Electronic Colloquium on Computational Complexity, Report No. 32 (2017)

Formulas with Large Weight: a New Technique for Genuine QBF Lower Bounds

Olaf Beyersdorff and Joshua Blinkhorn

School of Computing, University of Leeds, UK

{o.beyersdorff,scjlb}@leeds.ac.uk

Abstract. We devise a new technique to prove lower bounds for the proof size in resolution-type calculi for quantified Boolean formulas (QBF). The new technique applies to the strong expansion system IR-calc and thereby also to the most studied QBF system Q-Resolution.
Our technique exploits a clear semantic paradigm, showing the hardness of a QBF family by demonstrating that (1) the formulas require large witnessing functions for the universal variables, and (2) in the game semantics of QBF, the universal player is forced to make many responses on a set of variables that we identify as *critical*.
Based on these two conditions, our technique yields a *weight* for a QBF formula, which provides an absolute lower bound for the size of its IR-calc proofs (and hence also its Q-Resolution proofs). We exemplify our technique on a couple of known and new instances, among them the prominent formulas of Kleine Büning et al. [33], for which our method provides a hardness proof that is considerably simpler than previous approaches.
Our technique also provides the first separations for QBF dependency calculi. In particular, we construct a simple class of formulas that are hard for Q-Resolution, but easy in Q-Resolution parameterized by the reflexive resolution path dependency scheme, thus answering a question posed by Slivovsky and Szeider [49].

1 Introduction

Proof complexity is a subfield of computational complexity investigating the complexity of theorem proving. Its central problem is to determine the size of the smallest proof for a given formula in a specified proof system, typically defined through a set of axioms and inference rules. By placing a non-deterministic computational model (the existence of short proofs) into the centre of investigation, proof complexity closely relates to central questions in computational complexity (such as NP vs co-NP, or NP vs PSPACE, cf. [20, 24]). In addition, proof complexity is tightly linked to and inspired by first-order logic (bounded arithmetic [23, 34]) and developments in practical solving. In particular, the exciting developments in SAT solving during the past 15 years have been a main driver for the field.

SAT solvers have emerged as a universal tool for the solution of computationally hard problems. From a theoretical perspective, this success of solving is currently insufficiently understood. What makes huge industrial instances easy for SAT solvers, while even the best modern solvers fail on tiny instances of hand-crafted combinatorial problems?

The main theoretical approach towards understanding the power and limitations of SAT solving comes from proof complexity. Traces of runs of SAT solvers on unsatisfiable instances yield proofs for unsatisfiability, whereby each SAT solver implicitly defines a proof system. In particular, SAT solvers based on the DPLL procedure, and contemporary SAT solvers using conflict-driven clause learning (CDCL), are based on resolution, which is arguably the most studied proof system in proof complexity. Therefore proof lengths in resolution and its fragments correspond to the running time of SAT solvers on unsatisfiable instances (cf. [20]).

In the past decade the success of SAT solving has been extended to the more complex setting of *quantified Boolean formulas* (QBF). Due to its PSPACE completeness (even for restricted versions [1]), QBF is far more succinct than SAT and thus applies to further fields such as formal verification or planning [7, 26, 43]. As for SAT solvers, runs of QBF solvers

ISSN 1433-8092

produce witnesses of unsatisfiability, and there has been intense research activity to provide appealing proof-theoretic models for QBF solvers and to analyse their strength.

Modern QBF solvers are mainly modelled by resolution-type systems. However, due to a number of conceptually different approaches in QBF solving, the situation is somewhat more complex than in SAT (cf. [5, 11]). One of the central paradigms is to lift the CDCL approach to QBF [29]. The main corresponding proof system for this is Q-Resolution (Q-Res), defined in [33], together with various extensions such as long-distance Q-Resolution [3], universal Q-Resolution (QU-Res) [28], and their combinations [5]. Q-Res incorporates the classical resolution rule on existential variables. Additionally, under certain conditions it allows to eliminate universal variables from clauses by a universal reduction rule.

Another QBF solving paradigm comes from *expansion*, where universal variables are gradually replaced by constants 0/1 until a purely existentially quantified formula is obtained, which can be passed to a SAT solver. While in general the expansion yields an exponential increase in the formula size, careful (partial) expansion – as implemented in the solver RaREQs – yields a powerful approach [31]. The base system to model expansion-based solving is ∀Exp+Res [32], where proofs are clearly decomposed into an expansion phase and a subsequent propositional resolution phase. A more flexible model is provided by the system IR-calc [10], where expansion and resolution are mixed.

Exponential separations obtained in [11, 32] show that the base systems Q-Res for CDCL solving and ∀Exp+Res for expansion solving are incomparable in strength. However, they are both simulated[1] by the system IR-calc [10]. A host of stronger QBF proof systems have also been investigated recently , in the form of sequent systems [19,22,25,36], Frege systems [9,19], cutting planes [15], and the proof checking format QRAT [30].

In this paper we concentrate our investigation on QBF resolution, and on IR-calc in particular, as it combines the two base systems for CDCL and expansion solving.

1.1 Lower bound techniques

The strength of a proof system (and hence of the corresponding solving approach) is measured by which formulas admit short (i.e. polynomial-size) proofs in the system and for which formulas absolute lower bounds can be obtained. As is typical in computational complexity, it is the latter task of showing lower bounds that turns out to be the most challenging. In particular, proving lower bounds for strong systems such as Frege constitutes a major challenge, and in the QBF setting this task is intimately linked to progress in lower bounds for circuit complexity [19].

Arguably, what is even more important than the actual lower bounds, is to devise generally applicable *lower bound techniques*. For propositional resolution we have a number of such techniques (cf. [20, 45]) and it is illuminating to review their applicability in the QBF context. The most widely used classical technique is the size-width technique of [6], which shows lower bounds for size via lower bounds for the width in resolution. However, as shown in [14], this technique drastically fails even in the base system Q-Res.

On the positive side, feasible interpolation [35] is a propositional technique that also works for QBF resolution systems [13]. However, feasible interpolation only applies to a quite restricted class of formulas, as it imports lower bounds for monotone Boolean circuits (of which we only have very few). A further general approach is through games. While this is effective in both propositional resolution [18, 42] as well as in Q-Res [17], it only applies to the weak tree-like versions of these systems.

It is therefore fair to say that, at present, ideas from the propositional world have limited impact to the QBF framework. Moreover, there is a general feeling in the QBF community

[1] Proof system A simulates proof system B if A-proofs and be transformed into B-proofs with at most polynomial increase in proof size.

that lower bounds just lifted from classical propositional proof complexity are not particularly interesting. On purely existentially quantified formulas, all QBF resolution systems degrade to propositional resolution, hence every classically hard formula gives rise to a hard QBF. However, this is not the phenomenon we want to study (cf. also [21] for an elaboration). We are interested only in 'genuine' QBF lower bounds.

One approach in this direction is via *strategy extraction*, which originates from the game semantics of QBF. For this we consider a *two-player game* between an existential and a universal player, who in turn choose 0/1 values for the variables in the QBF in the order of the quantifier prefix, starting from the leftmost quantifier. The universal player wins as soon as one of the clauses in the QBF matrix gets falsified, otherwise the existential player wins. A fully quantified QBF is false if and only if the universal player has a winning strategy; likewise it is true if and only if the existential player has a winning strategy.

Strategy extraction computes a strategy from a proof of the formula. In particular, from a refutation of a false QBF in a QBF resolution system, a winning strategy for the universal player can be efficiently extracted [4, 10]. This is practically important, as strategies witness the answers of QBF solvers [3], but it can also be exploited for a lower bound technique [9, 11]. Namely, Q-Res allows strategy extraction even with bounded-depth circuits [3, 11], and therefore any AC^0 lower bound, suitably coded into a QBF, gives rise to a Q-Res lower bound [9, 11].

This is conceptually quite interesting, as it establishes a tight connection between circuit lower bounds and QBF proof size lower bounds [9, 19]. At the same time it also limits the applicability of the technique, as circuit lower bounds are very rare [51] and the QBFs expressing them of a rather special syntactic form.

2 Our contributions

In this section we explain our main results. We also outline the key technical ideas and provide pointers to later sections where these are fully elaborated.

2.1 A genuine lower bound technique for QBF

The need for new technique. Our main contribution is a new lower bound technique for the proof size in IR-calc that yields *genuine QBF lower bounds*. Before giving details on the technique, let us try to explain what we mean by that qualification.

For strong QBF Frege systems we can precisely classify the reasons for hardness: they either stem from a classical Frege lower bound (on an existentially quantified formula as explained above) or from a circuit lower bound via the strategy extraction technique [19]. Intriguingly, in QBF resolution a third case comes into play, where the hardness of the formula neither comes from propositional resolution nor from the hardness of the Herbrand functions that witness the universal quantifiers.

We currently do not understand this case very well, but we believe it to be crucial to assess the proof complexity of QBF resolution. In fact, the only example we are aware of are the formulas of Kleine Büning et al. [33], which have been proposed as the historically first formulas hard for Q-Res and form a recurring theme in QBF proof complexity [5, 11, 25, 27, 39]. Quite clearly, the hardness of these formulas does not stem from propositional resolution, and the witnessing functions for the winning strategy of the universal player are trivial.

Here we develop a general technique that applies to the formulas of [33] and opens the door for an understanding of the mysterious third case mentioned above, also providing further examples for that case.

Outline of the technique. Our technique has a clear semantic appeal grounded on the two-player game for QBF explained above. In the game, a winning strategy for the universal player determines responses to the choices made by the existential player. While the responses are possibly easy to compute, our lower bound method focuses on formulas where *many responses are necessary*, i.e., the range of each witnessing function for the collection of all universal quantifiers is large. For a QBF Φ let us refer to this condition as *Φ requiring large witnessing functions*. An easy example for such a formula is $\exists x_1, \ldots, x_n \forall u_1, \ldots, u_n.\ \bigvee_{i=1}^{n} (x_i \not\equiv u_n)$, where the unique witnessing function for $u_1, \ldots, u_n$ has full range $\{0,1\}^n$.

This condition alone does not suffice for lower bounds (cf. Example 6). We therefore identify a second condition, which guarantees that many of the universal responses must actually appear in the proof.

For this, we define the concept of a *restrictor* (Definition 7), which is an assignment to all existential variables, except for the rightmost existential block.[2] After applying a restrictor δ to a QBF Φ, it becomes a Π_2^b-formula of the form $\Phi|_\delta = \exists X \forall U.\, \phi|_\delta$. We now consider IR-calc refutations of $\Phi|_\delta$. We call a universal variable u from U *critical* (Definition 10) if it must appear in each refutation of $\Phi|_\delta$, i.e., every minimally unsatisfiable subset of $\phi|_\delta$ contains u.

Let us now fix an IR-calc refutation π of Φ and consider the winning strategy S extracted from that proof via the strategy extraction method of [10]. As we assume that Φ requires large witnessing functions, we know that S needs to contain many responses on universal variables. Our central argument establishes that all responses on critical variables must appear in each proof of $\Phi|_\delta$ (Lemma 13) and therefore also in the proof π of Φ (Proposition 2).

Thus, ranging over all different restrictors δ, we can collect all responses to S on critical variables. We include all such responses as nodes in a *critical response graph* $G(S, \Phi)$ (Definition 14), with edges connecting inconsistent responses. The proof π must then be at least as large as the *clique number of* $G(S, \Phi)$. We define the *weight of* Φ as the minimum clique number of all response graphs $G(S, \Phi)$ for all winning strategies S of the universal player (Definition 16). By this weight of Φ we obtain a precise lower bound to the IR-calc proof size for Φ.

Theorem 17 (Weight Theorem). *The size of any* IR-calc *refutation of a QBF Φ is at least the weight of Φ.*

A graphical representation of the idea behind our technique is given in Figure 6.

We remark that for the IR-calc lower bound method to take effect for Φ, we crucially need both conditions that

1. Φ requires large witnessing functions, and
2. for sufficiently many restrictors δ, the formula $\Phi|_\delta$ contains many critical variables.

While condition (1) alone yields a lower bound for the weaker expansion system $\forall$Exp+Res (Proposition 5), it is not difficult to find formulas fulfilling (1), but failing (2), which have short proofs in IR-calc (Example 6). On the other hand, a variation of the parity formulas from [11] provides an example for formulas fulfilling (2), but failing (1) (cf. Part A of the appendix). Interestingly, these parity formulas are hard for Q-Res, but easy in $\forall$Exp+Res and therefore also in IR-calc [11].

We emphasise that our lower bound technique directly applies to Q-Res, which is arguably the best-studied QBF resolution system. We have chosen to present it for IR-calc as this calculus simulates Q-Res [10], but is strictly stronger than Q-Res [11]. At the same time, the presentation of the lower bound appears technically easier for IR-calc, because IR-calc has a very neat way to deal with universal variables.

[2] W.l.o.g. we assume in the following that the rightmost quantifier block is existential.

2.2 Applications of the new technique

Applying our technique to show the hardness of specific formulas turns out to be rather straightforward: all one needs to do is to compute the weight of the formula by verifying conditions (1) and (2) above. We illustrate this with three examples.

Our *first example* is the formula family KBKF(n) of Kleine Büning et al. [33], mentioned above. Originally suggested as hard formulas for Q-Res [33], they have since appeared prominently in the QBF literature (e.g. [5,11,25,27,39]). A formal proof of their hardness for IR-calc (and hence for Q-Res) has been given in [11][3], where the rather technically involved argument is tailored towards the specific syntactic structure of the formulas.

We give a much easier proof of hardness for IR-calc (and hence for Q-Res) via our technique (Subsection 6.1). Exploiting semantic properties of the formulas, we prove that their weight is exponentially large.

Theorem 19. *For each $n \in \mathbb{N}$,* KBKF(n) *has weight 2^n.*

The proof-size lower bound is then obtained by applying the Weight Theorem. We believe that our argument sheds new light on KBKF(n), and provides the first clear appreciation of their hardness.

Our *second example* is a modification of formulas by Janota and Marques-Silva [32]. While these formulas are hard for the base expansion system ∀Exp+Res, they are easy for Q-Res (and thus also IR-calc), and hence separate these systems [32]. Whereas the weight of the original formulas is constant, we show that a simple reordering of the quantifier prefix yields modified formulas with exponential weight (Theorem 22), to which our technique is applicable. We therefore obtain a new lower bound for IR-calc. Interestingly, the original formulas have a linear number of quantifier alternations, while our new versions are just Σ^b_3.

Our *final example* is perhaps the most important. We introduce the following new class of QBFs.

Definition 24. *The* equality formulas *are the formula family*

$$\Psi(n) = \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \,.\, \psi(n)$$

where the matrix $\psi(n)$ consists of the clauses

$$B_n = \{\neg t_1, \ldots, \neg t_n\},$$
$$C_i = \{x_i, u_i, t_i\}, \quad D_i = \{\neg x_i, \neg u_i, t_i\}, \quad \text{for } i \in [n].$$

We call these QBFs the *equality formulas* because the unique winning strategy of the universal player is to choose $u_i \equiv x_i$ for each i. As in our previous two examples, semantic properties necessitate that the formulas require large weight.

Theorem 25. *For each $n \in \mathbb{N}$, the equality formula $\Psi(n)$ has weight 2^n.*

The equality formulas have been designed with the aim of providing simple and easily accessible formulas with exponential weight, and they are arguably the simplest formulas to which our technique applies. As such they form a welcome addition to the QBF portfolio of hard instances (which, unfortunately, remains rather limited at present). Additionally, they have an important application in the area of *dependency calculi*, which we describe next.

[3] the full proof is contained in the report [12]

2.3 Separation of QBF dependency calculi

The analysis of the equality formulas introduced in the previous subsection provides the first results in the proof complexity of QBF dependency calculi. To put these results into context, we provide a brief summary of the field.

SAT solvers can freely assign variables in an arbitrary order, and the ability to determine a 'good' order can yield an exponential improvement on running time. Indeed, the implementation of sophisticated *decision heuristics* – algorithms that decide the order of variable assignments – is a cornerstone of SAT solving, and there is a large volume of literature on the topic (e.g. [37, 40, 46, 47]).

QBF solvers, however, are not granted the same freedom; nested quantifier scopes entail variable dependencies that must be respected by the assignment order. As a result, the power of decision heuristics is greatly reduced, thus presenting a major challenge for practitioners.

Pioneered by the leading solver DepQBF [38], a recent development at the cutting edge of QBF solving is the introduction of *dependency schemes* [44]. A dependency scheme is an efficient algorithm that attempts to identify so-called *spurious* dependencies, thereby relaxing the restrictions on the assignment order and improving the power of decision heuristics.

The theoretical model for dependency-aware solving employs existing QBF systems parameterized by dependency scheme. For example, whereas a basic CDCL-based solver corresponds to Q-Res, the same solver augmented with dependency scheme $\mathcal{D}$ corresponds to Q($\mathcal{D}$)-Res [50], the parameterization of Q-Res by $\mathcal{D}$. In principle, any QBF proof system P can be parametrised by any dependency scheme $\mathcal{D}$, yielding the system P($\mathcal{D}$). Parameterizations of further CDCL-based systems have been studied in [8, 41]. (Prior to the present paper, a similar treatment of expansion-based systems had not been investigated.)

The *reflexive resolution path dependency scheme* ($\mathcal{D}^{\text{rrs}}$) [50] has assumed a prominent place in the literature, being the strongest known dependency scheme for which the parameterized CDCL systems are sound. However, as far as we are aware, nothing at all is known regarding the proof-theoretic strength of using this scheme.

Applying our lower bound technique, we provide the first separations for dependency calculi and initiate the study of expansion-based dependency systems. A clear picture emerges that demonstrates strong potential for dependency-aware solving in both major paradigms.

On the CDCL side, we show that Q($\mathcal{D}^{\text{rrs}}$)-Res is exponentially stronger than Q-Res, thereby answering a question first posed in 2014 by Slivovsky and Szeider [49]. In fact, we prove a stronger result, namely that Q-Res does not simulate *tree-like* Q($\mathcal{D}^{\text{rrs}}$)-Res, from which it follows that these two calculi are incomparable (the reverse separation lifts from propositional resolution). We also show that the same results hold when resolution over universal pivots is allowed; that is, we prove the analogous separations for QU-Res.

In fact, we also show the analogous results for both expansion-based calculi IR-calc and ∀Exp+Res, which we believe to be of particular practical relevance (cf. Section 8). Technically all these separations are achieved by our new equality formulas or a slight modification thereof.

Our results, stated in the following theorem, are depicted in Figure 1.

Theorem 29. *Let* P *be any of the proof systems* IR-calc, ∀Exp+Res, Q-Res, *and* QU-Res. *Then* P *simulates neither* P($\mathcal{D}^{\text{rrs}}$) *nor tree-like* P($\mathcal{D}^{\text{rrs}}$).

Organisation

The remainder of this paper is organised as follows. We start with relevant background information on QBFs, Q-Res, and strategies in Section 3 and on the system IR-calc in Section 4. The description of our new lower bound technique is then given in Section 5, followed by applications on known formulas, including the formulas of Kleine Büning et al. [33], (Section 6)

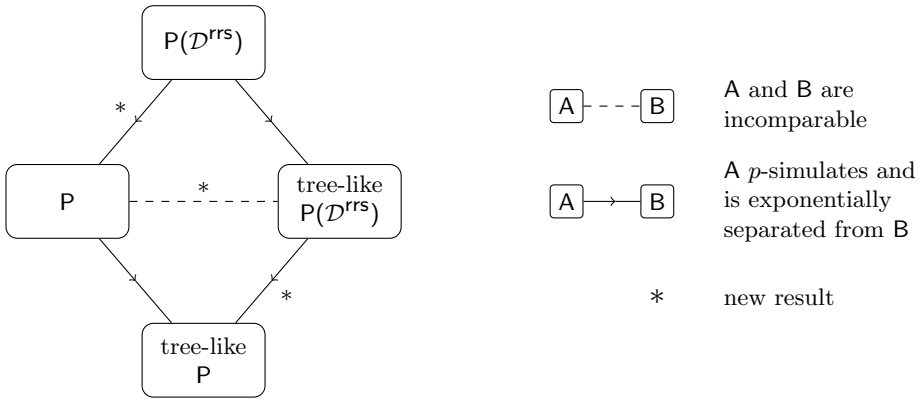

Fig. 1. Proof complexity of QBF dependency calculi. Here the proof system P is any one of IR-calc, ∀Exp+Res, Q-Res or QU-Res.

and on the new equality formulas and the separations between dependency calculi (Section 7). We conclude in Section 8 with a discussion on further research avenues opened by our results.

3 Preliminaries

Quantified Boolean formulas. We consider *quantified Boolean formulas* (QBFs) Φ over a set $\mathrm{vars}(\Phi) = \{z_1, \ldots, z_n\}$ of variables ranging over $\{0, 1\}$. Throughout we use only formulas in *prenex conjunctive normal form* (PCNF), denoted $\Phi = \mathcal{Q}\,.\,\phi$, in which all variables are quantified either existentially or universally in the *quantifier prefix* $\mathcal{Q} = \mathcal{Q}_1 z_1 \cdots \mathcal{Q}_n z_n$, $\mathcal{Q}_i \in \{\exists, \forall\}$ for $i \in [n]$, and ϕ is a propositional conjunctive normal form (CNF) formula called the *matrix*. The prefix $\mathcal{Q}$ imposes a linear ordering $<_\Phi$ on the variables of Φ, such that $z_i <_\Phi z_j$ holds whenever $i < j$, in which case we say that z_j *is right of* z_i.

A literal is a variable or its negation, a clause is a disjunction of literals, and a CNF is a conjunction of clauses. Throughout, we refer to a clause as a set of literals and to a CNF as a set of clauses. We typically write x for existential variables, u for universals, and z for either.

We assume w.l.o.g. that the rightmost literal in a clause is not universal (all such literals may be removed from clauses in polynomial time, while preserving the truth value of the formula).

For a literal l, we write $\mathrm{var}(l) = z$ iff $l = z$ or $l = \neg z$, for a clause C we write $\mathrm{vars}(C) = \{\mathrm{var}(l) \mid l \in C\}$, and for a PCNF Φ we write $\mathrm{vars}(\Phi)$ for the variables in the prefix of Φ. For any clause C, $C_\exists$ ($C_\forall$) denotes the existential (universal) literals of C. For any set of variables V, $V_\exists$ ($V_\forall$) denotes the existential (universal) variables of V. As is conventional, we represent assignments as sets of literals, where literal z (resp. $\neg z$) represents the assignment $z \mapsto 1$ (resp. $z \mapsto 0$).

QBF resolution. *Resolution* is a well-studied refutational proof system for propositional CNF formulas with a single inference rule: the *resolvent* $C_1 \cup C_2$ may be derived from clauses $C_1 \cup \{x\}$ and $C_2 \cup \{\neg x\}$ (variable x is the *pivot*). Resolution is *refutationally* sound and complete: that is, the empty clause can be derived from a CNF iff it is unsatisfiable.

There exist a host of resolution-based QBF proof systems – see [11] for a detailed account. *Q-resolution* (Q-Res, Fig. 2) introduced in [33] is the standard refutational calculus for PCNF. In addition to resolution over existential pivots, the calculus has a *universal reduction rule* which allows a clause C to be derived from $C \cup \{l\}$, where $\mathrm{var}(l)$ is a universal variable right of all existentials in C. Universal tautologies are explicitly forbidden; one may not derive a clause containing both l and $\neg l$ if $\mathrm{var}(l)$ is universal.

$\frac{}{C}$ (axiom)	C is a clause in the matrix of Φ.
$\frac{C \cup \{l\}}{C}$ ($\forall$-red)	Literal l is universal. Every existential literal in C is left of l.
$\frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}$ (res)	Variable x is existential. If $l \in C_1$ and var(l) is universal, then $\neg l \notin C_2$.

Fig. 2. The rules of Q-Res [33]. Note that $\Phi = \mathcal{Q} \,.\, \phi \in \mathsf{PCNF}$.

For a QBF resolution system P, a P *derivation* of a clause C from a PCNF Φ is a sequence $C_1, \ldots, C_m$ of clauses in which $C = C_m$, and each clause is either an axiom or is derived from previous clauses in the sequence using an inference rule. We insist that a derivation must have a unique conclusion; that is, C_m is the unique clause that is not the antecedent of another clause in the sequence. The *subderivation* of clause C_i is obtained by removing all such clauses from the subsequence $C_1, \ldots, C_i$. A *refutation* of Φ is a derivation of the empty clause from Φ.

Strategies. Semantics for PCNF are described neatly by the *two-player game*. Over the course of a game on a PCNF, the variables are assigned 0/1 values in the order of the prefix, with the $\exists$-player ($\forall$-player) choosing the values for the existential (universal) variables. When the game concludes, the players have constructed a total assignment α to the variables. The $\forall$-player wins if and only if α falsifies some clause of the matrix. A PCNF is false if and only if there exists a *winning strategy* for the universal player in the two-player game.

A *strategy* dictates how the $\forall$-player should respond to every possible choice of the $\exists$-player. Let $\Phi = \mathcal{Q} \,.\, \phi$ be a PCNF over variables V and let $\mathcal{Q} = \exists X_1 \forall U_1 \cdots \exists X_n \forall U_n \exists X_{n+1}$. A $\forall$-*strategy* S for Φ is a mapping from total assignments to $V_\exists$ to total assignments to $V_\forall$, such that, for each $i \in [n]$, $S(\alpha)$ and $S(\alpha')$ agree on the first i universal blocks whenever α and α' agree on the first i existential blocks. For any α in the domain of S, $\alpha \cup S(\alpha)$ is a total assignment to V referred to as a *game* of S. A strategy S is *winning* if and only if every game falsifies ϕ. Every winning $\forall$-strategy is equivalent to some *witnessing function*, and vice versa.

A strategy can be depicted naturally as a tree, as shown in Figure 4. Every game of the strategy is written on a unique path from the root of the tree to some leaf.

4 IR-calc fundamentals

In this section, we recall the definition of IR-calc [10] and discuss the underlying concepts of the calculus, including its use of annotations. We also cover restrictions of IR-calc proofs, and strategy extraction, both of which are central to the following section.

4.1 The calculus

To explain the concept of expansion, we consider an example. Let $\exists x \forall u \exists y \,.\, \phi(x, u, y)$ be a PCNF. The formula is semantically equivalent to $\exists x \exists y^0 \exists y^1 \,.\, \phi(x, 0, y^0) \wedge \phi(x, 1, y^1)$, where we have 'expanded' the universal variable u. (In our example this already leads to a purely existentially quantified formula.) Note that the variable x, which is left of u, remains unchanged, while we have to create two duplicate copies y^0 and y^1 for the variable y, which is right of u. To keep track of why we created these copies of y, we will annotate them with the reason

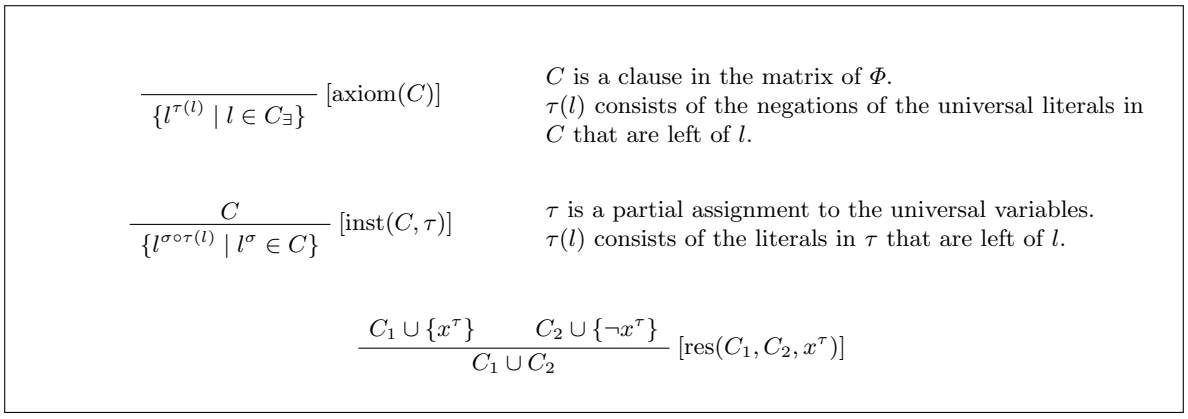

Fig. 3. The rules of IR-calc [10]. Note that $\Phi = \mathcal{Q} \,.\, \phi \in \mathsf{PCNF}$.

for their creation, i.e., we will use $y^{\neg u}$ instead of y^0 (where $\neg u$ corresponds to the assignment $u \mapsto 0$) and likewise y^u instead of y^1. Syntactically, $y^{\neg u}$ and y^u are just new, distinct existential variables.

Since a single expansion doubles the formula size in the worst case, the naive complete expansion of a PCNF is at the expense of a possible exponential blow-up. For example, given a formula with n universal variables, complete expansion produces a purely existentially quantified formula with 2^n conjuncts. Moreover, any existential in the scope of all n universals will require 2^n duplicate copies. As in our introductory example, we can keep track of all the duplicate variables with superscript annotations, but the annotations here are the appropriate assignments to *sets* of universal variables.

The study of expansion-based proof systems for QBF is motivated by developments in practice, most notably with the expansion-based QBF solver RaREQs [31]. In practice, it is frequently possible to maintain the falsity of a QBF by carefully expanding some universal variables only in one polarity, thus avoiding an exponential blow-up. The resulting (purely existential) formula is simply dispatched to a SAT solver.

In the basic theoretical model, exemplified by the fundamental system ∀Exp+Res [32], each existential x is first annotated with a *fixed, complete* assignment to the preceding universals on which it depends. The proof then proceeds exactly as a propositional resolution proof, with clauses in annotated variables. (Details on ∀Exp+Res are contained in the appendix.)

IR-calc, defined in [10], improves on this approach by working instead with *partial* assignments. In addition to resolution, the calculus is equipped with an *instantiation* rule by which partial annotations are grown throughout the course of the proof.

To facilitate instantiation, the $\circ$ operator describes how partial assignments are combined. Formally, for each PCNF Φ, we define the set of IR-calc annotations to be the set of partial assignments to the universal variables of Φ. For any such annotations τ and σ, we define $\tau \circ \sigma = \tau \cup \{l \in \sigma \mid \neg l \notin \tau\}$.

The rules of IR-calc are given in Figure 3. Due to space restriction in the superscript, it is beneficial to write partial assignments not as sets, but as literal strings, e.g. $u_1 \neg u_3 \neg u_6 u_7$. We explain the IR-calc rules and illustrate them with some simple examples.

Axiom clauses are introduced into the proof, or *downloaded*, by selecting a clause C from the matrix ϕ and applying the *download assignment* to the existential literals. By design, the download assignment σ for C is the smallest partial assignment that falsifies every universal literal in C. Represented as a set of literals, then, $\sigma = \{\neg l \mid l \in C_\forall\}$. When applying the download assignment, existentials are annotated only with universals *to their left* (i.e., those on which they depend). Consider the PCNF with prefix $\forall u \exists x \forall v \exists y$ and matrix

Line	IR-calc clause	Application of IR-calc rule
C_1	$\{\neg x_1, \neg x_2^{u_1}\}$	$\text{axiom}(\{\neg x_1, \neg u_1, \neg x_2\})$
C_2	$\{\neg x_1, x_2^{u_1}\}$	$\text{axiom}(\{\neg x_1, \neg u_1, x_2\})$
C_3	$\{\neg x_1\}$	$\text{res}(C_1, C_2, x_1)$
C_4	$\{x_1, x_3^{\neg u_1}\}$	$\text{axiom}(\{x_1, u_1, x_3\})$
C_5	$\{x_1, x_3^{\neg u_1 u_2}\}$	$\text{inst}(C_4, u_2)$
C_6	$\{x_1, \neg x_3^{u_2}\}$	$\text{axiom}(\{x_1, \neg u_2, \neg x_3\})$
C_7	$\{x_1, \neg x_3^{\neg u_1 u_2}\}$	$\text{inst}(C_6, \neg u_1)$
C_8	$\{x_1\}$	$\text{res}(C_5, C_7, x_3^{\neg u_1 u_2})$
C_9	$\bot$	$\text{res}(C_3, C_8, x_1)$

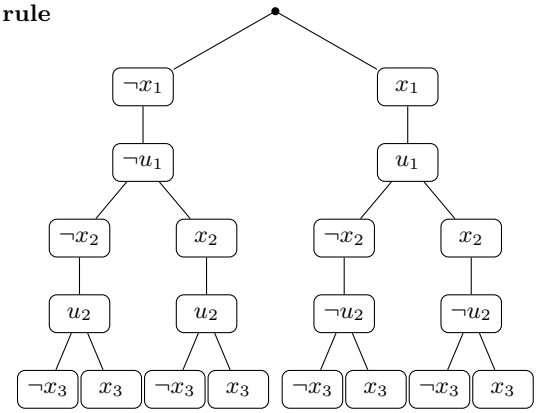

Fig. 4. An IR-calc refutation of a PCNF Φ (left) and the winning $\forall$-strategy extracted from it (right). The formula Φ has prefix $\exists x_1 \forall u_1 \exists x_2 \forall u_2 \exists x_3$ and clauses $\{\neg x_1, \neg u_1, \neg x_2\}$,$\{\neg x_1, \neg u_1, x_2\}$,$\{x_1, u_1, x_3\}$, $\{x_1, \neg u_2, \neg x_3\}$.

clauses $\{u, \neg x, \neg v, y\}$ and $\{\neg u, x, \neg y\}$. Downloading the two matrix clauses gives rise to axioms $\{\neg x^{\neg u}, y^{\neg u v}\}$ and $\{x^u, \neg y^u\}$. The matrix clause that gave rise to the IR-calc axiom C is referred to as the *download clause for* C.

Instantiation allows partial assignments to be combined during the course of the proof. A single partial assignment, called the *instantiation assignment* τ, is applied to all the literals in the clause. As in the axiom rule, universal variables to the right are omitted from the annotation. For example, given the prefix $\forall u_1 u_2 \exists x \forall u_3 u_4 \exists y$, and some clause $\{x^{u_1}, y^{\neg u_4}\}$ in an IR-calc derivation, instantiation by $\tau = \neg u_2 u_3 u_4$ derives $\{x^{u_1 \neg u_2}, y^{\neg u_2 u_3 \neg u_4}\}$. Note that u_3 and u_4, which are right of x, do not appear in that variable's annotation after the instantiation. Also note that u_4 does not appear in the annotation to y, which is already annotated with the negated literal $\neg u_4$ before the instantiation takes place (see the earlier definition of $\circ$).

Resolution in IR-calc is identical to propositional resolution. We emphasize that annotations are labelling distinct variables (e.g., x^u and x^v are different variables), so that a resolution step is only valid if the annotations of the pivot literals match.

A complete IR-calc refutation is shown in Figure 4. We point out that the annotation to variable x never features a universal variable to the right of x. Hence, whenever x^τ is written, it is considered implicit that each variable in τ is left of x.

4.2 IR-calc restrictions and strategy extraction

Restrictions. In the sequel, we make frequent use of PCNF and IR-calc restrictions, that derive from their propositional counterparts.

The restriction of a PCNF is performed similarly as for a propositional formula. For *any literal* l (representing an assignment to a variable), the restricted QBF $\Phi|_l$ is obtained by removing $\text{var}(l)$ from the quantifier prefix, removing the literal $\neg l$ from all clauses, and replacing all clauses containing l with the placeholder $C_\top$ (representing a satisfied clause). Restriction of Φ by a partial assignment ρ comprises successive restriction by the literals in ρ. In contrast to the propositional case, it is important to note that *PCNF restrictions do not preserve falsity in general.*

The purpose of restricting a refutation π by a partial assignment ρ is to obtain a refutation of the restricted formula $\Phi|_\rho$. Naturally, one applies the assignment to the refutation and simplifies the result, eliminating the satisfied clauses $C_\top$. The procedure differs depending on the quantification of the assigned variable.

For an *existential literal* l, the restricted refutation $\pi|_l$ is obtained as follows. First, define any download or instantiation of $C_\top$ to be $C_\top$ itself. Replace the axioms of π with axioms

downloaded from (the matrix of) $\Phi|_l$, and proceed as in the propositional case, repeating the steps of π where possible. If a resolution step is not possible, modify it as follows: select as the resolvent the unique antecedent that is not $C_\top$ and does not contain the pivot variable, if this antecedent exists; otherwise, mark the resolvent as $C_\top$. In the resulting sequence, taking the subderivation of the first empty clause and removing all occurrences of $C_\top$ yields the restricted refutation $\pi|_l$.

For a *universal literal* l that is *unopposed* in π (meaning that $\neg l$ does not appear in the annotations), the restricted derivation $\pi|_l$ is obtained from π simply by removing l from the annotations. Restriction is not defined here for opposed universal literals.

Finally, for restriction by a partial assignment $\rho = \{l_1, \ldots, l_n\}$ with $\mathrm{var}(l_i) <_\Phi \mathrm{var}(l_{i+1})$ for each $i \in [n-1]$, we define $\pi|_\rho := \pi_n$, where $\pi_0 = \pi$ and $\pi_i = \pi_{i-1}|_{l_i}$ for each $i \in [n]$, provided that each intermediate restriction is defined.

Restrictions of IR-calc refutations feature heavily in Section 5, for which the following two propositions are needed. For self-containment of the present work, we provide a proof for each in the appendix.

Proposition 1 ([10]). *Let π be an* IR-calc *refutation of a PCNF Φ and let l be a literal with $\mathrm{var}(l) \in \mathrm{vars}(\Phi)$. Then $\pi|_l$ is an* IR-calc *refutation of $\Phi|_l$ if (a) l is existential or (b) l is universal and unopposed in π.*

Proposition 2. *Let π be an* IR-calc *derivation from a PCNF Φ, and let ρ and ρ' be partial assignments to the existential variables of Φ. If $\rho \subseteq \rho'$, then every annotation of $\pi|_{\rho'}$ is an annotation of $\pi|_\rho$.*

IR-calc strategy extraction. Strategy extraction is a well-known paradigm in QBF proof complexity [4,9,27,41], and has already been studied for IR-calc [10]. In summary, there exists an algorithm that takes a refutation and returns a winning $\forall$-strategy that we call the *extracted strategy*.

Starting with an IR-calc refutation π of a PCNF $\Phi = \exists X_1 \forall U_1 \cdots \exists X_n \forall U_n \exists X_{n+1} \,.\, \phi$, we build a winning $\forall$-strategy S, viewing Φ as a game of n rounds. In round one, the $\exists$-player chooses some total assignment α_1 to X_1, and we determine the $\forall$-player's response β_1 simply by collecting the U_1 literals appearing in the annotations of $\pi|_{\alpha_1}$ (all such variables are unopposed in $\pi|_{\alpha_1}$ by Proposition 3 below). Any absent variables are assigned to 0, so that β_1 is a total assignment to U_1. Now, $\pi|_{\alpha_1 \cup \beta_1}$ is a refutation of $\exists X_2 \forall U_2 \cdots \exists X_n \forall U_n \exists X_{n+1} \,.\, \phi|_{\alpha_1 \cup \beta_1}$, so we repeat the process to obtain the $\forall$-player's response for the next round.

In this way, one can obtain a response $S(\alpha)$ to each total assignment α to the existentials, such that $\alpha \cup S(\alpha)$ falsifies ϕ. Moreover, $S(\alpha)$ and $S(\alpha')$ must agree up to block U_i if α and α' agree up to block X_i. This serves as a proof sketch for the following proposition. A formal proof is provided in the appendix.

Proposition 3 ([10]). *If π is an* IR-calc *refutation of a PCNF Φ, then the strategy extracted from π is a winning $\forall$-strategy for Φ.*

5 A new lower-bound technique

In this section, we provide the details of our new lower-bound technique for IR-calc. We begin in Subsection 5.1 with a high-level overview, intended to illuminate some of the technical details. This is followed in Subsection 5.2 by the proof of the Weight Theorem, the main result at the heart of our technique.

5.1 High-level description of the technique

The technique that we present exploits a fundamental relationship between strategies and refutations.

A winning $\forall$-strategy S details precisely how the $\forall$-player should respond to any possible choice of the $\exists$-player. Indeed, in each game of S, a total assignment to the existential variables (the $\exists$-player's *choice*) is associated with a total assignment to the universals (the $\forall$-player's *response*).

Our lower-bound technique works by proving that parts of the responses of the extracted strategy must appear within the annotations of the refutation, providing a lower bound on the number of annotations (and therefore on refutation size). For that reason, the technique applies to formulas that *require large witnessing functions*, that is, every winning strategy contains $2^{\Omega(n)}$ different responses. As we noted in Section 2, this alone necessitates a large proof in the basic expansion calculus $\forall$Exp+Res, but it is not enough to yield a lower-bound for IR-calc. Consider the following definition of the *response size* of a strategy.[4]

Definition 4 (Response size). *The response size of a $\forall$-strategy S is the number of distinct responses in S.*

It is very easy to prove that large response size implies a $\forall$Exp+Res lower bound.

Proposition 5. *The size of a $\forall$Exp+Res refutation of a false PCNF Φ is at least the minimum response size of a winning $\forall$-strategy for Φ.*

To prove this statement, it is enough to observe that we can extract a strategy from an $\forall$Exp+Res refutation even if the response is collected from the download assignment, rather than the annotations. Hence, there is at least one axiom per response in the extracted strategy. Indeed, large response size is the only genuine reason for QBF hardness in $\forall$Exp+Res; QBFs admitting small (i.e. polynomial) response sizes can be fully expanded out into a succinct propositional formula, and the hardness is (at least in part) due to a classical resolution lower bound.

The following example illustrates that the situation for IR-calc is more complex.

Example 6. Consider the formulas $\mathcal{Q}(n)\,.\,\phi(n)$ with quantifier prefixes $\mathcal{Q}(n) = \exists x_1 \forall u_1 \exists t_1 \cdots \exists x_n \forall u_n \exists t_n$ and matrices $\phi(n)$ consisting of the clauses $\{x_i, u_i, t_i\}$ and $\{\neg x_i, \neg u_i, t_i\}$ (for $i \in [n]$) along with the clause $\{\neg t_1, \ldots, \neg t_n\}$. These formulas have a unique winning $\forall$-strategy, in which each u_i must be assigned the same truth value as x_i. As such, there are exactly 2^n responses (namely, the total assignments to the universal variables), implying an $\forall$Exp+Res lower bound by Proposition 5. However, the formulas have linear size IR-calc refutations. A section of the refutation is shown in Fig. 5.

It is natural to seek an understanding of the interplay between semantics and hardness for IR-calc, analogous to Proposition 5. In light of the preceding example, which illustrates that responses do not necessarily appear as IR-calc annotations, our principal insight is the identification of *parts* of the responses that *must* appear in an IR-calc refutation. For this, we consider instead the annotations in restricted refutations. We show that some universal variables are in some sense *critical* in the restricted formula, and that the response on critical variables – the *critical response* – is what must appear in the IR-calc annotations. Central to the method are restrictions by total assignments to all existentials except those in the final block (which is existential by convention). We call such assignments *restrictors*.

Definition 7 (restrictor). *Let Φ be a PCNF over variables V and let $T \subseteq V\exists$ be the rightmost quantifier block of Φ. Any total assignment to the variables $V_\exists \setminus T$ is a* restrictor *of Φ.*

[4] The response size of a strategy S is equal to the size of the range of the equivalent witnessing function.

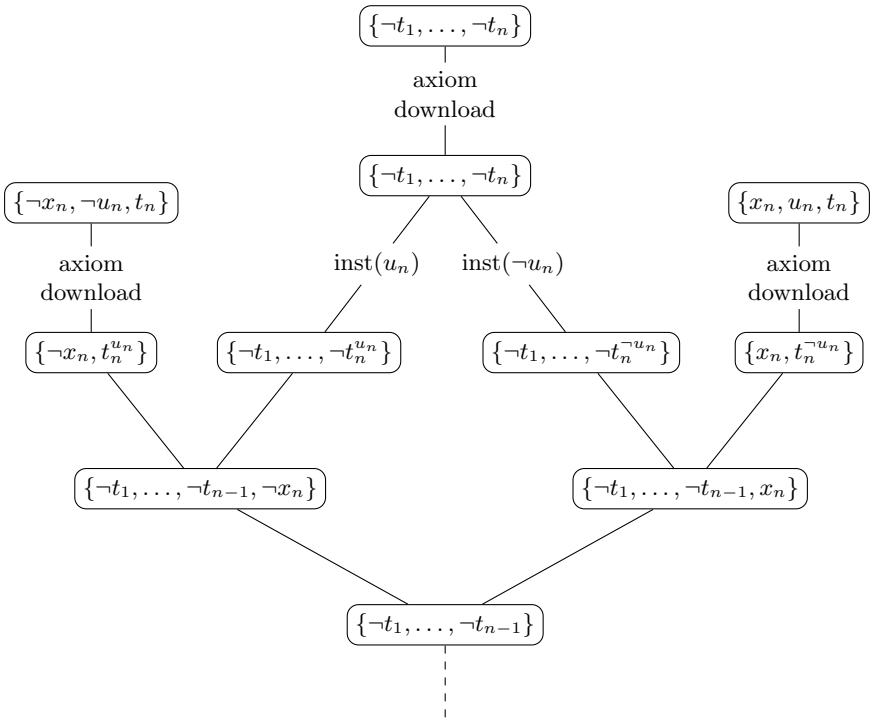

Fig. 5. The formulas from Example 6 have linear size IR-calc refutations. Here, the literal $\neg t_n$ is removed from the clause $\{\neg t_1, \ldots, \neg t_n\}$ in seven steps. After n repetitions, the empty clause is derived. Hence the n^{th} formula has a refutation of size $7n + 1$.

We refer to the annotation of the pivot variable of the final step of a refutation as the *final annotation*.

Definition 8 (final annotation). *Let π be an* IR-calc *refutation of a PCNF Φ, in which the empty clause is derived by resolution from the clauses $\{x^\tau\}$ and $\{\neg x^\tau\}$. Then τ is the* final annotation *of π.*

We may now provide a high-level summary of the technique. Given a refutation π of a PCNF Φ, we first select a restrictor δ. On the one hand, we apply strategy extraction to π to obtain the extracted strategy S, and take the response to δ, denoted $R(\delta, S)$. On the other hand, we restrict π by δ, and take the final annotation τ_δ of the restricted refutation $\pi|_\delta$. Now, we compare the response with the annotation. The definition of strategy extraction ensures that the literals in τ_δ are a subset of the response $R(\delta, S)$. We combine this with a proof that certain *critical variables* must occur in τ_δ. As a result, we obtain a subset of the response to δ, called the *critical response*, that must be contained in the annotation τ_δ. Since τ_δ appears as an annotation in the original refutation (Proposition 2), the critical response to a restrictor must always occur within some annotation in π. This is the central observation of our method, depicted in Figure 6.

Whereas some lower bound is always obtained by means of our technique, the magnitude of the bound is determined by the variation in the critical response as δ ranges through the restrictors of Φ. We introduce a measure of this variation that we call *weight*. In the next subsection we prove the Weight Theorem, that states that the size of an IR-calc refutation is lower-bounded by the weight of the formula.

5.2 Proof of the weight theorem

Following on from the discussion in the previous subsection, we work towards the definition of the critical response.

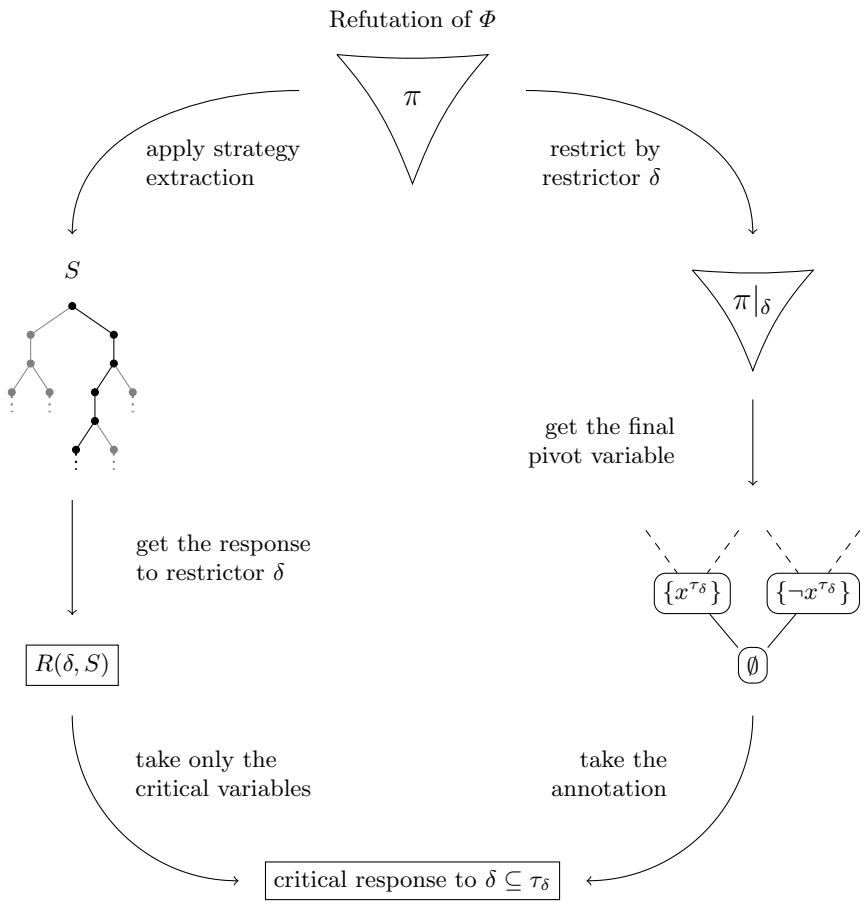

Fig. 6. Overview of the central observation of our lower-bound technique. The final statement is proved in Lemma 13.

We begin by proving a useful lemma that highlights a particular property of IR-calc derivations. Given a PCNF Φ whose first block U is universal, and an IR-calc derivation π from Φ, for a given clause C the U-literals are the same in each annotation. Moreover, the U-literals appearing in the annotations of C contain all those appearing in the subderivation of C.

Lemma 9. *Let π be an* IR-calc *derivation from a PCNF Φ whose first block U is universal. There exists a function f such that, for each clause C in π, (a) $f(C) = \{l \in \tau \mid \mathrm{var}(l) \in U\}$ for each annotation τ in C, and (b) $f(C') \subseteq f(C)$ for each C' in the subderivation of C.*

Proof. We prove the lemma by induction on subderivation depth. Let $\Phi = \mathcal{Q}\,.\,\phi$ and let C be a clause in π with subderivation depth d.

For the base case $d = 1$, clause C is an axiom and was downloaded from some clause $D \in \phi$. Since every existential variable in D is right of block U, every literal in C is annotated with the negation of every U-literal in D. Hence, setting $f(C) = \{\neg l \mid l \in D, \mathrm{var}(l) \in U\}$ satisfies both conditions (a) and (b).

For the inductive step, let $d \geq 2$, and assume that the lemma holds for all clauses with subderivation depth $d - 1$. There are two cases. (1) If $C = \mathrm{inst}(C', \tau)$, then $f(C')$ satisfies both conditions by the inductive hypothesis. Instantiation by any U-literal therefore applies identically to each annotation in C', so setting $f(C) = f(C') \circ \{l \in \tau \mid l \in \mathrm{vars}(U)\}$ satisfies both conditions. (2) If $C = \mathrm{res}(C', C'', x^\tau)$, then the U-literals of any annotation in $C' \cup C''$ are those of the pivot annotation, by the inductive hypothesis. Hence, setting $f(C) = f(C') = f(C'')$ satisfies both conditions.

This concludes the inductive step and the proof. □

We continue by identifying the set of universal variables, called the *critical variables*, that occur in every minimally unsatisfiable subformula of a QBF Φ. This is a useful notion in the present context, precisely because PCNF restrictions do not preserve minimal unsatisfiability. From the point of view of IR-calc, the critical variables of a formula are exactly the universal variables appearing in the annotations of every refutation.

Definition 10 (critical variables). *Let $\Phi = \mathcal{Q} \,.\, \phi$ be a false PCNF. The* critical variables *of Φ are the universal variables appearing in every matrix ϕ' for which (a) $\phi' \subseteq \phi$ and (b) $\mathcal{Q} \,.\, \phi'$ is false.*

We use the phrases 'u is a critical variable of Φ' and 'u is critical in Φ' synonymously. With the following corollary to Lemma 9, we prove that the critical variables of a Π_2-prefix PCNF must occur in the final annotation of any IR-calc refutation. This is a particularly useful result in the sequel, since restricting a formula by any restrictor yields a Π_2-prefix PCNF.

Corollary 11. *Let π be an* IR-calc *refutation of a PCNF $\Phi = \forall U \exists T \,.\, \phi$, and let τ be the final annotation of π. Then* $\mathrm{vars}(\tau)$ *contains every critical variable of Φ.*

Proof. Let the conclusion of π, which is the empty clause, be derived from clauses C and C'. Every critical variable of Φ must appear in a download clause of π, for otherwise the download clauses are true under the prefix $\forall U \exists T$ by definition of the critical variables. Hence, assuming w.l.o.g. that there are no trailing universal literals in the download clauses, every critical variable appears in some annotation of some axiom clause. Since every axiom clause of π is in the subderivation of at least one of C and C', the final annotation contains the U-literals of every annotation in every axiom, by Lemma 9. The corollary follows since every critical variable of Φ is in block U. □

We may now define the critical response for a restrictor δ, with respect to a strategy S and a PCNF Φ. We call a literal l in the response to δ 'critical' if and only if $\mathrm{var}(l)$ is a critical variable of the restricted formula $\Phi|_\delta$.

Definition 12 (critical response). *Let S be a winning $\forall$-strategy for a PCNF Φ, let δ be a restrictor of Φ and let $R(\delta, S)$ be the response to δ in S. The* critical response *to δ with respect to S and Φ is the set of literals $\{l \in R(\delta, S) \mid \mathrm{var}(l)$ is critical in $\Phi|_\delta\}$.*

Now we prove that the critical response to δ, with respect to the strategy extracted from a refutation π, satisfies the desired property: it is a subset of the final annotation of the restricted refutation $\pi|_\delta$.

Lemma 13. *Let π be an* IR-calc *refutation of a PCNF Φ and let S be the strategy extracted from π. Then, for each restrictor δ of Φ, the final annotation of $\pi|_\delta$ contains the critical response to δ with respect to S and Φ.*

Proof. If Φ contains no universal variables, then Φ has no restrictors and the lemma is vacuously true, so we assume otherwise.

Let δ be a restrictor of Φ, let τ_δ be the final annotation of $\pi|_\delta$, and let $R(\delta, S)$ be the response to δ in S. Observe that $\Phi|_\delta$ has a Π_2 prefix, and hence $\mathrm{vars}(\tau_\delta)$ contains the critical variables of $\Phi|_\delta$, by Corollary 11. We claim that $\tau_\delta \subseteq R(\delta, S)$. From this claim it follows that every literal in $R(\delta, S)$ whose variable is critical in the restricted formula $\Phi|_\delta$ must appear in the final annotation; that is, it follows that τ_δ contains the critical response $\{l \in R(\delta, S) \mid \mathrm{var}(l)$ is critical in $\Phi|_\delta\}$.

To prove the claim, we first introduce some notation. Let $\Phi = \mathcal{Q} \,.\, \phi$ where $\mathcal{Q} = \exists X_1 \forall U_1 \cdots \exists X_n \forall U_n \exists X_{n+1}$ (if the first block of Φ is universal, take X_1 as the empty set). Then, for each

$i \in [n]$, let α_i be the literals of δ from the first i existential blocks, and let β_i be the literals of $R(\delta, S)$ from the first i universal blocks. Finally, let β_0 be empty.

At the i^{th} stage of the strategy extraction process, the universal literals collected from the annotations of the restricted refutation are unopposed (this follows from Lemma 9). By Proposition 2, existential restrictions cannot introduce new literals into the annotations. It follows that, for each $i \in [n]$, the literals in β_{i-1} are unopposed in $\pi|_{\alpha_i}$. Hence, $\pi|_{\alpha_i \cup \beta_{i-1}}$ may be obtained from $\pi|_{\alpha_i}$ simply by removing the literals β_{i-1} from the annotations.

Now, let $l \in \tau_\delta$ with $\text{var}(l) \in U_i$. Since $\alpha_i \subseteq \delta$, τ_δ is an annotation in $\pi|_{\alpha_i}$, by Proposition 2. Then, since $l \notin \beta_{i-1}$, it follows that l appears in some annotation in $\pi|_{\alpha_i \cup \beta_{i-1}}$. Hence $l \in R(\delta, S)$. □

Our lower bound technique is concerned with identifying a certain amount of variation in the critical responses as δ ranges through the restrictors of Φ. More precisely, we are interested in the case where two critical responses are inconsistent with one another; that is, a literal l appears in one critical response and $\neg l$ appears in the other. For that reason, given a winning $\forall$-strategy S, we define the *critical response graph* that has a vertex for each critical response and an edge between each inconsistent pair.

Definition 14 (critical response graph). *Let S be a winning $\forall$-stragegy for a PCNF Φ. The* critical response graph *of S with respect to Φ is the undirected graph $G(S, \Phi)$ defined as follows:*

- *For each restrictor δ of Φ, $G(S, \Phi)$ has a vertex labelled with the critical response to δ with respect to S and Φ.*
- *$G(S, \Phi)$ has an edge between two vertices if and only if their labels are inconsistent with one another.*

The following lemma states that the number of annotations in a refutation is lower bounded by the clique number of the critical response graph for the extracted strategy. The clique number $\omega(G)$ of a graph G is the size of the largest clique in the graph.

Lemma 15. *Let S be the strategy extracted from an* IR-calc *refutation π of a PCNF Φ. Then there are at least $\omega(G(S, \Phi))$ distinct annotations in π.*

Proof. Let $k = \omega(G(S, \Phi))$, and let $\delta_1, \ldots, \delta_k$ be restrictors of Φ whose critical responses (with respect to S and Φ) are pairwise inconsistent. For each $i \in [k]$, the final annotation τ_{δ_i} of $\pi|_{\delta_i}$ contains the critical response to δ_i, by Lemma 13, and τ_{δ_i} appears as an annotation in π, by Proposition 2. Hence, for each $i, j \in [k]$ with $i \neq j$, τ_{δ_i} and τ_{δ_j} are distinct annotations appearing in π. Therefore π contains at least k distinct annotations. □

Lemma 15 leads to a natural proof-size measure for IR-calc, since the proof is at least as large as the number of distinct annotations. Hence, the minimal clique number of a critical response graph for a winning $\forall$-strategy of a PCNF Φ yields a size lower bound for an arbitrary refutation. This motivates the following definition, in which we define the *weight* of a PCNF, denoted $\mu(\Phi)$, to be equal to this minimal clique number.

Definition 16 (weight). *Let Φ be a false PCNF. The* weight *of Φ is $\mu(\Phi) = \min\{\omega(G(S, \Phi)) \mid S$ is a winning $\forall$-strategy for $\Phi\}$.*

We can now show the main theorem for our lower bound technique, stating that the weight provides a precise lower bound for the proof size in IR-calc.

Theorem 17 (Weight Theorem). *The size of any* IR-calc *refutation of a QBF Φ is at least the weight of Φ.*

Proof. Let S be the strategy extracted from a refutation π of Φ. By Lemma 15, there are at least $\omega(G(S,\Phi))$ distinct annotations appearing in π. A new annotation can be introduced into the refutation only by application of an IR-calc rule, so the size of π is at least $\omega(G(S,\Phi))$. Since S is a winning $\forall$-strategy by Proposition 3, the weight of Φ is at most $\omega(G(S,\Phi))$. □

6 Two applications to existing formulas

In this section, we apply the Weight Theorem to obtain hardness proofs for two formula families from the QBF literature. In Subsection 6.1, we show that our approach yields a much-simplified proof of a known result. In Subsection 6.2 we modify the prefix order of a known formula family to prove a new IR-calc lower bound.

6.1 An improved proof of hardness for the formulas of Kleine Büning et al.

We apply the Weight Theorem to a well-studied formula family KBKF(n) introduced by Kleine Büning et al. [33]. By showing that the formulas have large weight, we produce an alternative proof of their IR-calc hardness that is a great deal shorter than the original proof in [11]; moreover, we believe that the content of the Weight Theorem provides the first clear intuition for the lower bound. We recall the formulas below.

Definition 18. *We define the formula family*

$$\mathrm{KBKF}(n) = \exists x_1 y_1 \forall u_1 \cdots \exists x_n y_n \forall u_n \exists t_1 \cdots t_n \,.\, \xi(n)\,,$$

where the matrix $\xi(n)$ consists of the clauses

$$\begin{aligned}
&C_0 = \{\neg x_1, \neg y_1\}, && && \\
&C_i = \{x_i, u_i, \neg x_{i+1}, \neg y_{i+1}\}, && C'_i = \{y_i, \neg u_i, \neg x_{i+1}, \neg y_{i+1}\}\,, && \text{for } i \in [n-1], \\
&C_n = \{x_n, u_n, \neg t_1, \ldots, \neg t_n\}, && C'_n = \{y_n, \neg u_n, \neg t_1, \ldots, \neg t_n\}\,, && \\
&D_i = \{u_i, t_i\}, && D'_i = \{\neg u_i, t_i\} && \text{for } i \in [n]\,.
\end{aligned}$$

We pause to demonstrate that the formulas are indeed false, and, at the same time, provide some intuition about them. In the two-player game interpretation, KBKF(n) comprises n 'rounds', with variables x_i, y_i, u_i assigned in the i^{th} round. In round one, to avoid immediate loss on clause C_0, the $\exists$-player must assign either x_1 or y_1 false. If x_1 (resp. y_1) is assigned false, then y_1 (resp. x_1) is a pure literal and should be assigned true. In return, the $\forall$-player must assign u_1 false if and only if x_1 was assigned false, or otherwise allow the $\exists$-player to win by assigning all remaining existentials true. When round two begins, the formula on the board is syntactically equivalent to KBKF($n-1$), and hence best play dictates that the pattern repeats for n rounds. Immediately after round n, the board comprises the unsatisfiable clauses $\{t_1\}, \ldots, \{t_n\}$ and $\{\neg t_1, \ldots, \neg t_n\}$, and hence the $\exists$-player loses.

With best play, in round i the $\exists$-player has a choice of falsifying either x_i or y_i, and hence determines the winning assignment of u_i one way or the other. As such, even though $\exists$ cannot win the game, she can force $\forall$ to play any given assignment in order to win. As we noted in Subsection 5.1, this alone is not enough for an IR-calc lower bound; for that, we must show that the formulas have large weight.

Theorem 19. *For each $n \in \mathbb{N}$, KBKF(n) has weight 2^n.*

Proof. Let $n \in \mathbb{N}$, let S be a winning $\forall$-strategy for KBKF(n), let $\mathcal{Q}(n)$ denote the quantifier prefix of KBKF(n), and put $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$ and $U = \{u_1, \ldots, u_n\}$. Further, let Γ be the set containing exactly the games γ of S satisfying the following condition: either $\neg x_i \in \gamma$ or $\neg y_i \in \gamma$ but not both, for each $i \in [n]$. Finally, let $\gamma \in \Gamma$.

We first observe that we must have $\neg u_i \in \gamma \Leftrightarrow \neg x_i \in \gamma$ and $u_i \in \gamma \Leftrightarrow \neg y_i \in \gamma$ for each $i \in [n]$, for otherwise both C_i and C'_i are satisfied at the end of round i, and the $\exists$-player can win by assigning all remaining existential variables positively. Now, let $\delta = \{l \in \gamma \mid \mathrm{var}(l) \in X \cup Y\}$, and observe that δ is a restrictor of Φ. It is readily verified that

(i) $\xi(n)|_\delta = \{C, D_1, D'_1, \ldots, D_n, D'_n\}$ where $C = \{l, \neg t_1, \ldots, \neg t_n\}$ and $l \in \{u_n, \neg u_n\}$, and
(ii) if $\xi' \subseteq \xi(n)|_\delta$ and $\mathcal{Q}(n) \,.\, \xi'$ is false, then ξ' must contain C and some $E_i \in \{D_i, D'_i\}$ for each $i \in [n]$.

It follows that the critical variables of $\mathrm{KBKF}(n)|_\delta$ are exactly U, and therefore that the critical response to δ (w.r.t. S and Φ) is $\{l \in \gamma \mid \mathrm{var}(l) \in U\}$.

Finally, we observe that $\{\{l \in \gamma \mid \mathrm{var}(l) \in U\} \mid \gamma \in \Gamma\}$ is the set of total assignments to U, hence each total assignment to U is a critical response to some restrictor w.r.t. S and Φ. Since there are 2^n possible critical responses that are all pairwise inconsistent, we have $\omega(G(S, \mathrm{KBKF}(n))) = 2^n$. Therefore $\mu(\mathrm{KBKF}(n)) = 2^n$. □

The IR-calc hardness is immediate from Theorem 17 and Theorem 19.

Corollary 20 ([11]). *The size of an* IR-calc *refutation of* $\mathrm{KBKF}(n)$ *is at least* 2^n.

6.2 An adaptation of the formulas of Janota and Marques-Silva

As our second example we consider a formula family introduced by Janota and Marques-Silva [32]. In that paper, the authors showed that the formulas require exponential-size refutations in the basic expansion calculus ∀Exp+Res, which is simulated by IR-calc. Here we show that a simple adaptation, namely a reordering of the quantifier prefix[5], yields a formula family with large weight that is hard even for the stronger system IR-calc.

Definition 21 (adapted from [32]). *We define the formula family*

$$\Phi(n) = \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_{2n} \,.\, \phi(n),$$

where the matrix $\phi(n)$ *consists of the clauses*

$$\begin{aligned} &B_n = \{\neg t_1, \ldots, \neg t_{2n}\}, \\ &C_i = \{\neg x_i, t_{2i-1}\}, \qquad C'_i = \{\neg u_i, t_{2i-1}\}, \qquad \text{for } i \in [n], \\ &D_i = \{x_i, t_{2i}\}, \qquad D'_i = \{u_i, t_{2i}\}, \qquad \text{for } i \in [n]. \end{aligned}$$

A useful property of these formulas is that there exists a unique winning ∀-strategy. In assigning variable x_i, the $\exists$-player leaves exactly one of the unit clauses $\{t_{2i-1}\}$ or $\{t_{2i}\}$ on the board, and the ∀-player can only win the game by bringing in the other. There is only one way to do this: by assigning each u_i the opposite truth value to that of x_i. We use this fact to prove that the formulas have large weight.

Theorem 22. *For each* $n \in \mathbb{N}$, $\Phi(n)$ *has weight* 2^n.

Proof. Let $n \in \mathbb{N}$, let $\mathcal{Q}(n)$ denote the quantifier prefix of $\Phi(n)$, and put $X = \{x_1, \ldots, x_n\}$, $U = \{u_1, \ldots, u_n\}$ and $T = \{t_1, \ldots, t_{2n}\}$. Further, let S be a winning ∀-strategy for Φ and let γ be a game of S.

We observe that S must be the unique strategy satisfying $u_i \in \gamma \Leftrightarrow \neg x_i \in \gamma$ and $\neg u_i \in \gamma \Leftrightarrow x_i \in \gamma$ for each $i \in [n]$. Now, let $\delta = \{l \in \gamma \mid \mathrm{var}(l) \in X\}$, and observe that δ is a restrictor of Φ. It is readily verified that

(i) $\phi(n)|_\delta = \{B_n, C'_1, D'_1, E_1, \ldots, C'_n, D'_n, E_n\}$ where $E_i \in \{\{t_{2i-1}\}, \{t_{2i}\}\}$ for each $i \in [n]$, and

[5] The original formulas in [32] have the quantifier prefix $\exists x_1 \forall u_1 \exists t_1 \cdots \exists x_n \forall u_n \exists t_n$.

(ii) if $\phi' \subseteq \phi(n)|_\delta$ and $\mathcal{Q}(n) \,.\, \phi'$ is false, then either C'_i or D'_i is in ϕ', for each $i \in [n]$.

It follows that the critical variables of $\Phi|_\delta$ are exactly U, and therefore that the critical response to δ w.r.t. S and Φ is $\{l \in \gamma \mid \mathrm{var}(l) \in U\}$. This implies that $\{\{l \in \gamma \mid \mathrm{var}(l) \in U\} \mid \gamma \in S\}$ is the set of total assignments to U, and the conclusion of the proof is identical to that of Theorem 19. □

The IR-calc hardness is again immediate from Theorems 17 and 22.

Corollary 23. *The size of an* IR-calc *refutation of* $\Phi(n)$ *is at least* 2^n.

7 Application to proof complexity of QBF dependency calculi

In this section, we introduce the QBF family of equality formulas and apply the Weight Theorem to obtain a new IR-calc lower bound. By proving contrasting linear upper bounds in QBF dependency calculi, we present the first proof complexity results for QBF calculi parametrised by dependency scheme. Our results cover both expansion- and CDCL-based calculi. In particular, we answer the open problem of the relative complexities of Q-Res and Q($\mathcal{D}^{\mathsf{rrs}}$)-Res, first posed in [49] (the conference version of [50]), by proving an exponential separation.

7.1 New hard equality formulas

Defined below, our new equality formulas $\Psi(n)$ are superficially similar to the adapted formulas $\Phi(n)$ from the previous section, yet they are simpler, using fewer variables and clauses. Indeed, from the content of the Weight Theorem, we have distilled precisely what is needed to obtain an IR-calc lower bound; as a result, $\Psi(n)$ is arguably the simplest and most concise PCNF family with exponential weight.

Definition 24. *The* equality formulas *are the formula family*

$$\Psi(n) = \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \,.\, \psi(n)$$

where the matrix $\psi(n)$ *consists of the clauses*

$$B_n = \{\neg t_1, \ldots, \neg t_n\},$$
$$C_i = \{x_i, u_i, t_i\}, \quad D_i = \{\neg x_i, \neg u_i, t_i\}, \quad \text{for } i \in [n].$$

Proving the exponential weight of $\Psi(n)$ is aided by the fact that the ∀-player is forced to play $u_i \equiv x_i$ for each $i \in [n]$.

Theorem 25. *For each* $n \in \mathbb{N}$*, the equality formula* $\Psi(n)$ *has weight* 2^n.

Proof. Let $n \in \mathbb{N}$, let $\mathcal{Q}(n)$ denote the quantifier prefix of $\Psi(n)$, and put $X = \{x_1, \ldots, x_n\}$, $U = \{u_1, \ldots, u_n\}$ and $T = \{t_1, \ldots, t_n\}$. Further, let S be a winning ∀-strategy for $\Psi(n)$ and let $\gamma \in \mathrm{games}(S)$.

Observe that S must be the unique strategy satisfying $u_i \in \gamma \Leftrightarrow x_i \in \gamma$ and $\neg u_i \in \gamma \Leftrightarrow \neg x_i \in \gamma$ for each $i \in [n]$, for otherwise the ∃-player could win by assigning t_i false and all variables in $T \setminus \{t_i\}$ true. Now, let $\delta = \{l \in \gamma \mid \mathrm{var}(l) \in X\}$, and observe that δ is a restrictor of Ψ. It is readily verified that

(i) $\psi(n)|_\delta = \{B_n, E_1, \ldots, E_n\}$ where $E_i \in \{\{u_i, t_i\}, \{\neg u_i, t_i\}\}$ for each $i \in [n]$, and
(ii) $\mathcal{Q}(n) \,.\, \psi'$ is true for any proper subset $\psi' \subsetneq \psi(n)|_\delta$.

It follows that the critical variables of $\Psi|_\delta$, once again, are exactly those in the set U. The proof concludes identically to that of Theorem 19. □

Corollary 26. *The size of an* IR-calc *refutation of* $\Psi(n)$ *is at least* 2^n.

Proof. Immediate from Theorem 17 and Theorem 25. □

7.2 Dependency schemes and QBF dependency calculi

In this subsection, we complement the high-level overview of QBF dependency schemes from Section 2.3 by giving the definitions for dependency schemes and QBF dependency calculi, required to put our proof complexity results into context.

In summary, some of the dependencies implied by the quantifier prefix of a PCNF may be ignored, while preserving the truth value of the formula. Identification of so-called *spurious* dependencies is the role of the dependency scheme.

Dependency schemes. For each PCNF Φ, we define the *trivial dependency relation* $\mathcal{D}^{\mathrm{trv}}(\Phi) = \{(u,x) \in \mathrm{vars}_\forall(\Phi) \times \mathrm{vars}_\exists(\Phi) \mid u <_\Phi x\}$. A *proto-dependency scheme* (PDS) $\mathcal{D}$ is a mapping from the set of all PCNFs, satisfiying $\mathcal{D}(\Phi) \subseteq \mathcal{D}^{\mathrm{trv}}(\Phi)$ for each PCNF Φ.

Defined by Slivovsky and Szeider [49], the *reflexive resolution path dependency scheme* $\mathcal{D}^{\mathrm{rrs}}$ is arguably the most important[6] PDS. The scheme works by appeal to the syntactic form of an instance, and uses connections via the clauses of the matrix to identify dependent variables. Variable *independence* is therefore identified by the absence of such connections. We recall $\mathcal{D}^{\mathrm{rrs}}$ with the following definition.

Definition 27 (Reflexive resolution path dependency scheme [50]). *Let $\Phi = \mathcal{Q} \,.\, \phi$ be a PCNF, and let $(u,x) \in \mathcal{D}^{trv}(\Phi)$. Then $(u,x) \in \mathcal{D}^{rrs}(\Phi)$ if and only if there is a sequence of clauses $C_1, \ldots, C_n \in \phi$ and a sequence of existential literals $l_1, \ldots, l_{n-1}$ for which the following four conditions hold:*

(a) $u \in C_1$ *and* $\neg u \in C_n$,
(b) $x = \mathrm{var}(l_i)$, *for some* $i \in [n-1]$,
(c) $u <_\Phi \mathrm{var}(l_i)$, $l_i \in C_i$ *and* $\neg l_i \in C_{i+1}$, *for each* $i \in [n-1]$,
(d) $\mathrm{var}(l_i) \neq \mathrm{var}(l_{i+1})$ *for each* $i \in [n-2]$.

Dependency calculi. Due to variable dependencies implied by the quantifier prefix, a QBF calculus must use implicit references to the trivial dependency scheme $\mathcal{D}^{\mathrm{trv}}$. The parameterisation of a QBF calculus P by a general PDS $\mathcal{D}$ yields the corresponding *dependency calculus* $\mathsf{P}(\mathcal{D})$, in which the implicit references to the trivial dependencies are replaced by explicit references to the stronger scheme $\mathcal{D}$.

In Fig. 7, we introduce the dependency calculus $\mathsf{IR}(\mathcal{D})$-calc. The effect of the parameterisation by $\mathcal{D}$ is that literals are annotated only by the universal variables on which they depend *according to* $\mathcal{D}$. This is consistent with the observation that any existential x that is independent of a universal u need not be duplicated in the expansion by u.

The fact that $\mathsf{IR}(\mathcal{D}^{\mathrm{rrs}})$-calc is a sound and complete proof system is a non-trivial consequence of existing results, involving the related field of DQBF (cf. [8, 16]). Since proofs of soundness are off-topic for the current paper, we provide a proof sketch in the appendix. Definitions of the other dependency calculi we refer to (namely $\mathsf{Q}(\mathcal{D})$-Res, $\mathsf{QU}(\mathcal{D})$-Res and $\forall\mathsf{Exp}(\mathcal{D})$+Res) are also given in the appendix.

7.3 Proof complexity results for dependency calculi

We first prove a linear upper bound for the equality formulas in the tree-like versions of two dependency calculi.

Lemma 28. *There exist tree-like linear size* $\mathsf{Q}(\mathcal{D})$-Res *and* $\forall\mathsf{Exp}(\mathcal{D})$+Res *refutations of* $\Psi(n)$.

[6] $\mathcal{D}^{\mathrm{rrs}}$ is currently the strongest known PDS $\mathcal{D}$ for which $\mathsf{Q}(\mathcal{D})$-Res is sound.

$$\frac{}{\{a^{\tau(a)} \mid a \in C_\exists\}}\ (\text{axiom}(C)) \qquad \begin{array}{l} C \text{ is a clause in the matrix of } \Phi. \\ \tau(a) = \{\neg l \mid l \in C_\forall, (\text{var}(l), \text{var}(a)) \in \mathcal{D}(\Phi)\}. \end{array}$$

$$\frac{C}{\{a^{\sigma \circ \tau(a)} \mid a^\sigma \in C\}}\ (\text{inst}(\tau)) \qquad \begin{array}{l} \tau \text{ is a partial assignment to the universal variables.} \\ \tau(a) = \{l \in \tau \mid (\text{var}(l), \text{var}(a)) \in \mathcal{D}(\Phi)\}. \end{array}$$

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

Fig. 7. The rules of IR($\mathcal{D}$)-calc. Note that $\Phi = \mathcal{Q}\,.\,\phi \in$ PCNF.

Proof. We first show that $\mathcal{D}^{\text{rrs}}(\Psi(n)) = \emptyset$; that is, for each $i, j \in [n]$, we show that $(u_i, t_j) \notin \mathcal{D}^{\text{rrs}}(\Psi(n))$. Let u and t be variables from the second and third block of $\Psi(n)$, respectively. For the sake of contradiction, suppose that there exists a sequence of clauses $E_1, \ldots, E_m \in \psi(n)$ and a sequence of existentials literals $l_1, \ldots, l_{m-1}$ satisfying the following four conditions:

(a) $u \in E_1$ and $\neg u \in E_m$,
(b) $t = \text{var}(l_i)$, for some $i \in [m-1]$.
(c) $u <_\Phi \text{var}(l_i)$, $l_i \in E_i$ and $\neg l_i \in E_{i+1}$, for each $i \in [m-1]$,
(d) $\text{var}(l_i) \neq \text{var}(l_{i+1})$ for each $i \in [m-2]$.

By (a), E_1 must be the clause C_a from $\psi(n)$ for some $a \in [n]$, and it follows from (c) that l_1 is the positive literal t_a. Then, again by (c), we have $\neg t_a \in E_2$, so E_2 must be the clause B_n from $\psi(n)$. Then, by (d), $l_2 = \neg t_b$ for some $b \in [n]$ with $b \neq a$. Then, by (c), we have $t_b \in E_3$, so E_3 must be the clause D_b from $\psi(n)$. Observe that the only existential variable right of u in D_b is t_b. We reach a contradiction, since so no suitable literal l_3 exists, and neither of E_2 nor E_3 contains $\neg u$.

It remains to construct the refutations. In Q($\mathcal{D}^{\text{rrs}}$)-Res one can $\forall$-reduce any universal literal in any input clause, and hence one may derive in $2n$ steps all the clauses $C'_i = C_i \setminus \{u_i\}$ and $D'_i = D_i \setminus \{\neg u_i\}$ for $i \in [n]$. In further n steps, one derives the n clauses $\{t_1\}, \ldots, \{t_n\}$ by resolution of each pair C'_i, D'_i over pivot x_i. In a further n steps, one can obtain the empty clause by resolving every negative literal out of B_n. The complete refutation comprises $O(n)$ clauses and is a tree. The $\forall$Exp($\mathcal{D}$)+Res proof is identical. □

We may now pool our results and prove the following theorem. Part of the simulation order of dependency calculi is depicted in Figure 1 (in Section 2).

Theorem 29. *Let* P *be any of the proof systems* IR-calc, $\forall$Exp+Res, Q-Res, *and* QU-Res. *Then* P *simulates neither* P($\mathcal{D}^{\text{rrs}}$) *nor tree-like* P($\mathcal{D}^{\text{rrs}}$).

Proof. We note that, for any PDS $\mathcal{D}$, (i) IR($\mathcal{D}$)-calc trivially simulates $\forall$Exp($\mathcal{D}$)+Res and (ii) QU($\mathcal{D}$)-Res trivially simulates Q($\mathcal{D}$)-Res.

The theorem is proved easily for the first three proof systems: for IR-calc, the theorem is immediate from Corollary 26, Lemma 28 and the fact that IR($\mathcal{D}^{\text{rrs}}$)-calc simulates $\forall$Exp($\mathcal{D}^{\text{rrs}}$)+Res; for $\forall$Exp+Res, it is immediate from Corollary 26, Lemma 28 and the fact that IR-calc simulates $\forall$Exp+Res; and for Q-Res, it follows from Corollary 26, Lemma 28 and the fact that IR-calc simulates Q-Res [10].

Since QU-Res is incomparable with IR-calc [11], Corollary 26 does not immediately imply a lower-bound for QU-Res. However, we can use the trick of doubling universal variables, introduced in [5], to produce one.

Let $X = \{x_1, \ldots, x_n\}$ and define $U = \{u_1, \ldots, u_n\}$, $U' = \{u'_1, \ldots, u'_n\}$ and T similarly. Let $\Psi'(n)$ be the formula family with quantifier prefixes $\exists X \forall (U \cup U') \exists T$ and matrices consisting of the clauses $\{C_1, D_1, \ldots, C_n, D_n, B_n\}$, where $B_n = \{\neg t_1, \ldots, \neg t_n\}$ and $C_i = \{x_i, u_i, u'_i, t_i\}$, $D_i = \{\neg x_i, \neg u_i, \neg u'_i, t_i\}$ for each $i \in [n]$. Since universal tautologies are disallowed, universal resolution steps can be performed on variable u_i only after variable u'_i has been removed from both antecedent clauses by $\forall$-reduction. However, under such circumstances variable u_i could also be removed from either antecedent clause by $\forall$-reduction. It follows that QU-Res refutations of $\Psi'(n)$ are at least as large as Q-Res refutations of $\Psi(n)$.

Finally, we observe that the doubling of universal variables does not adversely affect the dependency scheme; in fact, $(u_i, t_j) \in \mathcal{D}^{\mathrm{rrs}}(\Psi') \Leftrightarrow (u'_i, t_j) \in \mathcal{D}^{\mathrm{rrs}}(\Psi')$, for each $i, j \in [n]$. Hence, $\Psi'(n)$ have linear size tree-like $\mathsf{QU}(\mathcal{D}^{\mathsf{rrs}})$-Res refutations.

The theorem then follows from Corollary 26, Lemma 28, the fact that IR-calc simulates Q-Res [10] and the fact that $\mathsf{QU}(\mathcal{D}^{\mathsf{rrs}})$-Res simulates $\mathsf{Q}(\mathcal{D}^{\mathsf{rrs}})$-Res. □

We conclude this section by showing that Theorem 29 can in fact be improved, delivering a stronger practical import. The following proposition states that the *standard dependency scheme* $\mathcal{D}^{\mathrm{std}}$ [44] is unable to identify any non-trivial independencies in the formulas $\Psi(n)$. We omit the proof as the proposition is immediate from the definition of $\mathcal{D}^{\mathrm{std}}$.

Definition 30 (Standard dependency scheme [44]). *Let $\Phi = \mathcal{Q} \,.\, \phi$ be a PCNF. The pair $(u, x) \in \mathcal{D}^{trv}(\Phi)$ is in $\mathcal{D}^{std}(\Phi)$ if and only if there exists a sequence of clauses $C_1, \ldots, C_n \in \phi$ with $u \in \mathrm{vars}(C_1)$, $x \in \mathrm{vars}(C_n)$, such that, for each $i \in [n-1]$, $\mathrm{vars}(C_i) \cap \mathrm{vars}(C_{i+1})$ contains an existential variable right of u.*

Proposition 31. *For each $n \in \mathbb{N}$, $\mathcal{D}^{\mathrm{std}}(\Psi(n)) = \mathcal{D}^{\mathrm{trv}}(\Psi(n))$.*

Immediate from this proposition is the fact that every $\mathsf{P}(\mathcal{D}^{\mathsf{std}})$ refutation of $\Psi(n)$ is a P refutation, yielding the following corollaries.

Corollary 32. *The size of an $\mathsf{IR}(\mathcal{D}^{\mathsf{std}})$-calc refutation of $\Psi(n)$ is at least 2^n.*

Corollary 33. *Let P be any of the proof systems IR-calc, ∀Exp+Res, Q-Res, and QU-Res. Then $\mathsf{P}(\mathcal{D}^{\mathsf{std}})$ simulates neither $\mathsf{P}(\mathcal{D}^{\mathsf{rrs}})$ nor tree-like $\mathsf{P}(\mathcal{D}^{\mathsf{rrs}})$.*

Given that the state-of-the-art dependency-aware solver DepQBF is currently using dependency analysis based on the standard dependency scheme [38], Corollary 33 illustrates that an implementation based on the reflexive resolution path dependency scheme would be exponentially stronger. Whereas $\mathcal{D}^{\mathrm{rrs}}$ is tractable [48], the cubic algorithm computing the exact scheme is prohibitively complex for use in practice. Corollary 33 therefore supports the notion that there is untapped potential in the development of better algorithms for QBF.

8 Conclusion

Our use of the Weight Theorem in Sections 6 and 7 reveals a further unification of the three formula families we consider there, namely that the critical literal sets of any winning strategy *include all total assignments to the universal variables*. The weight of such formulas is exponential in the number of universal variables.

Due to the semantic flavour of the weight measure, we believe that our new technique will have further applications beyond these initial examples we present here. As we identify a general reason for hardness in QBF, our method has the clear potential to guide the search for new formulas or even classes of formulas hard in IR-calc (and Q-Res) and possibly even in further QBF (resolution-type) systems.

The importance of constructing new hard QBF instances is already demonstrated through our newly established separations of QBF calculi with dependency schemes, and by these results our work also gains a practical impact.

Our separations between the expansion-based calculi IR-calc and ∀Exp+Res and their new dependency versions should be a particularly welcome discovery for practitioners, since dependency schemes are yet to be implemented in expansion-based solvers, and the authors of RaREQs plan to incorporate them [31]. Our results suggest that a move in that direction would be beneficial, and could promote advances in line with the those made by DepQBF on the CDCL side [38].

Acknowledgments. We thank Meena Mahajan and Anil Shukla for interesting discussions on this work. Research was supported by EPSRC grant EP/L024233/1 and the EU Marie Curie IRSES grant CORCON.

References

1. Atserias, A., Oliva, S.: Bounded-width QBF is PSPACE-complete. Journal of Computer and System Sciences 80(7), 1415–1429 (2014)
2. Balabanov, V., Chiang, H.K., Jiang, J.R.: Henkin quantifiers and Boolean formulae: A certification perspective of DQBF. Theoretical Computer Science 523, 86–100 (2014)
3. Balabanov, V., Jiang, J.H.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012)
4. Balabanov, V., Jiang, J.R., Janota, M., Widl, M.: Efficient extraction of QBF (counter)models from long-distance resolution proofs. In: Conference on Artificial Intelligence (AAAI). pp. 3694–3701 (2015)
5. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 154–169 (2014)
6. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow - resolution made simple. Journal of the ACM 48(2), 149–169 (2001)
7. Benedetti, M., Mangassarian, H.: QBF-based formal verification: Experience and perspectives. Journal on Satisfiability, Boolean Modeling and Computation (JSAT) 5(1-4), 133–191 (2008)
8. Beyersdorff, O., Blinkhorn, J.: Dependency schemes in QBF calculi: Semantics and soundness. In: Principles and Practice of Constraint Programming (CP). pp. 96–112 (2016)
9. Beyersdorff, O., Bonacina, I., Chew, L.: Lower bounds: From circuits to QBF proof systems. In: ACM Conference on Innovations in Theoretical Computer Science (ITCS). pp. 249–260 (2016)
10. Beyersdorff, O., Chew, L., Janota, M.: On unification of QBF resolution-based calculi. In: International Symposium on Mathematical Foundations of Computer Science (MFCS). pp. 81–93 (2014)
11. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. In: International Symposium on Theoretical Aspects of Computer Science (STACS). Leibniz International Proceedings in Informatics (LIPIcs), vol. 30, pp. 76–89 (2015)
12. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. Electronic Colloquium on Computational Complexity 21, 120 (2014)
13. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Feasible interpolation for QBF resolution calculi. In: International Colloquium on Automata, Languages, and Programming (ICALP). pp. 180–192 (2015)
14. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Are short proofs narrow? QBF resolution is not simple. In: Symposium on Theoretical Aspects of Computer Science (STACS). pp. 15:1–15:14 (2016)
15. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Understanding cutting planes for QBFs. In: Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS). pp. 40:1–40:15 (2016)
16. Beyersdorff, O., Chew, L., Schmidt, R.A., Suda, M.: Lifting QBF resolution calculi to DQBF. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 490–499 (2016)
17. Beyersdorff, O., Chew, L., Sreenivasaiah, K.: A game characterisation of tree-like Q-resolution size. Journal of Computer and System Sciences (2017), in press.
18. Beyersdorff, O., Galesi, N., Lauria, M.: A characterization of tree-like resolution size. Information Processing Letters 113(18), 666–671 (2013)
19. Beyersdorff, O., Pich, J.: Understanding Gentzen and Frege systems for QBF. In: ACM/IEEE Symposium on Logic in Computer Science (LICS). pp. 146–155 (2016)
20. Buss, S.R.: Towards NP-P via proof complexity and search. Annals of Pure and Applied Logic 163(7), 906–917 (2012)

21. Chen, H.: Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In: International Colloquium on Automata, Languages, and Programming (ICALP). pp. 94:1–94:14 (2016)
22. Cook, S.A., Morioka, T.: Quantified propositional calculus and a second-order theory for NC^1. Archive for Mathematical Logic 44(6), 711–749 (2005)
23. Cook, S.A., Nguyen, P.: Logical Foundations of Proof Complexity. Cambridge University Press (2010)
24. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. The Journal of Symbolic Logic 44(1), 36–50 (1979)
25. Egly, U.: On sequent systems and resolution for QBFs. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 100–113 (2012)
26. Egly, U., Kronegger, M., Lonsing, F., Pfandler, A.: Conformant planning as a case study of incremental QBF solving. In: International Conference on Artificial Intelligence and Symbolic Computation (AISC). pp. 120–131 (2014)
27. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR). pp. 291–308 (2013)
28. Gelder, A.V.: Contributions to the theory of practical quantified boolean formula solving. In: International Conference on Principles and Practice of Constraint Programming (CP). pp. 647–663 (2012)
29. Giunchiglia, E., Marin, P., Narizzano, M.: Reasoning with quantified boolean formulas. In: Handbook of Satisfiability, pp. 761–780. IOS Press (2009)
30. Heule, M.J.H., Seidl, M., Biere, A.: Solution validation and extraction for QBF preprocessing. Journal of Automated Reasoning 58(1), 97–125 (2017)
31. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. Journal of Artificial Intelligence 234, 1–25 (2016)
32. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. Theoretical Computer Science 577, 25–42 (2015)
33. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified boolean formulas. Information and Computation 117(1), 12–18 (1995)
34. Krajíček, J.: Bounded Arithmetic, Propositional Logic, and Complexity Theory, Encyclopedia of Mathematics and Its Applications, vol. 60. Cambridge University Press, Cambridge (1995)
35. 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)
36. Krajíček, J., Pudlák, P.: Quantified propositional calculi and fragments of bounded arithmetic. Zeitschrift für mathematische Logik und Grundlagen der Mathematik 36, 29–46 (1990)
37. Liang, J.H., Ganesh, V., Zulkoski, E., Zaman, A., Czarnecki, K.: Understanding VSIDS branching heuristics in conflict-driven clause-learning SAT solvers. In: Haifa Verification Conference (HVC). pp. 225–241 (2015)
38. Lonsing, F.: Dependency Schemes and Search-Based QBF Solving: Theory and Practice. Ph.D. thesis, Johannes Kepler University (2012)
39. Lonsing, F., Egly, U., Seidl, M.: Q-resolution with generalized axioms. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 435–452 (2016)
40. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Design Automation Conference (DAC). pp. 530–535 (2001)
41. Peitl, T., Slivovsky, F., Szeider, S.: Long distance Q-resolution with dependency schemes. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 500–518 (2016)
42. Pudlák, P., Impagliazzo, R.: A lower bound for DLL algorithms for SAT. In: Proc. 11th Symposium on Discrete Algorithms. pp. 128–136 (2000)
43. Rintanen, J.: Asymptotically optimal encodings of conformant planning in QBF. In: AAAI. pp. 1045–1050. AAAI Press (2007)
44. Samer, M., Szeider, S.: Backdoor sets of quantified Boolean formulas. Journal of Automated Reasoning 42(1), 77–97 (2009)
45. Segerlind, N.: The complexity of propositional proofs. Bulletin of Symbolic Logic 13(4), 417–481 (2007)
46. Shacham, O., Zarpas, E.: Tuning the VSIDS decision heuristic for bounded model checking. In: International Workshop on Microprocessor Test and Verification (MTV). p. 75 (2003)
47. Silva, J.P.M.: The impact of branching heuristics in propositional satisfiability algorithms. In: Portugese Conference on Progress in Artificial Intelligence (EPIA). pp. 62–74 (1999)
48. Slivovsky, F., Szeider, S.: Computing resolution-path dependencies in linear time. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 58–71 (2012)
49. Slivovsky, F., Szeider, S.: Variable dependencies and Q-resolution. In: International Conference on Theory and Applications of Satisfiability Testing (SAT). pp. 269–284 (2014)
50. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. Theoretical Computer Science 612, 83–101 (2016)
51. Vollmer, H.: Introduction to Circuit Complexity – A Uniform Approach. Texts in Theoretical Computer Science, Springer Verlag, Berlin Heidelberg (1999)

Appendix

The appendix contains supplementary material, mainly already contained elsewhere in the literature, but restated here for convenience and coherence of presentation.

A. Modified Q-parity formulas

In Section 2, we explained that the following two conditions must be fulfilled in order for our lower bound technique to be applicable to a formula family $\Phi(n)$:

1. $\Phi(n)$ requires large witnessing functions, and
2. for sufficiently many restrictors δ, the formula $\Phi(n)|_\delta$ contains many critical variables.

We show that there exists a family of PCNFs $\Phi(n)$ fulfilling (2) and failing (1) that admit short IR-calc refutations. We obtain such formulas by taking the Q-parity formulas of [11] and replacing the single universal variable u with a block of n universals $u_1, \ldots, u_n$.

For variables z_1, z_2 and z_3, let $\mathrm{xor}(z_1, z_2, z_3)$ denote the set containing the four clauses $\{z_1, z_2, \neg z_3\}$,$\{z_1, \neg z_2, z_3\}$,$\{\neg z_1, z_2, z_3\}$ and $\{\neg z_1, \neg z_2, \neg z_3\}$ (these clauses are satisfied if and only if z_3 is equal to $z_1 \oplus z_2$). The modified formulas $\Phi(n) = \mathcal{Q}(n) \,.\, \phi(n)$ have quantifier prefixes $\mathcal{Q}(n) = \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_2 \cdots t_n$ and matrices

$$\phi(n) = \{\{u_1, \ldots, u_n, t_n\}, \{\neg u_1, \ldots, \neg u_n, \neg t_n\}\} \cup \mathrm{xor}(x_1, x_2, t_2) \cup \bigcup_{i=3}^{n} \mathrm{xor}(t_{i-1}, x_i, t_i).$$

Similarly as in [11], the only strategy for the $\forall$-player is to set each u_i equal to the parity of the x_i variables; that is, she must play $u_i \equiv x_1 \oplus \cdots \oplus x_n$ for each $i \in [n]$. As a result, the range of the unique witnessing function for $\Phi(n)$ has size exactly 2, hence failing condition (1).

It is simple to verify that condition (2) is fulfilled. Observe that $\Phi(n)$ has 2^n restrictors, each of which is a total assignment to $\{x_1, \ldots, x_n\}$. We show that, for any restrictor δ, all n universal variables are critical in $\Phi|_\delta$. Note that $\phi|_\delta$ consists of the clauses

$$\{\{l_2\}, \{l_2, \neg l_3\}, \{\neg l_2, l_3\}, \ldots, \{l_{n-1}, \neg l_n\}, \{\neg l_{n-1}, l_n\}, \{u_1, \ldots, u_n, t_n\}, \{\neg u_1, \ldots, \neg u_n, \neg t_n\}\}$$

for some literals $l_2, \ldots, l_n$, with $\mathrm{var}(l_i) = t_i$ for $i \in \{2, \ldots, n\}$. Removing both clauses $\{u_1, \ldots, u_n, t_n\}$ and $\{\neg u_1, \ldots, \neg u_n, \neg t_n\}$ from $\phi|_\delta$ yields a satisfiable set ($\{l_2, \ldots, l_n\}$ is a satisfying assignment). Hence each u_i is critical in $\Phi|_\delta$, by Definition 10.

Syntactically, the only difference from the original formulas is that the single occurrence of the negative universal literal $\neg u$ is replaced by the n negative literals $\neg u_1, \ldots, \neg u_n$, and similarly for the positive literal u. It is easy to see that duplicating universal literals in this way cannot affect the complexity of refutations in IR-calc or Q-Res. Therefore, the fact that $\Phi(n)$ is hard for Q-Res and easy for $\forall$Exp+Res (and therefore also for IR-calc) follows from the corresponding results for Q-parity in [11].

B. The base expansion calculus $\forall$Exp+Res

The calculus $\forall$Exp+Res [32] is the base theoretical model for expansion-based QBF solving. The central observation is that a universal variable need not necessarily be expanded in both polarities to produce a false propositional formula. As a result, careful selection of assignments can considerably reduce the size increase due to expansion.

$\forall$Exp+Res is weaker than IR-calc for the following reason: the download assignments are total assignments to the universals. This means that, throughout the course of a proof, any given literal has for its annotation a fixed, total assignment to the universals to its left. (As in IR-calc, no literal is ever annotated with an assignment to a universal to its right.)

An ∀Exp+Res derivation, therefore, is clearly separated into two distinct phases: a download phase in which matrix clauses are introduced, and a resolution phase, which is nothing more than propositional resolution on duplicate variables.

Given a PCNF $\Phi = \mathcal{Q}\,.\,\phi$, the act of downloading clause $C \in \phi$ with total assignment τ to the universal variables should be viewed as follows: It is the introduction of the axiom $C|_\tau$, which is the corresponding clause from the conjunct $\phi|_\tau$ in the complete universal expansion of Φ. The annotations in $C|_\tau$ are simply recording the duplicate variables introduced by the expansion. Hence, ∀Exp+Res permits one to pick and choose single clauses from the complete expansion. This explains the strength of ∀Exp+Res over naive complete expansion followed by propositional resolution, and thereby models the principal feature of expansion solving.

The rules of ∀Exp+Res are given in Fig. 8

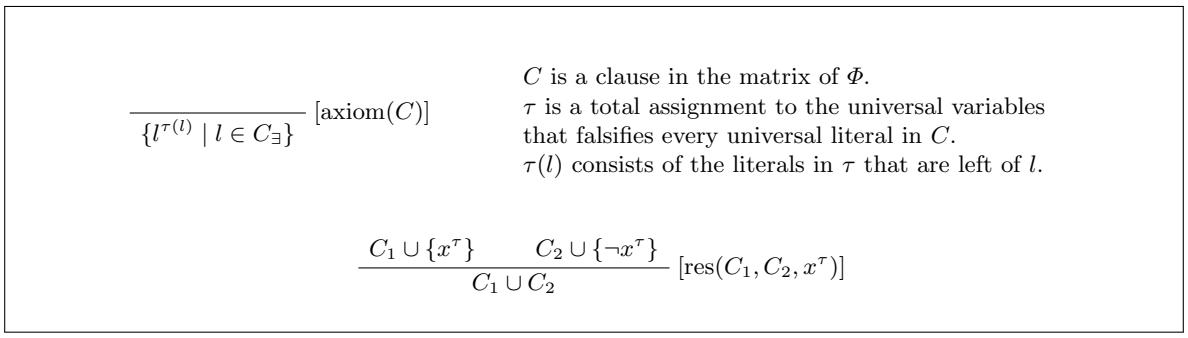

Fig. 8. The rules of ∀Exp+Res [32]. Note that $\Phi = \mathcal{Q}\,.\,\phi$ is a PCNF.

C. Proofs of propositions from Section 4

We provide proofs of Propositions 1, 2 and 3. Propositions 1 and 3 have existing proofs in the literature, namely those of [10, Thm. 4] and [10, Lem. 2 and 3] respectively. The results in that paper apply to stronger QBF calculi; we provide tight proofs consistent with the notation of this paper.

Proposition 1 ([10]). *Let π be an* IR-calc *refutation of a PCNF Φ and let l be a literal with* $\mathrm{var}(l) \in \mathrm{vars}(\Phi)$. *Then $\pi|_l$ is an* IR-calc *refutation of $\Phi|_l$ if (a) l is existential or (b) l is universal and unopposed in π.*

Proof. Let $\Phi = \mathcal{Q}\,.\,\phi$.

For case (a), let l be existential. We first give a formal definition of $\pi|_l$. Let $\pi = C_1, \ldots, C_m = \emptyset$ and let $\pi' = C'_1, \ldots, C'_m$, where

$$C'_i = \begin{cases} C_\top, & \text{if } l^\tau \in C_i \text{ for some annotation } \tau, \\ C_i \setminus L, & \text{otherwise,} \end{cases}$$

where $L = \{\neg l^\tau \mid \tau \text{ is an annotation}\}$. Noting that C'_m is the empty clause, let $e = \min\{i \in [m] \mid C'_e = \emptyset\}$. Then the restricted derivation $\pi|_l$ is the subderivation of C'_e.

We prove by induction on the depth of the subderivation of C_i that, if $C'_i \neq C_\top$, then the subderivation of C'_i is a valid IR-calc derivation from $\Phi|_l$. We hence prove the lemma, since the subderivation of C'_e is therefore a refutation of $\Phi|_l$.

For the base case, let C_i be an axiom downloaded from $D \in \phi$. If $C'_i \neq C_\top$, then $l \notin D$, and it follows that C'_i is the axiom downloaded from $D|_l \in \phi|_l$. The inductive step follows

trivially if C_i was derived by instantiation. Hence, suppose that $C_i = \text{res}(C_j, C_k, x^\sigma)$ and that $C'_i \neq C_\top$, so that $\neg l^\tau \notin C'_i$ for all annotations τ. There are two cases.

(1) Suppose that $\text{var}(l) \neq x$. Then $l^\tau \notin C_j \cup C_k$ for all annotations τ, and hence neither C'_j nor C'_k is $C_\top$. Moreover, we have $x^\sigma \in C'_j$ and $\neg x^\sigma \in C'_k$ so that $C'_i = C_i \setminus L = \text{res}(C_j, C_k, x^\sigma) \setminus L = \text{res}(C'_j, C'_k, x^\sigma)$ is a valid IR-calc inference.

(2) On the other hand, suppose w.l.o.g. that $l = x$. Then $l^\sigma \in C_j$, so $C'_j = C_\top$ and $C'_k \neq C_\top$. Moreover, $\neg l^\sigma \notin C'_k$, and it follows that $C'_i = (C'_j \setminus \{l^\sigma\}) \cup C'_k$. That is, C'_i is just a weakening[7] of C'_k.

For case (b), let l be universal and unopposed in π, and for each axiom C_i let $D_i \in \phi$ be the matrix clause from which C_i was downloaded. Observe that $l \notin D_i$ (for otherwise $\neg l$ would appear in the annotations of C_i), and therefore that $D_i|_l = D_i \setminus \{\neg l\}$ for each downloaded matrix clause D_i. Moreover, $\Phi|_l$ is a false PCNF, and $\pi|_l$ is obtained from π simply by removing the literal l from all annotations. It follows immediately that every step in the restricted proof is valid, and that the conclusion is unique. □

Proposition 2. *Let π be an* IR-calc *derivation from a PCNF Φ, and let ρ and ρ' be partial assignments to the existential variables of Φ. If $\rho \subseteq \rho'$, then every annotation of $\pi|_{\rho'}$ is an annotation of $\pi|_\rho$.*

Proof. Restricting π by an existential literal l may remove clauses from π and may remove literals from the clauses that remain. However, the annotations themselves are never changed. Therefore any annotation that appears in the restricted refutation $\pi|_l$ must appear in π. □

Proposition 3 ([10]). *If π is an* IR-calc *refutation of a PCNF Φ, then the strategy extracted from π is a winning $\forall$-strategy for Φ.*

Proof. Let $\Phi = \mathcal{Q}\,.\,\phi$, where $\mathcal{Q} = \exists X_1 \forall U_1 \cdots \exists X_n \forall U_n \exists X_{n+1}$, and let $V = \text{vars}(\Phi)$.

Let α_1 be an arbitrary total assignment to X_1. Then $\pi|_{\alpha_1}$ is a refutation of $\Phi|_{\alpha_1}$, by Proposition 1. Since U_1 is the first block of $\Phi|_{\alpha_1}$, every U_1 literal in $\pi|_{\alpha_1}$ is unopposed, by Lemma 9, so the U_1 literals of $\pi|_{\alpha_1}$ form a partial assignment β'_1. The total assignment β_1 to U_1 extending β'_1 with zero assignments must falsify $\Phi|_{\alpha_1}$, since the annotations in $\pi|_{\alpha_1}$ contain the negation of every U_1 literal in the download clauses. Thus $\Phi|_{\alpha_1 \cup \beta_1}$ is false.

After n iterations of this argument, we observe that ϕ is unsatisfiable under the restriction $\bigcup_{i=1}^n (\alpha_i \cup \beta_i)$. Ranging over all suitable assignments $\alpha_1 \cup \cdots \cup \alpha_{n+1}$, we obtain a function S from total assignments to $V_\exists$ into total assignments to $V_\forall$. Moreover, by construction, if α and α' agree on the first i existential blocks, then $S(\alpha)$ and $S(\alpha')$ agree on the first i universal blocks. Hence S is a $\forall$-strategy for Φ in which every game falsifies ϕ.

In fact, the equivalent statement, that every U_1 literal appearing in an annotation of $\pi|_\alpha$ is unopposed, follows from Lemma 9. □

D. QBF dependency calculi

The rules of Q($\mathcal{D}$)-Res and QU($\mathcal{D}$)-Res are presented in Fig. 9. The rules of $\forall$Exp($\mathcal{D}$)+Res are presented in Fig. 10.

Finally, we provide a proof sketch for the soundness of IR($\mathcal{D}^{\text{rrs}}$)-calc. The proof refers to a class of formulas called *dependency quantified Boolean formulas* (DQBF) in which the allowable dependencies between variables are given explicitly – see [2] for an introduction.

Proposition 34. IR($\mathcal{D}^{\text{rrs}}$)-calc *is a sound proof system.*

[7] We can add weakening as a valid inference rule to IR-calc without affecting the complexity of proofs.

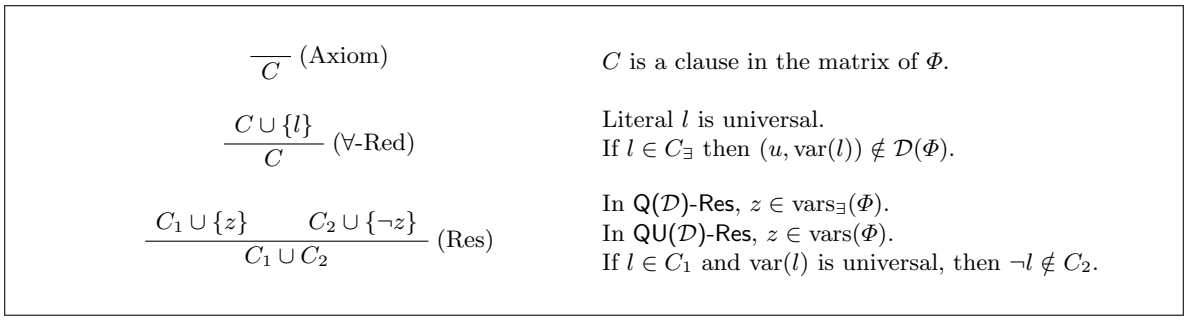

Fig. 9. The rules of Q($\mathcal{D}$)-Res [50] and QU($\mathcal{D}$)-Res [8]. Note that $\Phi = \mathcal{Q} \,.\, \phi$ is a PCNF.

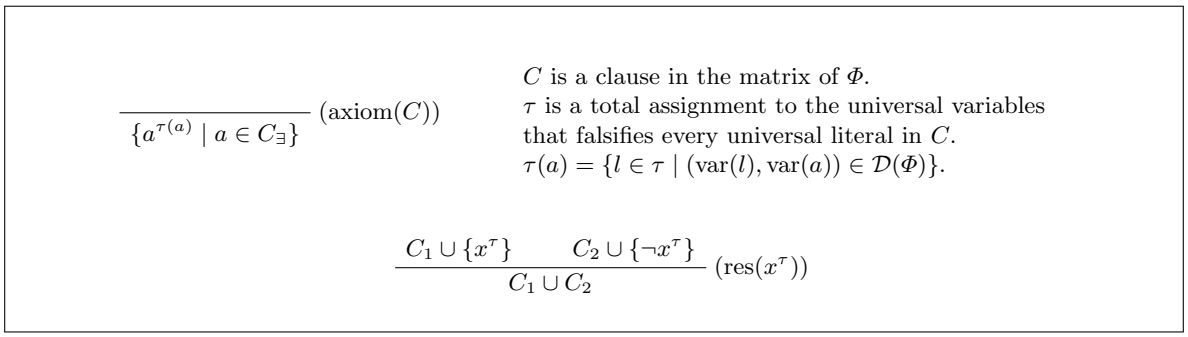

Fig. 10. The rules of ∀Exp($\mathcal{D}$)+Res. Note that $\Phi = \mathcal{Q} \,.\, \phi$ is a PCNF.

Proof (sketch). In [16] it was shown that D-IR-calc, the DQBF version of IR-calc, is sound, and in [8] it was shown that $\mathcal{D}^{\mathrm{rrs}}$ is fully exhibited. The proposition is proved from these two results as follows.

A dependency scheme $\mathcal{D}$ can be interpreted naturally as a mapping $f_\mathcal{D}$ from QBF into DQBF, in which the dependencies for $\Phi \in$ PCNF are given explicitly in the quantifier prefix of $f_\mathcal{D}(\Phi)$. Under this interpretation, we make two observations:

(1) A refutation of Φ in IR($\mathcal{D}$)-calc is also a refutation of $f_\mathcal{D}(\Phi)$ in D-IR-calc.
(2) For a fully exhibited scheme $\mathcal{D}$, Φ is true implies $f_\mathcal{D}(\Phi)$ is true.

Hence, if there existed an IR($\mathcal{D}^{\mathrm{rrs}}$)-calc refutation of a true PCNF Φ we would have a D-IR-calc refutation of a true DQBF $f_\mathcal{D}(\Phi)$, contradicting the soundness of that calculus. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il