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 conformance 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 B"uning 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 (Dstd [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-
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 Q(\(D\))-Res. We show that \(D^{n*}\) can identify crucial independencies in these formulas, leading to short proofs in the system Q(\(D^{n*}\))-Res. In contrast, we show that \(D^{nd}\) 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 Q(\(D^{nd}\))-Res.

A model of dynamic dependency analysis. We propose the new calculus dyn-Q(\(D\))-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-Q(\(D\))-Res is sound whenever the dependency scheme \(D\) 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-Q(\(D^{n*}\))-Res is exponentially stronger than Q(\(D^{n*}\))-Res.

2 Preliminaries

Quantified Boolean formulas. In this paper, we consider quantified Boolean formulas (QBFs) in prenex conjunctive normal form (PCNF), typically denoted \(\Phi = Q \cdot \phi\). A PCNF over Boolean variables \(z_1, \ldots, z_n\) consists of a quantifier prefix \(Q = Q_1 z_1 \cdots Q_n z_n\), \(Q_i \in \{\exists, \forall\}\) for \(i \in [n]\), in which all variables are quantified either existentially or universally, and a propositional conjunctive normal form (CNF) formula \(\phi\) called the matrix. The prefix \(Q\) imposes a linear ordering \(z_i < z_j\) on the variables of \(\Phi\), such that \(z_i < z_j\) 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 \(\text{var}(l) = z\) if \(l = z\) or \(l = \neg z\), for a clause \(C\) we write \(\text{vars}(C) = \{\text{var}(l) \mid l \in C\}\), and for a PCNF \(\Phi\) we write \(\text{vars}(\Phi)\) for the variables in the prefix of \(\Phi\).

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 \(C_2 \cup \{-x\}\) (variable \(x\) is the pivot). Resolution is refutationally sound and complete: that is, the empty clause can be derived from a CNF if 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 \(\Phi\) is a sequence \(C_1, \ldots, C_m\) of clauses in which \(C = C_m\), and each clause is either an axiom or is derived from previous clauses in the sequence using an inference rule. A refutation of \(\Phi\) is a derivation of the empty clause from \(\Phi\). \(P\) 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 \(\Phi = Q_1 z_1 \cdots Q_n z_n \cdot \phi\) be a PCNF over existential variables \(V_2\) and universal variables \(V_1\). A model \(f\) for \(\Phi\) is a mapping from total assignments to \(V_2\) to total assignments to \(V_1\) that satisfies two conditions: (a) whenever \(\alpha\) and \(\alpha'\) 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 \(\alpha\) in the domain of \(f\), \(\alpha \cup f(\alpha)\) satisfies every clause \(C \in \phi\) (that is, \(C \cap (\alpha \cup f(\alpha)) = \emptyset\)). A PCNF is true iff it has a model, otherwise it is false.
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 $\Phi$, the trivial dependency relation captures the linear order of the quantifier prefix of $\Phi$, and is given by $D^{\text{tv}}(\Phi) = \{(u, x) \in \text{vars}_v(\Phi) \times \text{vars}_e(\Phi) \mid u <_{\Phi} x\}$. Formally, a dependency scheme $D$ is a mapping from the set of all PCNFs that satisfies $D(\Phi) \subseteq D^{\text{tv}}(\Phi)$ for each PCNF $\Phi$. The existence of a pair $(u, x) \in D(\Phi)$ should be interpreted as ‘existential $x$ depends on universal $u$ in $\Phi$ according to dependency scheme $D$’. We say that $D$ is at least as general as $D'$ iff $D'(\Phi) \subseteq D(\Phi)$ for each PCNF $\Phi$, and is more general if the inclusion is strict for some PCNF.

We give definitions of the standard (Dstd) and reflexive resolution path (Drrs) dependency schemes below.

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

Definition 2 [Slivovsky and Szeider, 2016] Let $\Phi = Q \cdot \phi$ be a PCNF. The pair $(u, x) \in D^{\text{tv}}(\Phi)$ is in $D^{\text{std}}(\Phi)$ iff there is a sequence of existential literals $l_1, \ldots, l_{n-1}$ for which the following four conditions hold: (a) $u \in C_1$ and $\neg u \in C_n$; (b) $x = \text{var}(l_i)$, for some $i \in [n-1]$; (c) $u <_{\phi} \text{var}(l_i)$, $l_i \in C_i$, and $\neg l_i$ in $C_{i+1}$, for each $i \in [n-1]$; (d) $\text{var}(l_i) \neq \text{var}(l_{i+1})$ 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(D)-Res [Slivovsky and Szeider, 2016], shown in Fig. 1. Soundness of Q(D)-Res is not guaranteed, and hinges on the choice of the dependency scheme $D$. It is known that the concept of full exhibition, which imposes a natural condition on $D$, is sufficient to prove soundness in Q(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 $\alpha$ in the domain of $f$, the assignment to $x$ in $f(\alpha)$ remains unchanged when the assignment to $u$ in $\alpha$ is flipped.

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

In the remainder of this section, we prove that $Q(D^{\text{rrs}})$-Res is exponentially stronger than $Q(D^{\text{std}})$-Res. Given that $Q(D^{\text{std}})$-Res trivially p-simulates Q-Res, we thereby separate $Q(D^{\text{rrs}})$-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 $\Psi(n) := Q(n) \cdot \psi(n)$ has prefixes $Q(n) := \exists x_1 \exists y_1 \forall u_1 \cdots \exists x_n \exists y_n \forall u_n \exists t_1 \cdots \exists t_n$ and matrices $\psi(n)$ consisting of the clauses $A := \{\neg x_1, \neg y_1\}$, $B_i := \{x_i, u_i, \neg x_{i+1}, \neg y_{i+1}\}$, $B_i' := \{y_i, \neg u_i, \neg x_{i+1}, \neg y_{i+1}\}$, $B_n := \{x_n, u_n, \neg t_1, \ldots, \neg t_n\}$, $B_n' := \{y_n, \neg u_n, \neg t_1, \ldots, \neg t_n\}$, $C_i := \{u_i, t_i\}$, $C_i' := \{\neg u_i, t_i\}$.

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

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

The salient consequence of Prop. 1 is that every application of $\forall$-reduction in a $Q(D^{\text{std}})$-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 $Q(D^{\text{std}})$-Res. Theorem 1 The QBFs $\Psi(n)$ require exponential-size $Q(D^{\text{std}})$-Res refutations.

In contrast, the more general dependency scheme $D^{\text{rrs}}$ 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 $(u_i, t_j) \notin D^{\text{rrs}}(\Psi(n))$.

According to Prop. 2, a $Q(D^{\text{rrs}})$-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 $Q(D^{\text{rrs}})$-Res refutations. Theorem 2 The formulas $\Psi(n)$ have linear-size $Q(D^{\text{rrs}})$-Res refutations.

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

Theorem 3 $Q(D^{\text{rrs}})$-Res is exponentially stronger than $Q(D^{\text{std}})$-Res.
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{rs}}. We first define a particular kind of assignment to the variables of a PCNF that, in a clear sense, ‘respects’ the dependency scheme \mathcal{D}.

**Definition 5** Let \mathcal{D} be a dependency scheme and let \delta be a partial assignment to the variables of a PCNF \Phi. Then \delta is a \mathcal{D}-assignment for \Phi iff, whenever \delta assigns an existential variable \(x\), then \delta 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 \delta 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 \Phi and a fully exhibited dependency scheme \mathcal{D}, if \Phi is false under restriction by a \mathcal{D}-assignment \delta, then adding the largest falsified clause of \delta to the matrix of \Phi preserves satisfiability. Therefore, if the calculus is capable of refuting \(\Phi[\delta]\), it should be able to introduce the largest falsified clause of \delta.

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 \(\pi\) 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 \(\pi\) 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 \pi, the reference degree is \(d + 1\), where \(d\) is the largest reference degree of a refutation referenced from \pi.

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

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}^{\text{rs}})-Res is sound if \(\mathcal{D}\) is fully exhibited.

Since \mathcal{D}^{\text{rs}} is fully exhibited [Beyersdorff and Blinkhorn, 2016] and \mathcal{D}^{\text{rs}} 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{rs}})-Res are both sound.

To show the separation of dyn-Q(\mathcal{D}^{\text{std}})-Res and Q(\mathcal{D}^{\text{rs}})-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 \(\phi\) 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 \(\Psi(n) := Q(n) \cdot \psi(n)\) be the formulas of Kleine Büning et al. (as in Def. 4). The formula family \(\Xi(n)\) has prefixes \(\exists a Q(n) \exists b\) and matrices \(\{(a, b) \otimes \psi(n)\} \cup \{(\neg a, -b) \otimes \psi(n)\} \cup \{(a, -b), \neg a, b\}\).

To prove the lower bound for the static calculus, one first shows that \(Q(n) = Q(n)\), from which it follows that any Q(\mathcal{D}^{\text{rs}})-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{rs}})-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{rs}})-Res admits simple O(\(n\))-size refutations of \(\Xi(n)\).

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

Our final result is immediate from Theorems 5 and 6.

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


