Electronic Colloquium on Computational Complexity, Report No. 28 (2016)

Dependency Schemes in QBF Calculi: Semantics and Soundness

Olaf Beyersdorff and Joshua Blinkhorn

School of Computing, University of Leeds, UK

Abstract. We study the parametrization of QBF resolution calculi by dependency schemes. One of the main problems in this area is to understand for which dependency schemes the resulting calculi are sound. Towards this end we propose a semantic framework for variable independence based on 'exhibition' by QBF models, and use it to define a property of dependency schemes called *full exhibition*. We prove that all CDCL-based resolution calculi, including Q-resolution, universal and long-distance Q-resolution, are sound when parametrized by a fully exhibited dependency scheme. To illustrate proof of concept, we show that the standard dependency scheme is fully exhibited.

1 Introduction

The excellent success of SAT solvers in the realm of propositional Boolean formulae has motivated much interest in the corresponding search problem for quantified Boolean formulae (QBF). The greater expressive capacity of QBF, afforded by its PSPACE-completeness [21], presents novel challenges in solving, and the array of emerging techniques is motivating a wealth of research in the closely-related field of proof complexity [3–8, 10–13].

This relationship between practice and theory consists in the fact that the trace of any unsuccessful run of a solver serves as a witness to falsity. Understanding the refutational proof system that underpins a particular solving method, and thereby accounts for its correctness, motivates the proof-theoretic study of specific calculi; most notable is the enormous interest in propositional resolution, the calculus which underpins *conflict-driven clause learning* (CDCL) for SAT (cf. [9]). Similarly for QBF, recent work has led to a complete understanding of the relative strength of resolution-based QBF calculi [3, 6], including Q-resolution ($\mathsf{Q\text{-}Res}$) [14], universal Q-resolution ($\mathsf{QU\text{-}Res}$) [12], and long-distance Q-resolution ($\mathsf{LD\text{-}Q\text{-}Res}$) [2].

The value of this research is twofold. Foremost, the twin notions of *simulation* and *separation* (by proof-size bounds) of proof systems admit an illuminating overview of relative strengths [6]. Additionally, the research spawns ideas for new and stronger calculi which in turn might be utilised for improved solving.

Implemented in the state-of-the-art solver DepQBF [15, 17], one of the recent and exciting developments in QBF solving has seen the introduction of *dependency schemes*: algorithms that gather information on variable independence

ISSN 1433-8092

by prior appeal to the syntactic form of an instance. Independence, however, is usually presented as a semantic concept [15, 18]: the truth of a QBF Φ is witnessed by a Skolem-function model, in which each existential x is identified with a Boolean function f_x, such that substitution for each x produces a propositional tautology. The arguments to f_x are the universal variables U_x left of x in the quantifier prefix, but it may occur that some circuit computes f_x without using $u \in U_x$ as an input. In this case we say that *x is independent of u* – and a dual notion for false QBFs provides for independence of universals on existentials – even though the Skolem-function model is in general not unique.

This lack of uniqueness has consequences for soundness in QBF calculi. The impact of a dependency scheme in the proof system is to allow some logical steps which previously were prohibited; specifically, the $\forall$-reduction rule of Q-Res receives greater reign. This motivated the proposal of Q($\mathcal{D}$)-Res by Slivovsky and Szeider [20], a parametrization of the classical calculus by dependency schemes.

For which schemes this calculus is sound is an open problem; soundness fails for the so-called 'optimal dependency scheme' $\mathcal{D}^{\text{opt}}$ (defined semantically), but an argument by transformation of refutation [20] suffices to prove that Q($\mathcal{D}$)-Res is sound for the reflexive resolution path scheme $\mathcal{D}^{\text{rrs}}$ (defined syntactically).

Nothing is currently known about the intermediate dependency schemes; moreover, the lack of general methods may frustrate future developments. It is natural to propose the parametrization by dependency schemes of stronger QBF calculi, of the other CDCL-based QBF resolution systems and QBF Frege [4], whereupon methods for proofs of soundness based on *properties* of dependency schemes will carry over.

In this paper we demonstrate that semantic notions of independence are indeed equipped for this; our contributions are summarized below.

1. New QBF calculi parametrized by dependency schemes. We extend the parametrisation by dependency schemes to all the CDCL-based resolution calculi for QBF: with the new long-distance calculus LD-Q($\mathcal{D}$)-Res, with universal resolution QU($\mathcal{D}$)-Res, and with their combination LQU($\mathcal{D}$)-Res.

Our new long-distance calculus presents the greatest challenge. The merged literal u^*, while accounting for the strength of long-distance techniques [11], is something of a semantic obstacle, as its entrance in a refutational proof gives no clue as to how it should be evaluated under assignment. Our key contribution here is the proposal of *merged literal functions* that provide the missing semantics, giving a clear account of the role of the merged literal. Progressing from Q-resolution, variable independence and merging have a more subtle interaction; in LD-Q($\mathcal{D}$)-Res, we must supplant merged literals with *annotated literals*, which record existential pivots to prevent unsound $\forall$-reduction steps. Accordingly, we introduce analogous *annotated literal functions* to complete our approach.

2. A semantic framework and a sufficient condition for soundness. We address the issue of soundness in QBF dependency calculi, and propose a semantic framework for variable independence that builds on previous literature [15,18]. We show that, for a given instance, the use of distinct QBF models to assert (or 'exhibit') different independencies can be problematic for logical consequence in the proof system. This motivates our key definition of *full exhibition*, which applies to a dependency scheme whenever a single model exhibits all the existential independencies for an arbitrary true QBF. In view of the departure of semantic entailment, we proceed to a general method in which the concept of full exhibition proves sufficient for soundness. Our argument is widely applicable: it provides soundness for all resolution-type calculi considered here, but also applies to much stronger QBF Frege systems [4] when parametrized by a fully exhibited dependency scheme.

3. Demonstrating full exhibition. We conclude the paper with proof of concept, and show that the standard dependency scheme is fully exhibited. Specifically, this is achieved by an algorithmic transformation of an arbitrary model M for a true QBF Φ into a model M^* that exhibits all the required independencies. This reveals the possibility for QBF solving to implement, for example, long-distance techniques in tandem with the standard dependency scheme, or indeed with any other scheme shown to be fully exhibited by an extension of our method.

2 Preliminaries

Quantified Boolean Formulas. A *Quantified Boolean Formula* (QBF) Φ over a set $V = \{v_1, \dots, v_n\}$ of n variables is a formula in quantified Boolean logic with quantifiers ranging over $\{0, 1\}$. We consider only formulas in *prenex conjunctive normal form* (PCNF), denoted $\Phi = \mathcal{Q} . \phi$, in which all variables are quantified either existentially or universally in the *quantifier prefix* $\mathcal{Q} = \mathcal{Q}_1 v_1 \cdots \mathcal{Q}_n v_n$, $\mathcal{Q}_i \in \{\exists, \forall\}$ for $i \in [n]$, and ϕ is a propositional conjunctive normal form (CNF) formula called the *matrix*. We typically write x_i for existential variables and u_i for universals. Where convenient, we refer to a clause as a set of literals and a matrix as a set of clauses.

For any QBF Φ, we denote the set of existentially quantified variables of Φ by $V_\exists = \{x_i \in V \mid \mathcal{Q}_i = \exists\}$, and define the set $V_\forall = \{u_i \in V \mid \mathcal{Q}_i = \forall\}$ of universal variables similarly. The prefix $\mathcal{Q}$ imposes a linear ordering on the variables of Φ that is captured by the binary relation $<_\Phi$, such that $v_i <_\Phi v_j$ holds whenever $i < j$, in which case we say that v_j *is right of* v_i, or that v_i *is left of* v_j. The set of variables right of v is denoted $R(v) = \{v' \in V \mid v <_\Phi v'\}$.

Assignment Trees and Models. In order to develop the semantic definition of independence, we make extensive use of assignment trees as defined in [15], and give a precise restatement as follows.

Given a QBF Φ over n variables $V = \{v_1, \dots, v_n\}$, a *complete assignment* to Φ is a total function $\alpha : V \to \{\top, \bot\}$, which maps each variable of Φ to a

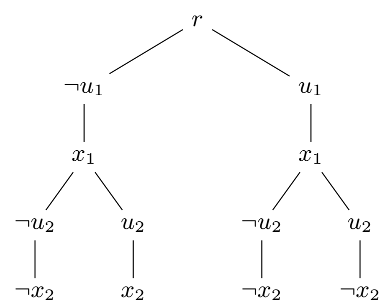

Fig. 1. An assignment tree T for a QBF $\forall u_1 \exists x_1 \forall u_2 \exists x_2 \,.\, \phi$, with arbitrary matrix ϕ.

truth value true $\top$ or false $\bot$. We may represent such an assignment α as a set of literals $\{l_1, \ldots, l_n\}$, where $\mathrm{var}(l_i) = v_i$ for each $i = 1, \ldots, n$, such that $l_i = v_i$ if $\alpha(v_i) = \top$ and $l_i = \neg v_i$ otherwise. We allow to specify variables and literals interchangeably as an argument to the assignment function, by defining $\alpha(\neg v) = \neg \alpha(v)$, where $\neg \top = \bot$, and vice versa.

An *assignment tree* T for Φ is a tree with root r in which every node except r is labelled with some literal l, with $\mathrm{var}(l) \in V$. Every node labelled with a universal literal has exactly one sibling labelled with the complementary literal, and every node labelled with an existential literal has no siblings. Every path P from r to a leaf of T defines a sequence of literals $\{l_1, \ldots, l_n\}$ which represents a total assignment to the variables of V, and which strictly acknowledges the linear ordering imposed by the prefix of Φ; that is, $\mathrm{var}(l_i) = v_i$ for all $i = 1, \ldots n$. We also use P to denote the assignment function corresponding to this sequence. When representing an assignment tree diagramatically, it is conventional to place the node labelled $\neg u$ to the left of that labelled u, for each $u \in V_\forall$, as shown in Fig. 1. Where convenient, we refer to an assignment tree as a set of paths. An assignment tree T for the QBF $\Phi = \mathcal{Q} \,.\, \phi$ is a *model* for Φ if $P(\phi) = \top$ for all paths P in T; that is every path in T satisfies every clause in the matrix of Φ. We typically use M to denote a model. A QBF which has a model is *true*, otherwise it is *false*.

Dependency Schemes. Closely following [20], a *proto-dependency scheme* $\mathcal{D}$ is a function that maps each QBF Φ to a binary relation $\mathcal{D}_\Phi$. The binary relation is a set of pairs of oppositely quantified variables[1] $\mathcal{D}_\Phi = \{(v_1, w_1), \ldots, (v_m, w_m)\}$ such that w_i is considered to depend on v_i, for each $i = 1, \ldots, m$. Such a pair is called a *dependency*, and w_i is a *dependent* of v_i. A proto-dependency scheme $\mathcal{D}'$ is said to be *at least as general* as another $\mathcal{D}$ (written $\mathcal{D}' \leq \mathcal{D}$) if $\mathcal{D}'_\Phi \subseteq \mathcal{D}_\Phi$ for all QBFs Φ, and is *strictly more general* (written $\mathcal{D}' < \mathcal{D}$) if the inclusion is strict for some formulae.

In practice, a proto-dependency scheme may include *spurious* dependencies (that is, pairs of variables that are in fact independent) which reduce the use-

[1] Similarly quantified variables are correctly considered independent.

fulness of the scheme, by inhibiting the strength of the calculus which uses it. Moreover, a proto-dependency scheme may also omit genuine dependencies.

There are two proto-dependency schemes of particular interest, which in a clear sense are the largest and smallest useful ones. The largest is the *trivial dependency scheme* $\mathcal{D}^{\text{trv}}$, which includes all the dependencies which are not explicitly ruled out by the linear ordering of the quantifier prefix; that is, $\mathcal{D}^{\text{trv}}_{\Phi} = \{(v, w) \mid v <_{\Phi} w\,, q(v) \neq q(w)\}$. The smallest is the *optimal dependency scheme* $\mathcal{D}^{\text{opt}}$, cf. [15], which includes only non-spurious dependencies by means of the rule $\mathcal{D}^{\text{opt}}_{\Phi} = \{(v, w) \mid v \text{ and } w \text{ are not independent}\}$. The definition of independence we use is given in Section 4 (Definition 2).

We restrict attention to those proto-dependency schemes which lie between these two extremes of usefulness; if a proto-dependency scheme $\mathcal{D}$ satisfies $\mathcal{D}^{\text{opt}} \leq \mathcal{D} \leq \mathcal{D}^{\text{trv}}$, we say that $\mathcal{D}$ is a *dependency scheme*.

QBF Resolution Calculi. We give a brief overview of four resolution-based CDCL QBF calculi, with the intention of augmenting them to utilise dependency schemes – for a more detailed survey of the original systems, see [6]. *Q-resolution* (Q-Res) introduced in [14] is the standard sound and complete refutational calculus for QBF in PCNF. In addition to resolution over existential pivots with non-tautologous resolvents, the calculus has a universal reduction rule

$$\frac{C \cup \{u\}}{C}\,,$$

where u is a universal literal, and all literals in C are left of u. *QU-resolution* (QU-Res) is a natural extension of Q-Res that allows universal resolution pivots, yielding a stronger calculus that is exponentially separated [12].

Long-distance Q-resolution (LD-Q-Res), formalised in [2], allows the introduction of a special merged literal u^*, denoting the appearance of complementary literals in u after a resolution step on an existential pivot x *which is left of* u:

$$\frac{C_1 \cup \{x\} \cup U_1 \qquad C_2 \cup \{\neg x\} \cup U_2}{C_1 \cup C_2 \cup U}$$

Here, U_1 and U_2 contain only universal literals with $\text{var}(U_1) = \text{var}(U_2)$, and for each $u \in \text{var}(U_1)$ we require $x <_{\Phi} u$. If for $l_1 \in U_1, l_2 \in U_2$, we have $\text{var}(l_1) = \text{var}(l_2) = u$, then $l_1 = \neg l_2$, or at least one of l_1, l_2 is already merged. The set U is defined as $\{u^* \mid u \in \text{var}(U_1)\}$. Merged literals do not prohibit $\forall$-reduction steps. The resulting system is exponentially stronger than Q-Res [11].

Finally, the calculus LQU$^+$-Res [3] combines naturally the features of QU-Res and LD-Q-Res, producing a system which introduces merged literals while allowing resolution over universal, as well as existential, pivots.

3 Dependency Schemes and Q-resolution

In this short section, we discuss the role of dependency schemes in QBF calculi, and recall the definition of Q($\mathcal{D}$)-Res [20], the first QBF dependency calculus to be put forward in the literature.

$$\frac{}{C}\ \text{(Axiom)}$$

C is a clause in the matrix of Φ.

$$\frac{D \cup \{l_u\}}{D}\ (\forall\text{-Red})$$

Literal l_u is universal. If $l \in D$ and $\mathrm{var}(l) = v$, then $(u, v) \notin \mathcal{D}_\Phi$.

$$\frac{C_1 \cup \{v\} \qquad C_2 \cup \{\neg v\}}{C_1 \cup C_2}\ \text{(Res)}$$

If $l \in C_1$, then $\neg l \notin C_2$. In Q($\mathcal{D}$)-Res, variable v is existential; in QU($\mathcal{D}$)-Res, v is existential or universal.

Fig. 2. The rules of Q($\mathcal{D}$)-Res [20] and QU($\mathcal{D}$)-Res

It is natural to try to strengthen the classical QBF calculi using dependency schemes. The motivation for the 'dependency version' comes from identifying the implicit use of the trivial dependency scheme in the rules of a calculus. Such restrictions are inevitably imposed by the linear ordering of the quantifier prefix; dependency calculi can relax these restrictions, replacing the implicit reference to $\mathcal{D}^{\text{trv}}$ with an explicit reference to a more general dependency scheme $\mathcal{D}$.

For example, in Q-Res, the universal reduction rule allows a universal u to be dropped from a clause C containing only variables left of u. In Q($\mathcal{D}$)-Res, u may be dropped whenever the clause contains none of its dependents, where the set of dependents of u is determined by $\mathcal{D}$.

We recall the formal definition of Q($\mathcal{D}$)-Res in Fig. 2, and observe that Q-Res and Q($\mathcal{D}^{\text{trv}}$)-Res are identical. It is natural to suggest relaxing the condition that the pivot is existential, and to allow resolution over universal pivots. Indeed, the resulting new system, which we call QU($\mathcal{D}$)-Res, is the analogous dependency version of QU-Res. We omit its formal definition, which is a simple modification to that of Q($\mathcal{D}$)-Res.

4 A Semantic Framework for Independence

In this section we extend the framework for semantic notions of independence in terms of assignment trees, and prove a condition sufficient for soundness in Q($\mathcal{D}$)-Res. Our key contribution is the notion of *full exhibition*.

4.1 The Genesis of Full Exhibition

In order to facilitate a clear understanding of independence as a concept, we feel it is helpful to adopt a fresh approach tailored to the task. To that end, we first introduce the new idea of *complementary paths* in an assignment tree, whose universal assignments differ for exactly one variable.

Definition 1 (Complementary path with respect to a universal variable). *Let Φ be a QBF over variables V, let T be an assignment tree for Φ and* *let P be the path in T determined by the total assignment to universal variables α. Then, for any $u \in V_\forall$, P^u is the path in T determined by the total assignment to universal variables α' given by*

$$\alpha'(v) = \begin{cases} \neg\alpha(v) & \text{if } v = u\,, \\ \alpha(v) & \text{otherwise}\,. \end{cases}$$

We may now reformulate the definition of independence from [15] so that it is better suited to our purposes. It is fortunate that, throughout this paper, we require only to consider the dependence of existentials on universals; this simplification, seen in the definition below, is the result of our dealing exclusively with refutational calculi, the rules of which remain unaffected by the dependence (or lack thereof) of universals on existentials.

Definition 2 (Independence of existentials from universals [15]). *Let Φ be a true QBF over variables V and let $u \in V_\forall$, $x \in V_\exists$. We say that x is independent of u in Φ if there exists a model M for Φ in which $P(x) = P^u(x)$ for all paths P in M. For such a model M we write $M \prec_\Phi (u, x)$, and we say that M* exhibits the independence *of x from u in Φ.*

Remark 3. It is not necessary for us to consider false QBFs in Definition 2, since, by definition, a false formula has no models. For a false formula, the condition for independence is satisfied vacuously, confirming our intuition that no existential variable can be dependent on any universal in such a formula.

As noted in [20], Definition 2 alone is too weak for soundness in $\mathsf{Q}(\mathcal{D})$-Res. The problem lies in the possibility for different models to exhibit different independencies, which are then used together in the same refutation. We illustrate this explicitly with the following formula, taken from [20], showing that $\mathsf{Q}(\mathcal{D}^{\mathsf{opt}})$-$\mathsf{Res}$ is unsound.

Example 4. Consider the true QBF $\Phi = \forall u_1 \forall u_2 \exists x \,.\, \{u_1, u_2, \neg x\}, \{\neg u_1, \neg u_2, x\}$. Figure 3 shows models $M_1 \prec (u_2, x)$ and $M_2 \prec (u_1, x)$ for Φ, and hence $\mathcal{D}^{\text{opt}}_\Phi$ is the empty set. In $\mathsf{Q}(\mathcal{D}^{\mathsf{opt}})$-$\mathsf{Res}$ one can therefore reduce the universal literals from both clauses, and then resolve over x to obtain the empty clause. Thus we have a $\mathsf{Q}(\mathcal{D}^{\mathsf{opt}})$-$\mathsf{Res}$ refutation of a true QBF.

In order to avoid this problem, it is natural to seek a model which exhibits simultaneously *all* the independencies used in a given proof. This is the intuition behind *full exhibition*, and motivates the following new definition which describes a key property for dependency schemes.

Definition 5 (Fully exhibited dependency scheme). *Let $\mathcal{D}$ be a dependency scheme. We say that $\mathcal{D}$ is* fully exhibited *if for each true QBF Φ there is a model M for Φ such that $M \prec (u, x)$ for each pair $(u, x) \notin D_\Phi$, with $u \in V_\forall$ and $x \in V_\exists$. For a given scheme $\mathcal{D}$ and formula Φ, any model satisfying this property is said to be* fully exhibiting.

As we will see, the power of this definition allows simple proofs of soundness for QBF dependency systems parametrized by fully-exhibited dependency schemes.

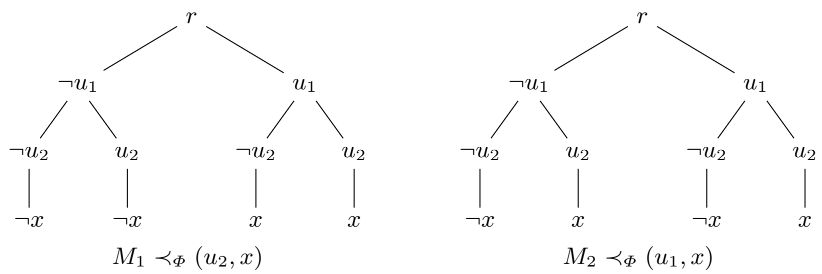

Fig. 3. Two models for the formula Φ exhibiting different independencies.

4.2 Soundness of Q($\mathcal{D}$)-Res and QU($\mathcal{D}$)-Res for Fully-Exhibited $\mathcal{D}$

As a first example of the application of Definition 5, we prove the following theorem.

Theorem 6. *Let $\mathcal{D}$ be a fully exhibited dependency scheme. Then* Q($\mathcal{D}$)-Res *is sound.*

Proof. Let $\Phi = \mathcal{Q} \,.\, \phi$ be a QBF over variables V, suppose that $\pi = \{C_1, \ldots, C_l\}$ is a Q($\mathcal{D}$)-Res refutation of Φ, and let

$$\phi_i = \begin{cases} \phi & \text{if } i = 0\,, \\ \phi \wedge C_1 \wedge \cdots \wedge C_i & \text{otherwise}\,, \end{cases}$$

for $i = 1, \ldots, l$.

Since $\mathcal{D}$ is fully exhibited, if Φ is true there exists a model M for Φ for which $M \prec_\Phi (u, x)$ for all pairs $(u, x) \notin D_\Phi$ with $u \in V_\forall$ and $x \in V_\exists$. We prove by induction on i that if Φ is true, M is a model for $\mathcal{Q} \,.\, \phi_i$, so $\mathcal{Q} \,.\, \phi_i$ is true. Hence at step $i = l$, we deduce that $\mathcal{Q} \,.\, \phi_l$ is true, a clear contradiction since ϕ_l contains the empty clause C_l.

Since $\mathcal{Q} \,.\, \phi = \mathcal{Q} \,.\, \phi_0$, if Φ is true then M is a model for $\mathcal{Q} \,.\, \phi_0$, thus the base for the induction is established.

We now show if M is a model for $\mathcal{Q}.\phi_i$, then M is a model for $\mathcal{Q} \,.\, \phi_{i+1}$. By definition, $\phi_{i+1} = (\phi_i \wedge C_{i+1})$, and C_{i+1} was introduced either by resolution or $\forall$-reduction. If C_{i+1} is the resolvent of clauses C_j, C_k for $j, k < i + 1$, then by hypothesis every path of M satisfies both C_j and C_k, and hence satisfies C_{i+1}. Thence M is a model for $\mathcal{Q} \,.\, \phi_{i+1}$.

On the other hand, suppose that C_{i+1} was obtained from $C_j, j < i + 1$, by $\forall$-reduction. Then $C_{i+1} = C_j \backslash \{u\}$, where $u \in V_\forall$ and $(u, x) \notin D_\Phi$ for all $x \in C_j \cap V_\exists$. Suppose that there exists some path P in M which satisfies C_j but falsifies C_{i+1}. Then $P(z) = \bot$ for all $z \in C_j$, and since $M \prec_\Phi (u, x)$ for all $(u, x) \notin D_\Phi$, we have $P^u(z) = P(z)$ for all literals $z \in C_j$. Also, since $P(u) = \top$, $P^u(u) = \bot$, and we deduce that $P^u(C_j) = \bot$, contradicting that M is a model for $\mathcal{Q}.\phi_i$. It follows that $P(u) = \top$ for some $u \in C_j$ with $\text{var}(u) \neq x$, and so every path in M satisfies C_{i+1}. Thence M is a model for $\mathcal{Q} \,.\, \phi_{i+1}$. $\square$

The reverse implication does not hold – namely it is not true that the soundness of Q($\mathcal{D}$)-Res implies that $\mathcal{D}$ is fully-exhibited – as evidenced by the following counterexample taken from the DQBF literature [1], reformulated in the language of QBF dependency schemes. Consider the formula $\Psi = \forall u_1 \forall u_2 \exists x_1 \exists x_2 \, . \, \psi$ with matrix

$$\psi \;=\; \big\{\{u_1, x_1, x_2\}, \{u_1, \neg x_1, \neg x_2\}, \{\neg u_1, u_2, x_1, x_2\}, \{\neg u_1, u_2, \neg x_1, \neg x_2\}, \{\neg u_1, \neg u_2, x_1, \neg x_2\}, \{\neg u_1, \neg u_2, \neg x_1, x_2\}\big\}\,,$$

and the dependency scheme $\mathcal{D}'$ defined by $\mathcal{D}'(\Phi) = \{(u_1, x_1), (u_2, x_2)\}$ if $\Phi = \Psi$ and $\mathcal{D}'(\Phi) = \mathcal{D}^{\text{trv}}(\Phi)$ otherwise. It can be verified that Ψ is true, but there is no model for Ψ which exhibits both independencies (u_1, x_2) and (u_2, x_2) simultaneously, and hence $\mathcal{D}'$ is not fully exhibited. However no Q($\mathcal{D}'$)-Res steps may be performed on clauses in ψ, so Q($\mathcal{D}'$)-Res remains sound.

Extension to QU($\mathcal{D}$)-Res. Since the proof of Theorem 6 makes no use of the fact that the pivot is existential, it suffices also to show the soundness of QU($\mathcal{D}$)-Res for fully exhibited $\mathcal{D}$.

Theorem 7. *Let $\mathcal{D}$ be a fully exhibited dependency scheme. Then* QU($\mathcal{D}$)-Res *is sound.*

The fact that the reverse implication does not hold is established by exactly the same counterexample Ψ above, which has no QU($\mathcal{D}'$)-Res refutation.

In fact, Theorem 7 even generalises further to far stronger QBF systems such as QBF Frege systems defined in [4]. We only need the correctness of the propositional rules other than the $\forall$-reduction rule in the proof of Theorem 6; hence the same argument also establishes the soundness of Frege + $\forall$red parametrized by a fully exhibited dependency scheme.

5 Dependency Schemes and Long-Distance Q-resolution

In this section, we define a new calculus LD-Q($\mathcal{D}$)-Res, the dependency version of long-distance Q-resolution, and prove that it is sound whenever $\mathcal{D}$ is fully exhibited.

5.1 Long-Distance Semantics

In the proof of Theorem 6, we validated the rules of Q($\mathcal{D}$)-Res by modifying the conventional argument by semantic entailment to use only a single fully exhibited model. Progressing to long-distance Q-resolution, we again use a single fully exhibited model M, but we validate logical steps on a path-by-path basis.

In this way, we provide a semantic account of the role of the merged literal; that is, we learn how to evaluate it under assignment. In each derived clause, the symbol u^*, by means of a *merged literal function*, represents a particular literal u or $\neg u$ for each path $P \in M$; as such, from the point of view of path P the proof itself contains no merged literals, but is still logically correct at the propositional level. The inevitable disappearance of u^* by $\forall$-reduction ties everything together, and the logical correctness for each individual path ensures the correctness at the level of the QBF model.

As a starting point, consider the following resolution step in an LD-Q-Res refutation of a QBF $\Phi = \mathcal{Q} \,.\, \phi$ over variables V, where $\mathrm{var}(l_u) = u \in V_\forall$ and $x \in V_\exists$.

$$\frac{C_1 \cup \{x\} \cup \{l_u\} \qquad C_2 \cup \{\neg x\} \cup \{\neg l_u\}}{C_1 \cup C_2 \cup \{u^*\}}$$

Let M be any model for the (conjunction of the) parent clauses prefixed by $\mathcal{Q}$, and let $P \in M$. For any universal v which is right of u, we must have $x <_\Phi u <_\Phi v$; therefore $P(x) = P^v(x)$, meaning that at least one of $C_1 \cup \{u\}$ and $C_2 \cup \{\neg u\}$ is satisfied by both P and P^v. In either case we can then choose a single literal u or $\neg u$ for u^* such that $C_1 \vee C_2 \vee u^*$ is satisfied by both P and P^v.

Generalising, let T be an assignment tree for Φ, and consider T as a set of paths. We observe that any resolution step producing u^* from complementary literals gives rise to a well-defined merged literal function $f^*_u : T \to \{u, \neg u\}$, with rule

$$f^*_u(P) = \begin{cases} l_u \,, & \text{if } P(x) = \bot \,, \\ \neg l_u \,, & \text{if } P(x) = \top \,. \end{cases} \tag{1}$$

For the time being, we just observe two features of such a definition. First, f^*_u simply reads the truth value of $P(x)$, selects the antecedent clause in which the pivot variable x is falsified, and takes the universal literal from that clause. In this way, any path $P \in T$ which satisfies both antecedent clauses is made to satisfy $C_1 \cup C_2 \cup f^*_u(P)$. Second, if v is any universal right of u, then $P(x) = P^v(x)$; hence v satisfies the *complementary property* $f^*_u(P) = f^*_u(P^v)$ for all $P \in T$.

To use these ideas more formally, we require a definition that accounts for 'successive merging' of merged literals with other merged literals, and with non-merged literals. We deal with this technicality in the next subsection, with a recursive extension that is equipped for the 'dependency version' of LD-Q-Res.

5.2 Defining LD-Q($\mathcal{D}$)-Res

We now introduce a new calculus LD-Q($\mathcal{D}$)-Res, which is the 'dependency version' of long-distance Q-resolution, and state the rules in Fig. 4.

Although the method of generalising the reference to the trivial dependency scheme remains, more care must be taken when defining LD-Q($\mathcal{D}$)-Res. Replacing the resolution condition $x <_\Phi u$ with $(u, x) \notin \mathcal{D}_\Phi$, one must annotate the merged literals with the corresponding existential pivots, and check during $\forall$-reduction that the reduced universal is independent of all the pivots for any annotated literals in the clause. This condition is always satisfied in LD-Q-Res, since reduced universals are always right of any merged variables in the clause, which are always right of any relevant existential pivots. However, harnessing independence opens up the potential to allow a universal variable u to be reduced from a clause containing a literal merged over a pivot x *which is not independent of* u. A simple counterexample [2] shows that reducing u in such a case is in general unsound for a fully-exhibited dependency scheme, and is not allowable in LD-Q($\mathcal{D}$)-Res.

$$\frac{}{C}\text{ (Axiom)}$$

C is a clause in the matrix of Φ.

$$\frac{D \cup \{l_u\}}{D}\text{ (}\forall\text{-Red)}$$

$$\frac{D \cup \{u^X\}}{D}\text{ (}\forall\text{-Red}^X\text{)}$$

Literals l_u and u^X are universal. If $l \in D$ and $\text{var}(l) = v$, then $(u, v) \notin \mathcal{D}_\Phi$, and if $l = v^{X'}$ then $(u, x) \notin \mathcal{D}_\Phi$ for all $x \in X'$.

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}\text{ (Res)}$$

Variable x is existential. If for $l_1 \in C_1, l_2 \in C_2, \text{var}(l_1) = \text{var}(l_2) = z$, then $l_1 = l_2 \neq z^*$. U_1, U_2 contain only universal literals with $\text{var}(U_1) = \text{var}(U_2)$. For each $u \in \text{var}(U_1)$ we require $(x, u) \notin \mathcal{D}_\Phi$. If for $u_1 \in U_1, u_2 \in U_2, \text{var}(u_1) = \text{var}(u_2) = u$, then $u_1 = \neg u_2$, or at least one of u_1, u_2 is annotated. U is defined as $\{u^X \mid u \in \text{var}(U_1)\}$, where X is the union of x with any annotations on u in U_1 and U_2.

Fig. 4. The rules of LD-Q($\mathcal{D}$)-Res

For any annotated literal u^X, we call X the *resolution set for* u. Non-merged literals have an empty resolution set. We give an example refutation below.

Example 8. Given the false QBF $\Phi = \forall u_1 \exists x_1 \exists x_2 \forall u_2 \,.\, \phi$, with matrix

$$\phi = \big\{\{u_1, x_1, x_2, u_2\}, \{x_1, \neg x_2, \neg u_2\}, \{\neg x_1, u_2\}\big\}$$

and a dependency scheme $\mathcal{D}$ for which $\mathcal{D}_\Phi = \{(u_1, x_1)\}$, we have the following LD-Q($\mathcal{D}$)-Res refutation.

$$\dfrac{\dfrac{\dfrac{\{u_1, x_1, x_2, u_2\} \qquad \{x_1, \neg x_2, \neg u_2\}}{\{u_1, x_1, u_2^{\{x_2\}}\}}\text{ A}}{\{x_1, u_2^{\{x_2\}}\}}\text{ B} \qquad \{\neg x_1, u_2\}}{\dfrac{\{u_2^{\{x_1, x_2\}}\}}{\bot}}\text{ C}$$

[2] To save space, we omit this counterexample.

Step A, producing an annotated literal $u_2^{\{x_2\}}$, is justified since $(u_2, x_2) \notin \mathcal{D}_\Phi$, and dropping u_1 in the presence of this annotated literal in step B is allowed since $(u_1, x_2) \notin \mathcal{D}_\Phi$. Step C is an example of successive merging.

Motivated by discussion of merged literal functions for LD-Q-Res in the previous subsection, we present a recursive definition of *annotated literal functions* in LD-Q($\mathcal{D}$)-Res that accounts for successive merging.

Definition 9 (Annotated literal functions for an assignment tree T). *Let $u^X \in U$ be a literal introduced by merging universal literals $l_1 \in U_1$ and $l_2 \in U_2$ in a resolution step*

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}$$

of an LD-Q($\mathcal{D}$)-Res *refutation of a formula Φ. Let X_1, X_2 be the resolution sets of l_1, l_2 respectively, and let T be an assignment tree for Φ. Then the annotated literal function $f_u^X : T \to \{u, \neg u\}$ for T is given by*

$$f_u^X(P) = \begin{cases} l_1\,, & \text{if } P(x) = \bot \text{ and } X_1 = \emptyset\,,\\ f_u^{X_1}(P)\,, & \text{if } P(x) = \bot \text{ and } X_1 \neq \emptyset\,,\\ l_2\,, & \text{if } P(x) = \top \text{ and } X_2 = \emptyset\,,\\ f_u^{X_2}(P)\,, & \text{if } P(x) = \top \text{ and } X_2 \neq \emptyset\,, \end{cases}$$

where $X = X_1 \cup X_2 \cup \{x\}$.

The following lemma states that the complementary property is preserved for annotated literal functions.

Lemma 10. *Let Φ be a QBF over variables V, let $u, v \in V_\forall$, let $X \subseteq V_\exists$ and let T be an assignment tree for Φ for which $T \prec (v, x)$ for all $x \in X$. Then any annotated literal function f_u^X for T satisfies $f_u^X(P) = f_u^X(P^v)$ for all paths $P \in T$.*

Proof. The lemma follows from the observation that, throughout the recursive definition of the annotated literal function f_u^X, complementary paths P and P^v always map to the same case, since $P(x) = P^v(x)$ for all $x \in X$ and for all $P \in T$. $\square$

Evaluation of annotated literals. We defined annotated literal functions for T specifically so that any $P \in T$ satisfying both antecedents of a resolution step also satisfies the resolvent. For that reason, we define u^X to have the same truth value as the concrete literal $f_u^X(P)$ when evaluated under α, the assignment represented by path P; that is, we define $(u^X)|_\alpha = (f_u^X(P))|_\alpha$. Representing assignments by paths, this would be written $P(u^X) = P(f_u^X(P))$, but we emphasize that the apparent circularity here is simply the result of the notational convenience of using the symbol P to represent both an assignment tree path and its corresponding assignment. The expression $P(f_u^X(P))$ is always well-defined because f_u^X can be computed for any given assignment tree T, so $f_u^X(P)$ is a well-defined non-annotated literal, which can then be evaluated under P in the usual way.

5.3 Soundness of LD-Q($\mathcal{D}$)-Res and LQU($\mathcal{D}$)-Res

We are now in a position to prove the following theorem.

Theorem 11. *Let $\mathcal{D}$ be a fully exhibited dependency scheme. Then* LD-Q($\mathcal{D}$)-Res *is sound.*

Proof. Let $\Phi = \mathcal{Q} . \phi$ be a QBF over variables V, suppose that $\pi = \{C_1, \dots, C_l\}$ is a LD-Q($\mathcal{D}$)-Res refutation of Φ, and let

$$\phi_i = \begin{cases} \phi & \text{if } i = 0\,, \\ \phi \wedge C_1 \wedge \cdots \wedge C_i & \text{otherwise}\,, \end{cases}$$

for $i = 1, \dots, l$. Since $\mathcal{D}$ is fully exhibited, if Φ is true there exists a model M for Φ for which $M \prec_{\Phi} (u, x)$ for all pairs $(u, x) \notin D_{\Phi}$ with $u \in V_{\forall}$ and $x \in V_{\exists}$. Our proof uses exactly the same induction on i as in the proof of Theorem 6, and the base case is no different. We only need confirm that if M is a model for $\mathcal{Q} . \phi_i$, then M is a model for $\mathcal{Q} . \phi_{i+1}$.

Suppose that $C_{i+1} = C_1 \cup C_2 \cup U$ is the resolvent of clauses $C_j = C_1 \cup U_1 \cup \{x\}$ and $C_k = C_2 \cup U_2 \cup \{\neg x\}$ for $j, k < i + 1$, and let P be an arbitrary path in M. By the inductive hypothesis, P satisfies C_j and C_k. Assume without loss of generality that $P(x) = \bot$. Then P satisfies $C_1 \cup U_1$. If P satisfies C_1 then P satisfies C_{i+1}. Otherwise, $P(l_u) = \top$ for some (annotated or non-annotated) literal $l_u \in U_1$ with $\mathrm{var}(l_u) = u$. The recursive definition of the annotated literal function ensures that $P(l_u) = \top \Rightarrow P(u^X) = \top$ for some annotated literal $u^X \in U$, and so P satisfies C_{i+1}. Therefore M is a model for $\mathcal{Q} . \phi_{i+1}$.

On the other hand, suppose that C_{i+1} was obtained from $C_j, j < i + 1$, by $\forall$-reduction on a non-annotated universal literal u. Then $C_{i+1} = C_j \backslash \{l_u\}$, where $\mathrm{var}(l_u) = u$, $(u, x) \notin \mathcal{D}_{\Phi}$ for all $x \in C_j \cap V_{\exists}$ and for all $x \in X$, where X is the union of the resolution sets of all universal literals in C_j. Suppose that there exists some path P in M which satisfies C_j but falsifies C_{i+1}. Let $z \in C_{i+1}$; then $P(z) = \bot$, and since $M \prec_{\Phi} (u, x)$ for all $(u, x) \notin \mathcal{D}_{\Phi}$, we have $P^u(z) = P(z) = \bot$ whenever z is a non-annotated literal. On the other hand, suppose that $z = v^X$, where $v \neq u$; then, since $M \prec_{\Phi} (u, x)$ for all $x \in X$, Lemma 10 gives $f_v^X(P) = f_v^X(P^u) = l_v$ with $\mathrm{var}(l_v) = v$, which implies $P^u(v^X) = P(v^X) = \bot$. Also, since $P(u) = \top$, we have $P^u(u) = \bot$, and we deduce that $P^u(C_j) = \bot$, contradicting that M is a model for $\mathcal{Q}.\phi_i$. It follows that P satisfies C_{i+1}, and that M is a model for $\mathcal{Q} . \phi_{i+1}$.

The same argument follows for an annotated literal u^X. Since $M \prec_{\Phi} (u, x)$ for all $x \in X$, the special case of Lemma 10 with $v = u$ gives $f_u^X(P) = f_u^X(P^u)$, hence $P(u^X) = \top$ implies $P^u(u^X) = \bot$. This completes the proof. □

Extension to LQU($\mathcal{D}$)-Res. Since the proof of Theorem 11 makes no use of the fact that the pivot is existential, it suffices also to show the soundness of LQU($\mathcal{D}$)-Res, the 'dependency version' of LQU$^+$-Res, for fully exhibited $\mathcal{D}$. We omit the formal definition of the system, which differs only from that of LD-Q($\mathcal{D}$)-Res in allowing universal resolution pivots.

Theorem 12. *Let $\mathcal{D}$ be a fully exhibited dependency scheme. Then* $\mathsf{LQU}(\mathcal{D})$-$\mathsf{Res}$ *is sound.*

6 Demonstrating Full Exhibition

In this section, to demonstrate proof of concept, we prove that the standard dependency scheme $\mathcal{D}^{\text{std}}$ is fully exhibited. First, we restate its definition.

Definition 13 (Standard dependency scheme [16, 19]). *Let $\Phi = \mathcal{Q} \,.\, \phi$ be a QBF over variables V, and let $v, w \in V$ such that $v <_{\Phi} w$. We say that v and w are* connected *in Φ if there exists a set of clauses $C_1, \ldots, C_k \in \phi$ such that $v \in C_1$, $w \in C_k$ and $C_i \cap C_{i+1} \cap X \neq \emptyset$ for $i = 1, \ldots, k-1$, where $X = R(v) \cap V_{\exists}$. The standard dependency scheme $\mathcal{D}^{std}$ is defined by*

$$\mathcal{D}^{std} : \Phi \mapsto \{(v, w) \mid v \text{ and } w \text{ are connected in } \Phi\,,\, q(v) \neq q(w)\}\,.$$

We proceed by showing the correctness of an algorithm which transforms an arbitrary model M of some true QBF Φ into a model M^*, where $M^* \prec (u, x)$ for all $(u, x) \notin \mathcal{D}^{\text{std}}_{\Phi}$. Figure 5 introduces the procedure `Flip()` which modifies M so that $P(x) = P^u(x)$ for some given path P, existential x and universal u.

1 `Flip` (P, u, X)
2 {
3   **for each** $x \in X$ **do** flip x in P
4   $X' \leftarrow \emptyset$
5   **for each** $C \in \phi$ **such that** $P(C) = \bot$ **do**
6   **for each** $v \in C$ **such that** $P(v) \neq P^u(v)$ **do**
7   **push** $v \to X'$
8   **if** $X' \neq \emptyset$ **then** `Flip` (P, u, X')
9 }

Fig. 5. The flipping procedure

Theorem 14. *The standard dependency scheme is fully exhibited.*

Proof. We fix a true QBF Φ and an arbitrary model M for Φ. Suppose we wish to modify M so that $M \prec (u, x)$ for some specific $(u, x) \notin \mathcal{D}^{\text{std}}_{\Phi}$, and to that end we identify some pair of paths P and P^u for which $P(x) \neq P^u(x)$ and then invoke `Flip`$(P, u, \{x\})$. On line 3, x is flipped in P and on line 5 the procedure checks for any clauses that P now falsifies. Any falsified clause C must contain x but cannot contain u due to $(u, x) \notin \mathcal{D}^{\text{std}}_{\Phi}$; also C must contain some variable v which P and P^u assign oppositely, and v must therefore be existential. All such existentials are added to the set X' in line 7. If no clauses were falsified, the procedure terminates on line 8 and M remains a model; otherwise the procedure repeats by flipping X' and checking again for falsified clauses. At any recursion depth, the collected set X' still contains only existentials, since the appearance of u would always define a chain of clauses implying that $(u, x) \in \mathcal{D}^{\text{std}}_{\Phi}$. Similarly, we observe that no flipped variable may be connected to u, hence

$$x' \text{ is flipped in } \texttt{Flip}(P, u, \{x\}) \Rightarrow (u, x') \notin \mathcal{D}^{\text{std}}_{\Phi} . \tag{2}$$

It is clear that `Flip()` yields a model if it terminates; the fact that it always terminates follows from the fact that no variable may be flipped twice.

Now we point out that using `Flip()` alone, we can transform M into a model M' where $M' \prec (u, x)$ for all $u \in U_x = \{u \mid (u, x) \notin \mathcal{D}^{\text{std}}_{\Phi}\}$; that is, M' exhibits all independencies associated with x. To see this, note that a model can be naturally partitioned into sets of paths, each of which is closed under the operation of taking the complementary path with respect to $u \in U_x$. Then any non-conformity of the assignment of x in a particular set can be erased by repeated application of `Flip()`.

We now describe how to obtain the desired model M^* which exhibits all the independencies. Suppose that the existential variables of Φ in prefix order are $x_1, \ldots, x_n$. Given an arbitrary model M, use the method described in the previous paragraph to obtain M_n which exhibits all the independencies associated with the rightmost existential x_n. Now repeat the process to obtain from M_n a model M_{n-1} which exhibits all the independencies associated with x_{n-1}. We observe that M_{n-1} preserves the exhibition of the independencies associated with x_n; due to condition (2), x_n cannot be flipped in obtaining M_{n-1} from M_n, since whenever $(u, x_n) \notin \mathcal{D}^{\text{std}}_{\Phi}$ we already have $P(x_n) = P^u(x_n)$. Continuing in this way, each model M_i exhibits all the independencies corresponding to $x_i, \ldots, x_n$, and at the n^{th} step we obtain the required model $M^* = M_1$. □

Our concluding result now follows immediately from Theorems 7, 11 and 12.

Corollary 15. QU($\mathcal{D}^{\text{std}}$)-Res, LD-Q($\mathcal{D}^{\text{std}}$)-Res *and* LQU($\mathcal{D}^{\text{std}}$)-Res *are sound proof systems.*

7 Conclusions and Open Problems

As we have shown, the parametrization by dependency scheme can be extended to all four CDCL QBF calculi, and the property of full exhibition – which is possessed by the standard dependency scheme – is sufficient for soundness in each case. Showing by counterexample that full-exhibition is not a necessary condition, our work leads naturally to the open problem of finding a characterization for soundness in this setting. We also leave open the question of whether the reflexive resolution path dependency scheme $\mathcal{D}^{\text{rrs}}$ is fully exhibited, and whether our methods of model transformation from Sect. 6 can be furthered to provide this result.

Acknowledgments. This research was supported by grant no. 48138 from the John Templeton Foundation and EPSRC grant EP/L024233/1.

References

1. Balabanov, V., Chiang, H.K., Jiang, J.R.: Henkin quantifiers and boolean formulae: A certification perspective of DQBF. Theor. Comput. Sci. 523, 86–100 (2014), http://dx.doi.org/10.1016/j.tcs.2013.12.020
2. Balabanov, V., Jiang, J.R.: Unified QBF certification and its applications. Formal Methods in System Design 41(1), 45–65 (2012), http://dx.doi.org/10.1007/s10703-012-0152-6
3. Balabanov, V., Widl, M., Jiang, J.R.: QBF resolution systems and their proof complexities. In: Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings. pp. 154–169 (2014), http://dx.doi.org/10.1007/978-3-319-09284-3_12
4. Beyersdorff, O., Bonacina, I., Chew, L.: Lower bounds: From circuits to QBF proof systems. In: ACM Conference on Innovations in Theoretical Computer Science (ITCS'16). pp. 249–260. ACM (2016)
5. Beyersdorff, O., Chew, L., Janota, M.: On unification of QBF resolution-based calculi. In: Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II. pp. 81–93 (2014), http://dx.doi.org/10.1007/978-3-662-44465-8_8
6. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. In: Mayr, E.W., Ollinger, N. (eds.) 32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 30, pp. 76–89. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2015), http://drops.dagstuhl.de/opus/volltexte/2015/4905
7. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Feasible interpolation for QBF resolution calculi. In: Proc. International Colloquium on Automata, Languages, and Programming (ICALP'15). pp. 180–192. Springer (2015)
8. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Are short proofs narrow? QBF resolution is not simple. In: Proc. Symposium on Theoretical Aspects of Computer Science (STACS'16) (2016)
9. Buss, S.R.: Towards NP-P via proof complexity and search. Ann. Pure Appl. Logic 163(7), 906–917 (2012)
10. Egly, U.: On sequent systems and resolution for QBFs. In: Theory and Applications of Satisfiability Testing (SAT'12). pp. 100–113 (2012)
11. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings. pp. 291–308 (2013), http://dx.doi.org/10.1007/978-3-642-45221-5_21
12. Gelder, A.V.: Contributions to the theory of practical quantified boolean formula solving. In: Principles and Practice of Constraint Programming - 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings. pp. 647–663 (2012), http://dx.doi.org/10.1007/978-3-642-33558-7_47
13. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. Theor. Comput. Sci. 577, 25–42 (2015)
14. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified boolean formulas. Inf. Comput. 117(1), 12–18 (1995), http://dx.doi.org/10.1006/inco.1995.1025

15. Lonsing, F.: Dependency Schemes and Search-Based QBF Solving: Theory and Practice. Ph.D. thesis, Johannes Kepler University (2012)
16. Lonsing, F., Biere, A.: A compact representation for syntactic dependencies in qbfs. In: Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings. pp. 398–411 (2009), http://dx.doi.org/10.1007/978-3-642-02777-2_37
17. Lonsing, F., Egly, U.: Depqbf: An incremental QBF solver based on clause groups. CoRR abs/1502.02484 (2015), http://arxiv.org/abs/1502.02484
18. Samer, M.: Variable dependencies of quantified CSPs. In: Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings. pp. 512–527 (2008), http://dx.doi.org/10.1007/978-3-540-89439-1_36
19. Samer, M., Szeider, S.: Backdoor sets of quantified boolean formulas. J. Autom. Reasoning 42(1), 77–97 (2009), http://dx.doi.org/10.1007/s10817-008-9114-5
20. Slivovsky, F., Szeider, S.: Soundness of Q-resolution with dependency schemes. Theor. Comput. Sci. 612, 83–101 (2016), http://dx.doi.org/10.1016/j.tcs.2015.10.020
21. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time: Preliminary report. In: Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA. pp. 1–9 (1973), http://doi.acm.org/10.1145/800125.804029

ECCC ISSN 1433-8092

http://eccc.hpi-web.de