Journal on Satisfiability, Boolean Modeling and Computation 15 (2024) 9–25

The Riis Complexity Gap for QBF Resolution

Olaf Beyersdorff olaf.beyersdorff@uni-jena.de
Institute of Computer Science
Friedrich Schiller University Jena
07743 Jena
Germany

Judith Clymo scjc@leeds.ac.uk
School of Computing
University of Leeds
Leeds LS2 9JT
U.K.

Stefan Dantchev s.s.dantchev@durham.ac.uk
Barnaby Martin barnaby.d.martin@durham.ac.uk
Department of Computer Science
University of Durham
Durham DH1 3LE
U.K.

Abstract

We give an analogue of the Riis Complexity Gap Theorem in Resolution 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 QBF Resolution systems are either of polynomial size (if ϕ has no models) or at least exponential in size (if ϕ has some infinite model). However, we show that this gap theorem is sensitive to the translation and different translations are needed for different QBF resolution systems. For tree-like Q-Resolution, the translation to QBF must be given additional structure in order for the polynomial upper bound to hold. This extra structure is not needed in the system tree-like ∀Exp+Res, where we see the complexity gap on a natural translation to QBF.

Keywords: *Complexity gap, proof complexity, quantified boolean formulas, resolution*

Submitted 6 November 2023; revised 15 October 2024; accepted 16 October 2024

1. Introduction

The Complexity Gap Theorem [23] considers a translation of a first-order sentence ϕ to a sequence of propositional formulas, and states that the complexity of refuting these propositional formulas in tree-like Resolution depends on whether ϕ has any [in]finite models. The nth member of the sequence of propositional formulas is satisfiable if and only if ϕ has a model of size n. When ϕ has an infinite model but no finite models then all tree-like Resolution refutations of related propositional formulas are exponential in size. When ϕ also has no infinite model then there must exist polynomial-size tree-like Resolution refutations of the propositional formulas.

Quantified Boolean logic is an extension of propositional logic in which variables may be existentially or universally quantified. 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 motivated the development of QBF solvers and proof systems [12]. Recent research has sought to understand which proof-theoretic techniques lift to the QBF setting [1,7,8] as well as developing QBF specific techniques [2,4,6,13].

We investigate whether the Complexity Gap Theorem holds in various QBF resolution systems [5,18,19,24]. 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 [16,17]. 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.

We demonstrate that tree-like Q-Resolution [19] 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 the propositional case, there exist formulas with no models but requiring exponential sized tree-like Q-Resolution refutations for the nth member of the sequence. We show that if the first-order formula ϕ is embellished with additional structure (precisely defined in Section 6) to obtain a formula ϕ^* before applying the translation then tree-like Q-Resolution is able to refute the nth member of the sequence in polynomial time 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 [24], which extends tree-like Q-Resolution. In contrast, in the QBF resolution system of tree-like $\forall$Exp+Res from [18], modelling QBF expansion solving, the gap holds naturally, that is without the embellishment. In this sense, $\forall$Exp+Res does not possess the same deficiency as tree-like Q-Resolution.

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 $Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \psi$ 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. If x_i and x_j are not in the same quantifier level and $i < j$, then we say x_j has higher quantification level than x_i. 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. Where convenient to do so we write the quantifier once per level rather than for each variable.

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

Finally, $\forall$Exp+Res [18] 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, and μ and ω are referred to as annotations of x^μ and x^ω. Every clause in a $\forall$Exp+Res refutation is either introduced in this way as an axiom, 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. We consider first-order logic to be relational and possibly with constants. Functions can be encoded as relations in the standard way. The method is inspired by the encoding of ϕ into propositional formulas in conjunctive normal form (CNF) previously given by Riis [23], and is similar to other translations used to encode QCSP instances as QBF in [16]. A more succinct "binary" or "logarithmic" form of encoding is discussed in [17] but for our purposes, since ϕ is fixed, the benefit of this more succinct encoding is not important.

We begin with a first-order sentence

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

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

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

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 refer to the set of existentially quantified variables by $\{x_j | Q_j = \exists\}$, or to the relevant indices by $\{j | Q_j = \exists\}$.

Each first-order variable x becomes n Boolean variables $x^1, \ldots, x^n$. If x^i is made true this indicates that x is evaluated as the ith element in a model of size n. We introduce existentially quantified variables associated with each instantiation of a relational predicate $R_i^j(\lambda_1, \ldots, \lambda_k)$ indicating that the tuple $(\lambda_1, \ldots, \lambda_k)$ is in the relation R_i^j.

In the original sentence a variable x can only take on one value at a time, and must be given some value. We introduce clauses so that if any existential variable is not given exactly one value the QBF is falsified, and if any universal variable is not given exactly one value then the QBF is made true. 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 $(\bigvee_{i \in [n]} x^i) \wedge \bigwedge_{j \neq i \in [n]} (\neg x^i \vee \neg x^j)$. Similarly $\neg(\sum_{i=1}^n x^i = 1)$ is shorthand for the conjunction of clauses $(\neg x^i \vee \bigvee_{j \neq i} x^j)$.

We can now build our sequence of QBFs

$$\begin{aligned}
\phi_n := {} & \exists_{\lambda_1, \ldots, \lambda_k \in [n]} R_1^1(\lambda_1, \ldots, \lambda_k) \ldots R_r^s(\lambda_1, \ldots, \lambda_k) \\
& Q_1 x_1^1 \ldots x_1^n \ldots Q_k x_k^1 \ldots x_k^n \\
& \bigwedge_{\{i | Q_i = \exists\}} \Big(\sum_{j \in [n]} x_i^j = 1 \Big) \\
& \wedge \Bigg[\bigwedge_{\{i | Q_i = \forall\}} \Big(\sum_{j \in [n]} x_i^j = 1 \Big) \\
& \rightarrow \Big(\bigwedge_{i \in [r] \lambda_1, \ldots, \lambda_k \in [n]} (x_1^{\lambda_1} \wedge \cdots \wedge x_k^{\lambda_k}) \rightarrow \mathcal{D}_i(\lambda_1, \ldots, \lambda_k) \Big) \Bigg] \quad ,
\end{aligned}$$

where the notation $\exists_{\lambda_1, \ldots, \lambda_k \in [n]} R_1^1(\lambda_1, \ldots, \lambda_k) \ldots R_r^s(\lambda_1, \ldots, \lambda_k)$ indicates that we existentially quantify over all propositional variables of the form $R_j^i(\lambda_1, \ldots, \lambda_k)$ for all tuples $\lambda_1, \ldots, \lambda_k \in [n]$. Where constants were involved the corresponding λ_is are fixed to those constants.

By construction, ϕ_n is true if and only if ϕ has a model of size n (the size of the domain). The quantifier-free part of ϕ_n can be expanded to CNF and 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 (e.g. see [23]), which states that given sets A and B with $|B| > |A|$, there does not exist an injective function $f : A \mapsto B$. We consider a first order formula that makes an opposing (false) claim that given two sets $A = B = \{1, \ldots, n\}$, there is an injective function such that 1 is not in the image. We represent the function f by relation P (where $P(i, j)$ is true if and only if $f(a) = b$).

- Every member of A must have some image in B: $\forall x \exists w P(x, w)$.
- No member of A is mapped to $1 \in B$: $\forall x \neg P(x, 1)$.
- Injectivity: $\forall x, y, z P(x, z) \wedge x \neq y \rightarrow \neg P(y, z)$.

Together, these give ϕ^{PHP}:

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

stating that the relation P contains the graph of a total injective function f from $\mathrm{dom}(f)$ to $\mathrm{dom}(f) \setminus \{1\}$. Clearly, ϕ^{PHP} has no finite models.

The translation to QBF gives us

$$\begin{aligned}
&\exists_{i,j \in [n]} P(i, j) \forall_{i \in [n]} x^i, y^i, z^i \exists_{i \in [n]} w^i \\
&\quad \Big(\sum_{i \in [n]} w^i = 1\Big) \wedge \Big(\Big(\sum_{i \in [n]} x^i = 1 \wedge \sum_{i \in [n]} y^i = 1 \wedge \sum_{i \in [n]} z^i = 1\Big) \rightarrow \\
&\quad x^i \wedge w^\ell \rightarrow P(i, \ell) \qquad i, \ell \in [n] \\
&\quad x^i \rightarrow \neg P(i, 1) \qquad i \in [n] \\
&\quad x^i \wedge y^j \wedge z^k \rightarrow [\neg P(i, k) \vee \neg P(j, k)] \ i \neq j, k \in [n] \\
&\quad \Big)
\end{aligned}$$

Note that, for the sake of readability, the indices expressed by $\lambda_1, \ldots \lambda_k$ in the general form are here denoted by i, j, k, l. This QBF can be written explicitly in prenex conjunctive normal form as

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

4. The Lower Bound

In this section we lift Riis' proof to show the following result.

Theorem 2. *Let ϕ be a first-order sentence which has an infinite model but no finite model. Then any tree-like Q-resolution refutation of QBF Φ_n, representing the statement that there is a model for ϕ of size n, has size at least $2^{\Omega(n)}$.*

We use the game of [9] 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. The game starts with the empty assignment and ends when the current assignment falsifies the matrix of Φ. Each round of the game has the following phases:

1. Setting universal variables: Prover can assign a value to any number of universal variables provided that every existential variable with a higher quantification level is currently unassigned.
2. Declare Phase: Delayer can assign values to any number of unassigned existential variables of his choice.
3. Query Phase: Prover queries the value of one existential variable x that is currently unassigned. Delayer replies with weights $p_0 \geqslant 0$ and $p_1 \geqslant 0$ such that $p_0 + p_1 = 1$. Prover assigns $x = b$ with $b \in \{0, 1\}$ and Delayer scores $\lg(\frac{1}{p_b})$ points. (If $p_b = 0$ then Delayer scores ∞ many points, thus forcing Prover not to play $x = b$.)
4. Forget Phase: Prover can choose any number of assigned variables (without regard to how they are quantified) to lose their assigned values.

Intuitively, the points scored by Prover correspond to the depth of the proof, and for full binary trees, the tree size is exponential in the depth of the tree. Thus exhibiting good Delayer strategies that score many points will lead to lower bounds for proof size. This is the intuition of the original Prover–Delayer game from [22]. Using the more refined version of the game with weights as described above (originating from [9,11]) the game exactly characterises tree-like Q-Resolution size. An example for a Delayer strategy for propositional tree-like Resolution on the pigeonhole formulas is contained in [10]. More examples for tree-like Q-Resolution can be found in [9].

Assignments made in the query phase correspond to branching points in the tree. In particular, if there exists a strategy and some choice of weighting, such that Delayer is guaranteed at least p points in a game on Φ, regardless of how Prover behaves, then any tree-like Q-Resolution refutation of Φ must have size at least 2^p. We give such a strategy for 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 ϕ has a model of size n, 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 ϕ. Delayer cannot win this game since Φ_n is false, but he can guarantee $\Omega(n)$ points, meaning that the tree-like Q-Resolution proof must have size $2^{\Omega(n)}$.

4.1. 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 $x^i \wedge y^j \rightarrow S(i, j)$. 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. This is the distinction between does hold and may hold – the latter may involve variables that are unassigned. At each point in the game we consider the subset $\widetilde{\mathcal{M}} \subseteq \mathcal{M}$ of models that agree with the current assignment.

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.

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.

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

Lemma 3. *Using this strategy, 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 Delayer to lose the game, and in particular these variables must between them reference all n of the elements in the universe.

Φ_n says that ϕ has a model of size n, and each variable in the QBF makes reference to some subset of those n elements: relational variables state that some relation holds between certain values; existential and universal variables state that the corresponding variables in ϕ take a certain value. A constant is mentioned during the query phase of the game if it is either a) referenced by a relational variable that is set during in the query phase or b) referenced by a main variable that is assigned true at the end of the query phase. Recall that k is the number of variables in the first-order sentence ϕ, which is a constant since ϕ is fixed.

Lemma 4. *At least $n-k$ of the universe's n elements are mentioned during the query phase of the game.*

Proof: Let the set $\{w^i\}_{i=1}^n$ be the failed witness. Part way through the game, $k' \leqslant k$ of the main variables have been assigned, and they are set to $c_1 \dots c_{k'}$. Consider some c_j with $j > k'$ and suppose that none of the variables assigned during the query phase have referenced c_j. As a result, there is no information known about c_j to distinguish it from other elements in an infinite universe.

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) assigns w a value that is outside of the n elements allowed by Φ_n. c_j cannot be distinguished from this value, so there is also a model that assigns c_j to w. Therefore the game cannot end yet and Prover is forced to make another query.

This demonstrates that all c_j with $j > k'$ must have been mentioned during the query phase at some point in the game. □

Lemma 5. *Delayer scores $\Omega(n)$ points by the given strategy.*

Proof: All relations have arity bounded above by k and at most k values can be set in the main variables so at most k constants can be mentioned in any one query. (Note that while a query can only mention one propositional variable, one such variable can refer to more than one element in the first-order model, e.g. by querying a propositional variable corresponding to the value of a relation. But the number of elements in the model to which a propositional variable may refer is bounded by k.)

With Lemma 4 this shows that at least $\frac{n-k}{k}$ queries are made during the game, with each scoring one point. □

5. A Surprising Lower Bound

Proposition 6. *Let $\theta := \forall x \exists y \forall z \exists u \forall v \exists w R(x,y,z) \wedge \neg R(u,v,w)$ and $\langle \Theta_n \rangle_{n \in \mathbb{N}}$ be the sequence of QBFs expressing that θ has a model of size n. Although θ has no models, any tree-like Q-Resolution refutation of Θ_n must have size $\Omega(2^n)$.*

Proof: We show a strategy that allows Delayer to score $\Omega(n)$ points. Delayer uses the rules below for responding to Prover queries.

1. No existentially quantified variable may have two values assigned simultaneously.
2. If $x = c$ and $\neg R(c, d, e)$, for some d, e then answer $y \neq d$.
3. If $x = c$ then answer $u \neq c$.
4. If $R(c, d, e)$ for some d and e then answer $u \neq c$.
5. If $u = c$ and $v = d$ and $R(c, d, e)$ for some e, then answer $w \neq e$.
6. If $x = c$ and $y = d$, then answer $R(c, d, e)$, for each e.
7. If $u = c$ and $v = d$ and $w = e$, then answer $\neg R(c, d, e)$.
8. When none of the above rules apply, Delayer gives weights 1/2 to both assignments (and will score one point whichever assignment Prover makes).

In items 1 to 7 Delayer forces Prover to answer according to his wish by setting weights 0 and 1 (Delayer's preferred choice gets weight 1), but Delayer will not score any points. Thus Delayer only scores a point when item 8 applies.

There are three ways that the QBF can become false. First, by having simultaneously that $x = c$, $y = d$, $z = e$ and $\neg R(c, d, e)$. This cannot happen because if $x = c$ and $y = d$ then $R(c, d, e)$ is made true by rule 6, and if $x = c$ and $\neg R(c, d, e)$ then y cannot be given the value d due to rule 2. x cannot be set after y by the rules of Q-Resolution. Second, the QBF could be made false by having simultaneously that $u = c$, $v = d$, $w = e$ and $R(c, d, e)$ and similarly this is not possible according to rules 4, 5 and 7. Third, the QBF may become false by failing to assign exactly one value to some existentially quantified variable. The first rule ensures that at most one value is given to each existential variable. Therefore the QBF must become false by failing to assign any value to some existential.

Now we need to show that Delayer scores linearly many points before all possible values are excluded for any existentially quantified variable. We consider the cases when u, w or y is the subject of the conflict.

Values for u are excluded by rules 3 or 4, or may have been excluded directly by a Prover choice scoring one point. Rule 3 can only exclude one value for u at a time. For rule 4 to exclude a value we must have one of the R variables assigned positively, and it must be a different variable for each excluded value of u. Either this was done in a Prover choice (scoring one point) or it was forced by rule 6, but then y must have been assigned its value by a Prover choice (since Delayer rules only exclude values for y). For rule 6 to force R variables that would be able to exclude different values for u it would be necessary to change the assignment to x, which requires forgetting and re-querying y. Therefore, if the game ends by ruling out all values of u then Delayer has scored at least $n - 1$ points.

If instead it is w that has every value excluded then for each of these values we have either that it was set in a Prover choice and Delayer scores one point, or else it was forced through rule 5. Rules 3 and 4 ensure that it is not possible to have simultaneously $u = c$ and $R(c, d, e)$ unless $R(c, d, e)$ was assigned in a Prover choice, and a different R assignment would be needed for each excluded value of w. If the game ends by exhausting all possible assignments to w then Delayer has scored at least n points.

Finally, if the game ends because no value is assigned to y then for each of the possible values either it was excluded in a choice made by Prover or it was excluded by rule 2. A different R variable would be needed for each excluded value of y, and they could only have been forced by rule 7 requiring a new assignment to v and so a new positive assignment to w for each one. In this case Delayer scores at least n points before the game ends.

Because Delayer must score $\Omega(n)$ points by the end of the game we have that any tree-like Q-Resolution proof of Θ_n has size $\Omega(2^n)$. ☐

This lower bound is surprising because if the result of [23] lifted directly to Q-Resolution on this natural translation to QBF then we would expect a formula without any models to yield a sequence of QBFs with polynomial size Q-Resolution proofs. We would expect these short proofs to use the refutation of the first-order formula itself as a basis, similar to the methods used in [14,23]. We briefly discuss why this approach fails for Q-Resolution.

Consider the tableau refutation in Fig. 1. The unification that closes the tableau suggests a strategy for 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. However, the strategy does not respect the order of the quantifier prefix. In the game representing 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 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 and is not sound

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

$$\downarrow$$

$$R(x_1, Y_1(x_1), z_1) \wedge \neg R(U_1(x_1, z_1), v_1, W_1(x_1, z_1, v_1))$$

$$\downarrow$$

$$R(x_2, Y_2(x_2), z_2) \wedge \neg R(U_2(x_2, z_2), v_2, W_2(x_2, z_2, v_2))$$

$$\downarrow$$

$$\neg R(U_1(x_1, z_1), v_1, W_1(x_1, z_1, v_1))$$

$$\downarrow \quad \textit{Unif}$$

$$R(x_2, Y_2(x_2), z_2)$$

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*: $x_2 \leftarrow U_1(x_1, z_1)$, $v_1 \leftarrow Y_2(x_2)$, $z_2 \leftarrow W_1(x_1, z_1, v_1)$. Through this substitution we have twin atoms $R(U_1(x_1, z_1), Y_2(x_2), W_1(x_1, z_1, v_1))$ and $\neg R(U_1(x_1, z_1), Y_2(x_2), W_1(x_1, z_1, v_1))$ which resolve to a contradiction. For more details on these tableau refutations please see [20].

in general. Our strategy for Delayer shows that this problem cannot be overcome in tree-like Q-Resolution with the proposed translation from the first-order formula to QBF. Instead, we will modify the translation to provide Prover with a mechanism for 'remembering' choices that have previously been made, while still respecting the rules of the game. Finally, we show that $\forall$Exp+Res is able to use the unification to construct a valid strategy and a short proof on the first, more natural, translation to QBF.

6. Embellishing the QBFs

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

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

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:

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

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 $S(\alpha, \beta, \gamma, a) = 1$. Forget u.
3. Set $x = a$ since $S(\alpha, \beta, \gamma, a) = 1$. Set also $x'' = \alpha$, $y'' = \beta$, $z'' = \gamma$, $u'' = a$.
4. Query y. Suppose $y = b$. Set $v = b$ to match.
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. Forget w
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 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

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

with atoms

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

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.

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

The sets X'_i and X''_i are copies of the set X_i. Since the sets X'_i and X_i do not appear together in any $\mathcal{D}_i$, there is some flexibility in how this additional statement may be converted to prenex form. We perform the prenexing such that:

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

Now the conjunction of the two parts can be 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.

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.

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 $n^{O(1)}$.*

Proof: Taking an analytic tableau refutation [20] 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 subsentence

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

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.

If instead some $S(\zeta_1, \eta_1, \dots, \zeta_i, \eta_i)$ is set true, then any remaining $S(\zeta_1, \eta_1, \dots, \zeta_i, \eta_i)$ 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.

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 $O(n^m \cdot n^{b^2})$. □

7. Extension to QU-Resolution

Although stated in terms of tree-like 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 Prover to query universally quantified variables as well as existentially quantified [9]. This may shorten the refutation, since it offers a way for 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 [24] via the formulas of Kleine Büning, Karpinski and Flögel [19]. 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.

8. A Polynomial Upper Bound for ∀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, which can be understood in the game description as Prover lacking memory of previous answers. We show that tree-like ∀Exp+Res does not share this weakness and enjoys short proofs for QBFs generated by our first translation whenever the underlying first-order formula has no models. Thus we have given both a specific example and a schema of QBFs separating tree-like ∀Exp+Res from tree-like Q-Resolution.

Theorem 8. *Let ϕ be a first-order sentence without any models. Then the sequence of QBFs $\langle \Phi_n \rangle$ have tree-like refutations in ∀Exp+Res of size $n^{O(1)}$.*

Proof: The idea is to use the unification witnessing that ϕ is false to structure the ∀Exp+Res proof. The unification is made up of assignments from the value of a Skolem function to a universal variable. The Skolem function represents an existential variable x and is evaluated for a setting of (a subset of) the universal variables prior x in the quantifier prefix.

We can mimic this process in ∀Exp+Res using the game description for Resolution [11] over annotated variables. Prover queries the value of x with an annotation α. If a variable in the domain of the Skolem function is appearing in the unification for the first time it can be set arbitrarily in α. Otherwise it is given the value already specified earlier in the unification. Because x has been split into $x^1 \dots x^n$ by the translation to QBF Prover will in fact query some or all of the $x^{i,\alpha}$ until one of them is made true. Then all other $x^{i,\alpha}$ would be forced to false so Prover can move on to the next assignment in the unification.

Once the assignments for the unification have been made Prover can query relational variables to quickly reach a contradiction. Each branch of the tableau contains two entries that are directly contradictory under (part of) the assignment given by the unification. For each branch in turn Prover queries the relations(s) that close the branch using the assignments determined in the first stage of the game. By construction, any sequence of Delayer answers results in an immediate contradiction, in particular some clause of the form $(\neg x_1^{\lambda_1} \vee \dots \vee x_k^{\lambda_k} \vee \mathcal{D}_i(\lambda_1, \dots, \lambda_k))$ is falsified.

We need to show that Delayer scores $O(\lg(n))$ points. The number of queries of the relational variables does not depend on n. For each query Prover can select the value that gives Delayer the lowest score so each choice has a maximum value of 1 point (when $p_0 = p_1 = \frac{1}{2}$).

There are also constantly many Skolem functions in the unification, but each of these requires (up to) n queries to assign a value to an (annotated) existential variable. The number of points remains limited to $\lg(n)$ points.

To assign a value to x^α Prover first queries $x^{1,\alpha}$. Delayer responds with weights p_0 and p_1. If $p_1 < 1/n$ then set $x^{1,\alpha} = 1$ so Delayer scores $\lg(n)$ points. Otherwise set $x^{1,\alpha} = 0$. $p_0 < \frac{n-1}{n}$ so Delayer scores $\lg(\frac{n}{n-1})$ points.

Over $x^{i,\alpha}$ for $i = 1, \dots, j$ Delayer has either scored $\lg(n)$ points and some $x^{i,\alpha} = 1$ or has scored $\lg(n) - \lg(n-j)$ points and all $x^{i,\alpha} = 0$. If some $x^{i,\alpha} = 1$ then we are done and Prover does not need to query $x^{j+1,\alpha}$. Otherwise Delayer sets p_0 and p_1, if $p_1 <= \frac{1}{n-j}$ then Prover sets $x^{j+1,\alpha} = 1$. Delayer scores at most $\lg(n-j)$ points for this, so a total of $\lg(n)$ points. If $p_1 > \frac{1}{n-j}$ then Prover sets $x^{j+1,\alpha} = 0$ and Delayer scores at most $\lg(\frac{n-j}{n-j-1})$ points for this, giving a total of $\lg(n) - \lg(n-j-1)$ points.

In total Delayer has scored $O(\lg(n))$ points, so the proof size is $n^{O(1)}$. □

9. 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 ∀-reduction must respect the order of variables in the prefix. This manifests when trying to construct short tree-like Q-Resolution proofs. Section 5 shows that this is in general not possible for the original FO translations, which is why we need to add additional structure to the translation (the embellishment of Section 6). This bypasses the constraint on ∀-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. Generating a series of QBFs 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 ∀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.

Finally, we remark that tree-like Resolution systems – both propositionally and for QBF – are rather weak calculi (which in particular are not strong enough to model solving paradigms such as (Q)CDCL [3,21]). It would be very interesting to explore whether similar gap theorems as shown here and previously in [14,23] can be obtained for stronger calculi, with dag-like (Q)-Resolution being a very interesting case. However, presently we only have quite limited knowledge on this (cf. [15] for some results on dag-like Resolution) with the case of general dag-like Q-Resolution completely unexplored.

Acknowledgement

Research was supported by grants from the Carl Zeiss Foundation and the John Templeton Foundation (grant no. 60842).

References

[1] O. Beyersdorff, Proof complexity of quantified Boolean logic – a survey, in: *Mathematics for Computation (M4C)*, M. Benini, O. Beyersdorff, M. Rathjen and P. Schuster, eds, World Scientific, Singapore, 2022, pp. 353–391.

[2] O. Beyersdorff, J. Blinkhorn and L. Hinde, Size, cost, and capacity: A semantic technique for hard random QBFs, *Logical Methods in Computer Science* **15**(1) (2019).

[3] O. Beyersdorff and B. Böhm, Understanding the relative strength of QBF CDCL solvers and QBF resolution, *Log. Methods Comput. Sci.* **19**(2) (2023).

[4] O. Beyersdorff, I. Bonacina, L. Chew and J. Pich, Frege systems for quantified Boolean logic, *J. ACM* **67**(2) (2020), 9:1–9:36. doi:10.1145/3381881.

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

[6] O. Beyersdorff, L. Chew and M. Janota, New resolution-based QBF calculi and their proof complexity, *ACM Transactions on Computation Theory* **11**(4) (2019), 26:1–26:42. doi:10.1145/3352155.

[7] O. Beyersdorff, L. Chew, M. Mahajan and A. Shukla, Feasible interpolation for QBF resolution calculi, *Logical Methods in Computer Science* **13** (2017).

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

[9] O. Beyersdorff, L. Chew and K. Sreenivasaiah, A game characterisation of tree-like Q-resolution size, *J. Comput. Syst. Sci.* **104** (2019), 82–101. doi:10.1016/j.jcss.2016.11.011.

[10] O. Beyersdorff, N. Galesi and M. Lauria, A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover–delayer games, *Information Processing Letters* **110**(23) (2010), 1074–1077. doi:10.1016/j.ipl.2010.09.007.

[11] O. Beyersdorff, N. Galesi and M. Lauria, A characterization of tree-like resolution size, *Information Processing Letters* **113**(18) (2013), 666–671. doi:10.1016/j.ipl.2013.06.002.

[12] O. Beyersdorff, M. Janota, F. Lonsing and M. Seidl, Quantified Boolean formulas, in: *Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications*, A. Biere, M. Heule, H. van Maaren and T. Walsh, eds, IOS Press, 2021, pp. 1177–1221.

[13] B. Böhm and O. Beyersdorff, Lower bounds for QCDCL via formula gauge, *J. Autom. Reason.* **67**(4) (2023), 35. doi:10.1007/s10817-023-09683-1.

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

[15] S.S. Dantchev and S. Riis, On relativisation and complexity gap, in: *Computer Science Logic (CSL), 12th Annual Conference of the EACSL*, M. Baaz and J.A. Makowsky, eds, Lecture Notes in Computer Science, Vol. 2803, Springer, 2003, pp. 142–154.

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

[17] I.P. Gent, P. Nightingale, A.G.D. Rowley and K. Stergiou, Solving quantified constraint satisfaction problems, *Artif. Intell.* **172**(6–7) (2008), 738–771. doi:10.1016/j.artint.2007.11.003.

[18] M. Janota and J. Marques-Silva, Expansion-based QBF solving versus Q-resolution, *Theoretical Computer Science* **577**(C) (2015), 25–42. doi:10.1016/j.tcs.2015.01.048.

[19] H. Kleine Büning, M. Karpinski and A. Flögel, Resolution for quantified Boolean formulas, *Inf. Comput.* **117**(1) (1995), 12–18. doi:10.1006/inco.1995.1025.

[20] R. Letz, *First-Order Tableau Methods*, Springer, Netherlands, Dordrecht, 1999, pp. 125–196.

[21] K. Pipatsrisawat and A. Darwiche, On the power of clause-learning SAT solvers as resolution engines, *Artif. Intell.* **175**(2) (2011), 512–525. doi:10.1016/j.artint.2010.10.002.

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

[23] S. Riis, A complexity gap for tree-resolution, *Computational Complexity* **10** (2001), 179–209. doi:10.1007/s00037-001-8194-y.

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