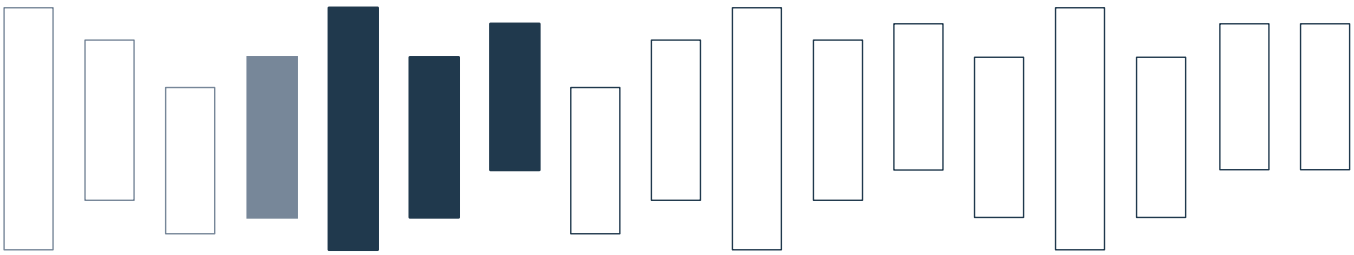
Technical Report AC-TR-20-011

September 2020

Fixed-Parameter Tractability of Dependency QBF with Structural Parameters

Robert Ganian, Tomáš Peitl, Friedrich Slivovsky, and Stefan Szeider



This is the authors' copy of a paper that will appear in the proceedings of KR2020, the 17th International Conference on Principles of Knowledge Representation and Reasoning.

www.ac.tuwien.ac.at/tr

Fixed-Parameter Tractability of Dependency QBF with Structural Parameters

Robert Ganian[1] , **Tomáš Peitl**[2] , **Friedrich Slivovsky**[1] , **Stefan Szeider**[1]
[1]TU Wien
[2]Friedrich-Schiller-Universität Jena
rganian@ac.tuwien.ac.at, tomas.peitl@uni-jena.de, {fslivovsky,sz}@ac.tuwien.ac.at

Abstract

We study *dependency quantified Boolean formulas* (*DQBF*), an extension of QBF in which dependencies of existential variables are listed explicitly rather than being implicit in the order of quantifiers. DQBF evaluation is a canonical NEXPTIME-complete problem, a complexity class containing many prominent problems that arise in Knowledge Representation and Reasoning. One approach for solving such hard problems is to identify and exploit structural properties captured by numerical parameters such that bounding these parameters gives rise to an efficient algorithm. This idea is captured by the notion of fixed-parameter tractability (FPT). We initiate the study of DQBF through the lens of fixed-parameter tractability and show that the evaluation problem becomes FPT under two natural parameterizations: the treewidth of the primal graph of the DQBF instance combined with a restriction on the interactions between the dependency sets, and also the treedepth of the primal graph augmented by edges representing dependency sets.

1 Introduction

Dependency quantified Boolean formulas (*DQBF*) are a powerful formalism that extends propositional and *quantified Boolean formulas* (*QBF*) by allowing an explicit specification of dependencies for individual existentially quantified variables, also known as *Henkin quantifiers* (Henkin 1961). The price of this additional syntactic tool is the NEXPTIME-completeness of DQBF evaluation (Peterson, Reif, and Azhar 2001), and as such DQBF evaluation is believed to be significantly more difficult than SAT and QBF evaluation, which are NP- and PSPACE-complete, respectively. As one of the canonical NEXPTIME-complete problems, DQBF can be used to succinctly encode problems for which no efficient encodings into QBF are likely to exist, such as synthesis of safe controllers (Bloem, Könighofer, and Seidl 2014), distributed synthesis for specifications in linear temporal logic (Chatterjee et al. 2013), consistency checking for certain description logics (Tobies 1999; Tobies 2001; Lutz 2001), checking the existence of secure plans (Eiter et al. 2004), and other problems that are central to knowledge representation and reasoning. We note that NEXPTIME-complete problems are provably intractable, while no such proofs for problems in the Polynomial or Boolean Hierarchies are known (Papadimitriou 1994).

In the past two decades we have seen, first in SAT solving and later in QBF as well, that these theoretically hard problems can often be successfully tackled in practice. A wide range of SAT and QBF solvers have been developed to date (Eén and Sörensson 2003; Audemard and Simon 2009; Biere and Lonsing 2010; Janota et al. 2012; Rabe and Tentrup 2015; Janota 2018; Peitl, Slivovsky, and Szeider 2019), some of which can solve instances with millions of variables, and progress in these areas has revolutionized application fields such as formal verification (Vizel, Weissenbacher, and Malik 2015). Inspired by this success story, researchers are now turning their attention even higher up the complexity hierarchy, to DQBF.

Indeed, research on DQBF is picking up steam in all directions—from investigations of restricted cases (Bubeck and Büning 2006) and DQBF proof systems and their proof complexity (Beyersdorff et al. 2016; Balabanov, Chiang, and Jiang 2014; Beyersdorff et al. 2018), through the development of powerful DQBF solvers and preprocessors (Fröhlich et al. 2012; Fröhlich et al. 2014; Finkbeiner and Tentrup 2014; Tentrup and Rabe 2019; Wimmer, Scholl, and Becker 2019), to the establishment of an independent DQBF track in the annual QBF Evaluation[1].

When dealing with computational problems that are hard in their full generality, such as evaluating DQBF, a natural question is whether we can identify restrictions under which the problem can be solved efficiently. One way to achieve this is through the identification of polynomially tractable classes of instances—an approach that has led to many classical results in past years, such as the identification of Horn and Krom as two tractable fragments of SAT. A more recent framework that allows us to identify and algorithmically use such restrictions is provided by the *parameterized complexity* paradigm (Cygan et al. 2015; Downey and Fellows 2013; Gottlob, Scarcello, and Sideri 2002), which associates each instance of the problem with an integer *parameter* k that can be used to quantify how structured the instance is. The goal is then to obtain algorithms that are efficient when the parameter is small: the best-case outcome is an algorithm with a runtime of the form $f(k) \cdot n^{\mathcal{O}(1)}$, where n is the input size, and f is a computable function. Such algorithms are called *fixed-parameter algorithms*, and parameterized prob-

[1]http://www.qbflib.org/qbfeval20.php

lems which admit such an algorithm are said to belong to the complexity class *FPT*.

Contribution. In this paper, we initiate the investigation of DQBF through the lens of parameterized complexity by developing the first two fixed-parameter algorithms for the problem. As for the choice of parameters, we follow up on previous successful work on SAT (Samer and Szeider 2009b; Ganian and Szeider 2017), QBF (Chen 2004; Atserias and Oliva 2014; Eiben, Ganian, and Ordyniak 2018; Eiben, Ganian, and Ordyniak 2020) as well as numerous other problems (Ganian and Ordyniak 2018; Samer and Szeider 2010) by considering restrictions to the *primal graph*[2]—a natural graph representation which captures how variables in the instance interact with each other. The by far most prevalent restriction in this context is the parameterization by *treewidth*, a graph parameter which measures how tree-like the graph is, and our first result aims at establishing the tractability of DQBF evaluation with respect to the treewidth of the primal graph (i.e., the *primal treewidth*).

It is important to note that DQBF (and also QBF) evaluation remains intractable even when restricted to instances with constant primal treewidth (Atserias and Oliva 2014), and the intuitive reason for this is the fact that instances of small primal treewidth still allow for arbitrary interactions between variables through the quantifier dependencies of those variables. Hence, in order to achieve tractability, it is natural to combine primal treewidth with a measure that bounds the complexity of the interactions between the dependency sets. A suitable "base case" for such a parameterization is the previously studied class of DQBFs which require each pair of dependency sets to be either disjoint or equal (Scholl et al. 2019); here, we refer to these as *modular DQBFs*. A natural parameter that suggests itself at this point is the total number of "omissions" from the dependency sets of a DQBF required to achieve modularity; this matches the notion of (deletion) *backdoors* (Williams, Gomes, and Selman 2003) whose size has been used as a parameter in the SAT, CSP, and QBF settings (Gaspers and Szeider 2012; Gaspers et al. 2014; Samer and Szeider 2009a). However, since our aim here is merely to restrict the interaction between variable dependencies, we show that parameterizing by the total size of the backdoor is not necessary for tractability.

Indeed, as our first result, we establish fixed-parameter tractability of DQBF evaluation when parameterized by the primal treewidth and the *local size of a backdoor to modularity*—which measures the largest number of omissions from a single dependency set in a backdoor instead of the total number of all omissions. We remark that, as a consequence of parameterizing by the local size of the backdoor, our result implies fixed-parameter tractability of DQBF parameterized by the primal treewidth and the maximum size of a dependency set (representing a rather trivial subcase of our parameterization).

Given this result, it is natural to ask the following questions: Is it possible to apply a structural parameter to solve DQBF instances without placing any explicit restrictions on the variable dependencies, and simply modeling dependencies directly in the graph representation? Furthermore, can we use such a parameter to solve instances where the dependency sets of variables are highly non-modular? Our second result provides a positive answer to these questions by using the parameter *treedepth*, a fundamental and well-studied restriction of treewidth. In particular, we establish fixed-parameter tractability of DQBF evaluation when parameterized by the treedepth of the *augmented primal graph* alone, where the augmented primal graph is obtained from the primal graph by representing dependencies as edges.

Methods and Techniques. The core of the first algorithmic result lies in a series of reductions. However, before applying these, we first develop a polynomial-time 2-approximation algorithm for detecting backdoors to modularity of minimum local size. We then make use of *partial expansion* of this backdoor, where a formula is instantiated with all assignments to a subset of the universal variables, and copies of the dependent existential variables and clauses in which they occur are created. We show that the combinatorial explosion this operation introduces is capped by a function of our parameters, and moreover that the increase in treewidth is bounded as well, which effectively gives us an FPT-reduction to a modular DQBF. On this class, we can invoke a result of Scholl et al. (2019), which shows that the formula can be reduced to a QBF with 2 quantifier alternations. We then prove that this translation is treewidth-preserving, which allows us to finally use fixed-parameter-tractability of bounded-alternation QBF to obtain our result.

The second result applies the branch-pruning technique to iteratively reduce the input DQBF into an equivalent one whose size is bounded by a function of the treedepth (Ganian and Ordyniak 2018; Gutin, Jones, and Wahlström 2016). Intuitively, the technique uses a treedepth "decomposition" to identify irrelevant parts of the instance, i.e., a set of variables and clauses whose deletion will not change the outcome of the evaluation. The difficulty here lies in the specific challenges posed by DQBFs: among others, during the proof, we need to alter a model of a pruned formula so that it can also work for the formula before the pruning step, without any a-priori knowledge of how the model operates.

Related Work. Treewidth is a famous structural parameter that has found ubiquitous applications across nearly all areas of theoretical computer science research. Among others, treewidth is known to give rise to fixed-parameter algorithms for SAT (Samer and Szeider 2009b) and also for QBF when combined with the number of quantifier alternations (Chen 2004). In the latter case, it is known that the dependence of the running time on the number of alternations is a tower of exponents and that this is unavoidable (Pan and Vardi 2006; Fichte, Hecher, and Pfandler 2019). Treedepth is a restriction of treewidth that is closely tied to the theory of graph sparsity (Nesetril and de Mendez 2012) and has been successfully used to solve problems that have proved resilient to the dynamic programming techniques commonly employed with treewidth (Ganian and Ordyniak 2018; Gutin, Jones, and Wahlström 2016; Iwata, Ogasawara, and Ohsaka 2018).

[2]Definitions are provided in Section 2.

To the best of our knowledge, there has been no investigation of treewidth, or indeed other structural parameters, in the context of DQBF. Perhaps the most closely related results involved the introduction of two refinements of treewidth that can be used to solve QBF instances without restricting the number of quantifier alternations. The more recent of these, called *dependency treewidth* (Eiben, Ganian, and Ordyniak 2018), is designed in a way which allows instances of small width to be solved by *Q-resolution* (Kleine Büning, Karpinski, and Flögel 1995)—a proof system for QBF which does not have any directly applicable equivalent in DQBF. The other result is centered around *prefix treewidth* (Eiben, Ganian, and Ordyniak 2020), which is designed in a way that facilitates the construction of a model by dynamic programming. While on a high level such an approach might seem fruitful also for DQBF evaluation, due to the lack of a linear quantifier order in DQBF it is not clear at all whether or how techniques from that paper can be adapted to our more general setting.

Organisation. After going through preliminaries about DQBF, treewidth, and treedepth in Section 2, we give the algorithm which uses treewidth and backdoors to modularity in Section 3. In Section 4 we then turn to the treedepth-based algorithm. Finally, we conclude with some remarks and open questions in Section 5.

2 Preliminaries

For an integer i, we let $[i] = \{1, 2, \ldots, i\}$ and $[i]_0 = [i] \cup \{0\}$. We refer to the handbook by Diestel (2012) for standard graph terminology, and to the recent books for a basic overview of parameterized complexity theory (Downey and Fellows 2013; Cygan et al. 2015).

Dependency QBF. We assume the existence of a countably infinite set containing propositional *variables*. A *literal* is either a variable x, or its negation $\overline{x}$, whereby $\overline{\overline{x}} = x$. For a variable x we let $\mathrm{var}(x) = \mathrm{var}(\overline{x}) = x$, and we say that x and $\overline{x}$ are literals *on* x. A *clause* is a disjunction of literals, and a *CNF* (i.e., a formula in *conjunctive normal form*) is a conjunction of clauses. Whenever convenient, we interpret a clause as a set of literals, and a CNF as a set of clauses. The empty clause is denoted by $\bot$. We write $\mathrm{vars}(C) = \{\mathrm{var}(l) : l \in C\}$ for the set of variables occurring in a clause, and $\mathrm{vars}(\phi) = \cup_{C \in \phi} \mathrm{vars}(C)$ for the set of variables occurring in a CNF ϕ. For a set of variables V, the *restriction* of a clause C to V is the clause $C|_V = \{l \in C : \mathrm{var}(l) \in V\}$. For a set of literals X, we write $\overline{X} = \{\overline{x} : x \in X\}$.

An *assignment* to a set of variables X is a mapping $\tau : X \to \{0, 1\}$, for a literal $\overline{x} \in \overline{X}$ we define $\tau(\overline{x}) = 1 - \tau(x)$. We sometimes interpret an assignment τ as the set of literals it sets to true, i.e., as $\{l \in X \cup \overline{X} : \tau(l) = 1\}$. The set of assignments to X is denoted by 2^X.

The substitution of an assignment τ into a clause C, denoted $C[\tau]$, is $\top$, if there is $l \in C$ such that $\tau(l) = 1$, and the clause $\{l \in C : \tau(l) \neq 0\}$ otherwise. The substitution of τ into a CNF ϕ is the CNF $\phi[\tau] = \{C[\tau] : C \in \phi, C[\tau] \neq \top\}$. We say that τ *satisfies* ϕ if $\phi[\tau] = \emptyset$.

A *DQBF*[3] is a formula Φ of the form

$$\Phi = \forall u_1 \cdots \forall u_m \exists x_1(S_{x_1}) \cdots \exists x_n(S_{x_n}) \cdot \phi,$$

where the *matrix* ϕ is a CNF, and each existential variable x_i has its associated *dependency set* $S_{x_i} \subseteq \{u_1, \ldots, u_m\}$—the specification of the dependency sets is called the *prefix*. By $\mathrm{vars}_\forall(\Phi)$ and $\mathrm{vars}_\exists(\Phi)$ we denote the universal and existential variables of Φ, respectively, and $\mathrm{vars}(\Phi) = \mathrm{vars}_\forall(\Phi) \cup \mathrm{vars}_\exists(\Phi)$. For $V \subseteq \mathrm{vars}(\Phi)$ and an assignment $\tau \in 2^V$, we write $\Phi[\tau]$ for the formula whose prefix is obtained from the prefix of Φ by deleting variables assigned by τ, and whose matrix is $\phi[\tau]$. We only deal with *closed* DQBF, i.e., where $\mathrm{vars}(\phi) \subseteq \mathrm{vars}(\Phi)$. The *size* $|\Phi|$ of a DQBF Φ with the matrix ϕ is defined as the total number of literals in all its clauses plus the total size of its dependency sets, i.e., as $\sum_{C \in \phi} |C| + \sum_{x \in \mathrm{vars}_\exists(\Phi)} |S_x|$.

A *candidate model* of a DQBF Φ is a set of functions

$$f = \{f_x : 2^{S_x} \to 2^{\{x\}} : x \in \mathrm{vars}_\exists(\Phi)\}.$$

For a candidate model f and $\tau \in 2^{\mathrm{vars}_\forall(\Phi)}$, we write $f(\tau) = \{f_x(\tau|_{S_x}) : x \in \mathrm{vars}_\exists(\Phi)\}$ for the assignment resulting from applying f to τ—also known as the *response* of f to τ. A *model* of Φ is a candidate model f such that for every $\tau \in 2^{\mathrm{vars}_\forall(\Phi)}$ the assignment $\tau \cup f(\tau)$ satisfies the matrix ϕ. A DQBF is *true* if it has a model, and *false* otherwise.

The *primal graph* G_Φ of a DQBF Φ is the graph whose vertices are $\mathrm{vars}(\Phi)$ and two variables a, b are adjacent if and only if there is a clause C such that $a, b \in \mathrm{vars}(C)$. Notice that the primal graph does not capture the prefix of Φ in any way. Hence, we also consider the *augmented primal graph* $G^\star_\Phi$ of Φ, which is the graph obtained from G_Φ by adding an edge between each pair u, x of variables such that $u \in S_x$.

Treewidth. Let G be a graph. A *tree decomposition* of G is a pair (T, χ) where T is a tree and $\chi : T \to 2^{V(G)}$ is a mapping from tree nodes to subsets of $V(G)$ such that:

- $\forall e = uv \in E(G), \exists t \in V(T) : \{u, v\} \subseteq \chi(t)$, and
- $\forall v \in V(G)$, $T[\{t \mid v \in \chi(t)\}]$ is a non-empty connected subtree of T.

We call the vertices of T *nodes* and the sets in $\chi(t)$ *bags* of the tree decomposition (T, χ). The *width* of (T, χ) is equal to $\max\{|\chi(t)| - 1 \mid t \in V(T)\}$ and the *treewidth* of G (denoted $\mathrm{tw}(G)$) is the minimum width over all tree decompositions of G. The primal treewidth of a DQBF Φ is the treewidth of its primal graph.

It is possible to efficiently construct near-optimal tree decompositions for graphs of low treewidth:

Proposition 1 (Bodlaender et al., 2016)**.** *There exists an algorithm which, given an n-vertex graph G and an integer k, in time $2^{\mathcal{O}(k)} \cdot n$ either outputs a tree-decomposition of G of width at most $5k + 4$ and $\mathcal{O}(n)$ nodes, or determines that $\mathrm{tw}(G) > k$.*

Treedepth. Treedepth is a structural parameter closely related to treewidth, and the structure of graphs of bounded

[3] We only consider what is otherwise also known as an S-form DQBF in CNF (Balabanov, Chiang, and Jiang 2014).

treedepth is well understood (Nesetril and de Mendez 2012). A useful way of thinking about graphs of bounded treedepth is that they are (sparse) graphs with no long paths.

We formalize a few notions needed to define treedepth. A *rooted forest* is a disjoint union of rooted trees. For a vertex x in a tree T of a rooted forest, the *height* (or *depth*) of x in the forest is the number of vertices in the path from the root of T to x. The *height of a rooted forest* is the maximum height of a vertex of the forest.

Definition 1 (Treedepth)**.** *Let the* closure *of a rooted forest $\mathcal{F}$ be the graph $clos(\mathcal{F}) = (V_c, E_c)$ consisting of the vertex set $V_c = \bigcup_{T\in\mathcal{F}} V(T)$ and the edge set $E_c = \{xy\colon x \text{ is an ancestor of } y \text{ in some } T \in \mathcal{F}\}$. A* treedepth decomposition *of a graph G is a rooted forest $\mathcal{F}$ such that $G \subseteq clos(\mathcal{F})$. The* treedepth $td(G)$ *of a graph G is the minimum height of any treedepth decomposition of G.*

We will later use T_x to denote the vertex set of the subtree of T rooted at a vertex x of T. Similarly to treewidth, it is possible to determine the treedepth of a graph in FPT time.

Proposition 2 (Nesetril and de Mendez, 2012)**.** *Given an n-vertex graph G and a constant w, it is possible to decide whether G has treedepth at most w, and if so, to compute an optimal treedepth decomposition of G in time $\mathcal{O}(n)$.*

3 DQBF Parameterized by Treewidth

In this section, we present algorithms for evaluating DQBFs with small treewidth. Since bounded treewidth on its own is not enough to ensure tractability (Atserias and Oliva 2014), we additionally introduce a parameter that imposes restrictions on the dependency sets. This allows us to reduce to the problem of evaluating a QBF with a Σ_3 prefix, which is known to be FPT parameterized by treewidth (Chen 2004; Capelli and Mengel 2019). Our parameter takes the value 0 for DQBFs with dependency sets that are pairwise identical or disjoint.

3.1 Modularity and Backdoors

The first notion we will need to define a suitable restriction of the dependency set is *modularity*.

Definition 2 (Modular)**.** *Let $S = (S_i)_{i\in[m]}$ be a family of subsets of a finite set X. The family S is called* modular *if $S_i \cap S_j \neq \emptyset \implies S_i = S_j$ for all $i, j \in [m]$.*

A DQBF with modular dependency sets can be rewritten as a QBF with a Σ_3 prefix (Scholl et al. 2019). An inspection of the proof shows that the rewriting preserves primal treewidth.

Theorem 1 (Scholl et al. 2019)**.** *A DQBF Φ with modular dependency sets can be rewritten into an equisatisfiable QBF Φ' with a Σ_3 prefix in polynomial time. Moreover, if Φ has primal treewidth w then Φ' has primal treewidth at most $w + 2$.*

Proof. Let $V_1, \dots, V_l$ be a partitioning of the existential variables of Φ such that, for any two existential variables, their dependency sets coincide if, and only if, they are from the same part. Each clause C of Φ can be expressed as a disjunction $C = C_1 \vee \cdots \vee C_l$ of variable-disjoint clauses with $var(C_i) \subseteq V_i \cup D(V_i)$. The rewriting now splits the clause C into clauses $C_1 \vee z_1, \neg z_1 \vee C_2 \vee z_2, \dots, \neg z_{l-1} \vee C_l$ by introducing new existential variables z_i such that $D(z_i) = \emptyset$. The dependency sets of the original existential variables are then inflated so as to include all universal variables. The resulting DQBF Φ' can be written as a QBF with the prefix $\exists Z \forall U \exists E$, where Z is the set of existential variables z_i introduced by splitting clauses, $U = \text{vars}_\forall(\Phi)$, and $E = \text{vars}_\exists(\Phi)$. Clearly, the entire rewriting can be done in polynomial time. For a proof of soundness, refer to the paper by Scholl et al. (2019).

We now show that the primal treewidth is increased by at most 2. First, consider a tree decomposition (T, χ) of the primal graph associated with Φ. To obtain a tree decomposition (T', χ') of the primal graph associated with Φ', we only need to take care of the auxiliary variables Z. Consider a clause $C = C_1 \vee \cdots \vee C_l$ split in the way described above and let $z_1, \dots, z_{l-1} \in Z$ be the newly introduced existential variables. Because $\text{vars}(C)$ is a clique in the primal graph, there has to be a node $t \in T$ such that $\text{vars}(C) \subseteq \chi(t)$.[4] In T', we add new nodes $t_1, \dots, t_l$ such that t_1 is adjacent to t and t_i is adjacent to t_{i-1}, for $1 < i \leq l$, so that the nodes $t, t_1, \dots, t_l$ form a path in T'. We let $\chi'(t) = \chi(t)$ for $t \in T$, $\chi'(t_1) = \chi(t) \cup \{z_1\}$, $\chi'(t_l) = \chi(t) \cup \{z_{l-1}\}$, and $\chi'(t_i) = \chi(t) \cup \{z_i, z_{i+1}\}$ for $1 < i < l$. Clearly, T' is a tree and (T', χ') still satisfies all conditions required of a tree decomposition for original variables $v \in \text{vars}(\Phi))$ and edges vw connecting variables $v, w \in \text{var}(\Phi)$ in the primal graph of Φ'. Also, each auxiliary variable z_i is contained in some bag, edges $z_i z_{i+1}$ in the primal graph are covered by the bag $\chi'(t_i)$, and the running intersection property is satisfied for each variable z_i by construction. Finally, if $z_i v$ is an edge in the primal graph and $v \in \text{vars}(\Phi)$ then $v \in \text{vars}(C)$ and $z_i, v \in \chi'(t_i)$. This proves that (T', χ') is a tree decomposition of the primal graph of Φ', and its width is at most $w + 2$. □

The parameter we will use to restrict dependencies measures the "distance" to a modular instance. To formalize this, we introduce a natural adaptation of backdoors to our setting.

Definition 3 (Deletion Backdoor to Modularity)**.** *Let X be a finite set and $S = (S_i)_{i\in[m]}$ a family of subsets of X. A* deletion backdoor to modularity *of S is a set $X' \subseteq X$ such that the family $S' = (S_i \setminus X')_{i\in[m]}$ is modular.*

A natural parameter traditionally used in the context of SAT and CSP is the size of a smallest backdoor to a certain tractable class. However, here we consider a considerably more relaxed restriction: the parameter will simply bound the maximum intersection between the backdoor and a dependency set. Formally, a deletion backdoor to modularity X' of S is said to be *locally k-bounded* (or to have *local size k*) if $|S_i \cap X'| \leq k$ for each $i \in [m]$.

For a DQBF Φ, we say that a set B of universal variables in Φ is a locally k-bounded backdoor to modularity if for

[4] For a proof of this statement refer to a standard textbook on graph theory (Diestel 2012) or parameterized complexity (Flum and Grohe 2006).

each $x \in \text{vars}_\exists(\Phi)$ it holds that $|S_x \cap B| \leq k$. The parameter we will use to restrict the dependencies is the minimum local size of a backdoor to modularity—but before we can use it, we first need to show how to efficiently compute it.

Lemma 1. *There is a polynomial-time algorithm that, given a family $S = (S_i)_{i\in[m]}$ of subsets of a finite set X and an integer $k \in \mathbb{N}$, either determines that S does not have a locally k-bounded backdoor to modularity, or computes a backdoor to modularity of S that is locally $2k$-bounded.*

Proof. The algorithm proceeds as follows. First, it sets $B = \emptyset$ and loops over all integers $i, j \in [m]$ and checks whether the modularity condition $S_i \cap S_j \neq \emptyset \implies S_i = S_j$ holds. If the condition holds for every such i, j, then the instance is already modular and $B = \emptyset$ is a locally k-bounded backdoor to modularity.

Otherwise, let i, j be a pair such that the condition is violated. Let $L = S_i \setminus S_j$, $C = S_i \cap S_j$ and $R = S_j \setminus S_i$, and recall that C as well as $L \cup R$ must be non-empty. Moreover, observe that every backdoor to modularity must either be a superset of C or a superset of $L \cup R$ (indeed, as long as a single variable remains undeleted in C and a single variable remains undeleted in $L \cup R$, the set system cannot be modular). Consider first the case where $|C| > k$; then every locally k-bounded backdoor must necessarily contain $L \cup R$. In this case, we add $L \cup R$ into B and check whether B remains locally k-bounded (if not, we correctly conclude that S does not have a locally k-bounded backdoor to modularity). Similarly, if $|L| > k$ or $|R| > k$, then every locally k-bounded backdoor must necessarily contain C, and hence we add C into B and check to ensure B remains locally k-bounded.

Using the above procedure, we process every pair S_i, S_j of sets in S such that their intersection C as well as both their non-intersecting parts L and R are larger than k, and have obtained a preliminary backdoor B that contains variables which must be present in *every* locally k-bounded backdoor to modularity. Now consider a set S_i such that $|S_i| > 2k$. We claim that for every set S_j it must hold that either $S_i \cap S_j = \emptyset$ or $S_i = S_j$. Indeed, if this were not the case, then the sets S_i and S_j would have been processed by the above procedure—after all, either $S_i \cap S_j$ or $S_i \setminus S_j$ must necessarily be larger than k—and the procedure would either result in $S_i \cap S_j$ being empty or in $S_i = S_j$. Since this property holds for the intersection between S_i and *every* set S_j and the property will not be violated by further additions to B, we may at this point safely remove every S_i of size greater than $2k$ from the family without changing the outcome of the algorithm.

At this point, the family only contains sets of size at most $2k$. To obtain our desired 2-approximation, we now simply add the contents of every remaining set into B. □

In view of Lemma 1, it is natural to ask why one needs to resort to approximation for computing the parameter—after all, finding a local backdoor of minimum local size could have been a polynomial-time computable problem. Or, perhaps it could at least be possible to compute the parameter exactly by using a fixed-parameter algorithm. Surprisingly, we settle these questions by showing that the problem is not only NP-complete, but remains NP-complete even when the task is to determine whether there exists a backdoor to modularity of local size 2.

Lemma 2. *Given a family $S = (S_i)_{i\in[m]}$ of subsets of a finite set X and an integer $k \in \mathbb{N}$, is it* NP*-complete to decide whether there exists a locally 2-bounded backdoor to modularity.*

Proof. Inclusion in NP is trivial, since we can easily verify whether a backdoor to modularity is locally k-bounded. To show hardness, we reduce from the NP-hard 1-IN-3 POSITIVE 3-SAT problem, where we are given a 3-CNF formula ϕ (i.e. each clause has 3 literals) without negative literals and are asked whether there exists a *1-in-3 satisfying assignment*, i.e. one which sets precisely one literal in each clause to true (Garey and Johnson 1979).

Let $\phi = \{C_1, \ldots, C_n\}$ be an instance of 1-IN-3 POSITIVE 3-SAT, where $C_i = \{x_1^i, x_2^i, x_3^i\}$. Let $S(\phi) = \phi \cup \{\{x_j^i\} : 1 \leq i \leq n;\ 1 \leq j \leq 3\}$. We claim that ϕ has a 1-in-3 satisfying assignment if and only if $S(\phi)$ has a locally 2-bounded backdoor to modularity.

For the first direction, let τ be a 1-in-3 satisfying assignment of ϕ, which we interpret as the set of variables it sets to true. Then $\tau' = \text{vars}(\phi) \setminus \tau$ is a locally 2-bounded backdoor to modularity of $S(\phi)$: indeed, τ' deletes exactly 2 variables from every clause, and the rest is an instance consisting of singletons only, which is trivially modular.

Conversely, let B be a locally 2-bounded backdoor to modularity of $S(\phi)$. Because $S(\phi)$ contains the sets $C_i = \{x_1^i, x_2^i, x_3^i\}$, and $\{x_1^i\}, \{x_2^i\}, \{x_3^i\}$, B must necessarily contain at least 2 variables from C_i. On the other hand, $B \cap C_i \leq 2$, so in fact B contains exactly 2 variables from every clause. Hence $B' = \text{vars}(\phi) \setminus B$ is a 1-in-3 satisfying assignment of ϕ. □

3.2 Universal Expansion

One method of deciding the truth value of a DQBF is by performing *expansion of universal variables* (Bubeck and Büning 2006; Balabanov, Chiang, and Jiang 2014). Expanding a universal variable u in a DQBF Φ results in the conjunction $\Phi[u \mapsto 1] \wedge \Phi[u \mapsto 0]$ of copies of Φ where u is removed from the quantifier prefix and replaced with the corresponding value in the matrix. Prenexing yields a DQBF with two copies (e^u and $e^{\overline{u}}$) of each existential variable e of Φ that has u in its dependency set. By repeating this step we can expand an entire set V of universal variables and get up to $2^{|V|}$ copies of the matrix and each existential variable. It is convenient to keep track of the universal assignment associated with each copy by working with *annotated literals* ℓ^σ that consist of a literal ℓ and a truth assignment $\sigma : V \to \{0, 1\}$. We introduce the concept of *instantiation* to describe the copy of the matrix induced by an assignment σ.

Definition 4 (Instantiation)**.** *Let Φ be a DQBF, $V \subseteq \text{vars}_\forall(\Phi)$, $\sigma \in 2^V$, and ℓ a literal. The* instantiation $\ell_\Phi^{[\sigma]}$ *of a literal ℓ with the assignment σ in Φ is the annotated literal $\ell^{\sigma'}$ with $\sigma' = \sigma|_{S_{var(\ell)}}$ if ℓ is existential, and $\ell[\sigma]$ if ℓ*

is universal. In particular, for a universal literal ℓ we get $\ell_\Phi^{[\sigma]} = \top$ if $\sigma(\ell) = 1$, $\ell_\Phi^{[\sigma]} = \bot$ if $\sigma(\ell) = 0$, and $\ell_\Phi^{[\sigma]} = \ell$ if var$(\ell) \notin V$. This allows us to define the instantiation of a clause C *with σ in Φ as $C_\Phi^{[\sigma]} = \bigvee_{\ell \in C} \ell_\Phi^{[\sigma]}$, and the* instantiation of a CNF ϕ *with σ in Φ as $\phi_\Phi^{[\sigma]} = \bigwedge_{C \in \varphi} C_\Phi^{[\sigma]}$.*

We omit the DQBF Φ from the subscript of an instantiated literal whenever it is understood.

Definition 5 (Expansion of Universal Variables)**.** *Given a DQBF $\Phi = \forall u_1 \cdots \forall u_m \exists x_1(S_{x_1}) \cdots \exists x_n(S_{x_n}) \cdot \phi$ and a subset $V \subseteq U = vars_\forall(\Phi)$, the* expansion *of V in Φ results in the DQBF*

$$\Phi' = \forall U' \exists x_1^{[V]}(S'_{x_1}) \ldots \exists x_n^{[V]}(S'_{x_n}). \bigwedge_{\sigma \in 2^V} \phi^{[\sigma]},$$

where $U' = U \setminus V$, $S'_{x_i} = S_{x_i} \setminus V$, and $\exists x_i^{[V]}(S'_{x_i})$ is a shorthand for the set $\{\exists x_i^{[\sigma]}(S'_{x_i})\}_{\sigma \in 2^V}$ of existential quantifiers created by instantiating the variable x_i with assignments to V, for $1 \leq i \leq n$.

It is well known that universal expansion is sound and complete for DQBF (Bubeck 2010).

Lemma 3. *Let Φ be a DQBF, let V be a subset of its universal variables, and let Φ' be obtained by expanding V in Φ. Then Φ is true if, and only if, Φ' is true.*

The following result shows that universal expansion of a locally k-bounded deletion backdoor set to modularity does not increase the size and the primal treewidth of a formula too much.

Lemma 4. *Let $\Phi = \forall u_1 \cdots \forall u_m \exists x_1(S_{x_1}) \cdots \exists x_n(S_{x_n}) \cdot \phi$ be a DQBF and let $V \subseteq vars_\forall(\Phi)$ be a locally k-bounded backdoor set to modularity of the family $(S_{x_i})_{i \in [n]}$ of the dependency sets of Φ. The DQBF Φ' obtained by expanding V in Φ has the following properties:*

1. *The size of Φ' is in $O(2^{k(w+1)} \|\Phi\|)$.*
2. *The family of dependency sets of Φ' is modular.*
3. *The primal treewidth of Φ' is at most $2^{k(w+1)} - 1$.*

Proof. Since V is locally k-bounded we can get at most 2^k existential variables $x_i^{[\sigma]}$ by instantiating an existential variable x_i with assignments $\sigma \in 2^V$ for $1 \leq i \leq n$, so the length of the quantifier prefix of Φ' is in $O(2^k \|\Phi\|)$. For each clause $C \in \phi$ we can get at most $2^{k(w+1)}$ distinct instantiations $C^{[\sigma]}$ for assignments $\sigma \in 2^V$: instantiations of C only differ in the annotations of existential literals ℓ, and by the above argument there are at most 2^k such instantiations $\ell^{[\sigma]}$ for each literal; moreover, each clause contains at most $w+1$ existential literals because the primal treewidth of ϕ is w. This proves the bound on the size of ϕ'.

By definition of expansion, every dependency set of an existential variable in Φ' is obtained by removing V from a dependency set of a variable from Φ. Because V is a deletion backdoor set to modularity the resulting set family is modular.

To obtain the desired bound on the primal treewidth of Φ', consider a tree decomposition (T, χ) of the primal graph of the DQBF Φ. We are going to construct a tree decomposition (T, χ') of the primal graph associated with Φ' by replacing variables by their annotated copies. Formally, we let $\chi'(t) = \{x^{[\sigma]} \mid v \in \chi(t), \sigma \in 2^V\} \setminus \{\top, \bot\}$. We have to verify that (T, χ') is indeed a tree decomposition.

First, we show that every variable of Φ' is contained in a bag. Each universal variable $u \in vars_\forall(\Phi')$ is in $vars(\Phi) \setminus V$ and $u^{[\sigma]} = u$ for any $\sigma \in 2^V$ by definition, so $u \in \chi'(t)$ whenever $u \in \chi(t)$. Each existential variable of Φ' is an annotated variable $e^{[\sigma]}$ for some $\sigma \in 2^V$. Since (T, χ) is a tree decomposition, there is a node $t \in T$ such that $e \in \chi(t)$ and thus $e^{[\sigma]} \in \chi'(t)$ by construction.

Second, we prove that every edge is covered by a bag. Every clause of Φ' is an annotated clause $C^{[\sigma]}$ for some assignment $\sigma \in 2^V$. Thus if $\ell_1^{[\sigma]}, \ell_2^{[\sigma]} \in C^{[\sigma]}$, then $\ell_1, \ell_2 \in C$ for a clause C of Φ. Again, there has to be a node $t \in T$ such that var(ℓ_1), var$(\ell_2) \in \chi(t)$ and var$(\ell_1)^{[\sigma]}$, var$(\ell_2)^{[\sigma]} \in \chi'(t)$ by construction.

Third, we verify that (T, χ') has the running intersection property. Take the subgraph $T_{v^{[\sigma]}} = \{t \in T | v^{[\sigma]} \in \chi'(t)\}$ of T induced by the set of nodes whose bags contain the variable $v^{[\sigma]}$ of Φ'. The subgraph $T_v = \{t \in T \mid v \in \chi(t)\}$ is a tree because (T, χ) is a tree decomposition, and the identity $T_{v^{[\sigma]}} = T_v$ holds by construction. Thus (T, χ') is a tree decomposition.

It remains to prove the bound on the width of this decomposition. We can get at most 2^k annotated variables $v^{[\sigma]}$ for each variable v of Φ by instantiating with assignments $\sigma \in 2^V$, and each bag $\chi(t)$ contains at most $w+1$ variables, so $|\chi'(t)| \leq (2^k)^{w+1} = 2^{k(w+1)}$. □

We refer to the smallest k such that the family of dependency sets of a DQBF has a locally k-bounded backdoor to modularity as its *distance to modularity*.

Theorem 2. *DQBF evaluation is fixed-parameter tractable parameterized by the combination of primal treewidth and distance to modularity.*

Proof. Let Φ be a DQBF, w its primal treewidth, and k its distance to modularity. By Lemma 1 there is a polynomial-time algorithm that computes a set $V \subseteq vars_\forall(\Phi)$ of universal variables that constitutes a locally $2k$-bounded backdoor set to modularity of the family of dependency sets of Φ. We expand V in Φ to obtain the DQBF Φ'. By Lemma 4, this can be done in time $O(2^{2k(w+1)} \|\Phi\|)$, the primal treewidth of Φ' is at most $2^{2k(w+1)} - 1$, and the family of dependency sets of Φ' is modular. Furthermore, the formulas Φ and Φ' are equisatisfiable by Lemma 3. We then apply Theorem 1 to obtain an equisatisfiable QBF with two quantifier alternations while increasing the primal treewidth by at most 2. Finally, we use the fact that QBF is FPT parameterized by primal treewidth for a fixed number of quantifier alternations (Chen 2004; Capelli and Mengel 2019). □

4 DQBF Parameterized by Treedepth

In this section we describe an FPT algorithm for evaluating DQBF parameterized by the treedepth of its augmented primal graph. The main result of this section is Theorem 3 at

the end, which describes the FPT algorithm on a high level. The key technical result used by Theorem 3 is Lemma 5, which however requires some introduction. We therefore start by describing the general proof strategy, setting the scene and preparing the vocabulary for the highly specific Lemma 5.

The main technique used in the algorithm is that of pruning—informally speaking, we will be reducing the instance by deleting irrelevant variables so that in the end we obtain an equivalent instance of bounded size (which can then be solved by any algorithmic technique, even brute force). For that purpose we define the following operation of deleting a set of variables from a formula: for a DQBF Φ and $V \subseteq \text{vars}(\Phi)$, let $\Phi - V$ be the formula obtained from Φ by removing (1) all clauses containing at least one variable from V, and (2) all variables from V. Note that since $\Phi - V$ is obtained by first removing clauses, and then removing variables that do not occur in the matrix, it follows that if Φ is true, then so is $\Phi - V$ for any V.

For the rest of this section, we assume that Φ is a DQBF, $G^\star_\Phi$ is its augmented primal graph, $\mathcal{F}$ is a rooted forest of depth k in which $G^\star_\Phi$ is embedded, and T is a tree in $\mathcal{F}$. We will use the aliases $U := \text{vars}_\forall(\Phi)$ and $X := \text{vars}_\exists(\Phi)$.

We will prove by induction that it is possible to reduce the number of children of any given node at depth $k - i$ to $g(k, i)$, and consequently to reduce the size of any subtree at depth $k - i$ to $h(k, i)$, where $g(k, i)$ and $h(k, i)$ are some suitable functions that we will define shortly. Together with the bound on the depth of T, we will then be able to bound the size of T as a function of k.

We define $g(k, 0) = 0$, $h(k, 0) = 1$, and we further define $g(k, i)$, $h(k, i)$ and the auxiliary functions s, r, m, c recursively as follows:

$$\begin{aligned}
s(k,i) &= 2^{3^{h(k,i)+k}} \left(2^{h(k,i)+k}\right)^{h(k,i)+k} \\
r(k,i) &= 2^{h(k,i)+k} \\
m(k,i) &= \left(2^{2^{h(k,i)+k}}\right)^{h(k,i)} \\
c(k,i) &= m(k,i)r(k,i) \\
g(k,i+1) &= s(k,i)c(k,i) \\
h(k,i+1) &= 1 + h(k,i)g(k,i+1)
\end{aligned}$$

These functions have the following intuitive meaning: $h(k, i)$ bounds the size of a subtree rooted at depth $k - i$ given that the number of children of any node at a larger or equal depth $k - j \geq k - i$ is bounded by $g(k, j)$; $s(k, i)$ bounds the number of DQBF formulas on $h(k, i) + k$ variables; $r(k, i)$ is the number of assignments to a set of size $h(k, i) + k$; and $m(k, i)$ bounds the number of combined model functions for a set of $h(k, i)$ existential variables that may depend on up to $h(k, i) + k$ universal variables.

Observe that our inductive hypothesis—that the number of children of any node at depth $k - i$ is at most $g(k, i)$—holds trivially for $i = 0$.

For a node $p \in T$, we define P_p to be the set which consists of p and all of its ancestors, and A_p to be the set of all connected components of $G^\star_\Phi[T_p] - P_p$.

Assume that the inductive hypothesis (number of children of any node at depth $k - j$ is bounded by $g(k, j)$) holds for all $j \leq i$. For a node $p \in T$ at depth $k - i$, we immediately get $|T_p| \leq h(k, i)$.

Now, assume that there is a node t at depth $k - (i + 1)$ with more than $g(k, i + 1)$ children. First, let us define a notion of equivalence of connected components of A_t.

For two sets of variables $A, B \in \text{vars}(\Phi)$, a *renaming function* is a bijection $\eta_{A,B} : A \to B$. We implicitly extend a renaming function $\eta_{A,B}$ and its inverse $\eta^{-1}_{A,B}$ to variables outside of A and B by setting $\eta_{A,B}(x) = x$, to literals by setting $\eta_{A,B}(\overline{x}) = \overline{\eta_{A,B}(x)}$, to assignments by setting $\eta_{A,B}(\tau) = \tau \circ \eta^{-1}_{A,B}$, to clauses as $\eta_{A,B}(C) = \{\eta_{A,B}(\ell) : \ell \in C\}$, to sets of clauses as $\eta_{A,B}(\phi) = \{\eta_{A,B}(C) : C \in \phi\}$, to a DQBF Φ' by renaming the prefix as well as the matrix, and finally to model functions as $\eta_{A,B}(f_x) = \eta_{A,B} \circ f_x \circ \eta^{-1}_{A,B}$.

Let A, B be connected components in A_t. Let us define the equivalence $\equiv$ as follows: $A \equiv B$ if and only if there is a renaming function $\eta_{A,B} : A \to B$ such that $\Phi - A = \eta_{A,B}(\Phi - B)$. Notice that $A \equiv B$ implies that A and B have precisely the same number of existentials and universals, where there is a one-to-one mapping between universals and one-to-one mapping between existentials such that 1) each clause on $B \cup P_t$ has a counterpart on $A \cup P_t$, and 2) the dependencies are preserved by these mappings.

Since $|P_t| \leq k$ and for each $A \in A_t$ we have $|A| \leq h(k, i)$ and the variables from A only have dependencies to/from $P_t \cup A$ and also only occur in clauses with $P_t \cup A$, the number of equivalence classes of $\equiv$ is upper-bounded by the number of DQBF instances on $h(k, i)+k$ variables, which is at most $s(k, i)$. Moreover, since each of the existential variables in A has at most $h(k, i) + k$ dependencies, the total number of existential strategies for each individual variable in $A \cap X$ is upper-bounded by $2^{2^{h(k,i)+k}}$. Since the number of variables in $A \cap X$ is upper-bounded by $h(k, i)$, we obtain that the total number of possible existential strategies for $A \cap X$ is at most $m(k, i)$.

Lemma 5. *Let t be a node of T with more than $g(k, i + 1)$ children in T, where each child of t satisfies the inductive hypothesis. Then we can compute in time $\mathcal{O}(h(k, i)! \cdot |\Phi|^2)$ a child a of t such that Φ is true if and only if so is $\Phi - T_a$.*

Proof. Since t has more than $g(k, i + 1) = s(k, i)c(k, i)$ children in T and there are at most $s(k, i)$ equivalence classes of $\equiv$, it follows that there will be over $c(k, i)$ components in A_t which are all equivalent. Let us denote this set Q_t. Note that since each component in A_t has size bounded by $h(k, i)$, it is possible to test all pairs of components for $\equiv$ (and in particular to compute Q_t) by brute-forcing over all renaming functions in time $\mathcal{O}(h(k, i)! \cdot |\Phi|^2)$.

Let a be the root of an arbitrary component in Q_t. We claim that a satisfies the properties of the lemma. Since if Φ is true, then trivially so is $\Phi - T_a$, we only need to show that if $\Phi - T_a$ has a model, then so does Φ.

Let f be a model of $\Phi - T_a$. Consider the model f^* for Φ obtained from f as follows. First, since $|Q_t \setminus \{T_a\}| \geq$

$c(k,i)$ and there are at most $m(k,i)$ many possible combined model functions for the existentials in each component of Q_t, there must be at least one component $T_b \in Q_t \setminus \{T_a\}$ whose strategy is reused at least $r(k,i)$ times by f (modulo renaming). Let Q_t^f be the restriction of Q_t to those components which use the same strategy as T_b, and once again note that $|Q_t^f| \geq r(k,i)$; formally, Q_t^f has the property that for each $T_y, T_z \in Q_t^f$, there is a renaming function $\eta_{y,z}$ such that $\Phi - T_y = \eta_{y,z}(\Phi - T_z)$ and furthermore for each $x \in T_y$ we have $f_{\eta_{y,z}(x)} = \eta_{y,z}(f_x)$. We now build f^* for the existentials in T_a by copying and renaming f from T_b—more precisely, for each variable $x_a \in X \cap T_a$ we set $f^*_{x_a} = \eta_{b,a}(f_{x_b})$.

It remains to define how f^* behaves on existential variables in P_t, which could now also depend on $T_a \cap U$. Unfortunately, it is not clear how one should extend the model function f_x (for each variable $x \in P_t \cap X$) to one for f^*_x in a way which results in f^* being a model—the issue that arises here is that the response of f^*_x must depend only on S_x, and cannot look at the entire assignment to all universals in T_a. So instead, we use a different approach: we will show that, in fact, all dependencies of every such x on Q_t^f can safely be omitted from the formula.

Claim 1. *Let Φ' be the DQBF obtained from Φ by setting for each $x \in P_t \cap X$ its dependency set in Φ' to $S'_x = S_x \setminus \bigcup_{T_b \in Q_t^f} T_b$. Then Φ' is true if and only if Φ is true; moreover, a model of Φ' can be obtained from a model of Φ in polynomial time.*

Proof of Claim. Clearly, if Φ' is true, then so is Φ, since the dependency sets are larger in Φ. We need to show that if Φ has a model, then so does Φ'.

Let f^1 be a model of Φ. Let $\gamma : Q_t^f \cap U \to \{0,1\}$ be an arbitrary assignment of the universals in Q_t^f which *encompasses all possible individual assignments* of Q_t^f. This means that γ has the following property: for an arbitrary $T_b \in Q_t^f$ and for each possible assignment $\beta : T_b \cap U \to \{0,1\}$, there exists $T_d \in Q_t^f$ such that $\forall u_b \in T_b \cap U : \gamma(\eta_{b,d}(u_b)) = \beta(u_b)$. Such a γ exists since $|Q_t^f| \geq r(k,i)$, and it can easily be constructed in polynomial time since we have already computed the renaming functions η for all pairs of components.

Now consider the candidate model f^2 for Φ' which behaves exactly like f^1 on all variables other than those in $P_t \cap X$. For each $x \in P_t \cap X$ and each $\lambda : S'_x \to \{0,1\}$, let $\lambda' = \lambda[S'_x] \cup \gamma$. We now set $f^2_x(\lambda) = f^1_x(\lambda')$; in other words, the response of f^2 for x simply copies the response of f^1 for x where we assume that the variables in Q_t^f were assigned according to γ. To conclude the proof, we want to show that f^2 is a model of Φ'.

Towards a contradiction, assume that there is a total universal assignment λ_2 such that $\lambda_2 \cup f^2(\lambda_2)$ falsifies the matrix, and in particular the clause C.

First, consider the case that C contains no variables from Q_t^f. Then consider the assignment $\lambda_1 = \lambda_2|_{U \setminus Q_t^f} \cup \gamma$ of $\mathrm{vars}_\forall(\Phi)$. By definition of f^2, for each existential $x \in P_t$ it holds that $f^2_x(\lambda_2|_{S_x}) = f^1_x(\lambda_1|_{S_x})$, and hence no existential in P_t can satisfy C under $\lambda_1 \cup f^1(\lambda_1)$. Moreover, all universals in C are assigned precisely the same way as in λ_2, and all other existentials in C must use the same assignment as dictated by f^2—hence C would be falsified when applying f^1 to λ_1, a contradiction.

Second, consider the case that C contains at least one variable from Q_t^f. In that case C can only contain variables from P_t and precisely one component of Q_t^f, say T_e. Consider once again the universal assignment $\lambda_1 = \lambda_2|_{U \setminus Q_t^f} \cup \gamma$. By the definition of γ, there must exist some component of Q_t^f, say T_f, such that λ_2's universal assignment on T_e is copied by γ on T_f—formally, we have $\forall u_e \in T_e \cap U : \gamma(\eta_{e,f}(u_e)) = \lambda_2(u_e)$. Moreover, by the definition of Q_t Φ must also contain a mirrored clause $\eta_{e,f}(C)$ of C on T_f.

Now, by the definition of Q_t^f and the fact that the assignment $\lambda_2|_{T_e}$ is mirrored by γ on T_f, we obtain that under $\lambda_1 \cup f^1(\lambda_1)$, $\eta_{e,f}(C)$ can be satisfied by neither (1) an existential nor (2) a universal in T_f. Moving on, the restriction $C|_{P_t}$ of C to P_t is precisely the same as the restriction $\eta_{e,f}(C)|_{P_t}$. Hence, since $\lambda_1|_{P_t} = \lambda_2|_{P_t}$, we see that (3) $\eta_{e,f}(C)$ cannot be satisfied by a universal in P_t. Finally, by the definition of f^2 we have that $f^2_x(\lambda_2|_{S'_x}) = f^1_x(\lambda_1|_{S_x})$, and so under $\lambda_1 \cup f_1(\lambda_1)$ the clause $\eta_{e,f}(C)$ cannot be satisfied by any $x \in P_t \cap X$ either. Altogether, this contradicts the fact that f^1 is a model. □

With Claim 1 in hand, we can proceed as follows. Let γ be an arbitrary universal assignment of Q_t^f which encompasses all individual assignments to the components in Q_t^f, as defined in the proof of Claim 1. Since f is a model of Φ, by following the proof of Claim 1 we can obtain a model f' for Φ which alters the functions f'_x for each $x \in P_t \cap X$ so that they do not depend on the universal assignments to Q_t^f (on all other variables, f' is the same as f). For each $x \in P_t \cap X$, we then simply set $f^*_x = f'_x$.

To conclude the proof, it remains to show that f^* is indeed a model. Assume for a contradiction that there is a universal assignment $\lambda_0 : U \to \{0,1\}$ in Φ such that a clause C is falsified by $\lambda \cup f^\star(\lambda)$. Observe that C must contain an existential variable from T_a, since all other existential variables use the same function as in f', which we already showed to be winning by Claim 1. Now consider the assignment $\lambda'_0 \in 2^{\mathrm{vars}_\forall(\Phi - T_a)}$ obtained from λ_0 by copying $\lambda_0|_{T_a}$ onto an arbitrary different component of Q_t^f, say T_b (modulo $\eta_{a,b}$). Consider the clause $C' = \eta_{a,b}(C)$ and note that C' is indeed a clause in $\Phi - T_a$ that contains variables from $T_b \cup P_t$.

Let us now ask: what happens to C' under $\lambda'_0 \cup f'(\lambda'_0)$? First, C' cannot be satisfied by any variable in P_t, since universals in P_t use the same assignment as λ_0 and f' assigns all existentials in P_t only based on dependencies outside of Q_t^f (which did not change at all compared to when we used f^* on λ_0). Second, since C was not satisfied previously and we copied the assignment of $\lambda_0|_{T_a}$ onto T_b, it follows that C' cannot be satisfied by any variable in T_b either. Hence, we reach a contradiction with f' being a model.

To summarize, we have shown that f^* must be a model for $\Phi - T_a$, and hence $\Phi - T_a$ is equivalent to Φ and the lemma holds. □

With Lemma 5, we can state and prove the main result of this section.

Theorem 3. *DQBF is fixed-parameter tractable parameterized by the treedepth of the augmented primal graph.*

Proof. Let Φ be a DQBF instance whose augmented primal graph $G^\star_\Phi$ has treedepth k. We begin by using Proposition 2 to compute an optimal treedepth decomposition T of $G^\star_\Phi$ of width k. We then exhaustively invoke Lemma 5 in a leaves-to-root fashion in order to reduce the number of children of any node at a given depth. Once that is no longer possible, we obtain an equivalent instance where the root r of T also satisfies the size bound given by Lemma 5; in particular, the resulting DQBF contains at most $h(k, k)$ many variables. To complete the proof, we can simply solve this bounded-size instance by an exhaustive brute-force search. □

5 Concluding Remarks

In this paper we have initiated the study of fixed-parameter tractability of the evaluation of dependency QBF (DQBF) under structural parameters. It turned out that the additional expressibility of DQBF compared to QBF has its price and requires new methods. We succeeded in obtaining two fixed-parameter tractability results: one that utilizes the combination of primal treewidth and distance to modularity as a parameter; and one that utilizes the treedepth of the augmented primal graph as a parameter.

One natural question is whether our parameterizations can be relaxed without losing fixed-parameter tractability. While our algorithms do exploit all the structural properties captured by these parameters, it would be desirable to obtain matching parameterized hardness results for weaker parameterizations. One particularly interesting case is that of using tree*width* (rather than treedepth as we did) of the augmented primal graph. We leave these as open questions for future work.

Another interesting question for future work is to come up with specific reductions from various NEXPTIME-complete Knowledge Representation and Reasoning problems to DQBF, and to analyze such reductions with respect to their preservation of structural properties.

Acknowledgments

This research was partially supported by the FWF grants J-4361, P31336, and P32441, as well as by the WWTF grant ICT19-065.

References

Atserias, A., and Oliva, S. 2014. Bounded-width QBF is PSPACE-complete. *J. of Computer and System Sciences* 80(7):1415–1429.

Audemard, G., and Simon, L. 2009. Predicting learnt clauses quality in modern SAT solvers. In Boutilier, C., ed., *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 399–404.

Balabanov, V.; Chiang, H.-J. K.; and Jiang, J. R. 2014. Henkin quantifiers and Boolean formulae: A certification perspective of DQBF. *Theoretical Computer Science* 523:86–100.

Beyersdorff, O.; Chew, L.; Schmidt, R. A.; and Suda, M. 2016. Lifting QBF resolution calculi to DQBF. In Creignou, N., and Le Berre, D., eds., *Theory and Applications of Satisfiability Testing – SAT 2016*, 490–499. Cham: Springer International Publishing.

Beyersdorff, O.; Blinkhorn, J.; Chew, L.; Schmidt, R.; and Suda, M. 2018. Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. *Journal of Automated Reasoning*.

Biere, A., and Lonsing, F. 2010. Integrating dependency schemes in search-based QBF solvers. In Strichman, O., and Szeider, S., eds., *Theory and Applications of Satisfiability Testing - SAT 2010*, volume 6175 of *Lecture Notes in Computer Science*, 158–171. Springer Verlag.

Bloem, R.; Könighofer, R.; and Seidl, M. 2014. SAT-based synthesis methods for safety specs. In McMillan, K. L., and Rival, X., eds., *Verification, Model Checking, and Abstract Interpretation - VMCAI 2014*, volume 8318 of *Lecture Notes in Computer Science*, 1–20. Springer Verlag.

Bodlaender, H. L.; Drange, P. G.; Dregi, M. S.; Fomin, F. V.; Lokshtanov, D.; and Pilipczuk, M. 2016. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.* 45(2):317–378.

Bubeck, U., and Büning, H. K. 2006. Dependency Quantified Horn formulas: Models and complexity. In Biere, A., and Gomes, C. P., eds., *Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *Lecture Notes in Computer Science*, 198–211. Springer Verlag.

Bubeck, U. 2010. *Model-based transformations for quantified Boolean formulas*. Ph.D. Dissertation, University of Paderborn.

Capelli, F., and Mengel, S. 2019. Tractable QBF by knowledge compilation. In Niedermeier, R., and Paul, C., eds., *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Chatterjee, K.; Henzinger, T. A.; Otop, J.; and Pavlogiannis, A. 2013. Distributed synthesis for ltl fragments. In *Formal Methods in Computer-Aided Design, FMCAD 2013*, 18–25. IEEE.

Chen, H. 2004. Quantified constraint satisfaction and bounded treewidth. In *Proceedings of ECAI 2004, the 16th Eureopean Conference on Artificial Intelligence*, 161–165. IOS Press.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Diestel, R. 2012. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer.

Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Eén, N., and Sörensson, N. 2003. An extensible SAT-solver. In Giunchiglia, E., and Tacchella, A., eds., *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, 502–518. Springer Verlag.

Eiben, E.; Ganian, R.; and Ordyniak, S. 2018. Small resolution proofs for QBF using dependency treewidth. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, 28:1–28:15.

Eiben, E.; Ganian, R.; and Ordyniak, S. 2020. Using decomposition-parameters for QBF: mind the prefix! *J. Comput. Syst. Sci.* 110:1–21.

Eiter, T.; Faber, W.; Leone, N.; Pfeifer, G.; and Polleres, A. 2004. A logic programming approach to knowledge-state planning: Semantics and complexity. *ACM Trans. Comput. Log.* 5(2):206–263.

Fichte, J. K.; Hecher, M.; and Pfandler, A. 2019. TE-ETH: lower bounds for qbfs of bounded treewidth. *CoRR* abs/1910.01047.

Finkbeiner, B., and Tentrup, L. 2014. Fast DQBF refutation. In Sinz, C., and Egly, U., eds., *Theory and Applications of Satisfiability Testing - SAT 2014*, volume 8561 of *Lecture Notes in Computer Science*, 243–251. Springer Verlag.

Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Berlin: Springer-Verlag.

Fröhlich, A.; Kovásznai, G.; Biere, A.; and Veith, H. 2012. A DPLL algorithm for solving DQBF. In *Pragmatics of SAT 2012*.

Fröhlich, A.; Kovásznai, G.; Biere, A.; and Veith, H. 2014. iDQ: Instantiation-based DQBF solving. In *Pragmatics of SAT 2014*.

Ganian, R., and Ordyniak, S. 2018. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.* 257:61–71.

Ganian, R., and Szeider, S. 2017. New width parameters for model counting. In *Theory and Applications of Satisfiability Testing - SAT 2017 - 20th International Conference, Melbourne, VIC, Australia, August 28 - September 1, 2017, Proceedings*, 38–52.

Garey, M. R., and Johnson, D. R. 1979. *Computers and Intractability*. San Francisco: W. H. Freeman and Company, New York.

Gaspers, S., and Szeider, S. 2012. Backdoors to satisfaction. In Bodlaender, H. L.; Downey, R.; Fomin, F. V.; and Marx, D., eds., *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, 287–317. Springer Verlag.

Gaspers, S.; Misra, N.; Ordyniak, S.; Szeider, S.; and Zivny, S. 2014. Backdoors into heterogeneous classes of SAT and CSP. In Brodley, C. E., and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, 2652–2658. AAAI Press.

Gottlob, G.; Scarcello, F.; and Sideri, M. 2002. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artificial Intelligence* 138(1-2):55–86.

Gutin, G. Z.; Jones, M.; and Wahlström, M. 2016. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM J. Discrete Math.* 30(4):2177–2205.

Henkin, L. 1961. Some remarks on infinitely long formulas. *Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959)* 167–183.

Iwata, Y.; Ogasawara, T.; and Ohsaka, N. 2018. On the power of tree-depth for fully polynomial FPT algorithms. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, 41:1–41:14.

Janota, M.; Klieber, W.; Marques-Silva, J.; and Clarke, E. M. 2012. Solving QBF with counterexample guided refinement. In Cimatti, A., and Sebastiani, R., eds., *Theory and Applications of Satisfiability Testing - SAT 2012*, volume 7317 of *Lecture Notes in Computer Science*, 114–128. Springer Verlag.

Janota, M. 2018. Towards generalization in QBF solving via machine learning. In McIlraith, S. A., and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence - AAAI 2018*. AAAI Press.

Kleine Büning, H.; Karpinski, M.; and Flögel, A. 1995. Resolution for quantified boolean formulas. *Inf. Comput.* 117(1):12–18.

Lutz, C. 2001. NEXPTIME-complete description logics with concrete domains. In Goré, R.; Leitsch, A.; and Nipkow, T., eds., *Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings*, volume 2083 of *Lecture Notes in Computer Science*, 45–60. Springer.

Nesetril, J., and de Mendez, P. O. 2012. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer.

Pan, G., and Vardi, M. Y. 2006. Fixed-parameter hierarchies inside PSPACE. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, 27–36. IEEE Computer Society.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

Peitl, T.; Slivovsky, F.; and Szeider, S. 2019. Dependency learning for QBF. *Journal of Artificial Intelligence Research* vol. 65:181–208.

Peterson, G.; Reif, J.; and Azhar, S. 2001. Lower bounds for multiplayer noncooperative games of incomplete information. *Computers & Mathematics with Applications* 41(7–8):957 – 992.

Rabe, M. N., and Tentrup, L. 2015. CAQE: A certifying QBF solver. In Kaivola, R., and Wahl, T., eds., *Formal Meth-*

ods in Computer-Aided Design - FMCAD 2015, 136–143. IEEE Computer Soc.

Samer, M., and Szeider, S. 2009a. Backdoor sets of quantified Boolean formulas. *Journal of Automated Reasoning* 42(1):77–97.

Samer, M., and Szeider, S. 2009b. Fixed-parameter tractability. In Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds., *Handbook of Satisfiability*. IOS Press. chapter 13, 425–454.

Samer, M., and Szeider, S. 2010. Constraint satisfaction with bounded treewidth revisited. *J. Comput. Syst. Sci.* 76(2):103–114.

Scholl, C.; Jiang, J. R.; Wimmer, R.; and Ge-Ernst, A. 2019. A PSPACE subclass of dependency quantified Boolean formulas and its effective solving. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 1584–1591. AAAI Press.

Tentrup, L., and Rabe, M. N. 2019. Clausal abstraction for DQBF. In Janota, M., and Lynce, I., eds., *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, 388–405. Springer.

Tobies, S. 1999. A NExpTime-complete description logic strictly contained in c2. In Flum, J., and Rodriguez-Artalejo, M., eds., *Computer Science Logic*, 292–306. Berlin, Heidelberg: Springer Berlin Heidelberg.

Tobies, S. 2001. *Complexity results and practical algorithms for logics in knowledge representation*. Ph.D. Dissertation, RWTH Aachen University, Germany.

Vizel, Y.; Weissenbacher, G.; and Malik, S. 2015. Boolean satisfiability solvers and their applications in model checking. *Proceedings of the IEEE* 103(11):2021–2035.

Williams, R.; Gomes, C.; and Selman, B. 2003. Backdoors to typical case complexity. In Gottlob, G., and Walsh, T., eds., *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003*, 1173–1178. Morgan Kaufmann.

Wimmer, R.; Scholl, C.; and Becker, B. 2019. The (D)QBF preprocessor hqspre - underlying theory and its implementation. *J. Satisf. Boolean Model. Comput.* 11(1):3–52.