. This even holds when the system ∀Exp+Res is restricted to formulas with constant quantifier complexity.*

Proof. Consider again the family of formulas QPARITY_N. As these formulas have polynomial-size ∀Exp+Res refutations by Lemma 21, we cannot extract strategies in AC^0 as these would compute parity. □

5 Extending the lower bound to long-distance and universal resolution

We now aim to extend the lower bound from the previous section to stronger QBF proof systems using long-distance resolution and resolution on universal variables. For this we cannot directly use the strategy extraction method from the last section. However, we will slightly modify the parity formulas and then reduce the hardness of the modified formulas in the stronger systems to the hardness of QPARITY in Q-Res. As the modified formulas remain easy for ∀Exp+Res, these lower bounds imply lots of new separations between the proof systems involved.

5.1 A lower bound for parity in long-distance resolution

We start by extending the lower bound to LD-Q-Res, which will provide a separation of LD-Q-Res and $\forall$Exp+Res. For this we consider a variant of the parity formulas from the last section. Let $\mathrm{xor}_l(o_1, o_2, o, y)$ be the set of clauses

$$\{y \vee \neg o_1 \vee \neg o_2 \vee \neg o,\ y \vee o_1 \vee o_2 \vee \neg o,\ y \vee \neg o_1 \vee o_2 \vee o,\ y \vee o_1 \vee \neg o_2 \vee o\}$$

(xor_l defines o to be equal to $o_1 \oplus o_2$ if $y = 0$). Define $\mathrm{LQP}\text{ARITY}_N$ as

$$\begin{aligned} &\exists x_1, \dots, x_N \forall z \exists t_2, \dots, t_N. \\ \mathrm{xor}_l(x_1, x_2, t_2, z) \cup &\bigcup_{i=3}^{N} \mathrm{xor}_l(t_{i-1}, x_i, t_i, z) \\ \cup\, \mathrm{xor}_l(x_1, x_2, t_2, \neg z) \cup &\bigcup_{i=3}^{N} \mathrm{xor}_l(t_{i-1}, x_i, t_i, \neg z) \\ &\cup \{z \vee t_N, \neg z \vee \neg t_N\}. \end{aligned}$$

It is easy to verify that the same arguments as for QParity in Section 4 also apply to LQParity, yielding:

Proposition 24. *The formulas* $\mathrm{LQP}\text{ARITY}_N$ *have polynomial-size* $\forall$*Exp+Res refutations, but require exponential-size Q-Res refutations.*

We now want to show that LQParity is hard for LD-Q-Res by arguing that long-distance steps do not help to refute these formulas. In the next two lemmas we will show that this actually applies to all QBFs Φ meeting the following condition.

Definition 25. *We say that z is* completely blocked in a QBF Φ, *if all clauses of Φ contain the universal variable z and some existential literal l such that* $\mathrm{lv}(z) < \mathrm{lv}(l)$.

Lemma 26. *Let Φ be a QBF and z be completely blocked in Φ. Let further C be a clause derived from Φ by LD-Q-Res. If C contains some existential literal l such that $\mathrm{lv}(z) < \mathrm{lv}(l)$, then $z \in C$ or $\neg z \in C$, or $z^* \in C$.*

Proof. We prove the lemma by induction on derivation depth. The hypothesis is established by the clauses in the matrix of Φ due to the condition that z must be in all matrix clauses and also that all these clauses contain some existential literal that blocks z.

The hypothesis is preserved by $\forall$-reduction because a literal over z cannot be $\forall$-reduced if the clause contains an existential literal l with $\mathrm{lv}(z) < \mathrm{lv}(l)$.

Consider now two clauses $C_1 = D_1 \vee x$ and $C_2 = D_2 \vee \neg x$ resolved into the clause C_3. If C_3 contains some literal l such that $\mathrm{lv}(z) < \mathrm{lv}(l)$, then one of C_1, C_2 must contain l and from induction hypothesis it must also contain the variable z, which then appears in C_3. □

Lemma 27. *Let Φ be a QBF such that z is completely blocked in Φ and let π be a refutation of Φ such that the variable z is $\forall$-reduced as early as possible. Then the derivation of the empty clause in π does not contain z^* in any of its clauses.*

Proof. Assume that we have a clause C in π that contains z^*. We will argue that C is not necessary to derive the empty clause $\bot$, i.e., there is no path in π from C to $\bot$. Since z^* does not appear in any of the matrix clauses, there must be a resolution step where it is introduced. Consider any such two clauses $C_1 = D_1 \vee x \vee z$ and $C_2 = D_2 \vee \neg x \vee \neg z$ resolved into $C = D_3 \vee z^*$. From the assumption that $\forall$-reductions are carried out as soon as possible, in both clauses C_1 and C_2 there must be some literals that block z and $\neg z$, respectively. From the conditions on LD-Q-Res, x or $\neg x$ cannot be the blocking literal (it must be that $\mathrm{lv}(x) < \mathrm{lv}(z)$ upon merging). This means that C contains at least one literal b that blocks z^*.

Now we argue that b cannot be resolved away. For contradiction assume that there is a resolution step of some C' and D on b where there is a path from C to C'. Moreover, let that be the first resolution step on b, i.e., b appears in all clauses on the path between C and C'. From Lemma 26, the clause D must contain a literal on the variable z. But this contradicts the conditions of LD-Q-Res because resolution steps are not permitted on literals b with $\mathrm{lv}(z) < \mathrm{lv}(b)$ if one of the antecedents contains a merged literal z^* and the other contains some literal on z. This means that C does not participate in the derivation of the empty clause $\bot$. □

This enables us to prove the hardness of LQParity in LD-Q-Res.

Theorem 28. *Any refutation of* $\mathrm{LQParity}_N$ *in LD-Q-Res is exponential in* N*.*

Proof. Any LD-Q-Res refutation π can be in polynomial time translated into a refutation π' such that $\forall$-reductions are carried out as soon as possible (such a refutation has clauses that are equal to the clauses of π or some universal literals are missing). From Lemma 27, the derivation of $\bot$ in π' contains no occurrences of the merged literal z^*, hence any such clauses can be removed from the refutation. Therefore π' is in fact also a Q-Res refutation. Hence, π must be exponential in N due to Proposition 24. □

As an immediate consequence we obtain:

Theorem 29. *LD-Q-Res does not simulate $\forall$Exp+Res, IR-calc, IRM-calc.*

5.2 A lower bound for parity in universal resolution

Our next goal is to extend the lower bound for the parity formulas even further to resolution systems that can resolve on universal variables. To do this we modify again the formulas, using a similar technique as in [3]. The trick is essentially to double the universal literals so they form tautological clauses when resolved. This way resolution on universal variables does not give any advantage.

Definition 30. *Let* $\mathrm{xor}_u(o_1, o_2, o, l_1, l_2)$ *be the set of clauses* $\{l_1 \vee l_2 \vee \neg o_1 \vee \neg o_2 \vee \neg o,\ l_1 \vee l_2 \vee o_1 \vee o_2 \vee \neg o,\ l_1 \vee l_2 \vee \neg o_1 \vee o_2 \vee o,\ l_1 \vee l_2 \vee o_1 \vee \neg o_2 \vee o\}$*. Define* $\mathrm{QUParity}_N$ *as*

$$\exists x_1, \dots, x_N \forall z_1, z_2 \exists t_2, \dots, t_N.$$

$$\mathrm{xor}_u(x_1, x_2, t_2, z_1, z_2) \cup \bigcup_{i=3}^{N} \mathrm{xor}_u(t_{i-1}, x_i, t_i, z_1, z_2)$$

$$\cup\, \mathrm{xor}_u(x_1, x_2, t_2, \neg z_1, \neg z_2) \cup \bigcup_{i=3}^{N} \mathrm{xor}_u(t_{i-1}, x_i, t_i, \neg z_1, \neg z_2)$$

$$\cup\, \{z_1 \vee z_2 \vee t_N, \neg z_1 \vee \neg z_2 \vee \neg t_N\}$$

It is clear that these formulae are false as the universal player should play both z_1 and z_2 as they would z in QParity. We will now assume that in an LQU^+-Res refutation we $\forall$-reduce immediately. It is easy to verify that this does not increase proof size (cf. also Proposition 1 in [3]). For QUParity we now show an analogous result to Lemma 26.

Lemma 31. *Let C be a clause in an LQU$^+$-Res refutation of* QUParity_N *where $\forall$-reduction steps are performed as soon as possible. If C contains some existential literal l such that* $\text{lv}(z_2) < \text{lv}(l)$*, then either $z_1, z_2 \in C$, or $\neg z_1, \neg z_2 \in C$, or $z_2^* \in C$.*

Proof. The proof is the same as for Lemma 26, except for the possibility of universal resolution steps. As $\forall$-reductions are required to happen immediately, in our induction hypothesis we know that a z_1 literal can only occur together with the corresponding z_2 literal. Therefore resolving on z_1 removes this variable and merges the complementary z_2 literals; hence we get the z_2^* literal.

The merged literals cannot be pivots. Neither can z_2. This holds because we know by induction hypothesis that when z_2 appears unmerged, then also z_1 appears unmerged with the same polarities. Hence resolving with z_2 as the pivot would merge z_1, which is illegal due to the index restriction. □

We can now argue that neither long-distance nor universal resolution steps help to refute QUParity.

Lemma 32. *Any LQU$^+$-Res refutation of* QUParity_N *does not contain any clauses with z_1^* or z_2^* or any application of resolution on universal pivots.*

Proof. We first argue for z_2^*. Let $z_2^* \in C$. As we assume that $\forall$-reductions are performed immediately, the literal z_2^* is blocked by an existential literal l when z_2^* is created in C by a long-distance resolution step. Then l cannot be removed from C by resolution as any clause with $\neg l$ in it contains a literal over z_2 by Lemma 31. Also z_2^* cannot be removed via universal resolution. So the empty clause can never be derived from any clause containing z_2^*.

Let us now argue for z_1^* and assume $z_1^* \in C$. If z_1^* is introduced into C by resolving clauses D_1 and D_2, the literals z_1 and $\neg z_1$ in D_1 and D_2, respectively, must be blocked by existential literals. Therefore by Lemma 31, the clauses D_1 and D_2 also contain z_2 and $\neg z_2$, respectively. Hence also z_2^* is introduced into C and we get back to the previous case.

Finally, universal resolution steps cannot be performed when deriving the empty clause. For universal resolution on z_1, using again Lemma 31 together with the assumption of performing $\forall$-reductions as early as possible leads to the introduction of z_2^*, and we again get back to the case above.

No resolution on z_2 is possible as from Lemma 31 it would cause both literals of z_1 to be merged, which is illegal due to the index restriction in long-distance resolution over universal variables. □

This immediately implies the hardness of QUParity for LQU^+-Res because by the previous lemma any LQU^+-Res refutation of QUParity_N is a Q-Res refutation, which by Theorem 20 is exponential in size.

Theorem 33. QUParity_N *require exponential-size refutations in LQU$^+$-Res.*

As QUParity_N still remains easy for $\forall$Exp+Res in a proof similar to Lemma 21 we get the following separations.

Theorem 34. *QU-Res, LQU-Res, LQU$^+$-Res do not simulate $\forall$Exp+Res, IR-calc, IRM-calc.*

6 Strategy extraction as a general lower bound technique

The results of Section 4 can be vastly generalised. We say that a QBF proof system P has *strategy extraction in complexity class* C if from each proof π of a QBF φ we can compute a winning strategy of the universal player for φ from π in C.

Let L be a language in $\mathsf{\Sigma_k^p/poly}$ for some $k \geq 0$. Let $L = \{x \in \Sigma^\star \mid \exists y_1 \forall y_2 \ldots Q y_k.\, A(x,y)\}$, where A is a predicate computable in $\mathsf{P/poly}$. We can thus compute A by a sequence of polynomial-size circuits A_n. The computation of each such circuit A_n can be described by a CNF $C_n(\bar{x}, \bar{y}, \bar{w})$, where $\bar{x}$ are the propositional variables associated with the input x, $\bar{y}_1, \ldots, \bar{y}_k$ are the propositional variables for the witnesses $y_1, \ldots, y_k$, and $\bar{w}$ are auxiliary propositional variables describing the gates of the circuit A_n.

We now define a QBF

$$\Phi_{L,n}(\bar{x}, \bar{y}_1, \ldots, \bar{y}_k, z, \bar{w}) = \exists \bar{x} \forall z \exists \bar{y}_1 \forall \bar{y}_2 \ldots Q \bar{y}_k \exists \bar{w}.\, (z \leftrightarrow C_n(\bar{x}, \bar{y}_1, \ldots, \bar{y}_k, \bar{w})).$$

Clearly, this is a false QBF as it expresses that x is both in and outside L. Moreover, from the construction of the formula it is clear that the only winning strategy for the universal player is to play $z = 1 - \chi_L(x)$, where χ_L is the characteristic function of L, and to supply arbitrary values for the remaining universal variables $\bar{y}_2$ etc. Therefore each winning strategy for the universal player for Φ_L will have to compute the characteristic function of L. This immediately yields conditional lower bounds for proof systems with strategy extraction:

Theorem 35. *Let P be QBF proof system with strategy extraction in $\mathsf{P/poly}$. Then P is not polynomially bounded, unless $\mathsf{PH} \subseteq \mathsf{P/poly}$.*

Proof. For arbitrary $k \geq 1$ consider the satisfiability problem SAT_k, which asks whether a given QBF (possibly with free variables) with k quantifier blocks and starting with an existential block of quantifiers is satisfiable. Note that SAT_k is the canonical $\mathsf{\Sigma_k^p}$-complete problem. Choose $L = \mathrm{SAT}_k$ and consider the formulas $\Phi_{\mathrm{SAT}_k,n}$. Assume that $\Phi_{\mathrm{SAT}_k,n}$ have polynomial-size proofs in P. These proofs might be non-uniform, but we can compute them in $\mathsf{P/poly}$. As P also has strategy extraction in $\mathsf{P/poly}$ we obtain $\mathrm{SAT}_k \in \mathsf{P/poly}$. If this holds for all $k \geq 1$ we get $\mathsf{PH} \subseteq \mathsf{P/poly}$. □

Note that the assumption $\mathsf{PH} \not\subseteq \mathsf{P/poly}$ is considered very weak. In fact, even $\mathsf{NP} \cap \mathsf{coNP} \subseteq \mathsf{P/poly}$ is considered unlikely as factoring is in $\mathsf{NP} \cap \mathsf{coNP}$. Also by the Karp-Lipton theorem [21], $\mathsf{NP} \subseteq \mathsf{P/poly}$ implies that the polynomial hierarchy collapses to the second level, and there are even stronger Karp-Lipton collapse consequences known (cf. [11,7]).

As one application of Theorem 35 we mention our calculus IRM-calc [6], which has strategy extraction in $\mathsf{P/poly}$.

Corollary 36. *IRM-calc is not polynomially bounded unless $\mathsf{PH} \subseteq \mathsf{P/poly}$.*

If the proof system allows for strategy extraction via weaker models, then we can improve the conditional lower bounds to unconditional lower bounds, possibly even exponential. We exemplify this paradigm in our next results.

Theorem 37. *Let P be a QBF proof system.*

1. *Let P have strategy extraction in a complexity class C such that the non-uniform version of C is strictly weaker than $\mathsf{NP/poly}$ (i.e. $\mathsf{NP/poly} \setminus \mathsf{C/poly} \neq \emptyset$). Then P is not polynomially bounded.*

2. *If P has strategy extraction in AC^0, then P has exponential lower bounds to the size of proofs.*

Proof. For item 1 let P be have strategy extraction in C and $L \in \mathsf{NP/poly} \setminus \mathsf{C/poly}$. Assume towards a contradiction that the formulas $\Phi_{L,n}$ have polynomial-size P-proofs. These proofs might be non-uniform. However, as strategies can be extracted in C, these proofs imply $L \in \mathsf{C/poly}$, contradicting our assumption.

For item 2 we can use the formulas $\Phi_{\textsc{Parity},n}$ (or directly $\text{Q}\textsc{Parity}_n$). If P admitted subexponential-size P-proofs for these formulas (even non-uniform), these would translate into subexponential-size bounded-depth circuits for $\textsc{Parity}$, contradicting Theorem 17. □

Our previous Theorem 20 is an instance of item 2 of Theorem 37. In contrast, we can show that the method of strategy extraction is not effective for $\forall$Exp+Res (and therefore neither for IR-calc nor IRM-calc), because all formulas that are potentially hard via the strategy extraction method are easy for $\forall$Exp+Res, similarly as in Lemma 21.

Proposition 38. *For every language $L \in \mathsf{P/poly}$ the formulas $\Phi_{L,n}$ have polynomial-size $\forall$Exp+Res refutations.*

Proof. For $L \in \mathsf{P/poly}$ the formula $\Phi_{L,n}$ has the form

$$\Phi_{L,n}(\bar{x}, \bar{y}, z, \bar{w}) = \exists \bar{x} \forall z \exists \bar{w}. \, (z \leftrightarrow C_n(\bar{x}, \bar{w}))$$

with the single universal variable z. In $\forall$Exp+Res we expand z in both polarities, obtaining two copies of the matrix clauses annotated with $0/z$ and $1/z$, respectively. Both of these copies describe the computation of the circuit C_n. Let w_i be the variable from $\bar{w}$ representing the output at gate i of the circuit C_n. By induction on i we derive clauses representing $w_i^{0/z} \leftrightarrow w_i^{1/z}$ in polynomial-size proofs. This finally gives a contradiction with the clauses $w_n^{1/z}$ and $\neg w_n^{0/z}$, stemming from instantiating the axiom $z \leftrightarrow w_n$. □

We remark that the same method of constructing short $\forall$Exp+Res proofs does not work once we have further universal or existential variables in the formulas, i.e. if L is a language from a level Σ_i^p or Π_i^p with $i \geq 1$.

7 Conclusion

In this paper we have shown new lower bounds for Q-Res, IR-calc, LD-Q-Res and LQU$^+$-Res, and thereby settled the relative complexity of the main resolution-based QBF calculi. This reveals an almost complete picture of the simulation order of these proof systems (cf. Figure 1). Most importantly, our results show striking separations between all proof systems modelling CDCL-based QBF solving vs. proof systems modelling expansion-based solving. This provides theoretical evidence that these two paradigms for QBF-solving are indeed complementary and should enhance the power of the solvers when carefully used in conjunction.

Two specific questions that remain open are to show explicit lower bounds for natural QBF formulas for IRM-calc and to fully explore the relationship of this system to universal resolution. With respect to lower bounds for IRM-calc we remark that it is easy to transfer classical resolution lower bounds to this system (e.g., use the existentially closed version of the pigeonhole principle) and thereby improve Corollary 36 to an unconditional lower bound.

However, it would be interesting to find meaningful classes of QBFs that are hard for IRM-calc. Regarding the relationship to universal resolution we leave open whether IRM-calc can simulate LQU$^+$-Res (but conjecture incomparability of the systems).

A more general and challenging open problem is to determine the extent of the applicability of our new lower bound method via strategy extraction. Here we have shown that this method is very effective for ∃∀∃-formulas in Q-Res, but fails for exactly these formulas in expansion-based systems as ∀Exp+Res and stronger. Is it possibly to use the technique for different types of QBFs even for unconditional lower bounds for stronger QBF proof systems?

Acknowledgments

This work was partially supported by FCT grants ATTEST (CMU-PT-/ELE/0009/2009), POLARIS (PTDC/EIA-CCO/123051/2010), INESC-ID's multiannual PIDDAC funding PEst-OE/EEI/LA0021/2011, EPSRC grant EP/L024233/1, and a grant from the John Templeton Foundation. The second author was supported by a Doctoral Training Grant from EPSRC.

References

1. Arora, S., Barak, B.: Computational Complexity - A Modern Approach. Cambridge University Press (2009)
2. Balabanov, V., Jiang, J.H.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012)
3. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Sinz, C., Egly, U. (eds.) Theory and Applications of Satisfiability Testing - SAT. vol. 8561, pp. 154–169. Springer (2014)
4. Benedetti, M.: Evaluating QBFs via symbolic Skolemization. In: Baader, F., Voronkov, A. (eds.) LPAR. vol. 3452, pp. 285–300. Springer (2004)
5. Benedetti, M., Mangassarian, H.: QBF-based formal verification: Experience and perspectives. JSAT 5(1-4), 133–191 (2008)
6. Beyersdorff, O., Chew, L., Janota, M.: On unification of QBF resolution-based calculi. In: MFCS, II. pp. 81–93 (2014)
7. Beyersdorff, O., Müller, S.: A tight Karp-Lipton collapse result in bounded arithmetic. ACM Transactions on Computational Logic 11(4) (2010)
8. Biere, A.: Resolve and expand. In: SAT. pp. 238–246 (2004)
9. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185. IOS Press (2009)
10. Buss, S.R.: Towards NP-P via proof complexity and search. Ann. Pure Appl. Logic 163(7), 906–917 (2012)
11. Cai, J.Y.: $S_2^p \subseteq ZPP^{NP}$. Journal of Computer and System Sciences 73(1), 25–35 (2007)
12. Cook, S.A., Nguyen, P.: Logical Foundations of Proof Complexity. Cambridge University Press (2010)
13. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. The Journal of Symbolic Logic 44(1), 36–50 (1979)
14. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: McMillan et al. [25], pp. 291–308
15. Furst, M.L., Saxe, J.B., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. Mathematical Systems Theory 17(1), 13–27 (1984)
16. Giunchiglia, E., Marin, P., Narizzano, M.: Reasoning with quantified boolean formulas. In: Biere et al. [9], pp. 761–780
17. Janota, M., Grigore, R., Marques-Silva, J.: On QBF proofs and preprocessing. In: McMillan et al. [25], pp. 473–489
18. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. In: Cimatti, A., Sebastiani, R. (eds.) SAT. vol. 7317, pp. 114–128. Springer (2012)
19. Janota, M., Marques-Silva, J.: ∀Exp+Res does not P-Simulate Q-resolution. International Workshop on Quantified Boolean Formulas (2013)

20. Janota, M., Marques-Silva, J.: On propositional QBF expansions and Q-resolution. In: Järvisalo, M., Van Gelder, A. (eds.) SAT. vol. 7962, pp. 67–82. Springer (2013)
21. Karp, R.M., Lipton, R.J.: Some connections between nonuniform and uniform complexity classes. In: Proc. 12th ACM Symposium on Theory of Computing. pp. 302–309. ACM Press (1980)
22. Kleine Büning, H., Bubeck, U.: Theory of quantified boolean formulas. In: Biere et al. [9], pp. 735–760
23. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. 117(1), 12–18 (1995)
24. Krajíček, J.: Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. The Journal of Symbolic Logic 62(2), 457–486 (1997)
25. McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.): Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings, vol. 8312. Springer (2013)
26. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1994)
27. Rintanen, J.: Asymptotically optimal encodings of conformant planning in QBF. In: AAAI. pp. 1045–1050. AAAI Press (2007)
28. Rivest, R.L.: Learning decision lists. Machine Learning 2(3), 229–246 (1987)
29. Segerlind, N.: The complexity of propositional proofs. Bulletin of Symbolic Logic 13(4), 417–481 (2007)
30. Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Milano, M. (ed.) CP. vol. 7514, pp. 647–663. Springer (2012)
31. Zhang, L., Malik, S.: Conflict driven learning in a quantified Boolean satisfiability solver. In: ICCAD. pp. 442–449 (2002)

ECCC ISSN 1433-8092

http://eccc.hpi-web.de