Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Dynamic Dependency Awareness for QBF*

Joshua Blinkhorn, Olaf Beyersdorff
School of Computing, University of Leeds, UK

Abstract

We provide the first proof complexity results for QBF dependency calculi. By showing that the reflexive resolution path dependency scheme admits exponentially shorter Q-resolution proofs on a known family of instances, we answer a question first posed by Slivovsky and Szeider (SAT 2014). Further, we introduce a new calculus in which a dependency scheme is applied dynamically. We demonstrate the further potential of this approach beyond that of the existing static system with an exponential separation.

1 Introduction

Proof complexity is the study of proof size in systems of formal logic. Since its beginnings the field has enjoyed strong connections to computational complexity [Cook and Reckhow, 1979; Buss, 2012] and bounded arithmetic [Cook and Nguyen, 2010], and has emerged in the past two decades as the primary means for the comparison of algorithms in automated reasoning.

Recent successes in that area, epitomised by progress in SAT solving, have motivated broader research into the efficient solution of computationally hard problems. Amongst them, the logic of quantified Boolean formulas (QBF) is an established field with a substantial volume of literature. QBF extends propositional logic with the addition of existential and universal quantification, and naturally accommodates more succinct encodings of problem instances. This gives rise to diverse applications in areas including conformant planning [Rintanen, 2007; Egly *et al.*, 2017], verification [Benedetti and Mangassarian, 2008], and ontologies [Kontchakov *et al.*, 2009].

It is fair to say that much of the early research into QBF solving [Zhang and Malik, 2002; Samulowitz and Bacchus, 2005; Giunchiglia *et al.*, 2006], and later the proof complexity of associated theoretical models [Beyersdorff *et al.*, 2017a; 2018; 2017b], was built upon existing techniques for SAT. For example, QCDCL [Giunchiglia *et al.*, 2009] is a major paradigm in QBF solving based on conflict-driven clause learning (CDCL [Silva and Sakallah, 1996]), *the* dominant paradigm for SAT. By analogy, the fundamental theoretical model of QCDCL, the calculus Q-resolution (Q-Res [Kleine Büning *et al.*, 1995]), is an extension of propositional resolution, the calculus that underpins CDCL. Given, however, that the decision problem for QBF is PSPACE-complete, it is perhaps unsurprising that the implementation of QCDCL presents novel obstacles for the practitioner, beyond those encountered at the level of propositional logic.

Arguably, the biggest challenge concerns the allowable order of variable assignments. In traditional QCDCL, the freedom to assign variables is limited according to a linear order imposed by the quantifier prefix. Whereas decision variables must be chosen carefully to ensure sound results, coercing the order of assignment to respect the prefix is frequently needlessly restrictive [Lonsing, 2012]. Moreover, limiting the choice adversely affects the impact of decision heuristics. In contrast, such heuristics play a major role in SAT solving [Silva, 1999; Shacham and Zarpas, 2003; Liang *et al.*, 2015; Moskewicz *et al.*, 2001], where variables may be assigned in an arbitrary order.

Dependency awareness, as implemented in the solver DepQBF [Lonsing and Biere, 2010], is a QBF-specific paradigm that attempts to maximise the impact of decision heuristics. By computing a *dependency scheme* before the search process begins, the linear order of the prefix is effectively supplanted by a partial order that better approximates the variable dependencies of the instance, granting the solver greater freedom regarding variable assignments. Use of the scheme is static; dependencies are computed only once and do not change during the search. Despite the additional computational cost incurred, empirical results demonstrate improved solving on many benchmark instances [Lonsing, 2012].

Dependency schemes themselves are tractable algorithms that identify dependency information by appeal to the syntactic form of an instance. From the plethora of schemes that have been proposed in the literature, two have emerged as principal objects of study. The *standard dependency scheme* ($\mathcal{D}^{\text{std}}$ [Samer and Szeider, 2009]), a variant of which is used by DepQBF, was originally proposed in the context of backdoor sets. This scheme uses sequences of clauses connected by common existential variables to determine a dependency relation between the variables of an instance. The *reflex-*

*This paper is an abridged version of a paper titled 'Shortening QBF Proofs with Dependency Schemes' that won the Best Paper Award at the SAT 2017 conference.

ive resolution path dependency scheme ($\mathcal{D}^{\text{rrs}}$ [Slivovsky and Szeider, 2016]) utilises the notion of a *resolution path*, a more refined type of connection introduced in [Van Gelder, 2011].

A solid theoretical model for dependency awareness was only recently proposed in the form of the calculus $\mathsf{Q}(\mathcal{D})$-Res [Slivovsky and Szeider, 2016], a parametrisation of Q-resolution by the dependency scheme $\mathcal{D}$. Whereas the body of work on $\mathsf{Q}(\mathcal{D})$-Res and related systems has focused on soundness [Slivovsky and Szeider, 2016; Peitl *et al.*, 2016; Beyersdorff and Blinkhorn, 2016], authors of all three papers have cited open problems in proof complexity. Indeed, prior to this paper there were no proof-theoretic results to support any claims concerning the potential of dependency schemes in the practice of QBF solving.

In this work, not only do we provide the first such results, we also demonstrate the potential of dependency schemes to further reduce the size of proofs if they are applied *dynamically*. We summarise our contributions below.

The first separations for dependency calculi. We use the well-known formulas of Kleine Büning et al. [Kleine Büning *et al.*, 1995] to prove the first exponential separation for $\mathsf{Q}(\mathcal{D})$-Res. We show that $\mathcal{D}^{\text{rrs}}$ can identify crucial independencies in these formulas, leading to short proofs in the system $\mathsf{Q}(\mathcal{D}^{\text{rrs}})$-Res. In contrast, we show that $\mathcal{D}^{\text{std}}$ cannot identify any non-trivial independencies, allowing us to lift the exponential lower bound for Q-Res [Kleine Büning *et al.*, 1995; Beyersdorff *et al.*, 2015] to $\mathsf{Q}(\mathcal{D}^{\text{std}})$-Res.

A model of dynamic dependency analysis. We propose the new calculus dyn-$\mathsf{Q}(\mathcal{D})$-Res that models the dynamic application of a dependency scheme in Q-resolution. The system employs a so-called 'reference rule' that allows new axioms, called reference clauses, to be introduced into the proof. The key insight is that the application of an assignment to an instance formula may allow the dependency scheme to unlock new independencies. As such, the reference rule alludes to an explicit refutation of the formula under an appropriate restriction, and is analogous to the recomputation of dependencies at an arbitrary point of the QCDCL procedure. We prove that dyn-$\mathsf{Q}(\mathcal{D})$-Res is sound whenever the dependency scheme $\mathcal{D}$ is fully exhibited.

Separation of static and dynamic systems. Our final contribution demonstrates that the dynamic application of dependency schemes can shorten Q-resolution proofs even further, yielding an exponential improvement over the static approach. Using a modification of the aforementioned formulas from [Kleine Büning *et al.*, 1995], we prove that dyn-$\mathsf{Q}(\mathcal{D}^{\text{rrs}})$-Res is exponentially stronger than $\mathsf{Q}(\mathcal{D}^{\text{rrs}})$-Res.

2 Preliminaries

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

A literal is a variable or its negation, a clause is a disjunction of literals, and a CNF is a conjunction of clauses. Throughout, we refer to a clause as a set of literals and to a CNF as a set of clauses. We typically write x for existential variables, u for universals, and z for either. For a literal l, we write $\text{var}(l) = z$ iff $l = z$ or $l = \neg z$, for a clause C we write $\text{vars}(C) = \{\text{var}(l) \mid l \in C\}$, and for a PCNF Φ we write $\text{vars}(\Phi)$ for the variables in the prefix of Φ.

A (partial) assignment δ to the variables of Φ is represented as a set of literals, typically denoted $\{l_1, \dots, l_k\}$, where literal z (resp. $\neg z$) represents the assignment $z \mapsto 1$ (resp. $z \mapsto 0$). The *restriction of* Φ *by* δ, denoted $\Phi[\delta]$, is obtained by removing from ϕ any clause containing a literal in δ, and removing the negated literals $\neg l_1, \dots, \neg l_k$ from the remaining clauses, while the variables of δ and their associated quantifiers are removed from the prefix $\mathcal{Q}$.

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

There exist a host of resolution-based QBF proof systems – see [Beyersdorff *et al.*, 2015] for a detailed account. *Q-resolution* (Q-Res) introduced in [Kleine Büning *et al.*, 1995] is the standard refutational calculus for PCNF. In addition to resolution over existential pivots, the calculus has a *universal reduction rule* which allows a clause C to be derived from $C \cup \{l\}$, provided $\text{var}(l)$ is a universal variable right of all existentials in C. Tautologies are explicitly forbidden; one may not derive a clause containing both z and $\neg z$.

For a QBF resolution system P, a P *derivation* of a clause C from a PCNF Φ is a sequence $C_1, \dots, C_m$ of clauses in which $C = C_m$, and each clause is either an axiom or is derived from previous clauses in the sequence using an inference rule. A *refutation* of Φ is a derivation of the empty clause from Φ. P is *complete* iff every false PCNF has a P refutation, and *sound* iff no true PCNF has a refutation.

A QBF proof system P *p-simulates* another Q iff each Q refutation can be transformed in polynomial time into a P refutation of the same formula [Cook and Reckhow, 1979].

QBF models. Let $\Phi = \mathcal{Q}_1 z_1 \cdots \mathcal{Q}_n z_n \cdot \phi$ be a PCNF over existential variables $V_\exists$ and universal variables $V_\forall$. A *model* f for Φ is a mapping from total assignments to $V_\forall$ to total assignments to $V_\exists$ that satisfies two conditions: (a) whenever α and α' agree on all universals left of a variable z_i, then $f(\alpha)$ and $f(\alpha')$ agree on all existential variables left of (and including) z_i; (b) for each α in the domain of f, $\alpha \cup f(\alpha)$ satisfies every clause $C \in \phi$ (that is, $C \cap (\alpha \cup f(\alpha)) \neq \emptyset$). A PCNF is true iff it has a model, otherwise it is false.

Axiom rule: $\mathrm{axiom}(\phi)$

$$\frac{}{C}$$

- C is a non-tautological clause in ϕ.

Reduction rule: $\mathrm{red}(C, l)$

$$\frac{C}{C \setminus \{l\}}$$

- literal l is universal.
- $(\mathrm{var}(l), x) \notin \mathcal{D}(\Phi)$ holds for each existential variable x in $\mathrm{vars}(C)$.

Resolution rule: $\mathrm{res}(C_1, C_2, x)$

$$\frac{C_1 \qquad C_2}{(C_1 \cup C_2) \setminus \{x, \neg x\}}$$

- variable x is existential.
- $x \in C_1$ and $\neg x \in C_2$.
- the resolvent is non-tautological.

Figure 1: The rules of Q($\mathcal{D}$)-Res . $\mathcal{D}$ is a dependency scheme and $\Phi = \mathcal{Q} \cdot \phi$ is a PCNF.

3 Static Dependencies in Q-Resolution

For the duration of this work, we deal only with the (in)dependence of existential variables on universal variables. For an arbitrary PCNF Φ, the *trivial dependency relation* captures the linear order of the quantifier prefix of Φ, and is given by $\mathcal{D}^{\text{trv}}(\Phi) = \{(u, x) \in \mathrm{vars}_\forall(\Phi) \times \mathrm{vars}_\exists(\Phi) \mid u <_\Phi x\}$. Formally, a *dependency scheme* $\mathcal{D}$ is a mapping from the set of all PCNFs that satisfies $\mathcal{D}(\Phi) \subseteq \mathcal{D}^{\text{trv}}(\Phi)$ for each PCNF Φ. The existence of a pair $(u, x) \in \mathcal{D}(\Phi)$ should be interpreted as 'existential x depends on universal u in Φ according to dependency scheme $\mathcal{D}$'. We say that $\mathcal{D}'$ is *at least as general* as $\mathcal{D}$ iff $\mathcal{D}'(\Phi) \subseteq \mathcal{D}(\Phi)$ for each PCNF Φ, and is *more general* if the inclusion is strict for some PCNF.

We give definitions of the standard ($\mathcal{D}^{\text{std}}$) and reflexive resolution path ($\mathcal{D}^{\text{rrs}}$) dependency schemes below.

Definition 1 [Samer and Szeider, 2009] *Let $\Phi = \mathcal{Q} \cdot \phi$ be a PCNF. The pair $(u, x) \in \mathcal{D}^{\text{trv}}(\Phi)$ is in $\mathcal{D}^{\text{std}}(\Phi)$ iff there exists a sequence of clauses $C_1, \dots, C_n \in \phi$ with $u \in \mathrm{vars}(C_1)$, $x \in \mathrm{vars}(C_n)$, such that, for each $i \in [n-1]$, $\mathrm{vars}(C_i) \cap \mathrm{vars}(C_{i+1})$ contains an existential variable right of u.*

Definition 2 [Slivovsky and Szeider, 2016] *Let $\Phi = \mathcal{Q} \cdot \phi$ be a PCNF. The pair $(u, x) \in \mathcal{D}^{\text{trv}}(\Phi)$ is in $\mathcal{D}^{\text{rrs}}(\Phi)$ iff there is a sequence of clauses $C_1, \dots, C_n \in \phi$ and a sequence of existential literals $l_1, \dots, l_{n-1}$ for which the following four conditions hold:* (a) *$u \in C_1$ and $\neg u \in C_n$;* (b) *$x = \mathrm{var}(l_i)$, for some $i \in [n-1]$;* (c) *$u <_\Phi \mathrm{var}(l_i)$, $l_i \in C_i$ and $\neg l_i \in C_{i+1}$, for each $i \in [n-1]$;* (d) *$\mathrm{var}(l_i) \neq \mathrm{var}(l_{i+1})$ for each $i \in [n-2]$.*

The theoretical model for the use of dependency schemes in dependency-aware solving is captured by the calculus Q($\mathcal{D}$)-Res [Slivovsky and Szeider, 2016], shown in Fig. 1. Soundness of Q($\mathcal{D}$)-Res is not guaranteed, and hinges on the choice of the dependency scheme $\mathcal{D}$. It is known that the concept of *full exhibition*, which imposes a natural condition on $\mathcal{D}$, is sufficient to prove soundness in Q($\mathcal{D}$)-Res [Slivovsky, 2015]. Following [Beyersdorff and Blinkhorn, 2016], we say that a model f *exhibits the independence of x on u* iff, for each α in the domain of f, the assignment to x in $f(\alpha)$ remains unchanged when the assignment to u in α is flipped.

Definition 3 [Slivovsky, 2015; Beyersdorff and Blinkhorn, 2016] *A model f for a PCNF Φ is a $\mathcal{D}$-model iff, for each $(u, x) \in \mathcal{D}^{\text{trv}}(\Phi) \setminus \mathcal{D}(\Phi)$, f exhibits the independence of x on u. A dependency scheme $\mathcal{D}$ is* fully exhibited *iff each true PCNF has a $\mathcal{D}$-model.*

In the remainder of this section, we prove that Q($\mathcal{D}^{\text{rrs}}$)-Res is exponentially stronger than Q($\mathcal{D}^{\text{std}}$)-Res. Given that Q($\mathcal{D}^{\text{std}}$)-Res trivially p-simulates Q-Res, we thereby separate Q($\mathcal{D}^{\text{rrs}}$)-Res and Q-Res, thus answering the question initially posed in [Slivovsky and Szeider, 2016]. The separating formulas are a well-studied family of PCNFs, originally introduced in [Kleine Büning *et al.*, 1995]. We recall the definition of this formula family, which is hereafter denoted $\Psi(n)$.

Definition 4 [Kleine Büning *et al.*, 1995] *The formula family $\Psi(n) := \mathcal{Q}(n) \cdot \psi(n)$ has prefixes $\mathcal{Q}(n) := \exists x_1 \exists y_1 \forall u_1 \cdots \exists x_n \exists y_n \forall u_n \exists t_1 \cdots \exists t_n$ and matrices $\psi(n)$ consisting of the clauses*

$$\begin{array}{rcll} A & := & \{\neg x_1, \neg y_1\}, & \\ B_i & := & \{x_i, u_i, \neg x_{i+1}, \neg y_{i+1}\}, & i \in [n-1], \\ B'_i & := & \{y_i, \neg u_i, \neg x_{i+1}, \neg y_{i+1}\}, & i \in [n-1], \\ B_n & := & \{x_n, u_n, \neg t_1, \dots, \neg t_n\}, & \\ B'_n & := & \{y_n, \neg u_n, \neg t_1, \dots, \neg t_n\}, & \\ C_i & := & \{u_i, t_i\}, & i \in [n], \\ C'_i & := & \{\neg u_i, t_i\}, & i \in [n]. \end{array}$$

We first show that the standard dependency scheme cannot identify any non-trivial independencies for $\Psi(n)$.

Proposition 1 *For each $n \in \mathbb{N}$, $\mathcal{D}^{\text{std}}(\Psi(n)) = \mathcal{D}^{\text{trv}}(\Psi(n))$.*

The salient consequence of Prop. 1 is that every application of $\forall$-reduction in a Q($\mathcal{D}^{\text{std}}$)-Res derivation from $\Psi(n)$ is also available in Q-Res. As a result, the Q-Res lower bound for $\Psi(n)$ [Kleine Büning *et al.*, 1995; Beyersdorff *et al.*, 2015] lifts directly to Q($\mathcal{D}^{\text{std}}$)-Res.

Theorem 1 *The QBFs $\Psi(n)$ require exponential-size* Q($\mathcal{D}^{\text{std}}$)-Res *refutations.*

In contrast, the more general dependency scheme $\mathcal{D}^{\text{rrs}}$ can identify some crucial non-trivial independencies in $\Psi(n)$.

Proposition 2 *For each $n \in \mathbb{N}$ and for each $i, j \in [n]$, if $i \neq j$ then $(u_i, t_j) \notin \mathcal{D}^{\text{rrs}}(\Psi(n))$.*

According to Prop. 2, a Q($\mathcal{D}^{\text{rrs}}$)-Res refutation of $\Psi(n)$ may contain $\forall$-reduction steps that are disallowed in Q-Res. Such steps permit the construction of $O(n)$-size Q($\mathcal{D}^{\text{rrs}}$)-Res refutations.

Theorem 2 *The formulas $\Psi(n)$ have linear-size* Q($\mathcal{D}^{\text{rrs}}$)-Res *refutations.*

The following result is an immediate consequence of Theorems 1 and 2.

Theorem 3 Q($\mathcal{D}^{\text{rrs}}$)-Res *is exponentially stronger than* Q($\mathcal{D}^{\text{std}}$)-Res.

Reference rule: $\mathrm{ref}(\delta, \pi)$

$$\overline{C}$$

- δ is a $\mathcal{D}$-assignment for Φ.
- π is a dyn-Q($\mathcal{D}$)-Res refutation of $\Phi[\delta]$
- C is the largest falsified clause of δ.

Figure 2: The reference rule of dyn-Q($\mathcal{D}$)-Res. $\mathcal{D}$ is a dependency scheme and $\Phi = \mathcal{Q} \cdot \phi$ is a PCNF.

4 Modelling Dynamic Dependency Awareness

In this section, we introduce the dynamic dependency calculus dyn-Q($\mathcal{D}$)-Res and prove that it is exponentially stronger than Q($\mathcal{D}$)-Res when $\mathcal{D}$ is $\mathcal{D}^{\text{rrs}}$. We first define a particular kind of assignment to the variables of a PNCF that, in a clear sense, 'respects' the dependency scheme $\mathcal{D}$.

Definition 5 *Let $\mathcal{D}$ be a dependency scheme and let δ be a partial assignment to the variables of a PCNF Φ. Then δ is a* $\mathcal{D}$-assignment *for Φ iff, whenever δ assigns an existential variable x, then δ assigns all universal variables in the set $\{u \mid (u, x) \in \mathcal{D}(\Phi)\}$.*

We also define the *largest falsified clause* of an assignment.

Definition 6 *Let $\delta = \{l_1, \ldots, l_k\}$ be an assignment. The* largest falsified clause *of δ is $\{\neg l_1, \ldots, \neg l_k\}$.*

Definition of the calculus. We define dyn-Q($\mathcal{D}$)-Res as the proof system that has the rules of Q($\mathcal{D}$)-Res in addition to the *reference rule* shown in Fig. 2. On an intuitive level, the reference rule is based on the following fact: Given a PCNF Φ and a fully exhibited dependency scheme $\mathcal{D}$, if Φ is false under restriction by a $\mathcal{D}$-assignment δ, then adding the largest falsified clause of δ to the matrix of Φ preserves satisfiability. Therefore, if the calculus is capable of refuting $\Phi[\delta]$, it should be able to introduce the largest falsified clause of δ.

We refer to a clause derived by application of the reference rule as a *reference clause*. As stated in the rule itself, a reference clause may only be introduced if an explicit refutation π of $\Phi[\delta]$ can be given. This feature allows the size of a dyn-Q($\mathcal{D}$)-Res derivation to be suitably defined. We refer to π as a *referenced refutation*.

The reference degree of a dyn-Q($\mathcal{D}$)-Res derivation is 0 iff it does not contain any reference clauses (i.e. it is a Q($\mathcal{D}$)-Res derivation). For all other derivations π, the reference degree is $d+1$, where d is the largest reference degree of a refutation referenced from π.

The size of a dyn-Q($\mathcal{D}$)-Res derivation π of reference degree 0 is the number of clauses in the proof. The size of a derivation π with non-zero reference degree is $a + b$, where a is the number of clauses in π and b is the sum of the sizes of refutations referenced from π.

Similarly as for static systems, one can prove that full exhibition of $\mathcal{D}$ is sufficient for soundness in dyn-Q($\mathcal{D}$)-Res. The task is essentially reduced to proving that the reference clauses derived from a true PCNF are satisfied by a $\mathcal{D}$-model.

Theorem 4 *The calculus* dyn-Q($\mathcal{D}$)-Res *is sound if $\mathcal{D}$ is fully exhibited.*

Since $\mathcal{D}^{\text{rrs}}$ is fully exhibited [Beyersdorff and Blinkhorn, 2016] and $\mathcal{D}^{\text{rrs}}$ is at least as general as $\mathcal{D}^{\text{std}}$, it follows that dyn-Q($\mathcal{D}^{\text{std}}$)-Res and dyn-Q($\mathcal{D}^{\text{rrs}}$)-Res are both sound.

To show the separation of dyn-Q($\mathcal{D}^{\text{rrs}}$)-Res and Q($\mathcal{D}^{\text{rrs}}$)-Res, we introduce the formula family $\Xi(n)$, a modification of $\Psi(n)$ which employs the following operation.

Definition 7 *Let C be a clause and let ϕ be a CNF matrix. The clause-matrix product $C \otimes \phi$ is the CNF matrix with clauses $\{C \cup C' \mid C' \in \phi\}$.*

Definition 8 *Let $\Psi(n) := \mathcal{Q}(n) \cdot \psi(n)$ be the formulas of Kleine Büning et al. (as in Def. 4). The formula family $\Xi(n)$ has prefixes $\exists a \mathcal{Q}(n) \exists b$ and matrices*

$$(\{a, b\} \otimes \psi(n)) \cup (\{\neg a, \neg b\} \otimes \psi(n)) \cup \{\{a, \neg b\}, \{\neg a, b\}\}.$$

To prove the lower bound for the static calculus, one first shows that $\mathcal{D}^{\text{rrs}}(\Xi(n)) = \mathcal{D}^{\text{trv}}(\Xi(n))$, from which it follows that any Q($\mathcal{D}^{\text{rrs}}$)-Res refutation of $\Xi(n)$ is also a Q-Res refutation. Then one shows that any Q-Res refutation of $\Xi(n)$ contains an embedded refutation of $\Psi(n)$, which has exponential size [Kleine Büning *et al.*, 1995; Beyersdorff *et al.*, 2015].

Theorem 5 *The QBFs $\Xi(n)$ require exponential-size* Q($\mathcal{D}^{\text{rrs}}$)-Res *refutations.*

The upper bound argument makes use of the construction of short refutations from the proof of Theorem 2. By referencing those refutations, dyn-Q($\mathcal{D}^{\text{rrs}}$)-Res admits simple $O(n)$-size refutations of $\Xi(n)$.

Theorem 6 *The formulas $\Xi(n)$ have linear-size* dyn-Q($\mathcal{D}^{\text{rrs}}$)-Res *refutations.*

Our final result is immediate from Theorems 5 and 6.

Theorem 7 *The calculus* dyn-Q($\mathcal{D}^{\text{rrs}}$)-Res *is exponentially stronger than* Q($\mathcal{D}^{\text{rrs}}$)-Res.

5 Conclusions

We demonstrated that the use of dependency schemes in Q-resolution can yield exponentially shorter proofs, and thereby provided strong theoretical evidence supporting the notion that dependency schemes can be utilised for improved solving. We also demonstrated that the dynamic use of schemes has further potential, beyond that of the static approach in existing implementations.

We emphasize that, at the present time, there is no implementation for $\mathcal{D}^{\text{rrs}}$, and that DepQBF uses (a refinement of) $\mathcal{D}^{\text{std}}$. Since we do not separate Q-Res and Q($\mathcal{D}^{\text{std}}$)-Res, our theoretical results are not *exactly* in line with experimental results. We believe, however, that our results should be viewed as sound motivation for further research into (dynamic) dependency-aware solving.

Finally, we suggest strongly that the results in this paper will lift to further QBF calculi, and most notably to expansion-based systems. We therefore highlight the potential for dependency schemes in expansion solving, and endorse the move in this direction mooted at the conclusion of [Janota *et al.*, 2016].

References

[Benedetti and Mangassarian, 2008] Marco Benedetti and Hratch Mangassarian. QBF-based formal verification: Experience and perspectives. *JSAT*, 5(1-4):133–191, 2008.

[Beyersdorff and Blinkhorn, 2016] Olaf Beyersdorff and Joshua Blinkhorn. Dependency schemes in QBF calculi: Semantics and soundness. In *CP*, pages 96–112, 2016.

[Beyersdorff *et al.*, 2015] Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *STACS*, pages 76–89, 2015.

[Beyersdorff *et al.*, 2017a] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Logical Methods in Computer Science*, 13, 2017.

[Beyersdorff *et al.*, 2017b] Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. *Journal of Computer and System Sciences*, 2017. in press.

[Beyersdorff *et al.*, 2018] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is *not* so simple. *ACM Transactions on Computational Logic*, 19(1):1:1–1:26, 2018.

[Buss, 2012] Samuel R. Buss. Towards NP-P via proof complexity and search. *Annals of Pure and Applied Logic*, 163(7):906–917, 2012.

[Cook and Nguyen, 2010] Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, Cambridge, 2010.

[Cook and Reckhow, 1979] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[Egly *et al.*, 2017] Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. Conformant planning as a case study of incremental QBF solving. *Annals of Mathematics and Artificial Intelligence*, 80(1):21–45, 2017.

[Giunchiglia *et al.*, 2006] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Clause/term resolution and learning in the evaluation of quantified Boolean formulas. *Journal of Artificial Intelligence Research*, 26:371–416, 2006.

[Giunchiglia *et al.*, 2009] Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano. Reasoning with quantified Boolean formulas. In *Handbook of Satisfiability*, pages 761–780. IOS Press, 2009.

[Janota *et al.*, 2016] Mikoláš Janota, William Klieber, Joao Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. *Artificial Intelligence*, 234:1–25, 2016.

[Kleine Büning *et al.*, 1995] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Information and Computation*, 117(1):12–18, 1995.

[Kontchakov *et al.*, 2009] Roman Kontchakov, Luca Pulina, Ulrike Sattler, Thomas Schneider, Petra Selmer, Frank Wolter, and Michael Zakharyaschev. Minimal module extraction from DL-lite ontologies using QBF solvers. In *IJCAI*, pages 836–841. AAAI Press, 2009.

[Liang *et al.*, 2015] Jia Hui Liang, Vijay Ganesh, Ed Zulkoski, Atulan Zaman, and Krzysztof Czarnecki. Understanding VSIDS branching heuristics in conflict-driven clause-learning SAT solvers. In *Haifa Verification Conference*, pages 225–241, 2015.

[Lonsing and Biere, 2010] Florian Lonsing and Armin Biere. DepQBF: A dependency-aware QBF solver. *JSAT*, 7(2-3):71–76, 2010.

[Lonsing, 2012] Florian Lonsing. *Dependency Schemes and Search-Based QBF Solving: Theory and Practice*. PhD thesis, Johannes Kepler University, 2012.

[Moskewicz *et al.*, 2001] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Design Automation Conference*, pages 530–535, 2001.

[Peitl *et al.*, 2016] Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Long distance Q-resolution with dependency schemes. In *SAT*, pages 500–518, 2016.

[Rintanen, 2007] Jussi Rintanen. Asymptotically optimal encodings of conformant planning in QBF. In *AAAI*, pages 1045–1050. AAAI Press, 2007.

[Samer and Szeider, 2009] Marko Samer and Stefan Szeider. Backdoor sets of quantified Boolean formulas. *Journal of Automated Reasoning*, 42(1):77–97, 2009.

[Samulowitz and Bacchus, 2005] Horst Samulowitz and Fahiem Bacchus. Using SAT in QBF. In *CP*, pages 578–592, 2005.

[Shacham and Zarpas, 2003] Ohad Shacham and Emmanuel Zarpas. Tuning the VSIDS decision heuristic for bounded model checking. In *International Workshop on Microprocessor Test and Verification*, page 75, 2003.

[Silva and Sakallah, 1996] João P. Marques Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227, 1996.

[Silva, 1999] João P. Marques Silva. The impact of branching heuristics in propositional satisfiability algorithms. In *EPIA*, pages 62–74, 1999.

[Slivovsky and Szeider, 2016] Friedrich Slivovsky and Stefan Szeider. Soundness of Q-resolution with dependency schemes. *Theoretical Computer Science*, 612:83–101, 2016.

[Slivovsky, 2015] Friedrich Slivovsky. *Structure in #SAT and QBF*. PhD thesis, Vienna University of Technology, 2015.

[Van Gelder, 2011] Allen Van Gelder. Variable independence and resolution paths for quantified Boolean formulas. In *CP*, pages 789–803, 2011.

[Zhang and Malik, 2002] Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *ICCAD*, pages 442–449, 2002.