Electronic Colloquium on Computational Complexity, Report No. 24 (2018)

The Riis Complexity Gap for QBF Resolution

Olaf Beyersdorff[1a], Judith Clymo[a], Stefan Dantchev[b], Barnaby Martin[b]

[a]School of Computing, University of Leeds, U.K.
[b]Department of Computer Science, University of Durham, U.K.

Abstract

We give an analogue of the Riis Complexity Gap Theorem for Quantified Boolean Formulas (QBFs). Every first-order sentence ϕ without finite models gives rise to a sequence of QBFs whose minimal refutations in tree-like Q-Resolution are either of polynomial size (if ϕ has no models) or at least exponential in size (if ϕ has some infinite model). However, differently from the translations to propositional logic, the translation to QBF must be given additional structure in order for the polynomial upper bound to hold in tree-like Q-Resolution. This extra structure is not needed in the system tree-like $\forall$Exp+Res, where we see the complexity gap on a natural translation to QBF.

Keywords: Complexity Gap, Proof Complexity, Quantified Boolean Formulas

1. Introduction

There is a standard way in which to translate a first-order sentence ϕ to a sequence of propositional formulas so that the nth member of the sequence is satisfiable if and only if ϕ has a model of size n [18], which additionally ensures that the nth member of the sequence has size at most polynomial in n. Suppose ϕ is a first-order sentence without finite models. The celebrated Complexity Gap Theorem of Riis [18] states that the minimal sized refutations in tree-like Resolution, of the nth member of the sequence, has growth rate either bounded above by some polynomial, or bounded below by some exponential. Further, the former case prevails precisely when ϕ has no

[1]Corresponding author. Address: School of Computing, University of Leeds, LS2 9JT, Leeds, UK. Email: `o.beyersdorff@leeds.ac.uk`

Preprint submitted to Elsevier *February 1, 2018*

ISSN 1433-8092

infinite models either. This theorem was subsequently lifted to various settings including Parameterized tree-like Resolution [11] and the Integer Linear Programming systems of Lovász-Schrijver and Sherali-Adams [12].

Quantified Boolean logic is an extension of propositional logic in which variables may be existentially or universally quantified. Therefore, the problem of determining the truth value of a quantified Boolean formula (QBF) naturally extends the satisfiability problem (SAT) on propositional formulas, and the success of SAT solving algorithms has laid the foundation for modern QBF solvers. Motivated by these exciting practical developments, a growing body of research has examined the proof complexity of QBF, including different versions of QBF Resolution [16, 15, 4, 19]. It is particularly interesting to understand which propositional ideas and techniques lift to the more complex QBF setting. In this respect, recent research has shown interesting effects, with major propositional approaches such as size-width [1] failing in QBF [6] and new genuine QBF techniques being developed [5, 3, 2].

In this article, we investigate whether the Riis Gap Theorem [18] extends to QBF resolution systems. We first introduce a method to translate a first-order sentence ϕ to a sequence of QBFs, which echoes similar translations of quantified constraint satisfaction problems (QCSPs) to QBFs that have appeared in [13, 14]. The translation will ensure that the nth member of the sequence has size at most polynomial in n, and is true precisely when ϕ has a model of size n. In Riis's Theorem, Resolution may be considered as a refutation system operating on CNF formulas whose literals are ground atoms. To allow operation on QBFs, the natural extension is the Q-Resolution of [16].

We demonstrate that tree-like Q-Resolution will always require exponential size to refute the nth member of the sequence of QBFs when ϕ has an infinite model but no finite model. However, unlike in previously considered contexts, it is not the case that if ϕ has no models, then there exists tree-like Q-Resolution refutations, of the nth member of the sequence, with size polynomial in n. We provide a counter example that demonstrates an anomaly of Q-Resolution. To achieve the polynomial upper bound we embellish some additional structure to the formula ϕ (without changing its models), to obtain a formula ϕ^*, before applying our translation to generate a new sequence of QBFs. The extra structure enables Q-Resolution to more easily refute the resultant QBFs precisely when ϕ has no models. Our main result is:

Theorem 1. *Let ϕ be a first-order sentence without finite models, ϕ^* its* *embellishment and $\langle \Phi_i^* \rangle_{i\in\mathbb{N}}$ the corresponding sequence of QBFs. If ϕ has no models, then there exist tree-like Q-Resolution refutations of $\langle \Phi_i^* \rangle_{i\in\mathbb{N}}$ of size $O(i^k)$, where k depends only on ϕ. If ϕ has some (infinite) model, then all tree-like Q-Resolution refutations of $\langle \Phi_i^* \rangle_{i\in\mathbb{N}}$ must have size $\Omega(2^{\epsilon i})$, where ϵ depends only on ϕ.*

Thus we obtain, à la Riis, a gap between polynomial and exponential in which certain growth behaviours (e.g. subexponential $2^{\sqrt{i}}$) are forbidden.

We prove that the same phenomenon holds in the system of tree-like QU-Resolution, whereas in the system of tree-like ∀Exp+Res from [15], the gap holds naturally, that is without the embellishment. In this sense, ∀Exp+Res does not possess the same deficiency as Q-Resolution.

On the technical side, our gap theorems exploit a Prover-Delayer game to show hardness in QBF resolution. While such a game already exists for tree-like Q-Resolution [7], we modify the game to obtain a full characterisation for the tree-like versions of the expansion-based QBF resolution systems ∀Exp+Res [15] and IR-calc [4].

2. Preliminaries

We restrict attention to QBFs in closed prenex conjunctive normal form, $\Psi = \mathcal{Q}\ \psi$, where ψ is a propositional formula (in CNF). The prefix $\mathcal{Q}$ takes the form $Q_1 x_1 Q_2 x_2 \dots Q_k x_k \psi$ where $Q_i \in \{\forall, \exists\}$, x_i are distinct Boolean variables. In closed formulas, all the variables in ψ must appear in $\mathcal{Q}$. The prefix also enforces a partial order on the variables. If $Q_i = Q_{i+1}$ we say x_i and x_{i+1} are in the same quantifier level in the prefix. Variables in the same level may be reordered arbitrarily to create another logically equivalent QBF, but otherwise changing the order that variables appear in the prefix may not preserve the truth value of Ψ.

Q-Resolution consists of a resolution rule and universal reduction. The resolution rule is

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D}$$

where C and D are clauses and x is an existentially quantified variable, and for all variables $y \neq x$ that appear in C, the negation of y does not appear in D. We call x the *pivot* of this resolution step.

The universal reduction rule is

$$\frac{C \vee x}{C}$$

where x is universally quantified and belongs to the inner-most quantifier level of all variables appearing in C.

A QBF is false if and only if it is possible to derive the empty clause by application of these rules. A Q-Resolution refutation of Ψ is a sequence of clauses $C_1 \dots C_n$ such that every C_i is either a clause from ψ, derived by resolution from C_j and C_k $(j, k < i)$ or derived by $\forall$-reduction from C_j $(j < i)$. A Q-Resolution proof has an underlying DAG structure, with edges denoting inference either by resolution or reduction. In a tree-like Q-Resolution proof this graph must be a tree. Each derived clause can therefore only be used once in the proof.

QU-Resolution [19] is similar to Q-Resolution except that the pivot of a resolution step is also permitted to be universally quantified.

Finally, $\forall$Exp+Res [15] describes an alternative approach to QBF solving in which existentially quantified variables are expanded according to different possible Boolean assignments to the universal variables. This produces an entirely existential formula that can be refuted by propositional Resolution. When an axiom is downloaded into a $\forall$Exp+Res proof, some complete assignment μ to the universal variables is implicitly being considered. For C a clause in ψ, the assignment will be one which does not automatically satisfy the clause (i.e. if universal literal u appears in C then μ will set $u = 0$). The universal literals in C are falsified by the assignment and so are removed, and each existential variable x in C is annotated with μ, to show which part of the expanded formula it relates to. Because x can only depend on universal variables that appear in an earlier level than x in the quantifier prefix, μ is truncated for each existential literal in C to only reference the part of the assignment that is relevant for this literal.

If μ and ω are distinct assignments to universal variables appearing before x in the prefix, then x^μ and x^ω are distinct, existentially quantified variables. Every clause in the refutation is either introduced in this way, or is the result of a propositional resolution step between some x^μ and $\neg x^\mu$.

3. Rendering a first-order sentence as a sequence of QBFs

We now give a method to translate a first-order sentence ϕ to a sequence $\langle \Phi_i \rangle_{i \in \mathbb{N}}$ of QBFs. The method is inspired by the encoding of ϕ into propositional formulas in conjunctive normal form (CNF) previously given by Riis

[18], but has rather more in common with other translations used to encode QCSP instances as QBF in [13]. A more succinct "binary" or "logarithmic" form of encoding is discussed in [14]. For our purposes, since ϕ is fixed, the benefit of this more succinct encoding is not important.

We will consider a first-order sentence

$$\phi := Q_1 x_1 \ldots\ldots Q_k x_k \; \mathcal{D}_1(x_1 \ldots, x_k) \wedge \ldots \wedge \mathcal{D}_r(x_1, \ldots, x_k)$$

with $Q_i \in \{\forall, \exists\}$, and where each $\mathcal{D}_i$ is a disjunction of the form

$$R_i^1(x_1, \ldots, x_k) \vee \ldots \vee R_i^s(x_1, \ldots, x_k).$$

Note that we do not lose significant generality by assuming all extensional relations to be of arity k and all disjunctions to be of width s. We can refer to the set of existentially quantified variables by $\{x_i \mid Q_i = \exists\}$, or to the relevant indices by $\{i \mid Q_i = \exists\}$.

We will take each first-order variable of the form x and create n propositional variables $x^1, \ldots, x^n$, where we will ask that precisely one of these is true, say x^i, and this indicates that x is evaluated as the ith element in a model of size n.

Let $[n] := \{1, \ldots, n\}$. $\sum_{i \in [n]} x^i = 1$ asserts that precisely one of the x^i is true, i.e. it is an abbreviation for $\left(\bigvee_{i \in [n]} x^i\right) \wedge \bigwedge_{i \neq j \in [n]} (\neg x^i \vee \neg x^j)$. Similarly $\neg(\sum_{i=1}^n x^i = 1)$ is shorthand for the conjuction of clauses $\left(\neg x^i \vee \bigvee_{j \neq i} x^j\right)$, ensuring that if any two of the x^i are true then the conjunction is satisfied, and if all are false then it is satisfied. In the original sentence a variable x can only take on one value at a time, and must be given some value. These conditions ensure the same restriction remains in the QBF. If any existential variable x is not given exactly one value, the QBF is falsified, and if any universal variable is not given exactly one value then the QBF evaluates to true.

Further, we will have propositional variables associated with each instantiation of a relational predicate $R_i^j(\lambda_1, \ldots, \lambda_k)$, where $\lambda_1, \ldots, \lambda_k \in [n]$, indicating that the tuple $(\lambda_1, \ldots, \lambda_k)$ is in the relation R_i^j. Thus, we have two types of propositional variable, one associated with elements and one associated with relations. $\exists_{\lambda_1, \ldots, \lambda_k \in [n]} R_i^j(\lambda_1, \ldots, \lambda_k)$ introduces the n^k possible relational Boolean variables for the relation R_i^j and shows that they are existentially quantified.

We now build a sequence of QBFs whose nth member ϕ_n begins

$$\exists_{\lambda_1,\ldots,\lambda_k \in [n]} R_1^1(\lambda_1,\ldots,\lambda_k) \ldots R_r^s(\lambda_1,\ldots,\lambda_k)$$

where each $R_i^j(\lambda_1,\ldots,\lambda_k)$ is a propositional variable, and continues

$$\begin{array}{r} Q_1\ x_1^1 \ldots x_1^n \ldots\ldots Q_k\ x_k^1 \ldots x_k^n \\ \bigwedge_{\{i\ \mid\ Q_i=\exists\}} \left(\sum_{j\in[n]} x_i^j = 1\right) \wedge \\ \Big[\bigwedge_{\{i\ \mid\ Q_i=\forall\}} \left(\sum_{j\in[n]} x_i^j = 1\right) \rightarrow \\ \left(\bigwedge_{i\in[r]\lambda_1,\ldots,\lambda_k\in[n]} (x_1^{\lambda_1} \wedge \ldots \wedge x_k^{\lambda_k}) \rightarrow \mathcal{D}_i(\lambda_1,\ldots,\lambda_k)\right) \Big] \end{array}$$

It is clear by construction that ϕ_n is true just in case ϕ has a model of size n. We will always imagine that the quantifier-free part of ϕ_n is expanded to CNF and so it is important to note that this expansion is not of size larger than polynomial in n. If the disjuncts $\mathcal{D}_i$ contain equality relationships between variables then these can be enforced by restriction of the $\lambda_1,\ldots,\lambda_k \in [n]$; indeed, if the disjuncts only involve some subset of $x_1,\ldots,x_k$ then plainly only those need be mentioned. We call Boolean variables of the form $R_i^j(\lambda_1,\ldots,\lambda_k)$, always existentially quantified outermost, *relational variables*.

Example. Recall the Pigeonhole principle (ϕ^{PHP}), which comes from the first-order sentence

$$\forall x, y, z\ \exists w\ P(x,w) \wedge \neg P(x,1) \wedge (\neg P(x,z) \vee \neg P(y,z) \vee x \neq y).$$

This has no finite models and a typical rendering of it, asserting that there is no model on a domain $[n]$, in CNF, would have the following form

$$\begin{array}{ll} \bigvee_{j\in[n]} P(i,j) & i \in [n] \\ \neg P(i,1) & i \in [n] \\ \neg P(i,k) \vee \neg P(j,k) & i \neq j \in [n], k \in [n] \end{array}$$

For a translation to QBF we proceed as follows.

$$\exists_{i,j\in[n]} P(i,j)\ \forall_{i\in[n]} x^i, y^i, z^i\ \exists_{i\in[n]} w^i$$

with quantifier free part of the form

$$(\sum_{i\in[n]} w^i = 1) \wedge \Big((\sum_{i\in[n]} x^i = 1 \wedge \sum_{i\in[n]} y^i = 1 \wedge \sum_{i\in[n]} z^i = 1) \rightarrow$$

followed by the conjunction of

$$
\begin{array}{lr}
x^i \wedge w^\ell \to P(i,\ell) & i,\ell \in [n] \\
x^i \to \neg P(i,1) & i \in [n] \\
x^i \wedge y^j \wedge z^k \to [\neg P(i,k) \vee \neg P(j,k)] & i \neq j, k \in [n] \\
)
\end{array}
$$

Then our QBF can be written explicitly in prenex conjunctive normal form

$$
\begin{aligned}
&\exists_{i,j\in[n]} P(i,j)\ \forall_{i\in[n]} x^i, y^i, z^i\ \exists_{i\in[n]} w^i \\
&\bigwedge_{i\neq j\in[n]} (\neg w^i \vee \neg w^j) \wedge (w^1 \vee \ldots \vee w^n) \wedge \\
&\bigwedge_{i,j,k,l\in[n]} \Big(\neg x^i \vee \bigvee_{i'\neq i} x^{i'} \vee \neg y^j \vee \bigvee_{j'\neq j} y^{j'} \vee \neg z^k \vee \bigvee_{k'\neq k} z^{k'} \vee \neg w^l \vee \neg P(i,l)\Big) \wedge \\
&\bigwedge_{i,j,k\in[n]} \Big(\neg x^i \vee \bigvee_{i'\neq i} x^{i'} \vee \neg y^j \vee \bigvee_{j'\neq j} y^{j'} \vee \neg z^k \vee \bigvee_{k'\neq k} z^{k'} \vee \neg P(i,1)\Big) \wedge \\
&\bigwedge_{i\neq j,k\in[n]} \Big(\neg x^i \vee \bigvee_{i'\neq i} x^{i'} \vee \neg y^j \vee \bigvee_{j'\neq j} y^{j'} \vee \neg z^k \vee \bigvee_{k'\neq k} z^{k'} \vee \neg P(i,k) \vee \neg P(j,k)\Big)
\end{aligned}
$$

4. The lower bound

There is a rich history of game-theoretic methods in Proof Complexity and these are especially applicable in the tree-like versions of refutation systems. For tree-like Resolution, such games have been known since [17] and the argument for the lower bound in [18] is itself game-theoretic. Subsequent work uncovered a full game characterisation for tree-like Resolution via an *asymmetric* game between Prover and Delayer [8, 9]. The previous games had been symmetric and symmetric games will suffice for our lower bounds, though we cannot expect them to be tight.

We invoke the game of [7], tailored for tree-like Q-Resolution, which we now recall. The game proceeds between a Prover and Delayer, who build a partial assignment to the variables of a QBF Φ. While the Prover tries to falsify the matrix of Φ, the Delayer aims to play consistently as long as possible and score points during the course of the game. The game starts with the empty assignment. Each round of the game has the following phases:

1. Setting universal variables: The Prover can assign values to any number of universal variables that satisfy the following condition: A universal variable u can be assigned a value if every existential variable with a higher quantification level than u is currently unassigned.

2. Declare Phase: The Delayer can choose to assign values to any number of unassigned existential variables of his choice. The Delayer does not score any points for this.
3. Query Phase: This phase has three stages, similar to the original game:
   (a) The Prover queries the value of one existential variable x that is currently unassigned.
   (b) The Delayer replies with weights $p_0 \geq 0$ and $p_1 \geq 0$ such that $p_0 + p_1 = 1$.
   (c) The Prover assigns a value for x. If she assigns $x = b$ for some $b \in \{0, 1\}$, the Delayer scores $\lg(\frac{1}{p_b})$ points. (If Prover picks a value b where $p_b = 0$, then we give the Delayer an infinite score.)
4. Forget Phase: The Prover can choose any number of assigned variables (without regard to how they are quantified) in this phase. Every variable chosen by the Prover in this phase will lose its assigned value and hence become an unassigned variable.

This game exactly characterises tree-like Q-Resolution, with free choices in the game corresponding to branching points in the tree. In particular, if there exists a strategy and some choice of weighting, such that the Delayer is guaranteed at least p points in a game on Φ, regardless of how the Prover behaves, then any tree-like Q-Resolution refutation of Φ must have size at least 2^p. We give such a strategy for the Delayer on any QBF generated through the above translation, for which the underlying first-order formula has an infinite model.

For QBF Φ_n, representing the (false) statement that the original first-order sentence ψ has a model of size n, the Delayer's strategy is stated in terms of the set of models that satisfy the original first-order sentence. Let $\mathcal{M}$ be the set of all models of ψ. The Delayer cannot win this game since Φ_n is false, but we seek to demonstrate that he can guarantee $\Omega(n)$ points, meaning that the tree-like Q-Resolution proof must have size $2^{\Omega(n)}$.

The Delayer's Strategy

At any point in the game some set of relational, existential, and universal variables have values assigned. We say that a model M *agrees* with this assignment if a) the relations *do* hold between the indicated constants in the relational variables, and b) the relations between values selected for universal and existential variables *may* hold.

For example, let $S(x, y)$ be the successor function, which is represented in Φ_n by relational variables $S(i, j)$ and in the conditions $x^i \wedge y^j \to S(i, j)$. If $S(i, j) = 1$ then all models agreeing with this assignment must have that the jth constant c_j in our universe is a successor of the ith constant c_i. If $x^i = 1$ and $y^j = 1$ all models agreeing with this assignment must not have that c_j cannot be the successor of c_i. Here, this is equivalent to requiring that the models have c_j as a successor of c_i. However, if $x^i = 1$ but y has not been assigned any value, then a model agreeing with this assignment must have some value c_j such that c_j is not forbidden from being the successor of c_i and $y^j \neq 0$. It is permitted for this c_j to be outside of the n elements referenced by the QBF. At each point in the game we consider the subset of models $\widetilde{\mathcal{M}}$ that agree with the current assignment.

The Delayer has an opportunity to declare any existential variables and should assign values wherever all $M \in \widetilde{\mathcal{M}}$ agree. For any existential variable, setting $x^i = 1$ immediately implies that $x^j = 0$ for all $j \neq i$, so these values should also be set in the declare phase.

The Prover can then query the value of any existential or relational variable. This query either asks "is the value of w equal to c_i?" or "does relationship r hold between these constants?" Since we have already assigned variables for which all models agree, we know that the models differ on the answer to this question. Set $p_0 = p_1 = \frac{1}{2}$ and let Prover decide on the assignment. Delayer scores 1 point.

Clearly no existential variable will be given more than one value at a time. If the Prover declares two values for some universal variable, i.e. $x^i = 1$ and $x^j = 1$ for $i \neq j$, treat this as if x has no value assigned. The Prover cannot win the game with this assignment, and will be forced to re-assign x at some point, so this strategy does not damage the Delayer. By ignoring the invalid assignment it is not possible for it to advantage the Prover during the game and so we can assume that each variable has only one value at any moment.

Lemma 2. *Using this strategy, the Delayer can only lose the game by violating a clause stating that, for some set of existential variables $\{w^i\}_{i=1}^n$, exactly one must be set to true.*

PROOF. Because we are following models that satisfy the original sentence, each such model must satisfy every clause of the QBF, except where the QBF makes a direct statement about the size of the model. The statements that reference the size of the model are those stating that exactly one variable from each set $\{w^i\}_{i=1}^n$ must be true (i.e. that the assignment to variable w in the original sentence must correspond to one of the n elements in the universe). For the same reason, the clause will be violated because all variables are assigned 0, never because more than one is assigned 1. There are still infinite models that agree with everything stated so far, and for which w has some value, but that value falls outside of the n elements permitted by the QBF. □

We call this set $\{w^i\}_{i=1}^n$ of existential variables the *failed witness*. As a result, at least n variables in the QBF must be assigned a value in order for the Delayer to lose the game, and in particular these variables must between them reference all n of the elements in the universe. We seek to show that the number of decisions made by the Prover, and so the number of points scored by the Delayer, is a constant proportion of the number of elements in the universe.

A constant is 'mentioned' in a free choice if it is referenced by a relational variable that is decided in a free choice, or if it is the assignment for one of the main variables when some existential is set in a free choice. This signifies that the value was relevant to the decision being made, and as such the part that it can play in the model may have been restricted. Let k be the number of variables in the first-order sentence.

Lemma 3. *At least $n-k$ of the universe's n elements appear in the questions asked by the Prover for which she is allowed to choose the assignment.*

Proof. Let the set $\{w^i\}_{i=1}^n$ be the failed witness. Say that $k' \leq k$ of the main variables have been assigned, and they are set to $c_1 \dots c_{k'}$. Relational variables may also be set, either resulting from earlier direct free choices (querying the relational variables themselves), from indirect free choices (a free choice on an existential variable that then forced the assignment of a relational variable), or from structural conditions. Consider some c_j with $j > k'$.

Suppose that c_j has not yet been mentioned in a free choice. When w^j is asked ("is w equal to c_j?"), the Delayer considers the models that satisfy everything decided so far, to see whether to allow a free choice. By construction, there is at least one *infinite* model that agrees with the choices made so far, and since w will be the failed witness we know that one of these models assigns w a value that is outside of the n elements allowed by the

QBF. Consider the model M with $w = a$, $a \notin \{c_1 \dots c_{k'}, c_j\}$. Because c_j is not mentioned in any of the variables currently assigned in the QBF, and has never been mentioned in a free choice, it is not distinguishable from a and so there is a model identical to M except that $w = c_j$. Therefore, w^j is given as a free choice.

This demonstrates that all c_j, with $j > k'$, must have been mentioned in a free choice at some point in the game. □

Lemma 4. *There are at least $(n-k)/k$ free choice nodes in the game.*

Proof. We have that any Prover strategy will result in at least $n - k$ constants being included in a free choice. Since all relations have arity bounded above by k, and at most k values can be set in the main variables, each free choice can only consider at most k constants, and the result follows. In the simple cost structure given, each such choice gives the Delayer one point. □

Therefore, at least $\Omega(n)$ points are scored by the Delayer.

Theorem 5. *For a first-order sentence ϕ, if the QBF Φ_n, representing the statement that there is a model for ϕ of size n, is unsatisfiable for all n, then any tree-like Q-resolution refutation of Φ_n has size at least $2^{\Omega(n)}$.*

5. A surprising lower bound

Let $\theta :=$

$$\forall x \exists y \forall z \exists u \forall v \exists w \; R(x, y, z) \wedge \neg R(u, v, w).$$

Clearly, θ has no models of any size. The translation of θ to QBF Θ_n, asserting that θ has a model of size n, by the mechanism given above, generates a sequence whose minimal refutations in tree-like Q-Resolution are of exponential size. We show this by giving a strategy for Delayer in the corresponding game. In general once some e.g. y^i has been answered as true, all subsequent y^j, for $j \neq i$, must be answered as false. Beyond this, we also use the following shorthands: "$x = c$" should be read as true if $x^c = 1$. Similarly, "answer $y \neq c$" should be read as allowing a free choice on y^i (if it is queried) except when $i = c$ (then answer $y^c = 0$).

Within the assumption that no existential variable is assigned two values, Delayer gives Prover a free choice except:

1. if $x = c$, or there is some d with $R(c, d, e)$ for all e, then answer $u \neq c$.

2. if $x = c$ and $\neg R(c, d, e)$, for some d, e then answer $y \neq d$.
3. if $x = c$ and $y = d$, then answer $R(c, d, e)$, for each e.
4. if $u = c$ and $v = d$ and $R(c, d, e)$ for some e, then answer $w \neq e$.
5. if $u = c$ and $v = d$ and $w = e$, then answer $\neg R(c, d, e)$.

If Delayer plays according to this strategy, a contradiction cannot be reached until, for some c, Prover has atoms

$$\neg R(c, 1, f(1)), \ldots, \neg R(c, n, f(n)),$$

for some Skolem function f. Take some path on which these atoms have been derived. Each atom $R(c, i, f(i))$ appeared in a free choice, unless it was subject to Rule 5 above. In this case it was preceded by some free choice $w = f(i)$ made after an assertion $v = i$. Therefore this strategy gives at least n free choices on each branch and so:

Proposition 6. *Any tree-like Q-Resolution refutation of Θ_n must have size at least 2^n.*

In order to demonstrate a polynomial upper bound on the size of Resolution refutations we would seek to use the refutation of the first-order formula itself as a basis, similar to the methods used in [11, 18].

Considering the tableau refutation in Figure 1 for this simple formula gives more insight into this counterexample, and the reason that the refutation of the first-order formula cannot be expanded to a tree-like Q-Resolution refutation of the translated formula.

The unification that closes the tableau suggests a strategy for the Prover, which is to query u and set x accordingly, then query y and set v accordingly, then query w and set z to match, at which point the contradiction is immediate. Unfortunately, this strategy does not respect the order of the quantifier prefix. Recall that in the game description of tree-like Q-Resolution, all existential assignments at a higher level must be forgotten in order to make a universal assignment at a lower level. Therefore it is not possible for the prover to set x matching u.

Disobeying this rule in the game corresponds to using $\forall$-reduction while existential variables with a higher quantification level remain in the clause. This is not sound. Our strategy for Delayer shows that this problem cannot be overcome with the proposed translation from the first-order formula to QBF. Instead, we will modify the translation to provide the Prover with a mechanism for 'remembering' choices that have previously been made, while still respecting the rules of the game.

$$
\begin{tikzcd}
\forall x \exists y \forall z \exists u \forall v \exists w \; R(x,y,z) \wedge \neg R(u,v,w) \arrow[d] \\
R(x_1, Y_1(x_1), z_1) \wedge \neg R(U_1(x_1,z_1), v_1, W_1(x_1,z_1,v_1)) \arrow[d] \\
R(x_2, Y_2(x_2), z_2) \wedge \neg R(U_2(x_2,z_2), v_2, W_2(x_2,z_2,v_2)) \arrow[d] \\
\neg R(U_1(x_1,z_1), v_1, W_1(x_1,z_1,v_1)) \arrow[d] \\
R(x_2, Y_2(x_2), z_2) \arrow[u, bend right, "\mathit{Unif}"']
\end{tikzcd}
$$

Figure 1: Universal variables are replaced by free variables (lower case with indices), existential variables are written as functions (upper case) over those free variables. The tableau is closed by the unification *Unif*: $x_2 \leftarrow U_1(x_1, z_1)$, $v_1 \leftarrow Y_2(x_2)$, $z_2 \leftarrow W_1(x_1, z_1, v_1)$.

6. Embellishing the QBFs

Continuing with the same example, expand the formula by introducing a side condition

$$
\begin{aligned}
&\forall x \exists y \forall z \exists u \forall v \exists w R(x,y,z) \wedge \neg R(u,v,w) \\
&\wedge \forall x'' y'' z'' u'' S(x'', y'', z'', u'') \to (\forall v \exists w R(x'', y'', z'') \wedge \neg R(u'', v, w)) \\
&\wedge \forall x'' y'' z'' u'' \neg S(x'', y'', z'', u'') \to (\exists v' \forall w' \neg R(x'', y'', z'') \vee R(u'', v', w'))\,.
\end{aligned}
$$

The new S relations record whether, given some values for x, y, z, u, the original formula is true or false. As such, their addition does not affect the models of the formula (notwithstanding the interpretation of S).

We put this expanded formula into prenex form:

$$
\begin{aligned}
&\forall x'' y'' z'' u'' \forall x \exists y \forall z \exists u \exists v' \forall v \exists w \forall w' \\
&R(x,y,z) \wedge \neg R(u,v,w) \\
&\wedge S(x'', y'', z'', u'') \to (R(x'', y'', z'') \wedge \neg R(u'', v, w)) \\
&\wedge \neg S(x'', y'', z'', u'') \to (\neg R(x'', y'', z'') \vee R(u'', v', w'))
\end{aligned}
$$

and apply the original translation to it. The S relations become existential variables in the outermost quantifier block.

The idea is that when the existential variable u is queried and given the value a, Prover can then ask Delayer to identify some specific sub-problem with $u = a$ that evaluates to true. If Delayer refuses to do this, their choice of u in the original formula quickly generates a contradiction, and otherwise x can be set based on the S variable that was made true. In this way, the S variables act as a memory of Delayer's choices.

We describe the decision tree for this formula. Recall that the QBF is constructed so that if all of the existential variables $\{x_i\}_{i=1}^n$ are assigned 0 then the formula is immediately falsified; similarly no universal set $\{y_i\}_{i=1}^n$ may have more than one value given at a time, else the formula is immediately satisfied.

1. Set $x = \alpha$, $z = \gamma$ arbitrarily. Query u^i for $i = 1 \dots n$ until u is given a value. That is, branch on u^1. If $u^1 = 0$ branch on u^2. If all $u^i = 0$ we have a contradiction. Now consider the subtree with $u^a = 1$.
2. Query $S(\alpha, *, \gamma, a)$, for $* = 1 \dots n$, until some S is set to true. If all such S are made false, skip to line 8. Suppose $S(\alpha, \beta, \gamma, a) = 1$.
3. Set $x = a$ since $S(\alpha, \beta, \gamma, a) = 1$.
4. Query y. Suppose $y = b$. Set $v = b$ to match, as well as $x'' = \alpha$, $y'' = \beta$, $z'' = \gamma$, $u'' = a$.
5. Query w'. Suppose $w' = c$.
6. Since $S(\alpha, \beta, \gamma, a) = 1$ we now have $R(\alpha, \beta, \gamma) = 1$ and, importantly, $R(a, b, c) = 0$ forced.
7. $x = a$ and $y = b$ are still set, and $R(a, b, c) = 0$ prompts setting $z = c$ for a contradiction.
8. Suppose instead that $S(\alpha, *, \gamma, a) = 0$ for all values of $*$. Query $R(\alpha, *, \gamma)$ for $* = 1 \dots n$.
9. If all $R(\alpha, *, \gamma)$ are made false then with $x = \alpha$, query y for a contradiction.
10. If some $R(\alpha, \beta, \gamma) = 1$, set $x'' = \alpha$, $y'' = \beta$, $z'' = \gamma$, $u'' = a$ and since $S(\alpha, \beta, \gamma, a) = 0$ we have $\exists v' \forall w' R(a, v', w')$. Query v'. Suppose $v' = d$.
11. Now $R(a, d, 1) \dots R(a, d, n) = 1$. This contradicts the original choice to set $u = a$, so return to the main formula and set $v = d$, and query w for a contradiction.

For each instance of an existential variable e in the unification closing the tableau refutation, the decision tree has branched once on either e, or e', as well as branching once on the n variables $S(\alpha, *, \gamma, a)$.

This motivating example shows how additional structure derived from the original sentence can aid the Prover in the resulting sequence of QBFs. To generalise this method we will introduce new relational variables for each level of the quantifier prefix.

We are now more interested in blocks of variables than individual variables, so represent our general formula with slightly different notation to emphasise this. Take the first-order sentence

$$\phi := \forall X_1 \exists Y_1 \ldots \forall X_k \exists Y_k \ \mathcal{D}_1(X_1, Y_1, \ldots, X_k, Y_k) \wedge \ldots \wedge \mathcal{D}_r(X_1, Y_1, \ldots, X_k, Y_k)$$

with atoms

$$R_i^1(X_1, Y_1, \ldots, X_k, Y_k) \vee \ldots \vee R_i^s(X_1, Y_1, \ldots, X_k, Y_k)$$

where X_i and Y_i are mutually disjoint sets of variables.

It is modified to include new relations $S_k, S'_k, \ldots,\ S_1, S'_1$. The following statement is conjoined to the original.

$$\begin{array}{ll}
\forall X''_1, Y''_1, \ldots, X''_k, Y''_k & \neg S_k(X''_1, Y''_1, \ldots, X''_k, Y''_k) \ \vee\ \bigwedge_{i\in[r]} \mathcal{D}_i(X''_1, Y''_1, \ldots, X''_k, Y''_k) \\
\forall X''_1, Y''_1, \ldots, X''_k, Y''_k & S_k(X''_1, Y''_1, \ldots, X''_k, Y''_k) \ \vee\ \bigvee_{i\in[r]} \neg\mathcal{D}_i(X''_1, Y''_1, \ldots, X''_k, Y''_k) \\
\forall X''_1, Y''_1, \ldots, X''_k & \neg S'_k(X''_1, Y''_1, \ldots, X''_k) \ \vee\ \exists Y_k \ \bigwedge_{i\in[r]} \mathcal{D}_i(X''_1, Y''_1, \ldots, X''_k, Y_k) \\
\forall X''_1, Y''_1, \ldots, X''_k & S'_k(X''_1, Y''_1, \ldots, X''_k) \ \vee\ \forall Y'_k \ \bigvee_{i\in[r]} \neg\mathcal{D}_i(X''_1, Y''_1, \ldots, X''_k, Y'_k) \\
\vdots & \qquad\qquad\vdots \\
\forall X''_1, Y''_1 & \neg S_1(X''_1, Y''_1) \ \vee\ \forall X_2 \exists Y_2 \ldots \forall X_k \exists Y_k \ \bigwedge_{i\in[r]} \mathcal{D}_i(X''_1, Y''_1, \ldots, X_k, Y_k) \\
\forall X''_1, Y''_1 & S_1(X''_1, Y''_1) \ \vee\ \exists X'_2 \forall Y'_2 \ldots \exists X'_k \forall Y'_k \ \bigvee_{i\in[r]} \neg\mathcal{D}_i(X''_1, Y''_1, \ldots, X'_k, Y'_k) \\
\forall X''_1 & \neg S'_1(X''_1) \ \vee\ \exists Y_1 \forall X_2 \exists Y_2 \ldots \forall X_k \exists Y_k \bigwedge_{i\in[r]} \mathcal{D}_i(X''_1, Y_1, \ldots, X_k, Y_k) \\
\forall X''_1 & S'_1(X''_1) \ \vee\ \forall Y'_1 \exists X'_2 \forall Y'_2 \ldots \exists X'_k \forall Y'_k \bigvee_{i\in[r]} \neg\mathcal{D}_i(X''_1, Y'_1, \ldots, X'_k, Y'_k)
\end{array}$$

The sets X'_i and X''_i are copies of the set X_i. To put this additional statement into prenex form, follow the rules:

- X''_i, Y''_i outermost
- X'_i immediately before X_i
- Y_i immediately before Y'_i

And then the conjunction of the two parts is returned to the form required for our original translation. This embellished sentence ϕ^* is syntactically ugly but enjoys the same models as ϕ up to reduction to the original signature σ; thus, the semantic change is slight.

7. The lower bound revisited

The models are essentially unchanged by the proposed modification, the number of variables has only increased polynomially, and the arity of the new S relations is still bounded above by the number of variables in the original first-order sentence. Therefore, the proof of the exponential lower bound in the case that ϕ (and so ϕ^*) has an infinite model still applies exactly as given in Section 4.

8. The upper bound

Taking an analytic tableau refutation [10] of a logical contradiction ϕ, we explain how to generate a decision tree for Φ_n. The unification that closes the tableau shows how to determine universal assignments from choices made for the existential variables. Follow the unification in order, expanding existential variables with a branching factor of n. When it is necessary to set a universal variable (unless this can be done within the rules for $\forall$-reduction), first use the S relations to find a specific sub-problem claimed to be correct for the variables that have been assigned so far. Once in a position to derive R variables (recall these are outermost and existential in our QBF), we do so.

Let ζ_i (resp. η_i) range over all assignments to variables in the block X_i (resp. Y_i). If all $S(\zeta_1, \eta_1, \ldots, \zeta_j, \eta_j)$ (similarly $S'(\zeta_1, \eta_1, \ldots, \zeta_j)$) are set to false, we work through the sub-sentence

$$\begin{aligned}
&S(\zeta_1, \eta_1, \ldots, \zeta_j, \eta_j) \vee \exists X'_{j+1} \forall X_{j+1} \exists Y_{j+1} \forall Y'_{j+1} \ldots \exists X'_k \forall X_k \exists Y_k \forall Y'_k \\
&\textstyle\bigvee_{i \in [r]} \neg \mathcal{D}_i(\zeta_1, \eta_1, \ldots, \zeta_j, \eta_j, X'_{j+1}, Y'_{j+1}, \ldots, X'_k, Y'_k) \\
&\wedge \textstyle\bigwedge_{i \in [r]} \mathcal{D}_i(\zeta_1, \eta_1, \ldots, \zeta_j, \eta_j, X_{j+1}, Y_{j+1}, \ldots, X_k, Y_k).
\end{aligned}$$

Note the quantifier order of this sentence means that the universal variables can simply copy the choice made for the immediately preceding existential, and so a contradiction is reached in polynomial expansion of size $O(n^b)$, where b is the total number of variables in the first-order sentence.

Assume instead that some $S(\zeta_1, \eta_1, \ldots, \zeta_i, \eta_i)$ is set true, then any remaining $S(\zeta_1, \eta_1, \ldots, \zeta_i, \eta_i)$ do not need to be considered in this branch. The assignments to relational variables (S and R) are never changed on a given branch, and they will form a memory during backtracking, when later existential assignments need to be forgotten in order to make universal assignments. When we backtrack we will need to forget some variables, yet when we jump again forward we will have some memory of them in the relational variables, in the obvious fashion.

Let m be the number of Skolem functions in the unification, b the number of variables in the original first-order sentence, n the size of model being searched for. The decision tree branches m times on existential variables, with a branching factor of n. Up to b sets of S variables have been added, each with up to n^b members, and we may branch on any of these sets, once only. The size of the decision tree refutation is therefore $O(n^m \cdot n^{b^2})$. Thus we have obtained the following.

Theorem 7. *Let ϕ be a first-order sentence without any models, and ϕ^* be its embellishment. Then the sequence of QBFs $\langle \Phi_n^* \rangle$ enjoy refutations in tree-like Q-Resolution of size $n^{O(1)}$.*

9. Extension to QU-Res

Although stated in terms of Q-Resolution, our result also holds for tree-like QU-Resolution, in which the Resolution rule may be applied to universally, as well as existentially, quantified variables.

Since QU-Resolution contains Q-Resolution, our upper bound immediately transfers. For the exponential lower bound, we note that the game description of tree-like Q-Resolution can be extended to describe QU-Resolution by allowing the Prover to query universally quantified variables as well as existentially quantified [7]. This may shorten the refutation, since it offers a way for the Prover to set universal variables after existential variables that are later in the prefix have already been assigned. However, it does not affect the crux of our argument, that $\Omega(n)$ values must be considered in a free choice at some point during the game, and only constantly many values can be considered in each free choice. Thus, the analogous version of Theorem 1 holds for QU-Resolution as well.

QU-Resolution is exponentially stronger than Q-Resolution in the DAG-like case. This is demonstrated in [19] via the formulas of Kleine Büning, Karpinski and Flögel [16]. It is not known whether a separation exists between the tree-like variants. Our results here mean that such a separation — if it exists — cannot be shown by using translations of first-order formulas as considered here.

10. The gap theorem for the expansion-system ∀Exp+Res

Our observation of the behaviour of tree-like Q-Resolution on the initial translation of these formulas, reveals a weakness in the proof system. This weakness can be characterised in the game description as the Prover lacking memory of previous answers. Further, it suggests that if the game was modified to allow the Prover to remember answers previously given and return to these later in the game, we would have a stronger proof system. In fact, such a game characterises the tree-like version of the QBF resolution system ∀Exp+Res, defined in [15] to model expansion-based QBF solving (cf. also Section 2).

A new Prover-Delayer game

In each round of the game

1. The Prover assigns some number of universal variables.
2. If the Prover has remembered previous answers that were given under a universal assignment matching the current assignment, the Delayer must immediately set those values accordingly.
3. The Prover queries the value of some unassigned existential variable x, and Delayer responds with weights p_0 and p_1 such that $p_0 + p_1 = 1$. The Prover chooses a value for x from $\{0, 1\}$, and the Delayer scores $\lg\left(\frac{1}{p_i}\right)$ points if the Prover set $x = i$.
4. The Prover remembers the choice for x and the universal assignment under which that choice was made. All assignments are removed.

The Prover wins the game if any clause in Φ is falsified, and the game ends as soon as this occurs.

Theorem 8. *Suppose there is a tree-like ∀Exp+Res refutation of QBF Φ, having size at most S. Then there is a strategy for the Prover, such that any Delayer strategy scores at most* $\lg\lceil\frac{S}{2}\rceil$ *points.*

Proof. Let π be a tree-like ∀Exp+Res refutation of Φ of size $\leq S$. Informally, the Prover plays according to π, starting at the empty clause and following a path in the tree to one of the axioms. At a Resolution inference the Prover will set the universal variables according to the annotation of the pivot, and query the existential variable. The Prover will keep the invariant that at each moment in the game, they have in memory a set of choices that contradicts each of the annotated literals in the current clause. This invariant holds in the beginning at the empty clause. In the end the Prover wins when they reach a clause that was introduced to the proof by axiom download. The axiom clauses are downloaded with a single annotation across all variables. Therefore the Prover can set the universal variables according to this assignment, and the Delayer is forced to set the existential variables according to their earlier, recalled, choices. None of the existential assignments can satisfy the clause, since we know that the Delayer's previous choices must disagree with the annotated literals in the clause, and because there is a single annotation that must include setting universal literals in the original clause to 0, it is not satisfied by the universal assignments either.

The Prover is at a clause C that was derived by Resolution. Let the pivot variable be x^{μ}. The Prover sets universal variables according to μ, that is, $u = 0$ if and only if $u \in \mu$. If the Delayer chooses to set a value for x, Prover continues to the next phase.

Otherwise, the Prover queries x and, having been provided with weights p_0 and p_1, chooses $x = i$. The model for making this choice is described below. If x is set to 0 then the Prover moves to the parent clause with x^{μ}, if $x = 1$ then they move to the clause containing $\neg x^{\mu}$. The Prover remembers the value of x under μ, and forgets all current assignments. The invariant is maintained, since the new clause is a strict subset of the previous clauses, plus (wlog) x^{μ}, which is contradicted by the Delayer's most recent choice.

When the Prover is required to decide the value of x under the universal assignment μ she considers the subtree rooted at each of the parents of the current clause C. Recall that if the Prover chooses $x = 0$ then she will move on to the parent clause C_0 containing x^{μ}, so the subtree corresponding to setting $x = 0$ is the subtree rooted at C_0. The subtree for choosing $x = 1$ is defined analogously, with C_1 denoting the parent of C containing $\neg x^{\mu}$. Let L_0 and L_1 be the number of leaves in the subtrees deriving C_0 and C_1 respectively. If $\frac{L_i}{L_0+L_1} \leq p_i$ then choose $x = i$. This must hold for (at least) one of the p_i since $p_0 + p_1 = 1 = \frac{L_0}{L_0+L_1} + \frac{L_1}{L_0+L_1}$. Delayer scores $\lg\left(\frac{1}{p_i}\right) \leq \lg\left(\frac{L_0+L_1}{L_i}\right)$ points.

Suppose the game ends at leaf l after n decision points. Let L^j be the total number of leaves in the tree rooted at the active clause C_j after j choices. Then L^0 is the number of leaves in the whole tree. From C_j the Prover makes assignments and queries the pivot variable of the next step. Let L_i^j be the number of leaves in the subtree corresponding to setting the pivot to i at step

$j+1$. Note that $L^j = L^j_0 + L^j_1$ for every $j < n$, and $L^j = L^{j-1}_{i_j}$ where i_j was the choice made at step j. Delayer scores at most $\lg\left(\frac{L^{j-1}}{L^{j-1}_{i_j}}\right) = \lg\left(\frac{L^{j-1}}{L^j}\right)$, if the pivot is set to i at step j.

By assumption n choices are made in the game, so the total number of points scored is at most $\displaystyle\sum_{j=1}^{n} \lg\left(\frac{L^{j-1}}{L^j}\right) = \lg\left(\prod_{j=1}^{n} \frac{L^{j-1}}{L^j}\right)$, which simplifies to $\lg\left(\frac{L^0}{L^n}\right)$. After the final decision the subtree corresponding to the choice made has exactly one leaf, because the game ends here and the game must end at an axiom clause. Therefore $L^n = 1$. L^0 is the total number of leaves in the tree, so $L^0 \leq \lceil \frac{S}{2} \rceil$ for a tree of size S . The total score is at most $\lg\lceil \frac{S}{2} \rceil$. □

Theorem 9. *Suppose the shortest tree-like $\forall$Exp+Res refutation of QBF Φ has size S. Then there is a strategy for the Delayer allowing him to score at least $\lg\lceil \frac{S}{2} \rceil$ points against any Prover strategy.*

PROOF. For QBF Φ, $\Phi[u/i]$ indicates that every occurrence of u in Φ has been substituted with constant $i \in \{0, 1\}$. The expansion of QBF $\forall u \Phi$ on u is $\Phi[u/0] \wedge \Phi[u/1]$. New copies of every variable are made, and we might denote the copy of variable x in $\Phi[u/0]$ as $x^{\neg u}$ and the copy of x in $\Phi[u/1]$ as x^u, for example. Supposing the next universally quantified variable was v, now both copies of the formula have a copy of v, and both can expand on v, causing each remaining variable to be duplicated again. Now we have four copies of each variable x, and might label the copy resulting from $\Phi[u/1][v^u/1]$ as x^{uv}, the copy from $\Phi[u/1][v^u/0]$ as $x^{u\neg v}$, etc. The naming for the new variables is not important; we simply seek to highlight the correspondence between variables in this expanded formula and variables in a $\forall$Exp+Res proof. When the outermost variables are existentially quantified they can be brought to the beginning of the whole formula, leaving a conjunction of sub-formulas each beginning with universally quantified variables that can be further expanded. Continuing this process until only existentially quantified variables remain produces the full universal expansion of Φ, denoted $E(\Phi)$, a purely propositional formula.

In creating a $\forall$Exp+Res proof there is an implicit expansion stage. It may not be necessary to create the full universal expansion of Φ, if the refutation does not require it. Instead, the formula may be expanded according to an expansion tree, which generates a subset of the full expansion.

The set of tree-like $\forall$Exp+Res refutations of Φ and the set of tree-like Resolution refutations of $E(\Phi)$ are isomorphic, so in particular the shortest such proofs can be identified with one another. The Delayer's strategy is to keep a partial assignment $\mathbf{a}$ to the variables in $E(\Phi)$, indicating the answers that have already been given in the game. For a partial existential assignment a and propositional formula ϕ, $\phi|_a$ is the restriction of ϕ by a.

The existential variable x^μ is queried in $E(\Phi)$ whenever the universal variables have been assigned in accordance with μ, and the Prover queries x in the game on Φ. On receiving such a query, the Delayer finds the number of leaves in the shortest refutation of each of $E(\Phi)|_{\mathbf{a},x^\mu}$ and $E(\Phi)|_{\mathbf{a},\neg x^\mu}$, denoted L_1 and L_0 respectively. Set $p_i = \frac{L_i}{L_1+L_0}$.

We use induction on the number of existential variables n in $E(\Phi)$ to show that the number of points scored is at least $\lg L(E(\Phi \vdash \bot))$ where $L(E(\Phi \vdash \bot))$ is the number of leaves in the shortest tree-like Resolution refutation of $E(\Phi)$. When there are no variables in the expanded formula then the shortest proof is empty and the Delayer scores no points in the game. When there is one variable in the expanded formula, the shortest refutation must be resolution on that one variable. There are two leaves in the refutation tree and the game consists of a single, balanced, choice scoring one point.

For the inductive step, let now π be the shortest refutation of $E(\Phi)$. Assume the last step has pivot x^μ. Restrict π by x^μ and $\neg x^\mu$, producing refutations of $E(\Phi)|_{x^\mu}$ and $E(\Phi)|_{\neg x^\mu}$. These must be the shortest such refutations or we contradict the assumption that π is minimal. If $E(\Phi)$ has n variables then each of the restricted formulas has $n-1$ variables. Let L be the number of leaves in π, L_0 and L_1 the number of leaves in the refutations of $E(\Phi)|_{x^\mu}$ and $E(\Phi)|_{\neg x^\mu}$ respectively. $L = L_0 + L_1$. Answering $x^\mu = i$ will score $\lg\left(\frac{L_1+L_0}{L_i}\right)$, and the game on the sub formula scores at least $\lg(L_i)$. Therefore the overall score is at least $\lg(L_1 + L_0) = \lg(L)$.

Since every Resolution refutation of $E(\Phi)$ is equivalent to a $\forall$Exp+Res refutation of Φ, it follows that the Delayer scores at least $\lg L(\Phi)$, for $L(\Phi)$ the number of leaves in the shortest tree-like $\forall$Exp+Res refutation of Φ. The refutation forms a binary tree, so the number of leaves is $\lceil \frac{S}{2} \rceil$. $\square$

This game characterisation provides an independent proof that tree-like $\forall$Exp+Res simulates tree-like Q-Resolution, shown by a rather involved argument in [15] (cf. also [6]). It is also not difficult to see that the characterisation similarly works for the more powerful expansion-based QBF system

IR-calc from [4]. In particular, in the proof of Theorem 8 the Prover can similarly use a tree-like IR-calc proof to guide her strategy. Interestingly, $\forall$Exp+Res and IR-calc are separated in their DAG-like versions [5], while they are equivalent in their tree-like versions [6]. Our game characterisation here provides an alternative approach towards this equivalence of tree-like $\forall$Exp+Res and IR-calc.

Returning to the topic of the gap theorem, we now argue that this new game can be used to prove the lower bound part of the gap theorem on our original translation, where a constant is 'mentioned' if it appears positively in the annotation of the variable being queried, or is the subject of the question itself.

The idea for the upper bound is again to mimic the unification, we demonstrate the technique on the example in Figure 1. Recall that the first-order formula $\forall x \exists y \forall z \exists u \forall v \exists w\ R(x, y, z) \wedge \neg R(u, v, w)$ is disproved using the unification $x_2 \leftarrow U_1(x_1, z_1)$, $v_1 \leftarrow Y_2(x_2)$, $z_2 \leftarrow W_1(x_1, z_1, v_1)$ in which lower case letters indicate free variables and upper case letters indicate functions.

$x_2 \leftarrow U_1(x_1, z_1)$ is the first part of the unification, stating that the free variable x_2 should mimic the behaviour of function U_1 on x_1 and z_1 (previously set free variables). In our new game, this means that x and z are first given arbitrary values by the Prover, say $x = \alpha$ and $z = \gamma$, then u is queried and must be given some value to avoid a contradiction, say $u = a$. This is equivalent to discovering how the function U_1 behaves on input (α, γ). The fact that $u = a$ when $x = \alpha$ and $z = \gamma$ is remembered, and all variables are unset.

The unification continues with $v_1 \leftarrow Y_2(x_2)$. We know from the previous step that x_2 should match the behaviour of U_1, so x is set to a, and y queried. Suppose the Delayer sets $y = b$. Remember that $y = b$ when $x = a$, and forget all assignments. The Prover has discovered something about the behaviour of function Y_2, and will use this to set the assignment to v wherever v_1 appears in the unification.

Finally the unification sets $z_2 \leftarrow W_1(x_1, z_1, v_1)$, meaning the Prover should query the value of w while $x = \alpha$, $z = \gamma$ (returning to the arbitrary assignments used in the first step) and $v = b$ (to match the choice of y determined in the second step). Suppose the Delayer sets $w = c$, the Prover remembers this choice.

Under universal assignments $x = \alpha$, $z = \gamma$ and $v = b$ we now have that $u = a$, $v = b$, $w = c$. Together these allow the Prover to discover that $R(a, b, c) = 0$. Under $x = a$, $z = c$ we have $y = b$, implying that

$R(a, b, c) = 1$. This generates the contradiction.

Generalising this example we obtain:

Theorem 10. *Let ϕ be a first-order sentence and $\langle \Phi_n \rangle$ its translations into QBFs (as obtained in the first translation in Section 3).*

If ϕ has no models, then there exist tree-like $\forall$Exp+Res refutations of $\langle \Phi_n \rangle$ of size $n^{O(1)}$. If ϕ has some (infinite) model, then all tree-like $\forall$Exp+Res refutations of $\langle \Phi_n \rangle$ must have size $2^{\Omega(n)}$.

11. Conclusion

We have demonstrated a translation from first-order formulas to QBF families for which a complexity gap exists in tree-like Q-Resolution. Our translation is not as natural as that used in Riis' original translation to propositional logic, due to an inherent constraint in Q-Resolution that $\forall$-reduction must respect the order of variables in the prefix. We demonstrate how to manage the constraint on $\forall$-reduction in this setting so that short proofs can be achieved where the original formula had no models. We have also noted that in this setting, tree-like QU-Resolution and Q-Resolution coincide, with the additional power of QU-Resolution providing at most a polynomial improvement in the proof length. It is not currently known whether there are any situations in which tree-like QU-Resolution is exponentially stronger than tree-like Q-Resolution, the separation of these two systems has only been demonstrated in the DAG-like variant.

Our investigation has led to a game description of proof size in tree-like $\forall$Exp+Res, which may be used to demonstrate that the gap theorem lifts to tree-like $\forall$Exp+Res with the more natural QBF encoding. Generating a series of QBFs generated from the unsatisfiable first-order formula $\forall x \exists y \forall z \exists u \forall v \exists w\ R(x, y, z) \wedge \neg R(u, v, w)$, that has short proofs in tree-like $\forall$Exp+Res but exponential sized proofs in tree-like Q-Resolution and in fact tree-like QU-Resolution, we have exhibited new formulas that separate the two systems. It is likely that other separating formulas could be generated in the same way.

Acknowledgements

Research was supported by grant no. 60842 by the John Templeton Foundation.

[1] Ben-Sasson, E., and Wigderson, A. Short proofs are narrow - resolution made simple. *Journal of the ACM 48*, 2 (2001), 149–169.

[2] Beyersdorff, O., Blinkhorn, J., and Hinde, L. Size, cost, and capacity: A semantic technique for hard random QBFs. In *Proc. Conference on Innovations in Theoretical Computer Science (ITCS)* (2018).

[3] Beyersdorff, O., Bonacina, I., and Chew, L. Lower bounds: From circuits to QBF proof systems. In *Proc. ACM Conference on Innovations in Theoretical Computer Science (ITCS)* (2016), ACM, pp. 249–260.

[4] Beyersdorff, O., Chew, L., and Janota, M. On unification of QBF resolution-based calculi. In *Proc. Symposium on Mathematical Foundations of Computer Science (MFCS)* (2014), pp. 81–93.

[5] Beyersdorff, O., Chew, L., and Janota, M. Proof complexity of resolution-based QBF calculi. In *Proc. Symposium on Theoretical Aspects of Computer Science (STACS)* (2015), LIPIcs, pp. 76–89.

[6] Beyersdorff, O., Chew, L., Mahajan, M., and Shukla, A. Are short proofs narrow? QBF resolution is not so simple. *ACM Transactions on Computational Logic 19* (2018).

[7] Beyersdorff, O., Chew, L., and Sreenivasaiah, K. A game characterisation of tree-like Q-resolution size. *Journal of Computer and System Sciences* (2017). in press.

[8] Beyersdorff, O., Galesi, N., and Lauria, M. A characterization of tree-like resolution size. *Inf. Process. Lett. 113*, 18 (2013), 666–671.

[9] Beyersdorff, O., Galesi, N., and Lauria, M. Parameterized complexity of DPLL search procedures. *ACM Transactions on Computational Logic 14*, 3 (2013).

[10] D'Agostino, M., Gabbay, D., Hähnle, R., and Posegga, J. *Handbook of Tableau Methods*. Springer Netherlands, 1999.

[11] Dantchev, S., Martin, B., and Szeider, S. Parameterized proof complexity. *Computational Complexity 20* (2011). An extended abstract appeared at FOCS 2007.

[12] Dantchev, S. S., and Martin, B. Rank complexity gap for Lovász-Schrijver and Sherali-Adams proof systems. *Computational Complexity 22*, 1 (2013), 191–213. An extended abstract appeared at STOC 2007.

[13] Gent, I. P., Nightingale, P., and Rowley, A. G. D. Encoding quantified CSPs as quantified Boolean formulae. In *Proc. 16th European Conference on Artificial Intelligence (ECAI)* (2004), pp. 176–180.

[14] Gent, I. P., Nightingale, P., Rowley, A. G. D., and Stergiou, K. Solving quantified constraint satisfaction problems. *Artif. Intell. 172*, 6-7 (2008), 738–771.

[15] Janota, M., and Marques-Silva, J. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science 577*, Supplement C (2015), 25 – 42.

[16] Kleine Büning, H., Karpinski, M., and Flögel, A. Resolution for quantified Boolean formulas. *Inf. Comput. 117*, 1 (1995), 12–18.

[17] Pudlák, P., and Impagliazzo, R. A lower bound for DLL algorithms for SAT. In *Proc. 11th Symposium on Discrete Algorithms (SODA)* (2000), pp. 128–136.

[18] Riis, S. A complexity gap for tree-resolution. *Computational Complexity 10* (2001), 179–209.

[19] Van Gelder, A. Contributions to the theory of practical quantified Boolean formula solving. In *Proc. Principles and Practice of Constraint Programming (CP)* (2012), pp. 647–663.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il