Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Clausal Deletion Backdoors for QBF: a Parameterized Complexity Approach

Leif Eriksson[1] , **Victor Lagerkvist**[2] , **Sebastian Ordyniak**[3] , **George Osipov**[2,4]
Fahad Panolan[3] , **Mateusz Rychlicki**[1]
[1]Independent researcher
[2]Linköping University, Sweden
[3]University of Leeds, UK
[4]Royal Holloway, University of London, UK

Abstract

Determining the validity of a quantified Boolean formula (QBF) is a PSPACE-complete problem with rich expressive power. Despite interest in efficient solvers, there is, compared to problems in NP, a lack of positive theoretical results, and in the parameterized complexity setting one often has to restrict the quantifier prefix (e.g., bounding alternations) to obtain fixed parameter tractability (FPT). We propose a new parameter: the number of variables in clauses that has to be removed before reaching a tractable class (a *clause covering (CC) backdoor*). We are then interested in solving QBF in FPT time given a CC-backdoor of size k. We consider the three classical, tractable cases of QBF as base classes: *Horn*, *2-CNF*, and *linear equations*. We establish $\mathrm{W}[1]$-hardness for Horn but prove FPT for the others, and prove that in a precise, algebraic sense, we are only missing one important case for a full dichotomy. Our algorithms are non-trivial and depend on propagation, and Gaussian elimination, respectively, and are comparably unexplored for QBF.

1 Introduction

The *Boolean satisfiability problem* (SAT) is the canonical NP-complete problem. Many hard problems can today be solved efficiently by a suitable SAT encoding together with off-the-shelf solvers (Biere, Heule, and van Maaren 2021). While this often works well for problems in NP, many fundamental problems in AI and machine learning are *harder* than NP, with PSPACE crystallizing as the most fundamental class. Examples include neural network training (Abrahamsen, Kleist, and Miltzow 2021), graph neural network satisfiability (Nunn et al. 2024), and quantized graph neural network verification (Sälzer, Schwarzentruber, and Troquard 2025), all of which are PSPACE-complete or contained in PSPACE. In symbolic AI, planning problems like STRIPS and SAS$^+$ also fall inside PSPACE.

Solving PSPACE problems is thus a pressing issue. The canonical hard problem is the *quantified Boolean formula* problem (QBF), deciding the validity of $\Psi = Q_1 x_1 \dots Q_n x_n . \phi(x_1, \dots, x_n)$ where $Q_i \in \{\forall, \exists\}$ and ϕ is in CNF. Unfortunately, we lack fast techniques for solving QBF instances. Theoretically, Ψ can be solved in 2^n time by branching: $\exists$ requires one successful branch while $\forall$ requires both. While SAT's 2^n complexity can be improved to c_k^n for $c_k < 2$ for any k-CNF, no such improvement is known even for $k = 3$ for QBF, and it is conjectured impossible unless the *strong exponential-time hypothesis* (SETH) (Calabro, Impagliazzo, and Paturi 2009) fails (Calabro, Impagliazzo, and Paturi 2013).

Hence, we move to *parameterized complexity* to identify parameters k (corresponding to a "hidden structure") leading to tractability. We seek *fixed-parameter tractability* (FPT), i.e., $f(k) \cdot ||\Psi||^{\mathcal{O}(1)}$ time for a computable f. FPT is also characterized by a *kernel*: a polynomial-time procedure reducing an instance (Ψ, k) to (Ψ', k') where $||\Psi'|| + k' \leq f(k)$. A well-known parameter is *treewidth*, measuring how tree-like a formula is. For instance, while QBF remains PSPACE-complete even for constant treewidth (Atserias and Oliva 2014) it is FPT parameterized by treewidth plus quantifier alternations. However, even in this case, f is a *power tower* which, even for small values, requires more operations than there are atoms in the universe.

The most successful parameter besides treewidth is arguably *backdoor size*, which measures how close an instance is to a tractable class. For SAT, a (strong) backdoor for ϕ is a set $B \subseteq V(\phi)$ such that $\phi[\tau]$ belongs to a fixed, tractable class (e.g., 2-CNF) for each partial assignment $\tau \colon B \to \{0, 1\}$. Here, $\phi[\tau]$ is the formula obtained by replacing variables in B with their values under τ and simplifying, and $V(\phi)$ is the set of variables in ϕ. However, backdoors cannot readily be applied to QBF since branching on B may violate the quantifier prefix. This can be circumvented with restrictions on the quantifier prefix (Samer and Szeider 2009) (e.g., that backdoor variables are to the left), but this only provides a meaningful speed-up for a very narrow class of QBF formulas.

In this paper we investigate an alternative QBF backdoor notion which avoids any such restrictions. We primarily consider 2-CNF, HORN/DualHORN (at most one positive/negative literal in each clause), and AFF (systems of linear equations over GF(2)). We assume that the formulas in each class are conjunctions of clauses/atoms, e.g., each formula in 2-CNF is a conjunction of clauses of arity 2. For a class of formulas $\mathcal{C}$ we write $\mathrm{QBF}(\mathcal{C})$ for the QBF problem restricted to instances in $\mathcal{C}$. A first idea could be to measure the distance to a tractable class $\mathcal{C}$ with respect to the number of *clauses* that are outside $\mathcal{C}$. I.e., if we view ϕ as a set of clauses, we let $\phi_{\mathcal{C}} \subseteq \phi$ be the maximal subset of clauses in ϕ such that $\phi_{\mathcal{C}}$ belongs to $\mathcal{C}$, and parameterize

by $|\phi \setminus \phi_{\mathcal{C}}|$. Unfortunately, allowing even one out-of-class clause to be added to a 2-SAT or a Horn formula makes the problem PSPACE-complete (Fichte et al. 2023). Thus, we propose parameterization by $|V(\phi \setminus \phi_{\mathcal{C}})|$, i.e., the number of variables that cover all out-of-class clauses, leading to the following definition.

Definition 1.1. Let $\mathcal{C}$ be a class of formulas. Let $\Phi = \mathcal{Q}.\phi$ be a QBF formula. A set B of variables is a *clause covering backdoor* (CC-backdoor) for Φ if $\phi \setminus C \in \mathcal{C}$ for a subset $C \subseteq \phi$ of clauses such that $V(C) = B$.

Note that the *CC-backdoor detection problem* is trivial for all classical base classes: if the tractable class (e.g., 2-CNF) is fixed then we simply collect all out-of-class clauses, compute the set of variables covering them, and see whether it has the required size. Hence, the computationally interesting problem is *CC-backdoor evaluation*: given a QBF formula Ψ and CC-backdoor of size k, can Ψ be evaluated in FPT time with respect to k?

Given the lack of FPT results for QBF with unbounded quantifier alternations, there is perhaps not much reason to be optimistic. Furthermore, the standard quantifier elimination does not seem directly compatible with CC-backdoor evaluation problem: by eliminating backdoor variables we get a complicated disjunctive formula which does not appear to be easier to solve than the original formula. The main complication, then, is that the backdoor variables may be involved in the in-the-class clauses, i.e., some of them may be in $V(\phi_{\mathcal{C}})$, so it is not possible to solve the backdoor part independently. Moreover, they can occur anywhere in the quantifier prefix.

Despite this grim outlook we manage to prove FPT for 2-CNF and AFF. For 2-CNF we use an elaborate branching strategy with look-ahead. Essentially, by going from left to right in the quantifier prefix we prove that one either (1) can branch and always "hits" a variable in the backdoor and therefore decreases the parameter k, or (2) if branching would not affect the backdoor variables, then one can disregard one of the branches. Then, depending on the current quantifier ($\exists$ or $\forall$) we either continue the process or correctly report unsatisfiability. Concurrently, we repeatedly propagate the input formula to ensure that we do not miss any immediate implications. Put together, this results in an FPT algorithm where $f(k) = 2^k$ and with a polynomial factor dominated by n^3 from propagation. Under SETH, the dependence on k cannot be improved: in the special case when all clauses are out-of-class, the CC-backdoor evaluation problem is as hard as solving an arbitrary QBF instance, which cannot be solved substantially faster than 2^n time under SETH. We stress that, a priori, it is rather surprising that non-trivial branching is applicable: while the algorithm can be stated in a rather succinct form it is non-trivial to prove correctness.

The affine case, however, cannot be solved in FPT-time by branching in the order of quantifier prefix, and requires a drastically different strategy. For this case it is conceptually easier to construct a kernel instead of deriving an FPT algorithm. Naturally, the main complication is that our input instance $\Psi = Q_1 x_1 \dots Q_n x_n.\phi$ contains both affine equations (easily solvable by Gaussian elimination) and arbitrary clauses (where Gaussian elimination is not applicable). To handle this we isolate the influence of backdoor variables by a carefully orchestrated sequence of choosing which variables to pivot on, and finally reach the desired conclusion where at most $2k$ variables remain. By branching on these variables and observing that at most k of them are not determined by previous choices, i.e., only k need to be branched upon, we can then solve the problem in $\mathcal{O}^*(2^k)$ time, and the running time cannot be improved under the SETH.

Inspired by these two positive results it is natural to consider to which extent our classification is complete, i.e., are there more FPT cases among the natural tractable classes of QBF? We begin studying this in Section 5 where we as a first step prove a negative result for HORN (and DualHORN): CC-backdoor evaluation is *not* FPT unless a widely believed conjecture in parameterized complexity is false (FPT $\neq$ W[1]). This does not cover everything, however, and we manage to extend W[1]-hardness to the class of *implicative hitting set-bounded* formulas IHSB$_-$ $\subseteq$ HORN (negative clauses of any length, and binary implication) and its dual IHSB$_+$ $\subseteq$ DualHORN (positive clauses of any length, and binary implication). It is easy to show that the problem becomes FPT if we remove implication. However, in contrast, keeping implication and all positive (or negative) clauses of arity at most d (d-IHSB$_+$, d-IHSB$_-$) results in a problem where it seems much more challenging to obtain FPT. With this troublesome case in mind we investigate (in Section 6) exactly how far away we are from a complete CC-evaluation dichotomy (FPT versus W[1]-hard). We do this not only for all possible sets of backdoor clauses, but for the extension of QBF to arbitrary constraints, the *quantified constraint satisfaction problem* (QCSP). Here, we show that the so-called *algebraic approach* is compatible with CC-backdoors, which greatly reduces the number of cases that we need to consider. With this machinery we manage to show that the CC-evaluation problem is always either FPT or W[1]-hard *except* for d-IHSB$_+$ (and d-IHSB$_-$) for $d \geq 3$ where the complexity status is unknown.

Put together, we thus manage to solve arbitrary QBF instances in optimal time with respect to the number of variables in the CC-backdoor, *without* any assumptions on the quantifier prefix. We accomplished this with two different algorithmic schemes, branching and Gaussian elimination, and the latter is overlooked in the QBF literature. CC-backdoors may thus open up the usage of algorithmic schemes that do not work for the full QBF problem, while still being reasonably expressive since we do not have to make any assumptions on the quantifier prefix. Hence, we have obtained a novel paradigm for solving QBF, and in the long term, possibly more PSPACE-complete problems. We discuss this and other open questions in Section 7.

2 Preliminaries

We begin by formally introducing the quantified Boolean formula problem and some additional notation.

2.1 Clauses and Relations

Boolean expressions, formulas, variables, literals, conjunctive normal form (CNF), clauses and atoms are defined in the standard way (cf. (Biere, Heule, and van Maaren 2021)). We treat $1/\top$ and $0/\bot$ as the truth values "true" and "false", respectively, and formulas in CNF as sets of clauses where each clause is a set of literals. More generally, it is convenient to use a relational perspective where atoms are expressions of the form $R(x_1, \dots, x_r)$ where $x_1, \dots, x_r$ are variables and $R \subseteq \{0,1\}^r$ is an r-ary relation. A function $f\colon \{x_1, \dots, x_r\} \to \{0,1\}$ satisfies an atom $R(x_1, \dots, x_r)$ if $(f(x_1), \dots, f(x_r)) \in R$. Note that clauses can be defined in this way, e.g., the clause $(x \vee y \vee \neg z)$ is equivalent to an atom $R(x, y, z)$ where $R = \{0,1\}^3 \setminus \{(0,0,1)\}$.

Given a Boolean formula ϕ, we write $V(\phi)$ for the set of variables in ϕ. For a (partial) assignment $\tau : V' \to \{0,1\}$, where $V' \subseteq V(\phi)$, and an atom $R(x_1, \dots, x_r)$, we let $R(x_1, \dots, x_r)[\tau]$ be the atom obtained by (1) removing any tuple $(b_1, \dots, b_r) \in R$ if there is an $1 \leq i \leq r$ where $x_i \in V'$ and $\tau(x_i) \neq b_i$, and then (2) projecting away any x_i such that $x_i \in V'$. This easily extends to conjunctions of atoms, and for a formula ϕ we thus let $\phi[\tau]$ be the formula obtained by simplifying each atom according to τ. If $V' = \{x\}$ and $\tau(x) = b$, then we simply write $\phi[x = b]$. In the special case when ϕ is in CNF, $\phi[\tau]$ is the formula obtained from ϕ by removing all clauses satisfied by τ and removing all literals from clauses that are set to 0 by τ.

Let us now define the following classes of conjunctive Boolean formulas.

1. d-CNF for the class of formulas in CNF where each clause has at most d literals.
2. HORN for the class of formulas in CNF where each clause has at most one positive literal.
3. IHSB$_-$ $\subseteq$ HORN restricted to purely negative clauses, unit clauses, and implications $(x \Rightarrow y)$, and d-IHSB$_-$ $\subseteq$ IHSB$_-$ for the restriction to at most d-ary clauses.
4. AFF for the class of conjunctive formulas with atoms of the form $(\ell_1 \oplus \ell_2 \oplus \cdots \oplus \ell_k) = b$ where $b \in \{0,1\}$ and each $\ell_i = b_i x_i$ for a variable x_i and coefficient $b_i \in \{0,1\}$. We represent the atoms of affine formulas by a pair (A, b), where A is a set of variables and $b \in \{0,1\}$. We refer to such atoms as *equations*.
5. If $\mathcal{C}$ and $\mathcal{C}'$ are both classes of formulas then we write $\mathcal{C} + \mathcal{C}'$ for the class of formulas where each atom is in $\mathcal{C} \cup \mathcal{C}'$. CNF+AFF for the class of conjunctive formulas where each atom is either in CNF or in AFF.

We sometimes also consider dual cases and write DualHORN, IHSB$_+$ and d-IHSB$_+$ for the classes obtained by flipping the literals of HORN, IHSB$_-$, and d-IHSB$_-$. We say that a conjunctive formula is in *clausal form* if all its atoms are clauses.

Example 2.1. Consider the propositional formula

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \oplus x_2 = 1)$$

in CNF+AFF. Here $(x_1 \vee x_2 \vee x_3) \in$ CNF and $(x_1 \oplus x_2 = 1) \in$ AFF. In the relational perspective we view ϕ as $R(x_1, x_2, x_3) \wedge R'(x_1, x_2)$ where $R = \{0,1\}^3 \setminus \{(0,0,0)\}$ and $R' = \{(0,1), (1,0)\}$.

The *satisfiability* problem (SAT) for ϕ is then simply to determine whether there exists a satisfying assignment $f\colon V(\phi) \to \{0,1\}$, and by restricting ϕ to formulas in d-CNF we obtain the well-known *d-SAT* problem.

When working with 2-CNF it will be convenient to assume that the set of clauses is maximally propagated in the sense that no further clauses can be inferred via resolution. For example, the formula $(\neg x \vee y) \wedge (\neg y \vee z)$ is not propagated because it also implies $(\neg x \vee z)$. Adding the latter makes the formula propagated. We formalize this as follows: let $C_1 = (\ell \vee A)$ and $C_2 = (\neg \ell \vee B)$ be two clauses in a CNF formula, where ℓ is a literal. The *resolution* of C_1 and C_2 derives the *resolvent* clause $C = (A \vee B)$. We say that a 2-CNF formula ϕ is *propagated* if (1) $\phi = \{\bot\}$ only if $\bot \in \phi$ or $\emptyset \in \phi$[1], (2) ϕ is closed under resolution, and (3) no clause is a super-clause of another clause. Let $\texttt{prop}(\phi)$ denote the 2-CNF formula ϕ' such that ϕ' is propagated and $\phi \equiv \phi'$. It is well-known that given a 2-CNF formula ϕ, the propagated formula ϕ' can be computed in polynomial time (see e.g (Biere, Heule, and van Maaren 2021)).

Proposition 2.2. *Let ϕ be a* 2-CNF *formula. For any* 2-CNF *formula ϕ, $\texttt{prop}(\phi)$ can be computed in $\mathcal{O}(|V(\phi)|^3)$ time, and $\texttt{prop}(\phi)$ has at most $|V(\phi)|^2$ clauses.*

2.2 The Quantified Boolean Formula Problem

A *quantified Boolean formula (QBF)* is of the form $\mathcal{Q}.\phi$, where $\mathcal{Q} = Q_1 x_1 \dots Q_n x_n$ with $Q_i \in \{\forall, \exists\}$ for all $1 \leq i \leq n$ is the *(quantifier) prefix*, $x_1, \dots, x_n$ are variables, and ϕ is a Boolean formula on these variables called the *matrix*. For any set $S = \{x_i, \dots, x_j\} \subseteq V(\phi)$ the variable x_i said to be the *outermost* variable if no other variable in S occurs before x_i in $\mathcal{Q}$ and similarly x_n the *innermost* if no other variable in S occurs after x_i in $\mathcal{Q}$. When speaking of the entire quantifier prefix without any specific subset then x_1 is the outermost and x_n the innermost. If $Q_i = \forall$, we say that x_i is a *universal variable*, and if $Q_i = \exists$, we say that x_i is an *existential variable*.

The truth value of a QBF $\mathcal{Q}.\phi$ is defined recursively. Let $\mathcal{Q} = Q_1 x_1 \dots Q_n x_n$ and $\mathcal{Q}' = Q_2 x_2 \dots Q_n x_n$. Then $\mathcal{Q}.\phi$ is *true* if

- ϕ is a true Boolean expression, or
- $Q_1 = \exists$ and $\mathcal{Q}'.\phi[x_1 = 0]$ *or* $\mathcal{Q}'.\phi[x_1 = 1]$ is true, or
- $Q_1 = \forall$ and $\mathcal{Q}'.\phi[x_1 = 0]$ *and* $\mathcal{Q}'.\phi[x_1 = 1]$ are true.

Otherwise, $\mathcal{Q}.\phi$ is *false*. For a QBF $\Phi = \mathcal{Q}.\phi$ and a partial assignment $\tau : V' \to \{0,1\}$, where $V' \subseteq V(\Phi)$, we write $\Phi[\tau]$ for the QBF formula $\mathcal{Q}'.\phi[\tau]$, where $\mathcal{Q}'$ is obtained from $\mathcal{Q}$ after removing the variables in V'.

Definition 2.3. For a class of formulas $\mathcal{C}$ we write QBF($\mathcal{C}$) for the computational problem of deciding whether a QBF formula $\mathcal{Q}.\phi$, where $\phi \in \mathcal{C}$, is true. We may also refer to an instance of this problem as a QBF($\mathcal{C}$) formula.

[1] Recall that $\bot$ is the constant 0, while $\emptyset$ is the empty clause, logically equivalent to $\bot$.

The evaluation of a QBF $Q_1x_1 \dots Q_nx_n.\phi$ can be seen as a two-player game between the *universal* and *existential* players. In the ith step the player Q_i assigns value to the variable x_i. The existential player wins the game if ϕ evaluates to true under the assignment constructed in the game and the universal player wins if ϕ evaluates to false.

A *strategy* for a player is a rooted binary tree T of depth $n+1$ where nodes at level $j \in \{0, \dots, n-1\}$ are labeled x_{j+1}, leaves are labeled $\top$ or $\bot$, and edges are labeled 0 or 1. If T is an existential strategy, nodes labeled by existential variables have one child, and those by universal variables have two; the roles are reversed for universal strategies. Each node t corresponds to the assignment τ_t^T given by the vertex and edge labels on the root-to-t path, defining $\phi[\tau_l^T]$ as the label of each leaf l. An existential strategy is *winning* if all leaves are $\top$, and a universal strategy is *winning* if all leaves are $\bot$.

Proposition 2.4. *A QBF formula Φ is true ($\Phi \Leftrightarrow \top$) if and only if the existential player has a winning strategy, and Φ is false ($\Phi \Leftrightarrow \bot$) if and only if the universal player has a winning strategy.*

Example 2.5. Consider the QBF formula

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \exists x_4 \exists x_5 \,.\, (\phi_1 \wedge \phi_2),$$

where

$$\phi_1 = (x_1 \vee x_3) \wedge (\neg x_1 \vee x_4) \wedge (x_3 \vee x_4) \wedge (x_2 \vee x_5)$$

is in 2-CNF (and is propagated) and

$$\phi_2 = (\neg x_3 \vee \neg x_4 \vee \neg x_5).$$

Then $V(\phi_2) = \{x_3, x_4, x_5\}$ is a CC-backdoor of size 3 into 2-CNF. Note that the backdoor variables also appear in ϕ_1 and lie in the inner part of the prefix, behind the universal x_2, so they cannot be branched on directly without first handling earlier quantifiers.

2.3 Parameterized Complexity

We use standard notation and primarily follow (Downey, Fellows, and others 2013) and (Fomin et al. 2019). For a finite alphabet Σ a *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$. The problem L is *fixed-parameter tractable* (or, in FPT) if there is an algorithm deciding whether an instance $(I, k) \in \Sigma^* \times \mathbb{N}$ is in L in time $f(k) \cdot |I|^c$, where f is some computable function and c is a constant independent of (I, k). To state FPT results we sometimes use the $\mathcal{O}^*$ notation which suppresses polynomial factors in the input size, i.e., we simply write $\mathcal{O}^*(f(k))$ rather than $f(k) \cdot |I|^c$. An equivalent definition of FPT can be given by a *kernelization (algorithm)* for L, i.e., an algorithm that takes $(I, k) \in \Sigma^* \times \mathbb{N}$ as input and in time polynomial in $|I| + k$, outputs $(I', k') \in \Sigma^* \times \mathbb{N}$ such that $(I, k) \in L$ if and only if $(I', k') \in L$, and $|I'|, k' \leq h(k)$ for some computable function h.

Let $L, L' \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A mapping $P : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ is a *fixed-parameter (*FPT*) reduction from L to L'* if there exist computable functions $f, p : \mathbb{N} \to \mathbb{N}$ and a constant c such that (1) $(I, k) \in L$ if and only if $P(I, k) = (I', k') \in L'$, (2) $k' \leq p(k)$, and (3) $P(I, k)$ can be computed in $f(k) \cdot |I|^c$ time. We write $L \leq_{\text{FPT}} L'$ if this holds, and if, additionally, $L' \leq_{\text{FPT}} L$ then L and L' are said to be *FPT-equivalent*. Parameterized complexity also contains a complementary theory of hardness. Here, the hard classes $\text{W}[1] \subseteq \text{W}[2] \subseteq \dots$, form a hierarchy of classes. We say that a problem is W[1]*-hard* if it admits an FPT-reduction from INDEPENDENT SET (parameterized by the number of vertices in the independent set). For sharper lower bounds, stronger assumptions are sometimes necessary. For $d \geq 3$, let c_d denote the infimum of all constants c such that d-SAT is solvable in 2^{cn} time by a deterministic algorithm. The *exponential-time hypothesis* (ETH) then states that $c_3 > 0$, i.e., that 3-SAT is not solvable in subexponential time. The *strong exponential-time hypothesis* (SETH) additionally conjectures that the limit of the sequence $c_3, c_4, \dots$ tends to 1, which in particular is known to imply that the satisfiability problem for clauses of arbitrary length is not solvable in 2^{cn} time for *any* $c < 1$ (Calabro, Impagliazzo, and Paturi 2009).

3 Backdoors into 2-CNF

Let $\Phi = Q_1x_1 \dots Q_nx_n.\phi$ be a QBF(CNF) formula. Suppose $\phi = \phi_1 \wedge \phi_2$, where $\phi_1 \in$ 2-CNF and $|V(\phi_2)| = k$, i.e., $V(\phi_2)$ is a CC-backdoor to 2-CNF. Before presenting our FPT algorithm we introduce some simplifying notation and formalia when branching on $b \in \{0, 1\}$ for the outermost variable x_1, and analyzing the unit propagations caused by this assignment on the remaining instance. For simplicity we will assume $\phi_1 = \texttt{prop}(\phi_1)$ and that ϕ does not contain any unit-clauses, as if any such exist, we can assign values to any variable in such a clause trivially. We first define the two 1-CNF formulas

$$u^0_{\phi_1} = \texttt{prop}(\phi_1 \wedge (\neg x_1)) \setminus \phi_1$$

and

$$u^1_{\phi_1} = \texttt{prop}(\phi_1 \wedge (x_1)) \setminus \phi_1,$$

which are thus simply the variables affected by assigning 0 or 1 to x_1. Note that $u^0_{\phi_1}$ and $u^1_{\phi_1}$ define partial assignments. Formally, for $b \in \{0, 1\}$ we define the function $U^b_\Phi \colon V_\exists(u^b_{\phi_1}) \cup \{x_1\} \to \{0, 1\}$ as follows. For x_1 the function U^b_Φ is defined as $U^b_\Phi(x_1) = b$. For all $x \in V_\exists(u^b_{\phi_1})$, $U^b_\Phi(x) = 1$ if $u^b_{\phi_1}$ contains a unit clause (x) and $U^b_\Phi(x) = 0$ if $u^b_{\phi_1}$ contains a unit clause $(\neg x)$. Clearly, $u^0_{\phi_1}$ and $u^1_{\phi_1}$ are computable in polynomial time, and so are U^0_Φ and U^1_Φ.

Let $\text{D}(U^b_\Phi) = V_\exists(u^b_{\phi_1}) \cup \{x_1\}$ denote the domain of U^b_Φ, i.e., variables affected by setting $x_1 = b$. The basic idea is now to carefully branch on the first variable: if any choice affects a backdoor variable in the 2-CNF fragment, then we have decreased the backdoor size; otherwise, no branching is necessary under certain technical assumptions.

Lemma 3.1. *Let $\Phi = \mathcal{Q}.(\phi_1 \wedge \phi_2)$ be a* QBF(CNF) *instance with $\phi_1 \in$ 2-CNF. If $\text{D}(U^b_\Phi) \cap V(\phi_2) = \emptyset$ and $\mathcal{Q}.\phi_1[U^b_\Phi] \Leftrightarrow \top$ for some $b \in \{0, 1\}$ then $\Phi[U^{\neg b}_\Phi] \Rightarrow \Phi[U^b_\Phi]$.*

Proof. Suppose $\Phi[U^{\neg b}_\Phi] \Leftrightarrow \top$, i.e., the existential player has a winning strategy $T^{\neg b}_\exists$ on $\Phi[U^{\neg b}_\Phi]$. It suffices to show that in

this case the existential player also has a winning strategy $T^b_\exists$ on $\Phi[U^b_\Phi]$. We define the strategy as follows. Let $\alpha : U^0_\Phi \cup U^1_\Phi \to \{0,1\}$ be an assignment such that $\alpha(x) = U^b_\Phi(x)$ if $x \in \mathrm{D}(U^b_\Phi)$ and $\alpha(x) = U^{\neg b}_\Phi(x)$ otherwise, i.e., α defaults to the assignment U^b_Φ and uses $U^{\neg b}_\Phi$ otherwise. Then the existential player will follow α on the variables in $\mathrm{D}(\alpha)$ and reuse $T^{\neg b}_\exists$ on the remaining existential variables.

Formally, the strategy $T^b_\exists$ is defined as follows. Consider an existential variable x in the node t of $T^b_\exists$. If $x \in \mathrm{D}(\alpha)$, then choose $x = \alpha(x)$. Otherwise, let τ_t be the root-to-t assignment in $T^b_\exists$. Since $\Phi[U^b_\Phi]$ and $\Phi[U^{\neg b}_\Phi]$ have the same set of universal variables appearing in the same order in the quantifier prefix, there is a node t' in $T^{\neg b}_\exists$ labeled with x such that the root-to-t' assignment $\tau'_{t'}$ in $T^{\neg b}_\exists$ coincides with τ_t on all universal variables. There may be multiple such nodes t', so pick an arbitrary one. Assign $x = \tau'_{t'}(x)$.

We claim that $T^b_\exists$ is a winning strategy. Consider a root-to-leaf assignment τ. It suffices to show that τ satisfies every clause c in $\phi_1 \wedge \phi_2$ that is not satisfied by U^b_Φ. We claim that no such clause c contains a variable from $\mathrm{D}(U^b_\Phi)$. Indeed, if $c \in \phi_1$, then U^b_Φ neither satisfies nor falsifies c because $\mathcal{Q}.\phi_1[U^b_\Phi] \Leftrightarrow \top$. Moreover, by definition of unit propagation, U^b_Φ cannot falsify a literal in c without satisfying the other literal, hence it does not assign a value to either. Furthermore, if $c \in \phi_2$, then the claim holds because $\mathrm{D}(U^b_\Phi)$ is disjoint from $V(\phi_2)$.

By construction of $T^b_\exists$, there exists a root-to-leaf assignment τ in $T^{\neg b}_\exists$) such that $\tau(x) = \alpha(x)$ for all $x \in \mathrm{D}(\alpha)$ and $\tau(x) = \tau'(x)$ otherwise. Moreover, every clause in $\phi_1 \wedge \phi_2$ is satisfied by $U^{\neg b}_\Phi$ or τ' because $T^{\neg b}_\Phi$ is a winning strategy. If c is satisfied by $U^{\neg b}_\Phi$, then $U^{\neg b}_\Phi$ must satisfy a literal in c over a variable $x \in \mathrm{D}(U^{\neg b}_\Phi) \setminus \mathrm{D}(U^b_\Phi)$, and $\tau(x) = \alpha(x) = U^{\neg b}_\Phi$, hence τ satisfies c. Otherwise, c is not satisfied by either $U^{\neg b}_\Phi$ or U^b_Φ and is satisfied by τ'. Then c does not contains any variables from $\mathrm{D}(\alpha)$, and τ agrees with τ' on all variables in c, implying that τ satisfies c. □

We obtain the following corollary depending on the quantifier of the outermost variable.

Corollary 3.2. *Let $\Phi = Q_1 x_1 \dots Q_n x_n.(\phi_1 \wedge \phi_2)$ be a* QBF(CNF) *instance with a CC-backdoor $V(\phi_2)$ into* 2-CNF. *Let $b \in \{0,1\}$. If $\mathrm{D}(U^b_\Phi) \cap V(\phi_2) = \emptyset$ and $\phi_1[U^b_\Phi] \Leftrightarrow \top$, then*

- $\Phi \Leftrightarrow \Phi[U^b_\Phi]$ *if* $Q_1 = \exists$, *and*
- $\Phi \Leftrightarrow \Phi[U^{\neg b}_\Phi]$ *if* $Q_1 = \forall$.

We solve Φ by greedily assuming an assignment to the first unassigned variable in the quantifier prefix, and use one step look-ahead combined with unit propagation. If an assignment directly fails, or affects the backdoor, we try the other assignment. If both assignments affect the backdoor variables we branch on both of them but in this case successfully reduce k. If both assignments fail, we simply answer no. For each variable in the quantifier prefix we thus use only polynomial time to do propagation, and either do not branch at all, or branch while simultaneously decreasing the backdoor size.

Theorem 3.3. QBF(CNF) *is solvable in $\mathcal{O}^*(2^k)$ time, where k is the size of CC-backdoor to* 2-CNF.

Proof. Let $\Phi = \mathcal{Q}.(\phi_1 \wedge \phi_2)$, where $\phi_1 \in$ 2-CNF and $k = |V(\phi_2)|$. Let $\mathcal{Q} = Q_1 x_1 \dots Q_n x_n$. We decide whether to branch on x_1 as follows.

1. If $\mathcal{Q}.\phi_1[U^1_\Phi] \Leftrightarrow \bot$ and $\mathcal{Q}.\phi_1[U^0_\Phi] \Leftrightarrow \bot$, then reject.
2. If $\mathcal{Q}.\phi_1[U^b_\Phi] \Leftrightarrow \top$ and $\mathcal{Q}.\phi_1[U^{\neg b}_\Phi] \Leftrightarrow \bot$ for some $b \in \{0,1\}$ then there is no need to branch and we continue with $\Phi[U^b_\Phi]$ if $Q_1 = \exists$ and otherwise reject.
3. If $\mathcal{Q}.\phi_1[U^b_\Phi] \Leftrightarrow \top$, $\mathcal{Q}.\phi_1[U^{\neg b}_\Phi] \Leftrightarrow \top$ and $\mathrm{D}(U^b_\Phi) \cap V(\phi_2) = \emptyset$ for some $b \in \{0,1\}$ then we use the assignment prescribed by Corollary 3.2.

Otherwise, $\mathrm{D}(U^1_\Phi) \cap V(\phi_2) \neq \emptyset$ and $\mathrm{D}(U^0_\Phi) \cap V(\phi_2) \neq \emptyset$, so regardless of whether we branch on $x_1 = 1$ or $x_1 = 0$, the parameter k decreases by at least one. Hence, if we repeat this for every outermost variable then we in each iteration either do not branch at all, or branch on two possible values where we in each branch shrink the parameter. The complexity is thus bounded by 2^k and a polynomial factor, which is dominated by computing $\mathtt{prop}(\phi_1)$ (recall Proposition 2.2) for every variable in the quantifier prefix, via U^b_Φ.

For correctness, step 1 and 2 are trivial, 3 follows by Lemma 3.1 and Corollary 3.2, and otherwise we are branching and hence inherit correctness directly from the definition of evaluating QBFs. As these four cases are exhaustive, correctness for the full algorithm follows directly. □

Example 3.4. We illustrate the algorithm of Theorem 3.3 on the formula Φ of Example 2.5. The outermost variable x_1 is existentially quantified. Setting $x_1 = 0$ propagates $(x_1 \vee x_3)$ to the unit clause (x_3), hence $\mathrm{D}(U^0_\Phi) = \{x_1, x_3\}$ with $U^0_\Phi(x_1) = 0$ and $U^0_\Phi(x_3) = 1$. Symmetrically, $x_1 = 1$ propagates $(\neg x_1 \vee x_4)$ to (x_4), hence $\mathrm{D}(U^1_\Phi) = \{x_1, x_4\}$. Both $\mathrm{D}(U^0_\Phi)$ and $\mathrm{D}(U^1_\Phi)$ intersect $V(\phi_2)$, so we branch on x_1, but in each branch the parameter k strictly decreases. For instance, on the branch $x_1 = 0$, $x_3 = 1$, the clause ϕ_2 simplifies to $(\neg x_4 \vee \neg x_5) \in$ 2-CNF and the formula

$$\forall x_2 \, \exists x_4 \, \exists x_5 \, . \, (x_2 \vee x_5) \wedge (\neg x_4 \vee \neg x_5)$$

is in 2-CNF and can be solved in polynomial time. The branch $x_1 = 1$ is symmetric, so Φ evaluates to true.

This running time is optimal under the SETH since an asymptotically faster algorithm immediately implies a faster algorithm for QBF(CNF) by letting the backdoor cover the entire instance.

4 Backdoors into Affine

We proceed by analyzing the affine case which is notably different from 2-CNF. The main idea is to simultaneously carefully manipulate which variables only occur in a single row while computing a kernel via repeated applications of Gaussian elimination.

Let $\phi = \{(A_1, b_1), \dots, (A_m, b_m)\}$ be a system of affine constraints. We define basic linear-algebraic operations which will be used throughout the algorithm. Suppose $A_i \neq \emptyset$ and $x \in A_i$. By *pivoting on x and i* we denote the

operation of removing x from every equation except (A_i, b_i) and substituting its value according to (A_i, b_i). Formally,

$$\begin{aligned}\mathrm{pivot}(\phi, x, i) = &\{(A_j, b_j) \mid j \in [m],\ x \notin A_j\} \\ &\cup \{(A_j \triangle A_i,\ b_j \oplus b_i) \mid j \in [m],\ j \neq i,\ x \in A_j\} \\ &\cup \{(A_i, b_i)\}.\end{aligned}$$

where $\triangle$ denotes the symmetric difference of two sets. We say that a variable is *private* if it only occurs in one equation. Observe that x is a private variable in $\mathrm{pivot}(\phi, x, i)$. It is also easy to see that for every quantifier prefix $\mathcal{Q}$ on $V(\phi)$, we have $\mathcal{Q}.\phi \Leftrightarrow \mathcal{Q}.\mathrm{pivot}(\phi, x, i)$ because ϕ and $\mathrm{pivot}(\phi, x, i)$ have the same set of satisfying assignments. Moreover, if x is the innermost variable in A_i and it is existential, we can now *eliminate* it from the system without affecting the truth value of the formula. We define $\mathrm{elim}(\phi, x, i) = \mathrm{pivot}(\phi, x, i) \setminus (A_i, b_i)$ and note that if x is innermost in A_i and existential, then for all quantifier prefixes $\mathcal{Q}$, we have $\mathcal{Q}.\phi \Leftrightarrow \mathcal{Q}'.\mathrm{elim}(\phi, x, i)$, where $\mathcal{Q}'$ is obtained by removing x from $\mathcal{Q}$. Indeed, since x is the innermost variable in (A_i, b_i) and private, for any partial assignment $\alpha : A_i \setminus \{x\} \to \{0, 1\}$, the existential player can set x to $\bigoplus_{y \in A_i \setminus \{x\}} \alpha(y) \oplus b_i$ to satisfy (A_i, b_i) without affecting any other equation.

We are now ready to introduce our first step towards a kernel. Given a CNF+AFF, we will effectively remove every equation from the AFF part where the innermost variable does not occur in the CNF part. If the backdoor set is of size k, this already yields a formula with at most k equations in the AFF part. With an additional step we ensure that every such formula contains at most one non-backdoor variable.

Definition 4.1. Let $\mathcal{Q}.\phi$ be in AFF and $X \subseteq V(\phi)$ be a subset of variables. Define $\mathrm{Kernel}(\mathcal{Q}, \phi, X)$ as follows:

1. While there is an equation $(A_i, b_i) \in \phi$ with innermost variable x such that $x \notin X$, eliminate x, i.e., let $\phi := \mathrm{elim}(\phi, x, i)$.
2. While there is an equation $(A_i, b_i) \in \phi$ such that $A_i \setminus X \neq \emptyset$ and the innermost variable in $A_i \setminus X$ is not private, pivot on said variable, i.e., let $\phi := \mathrm{pivot}(\phi, x, i)$.
3. For every equation (A, b) in ϕ with $A \setminus X \neq \emptyset$, remove every variable of $A \setminus X$ except the innermost.

Now we verify correctness of the reduction.

Lemma 4.2. *Let $\mathcal{Q}.(\phi_1 \wedge \phi_2)$ be in* CNF+AFF *with $\phi_1 \in$* AFF *and $X = V(\phi_2)$. If $\mathcal{Q}.\phi_1 \Leftrightarrow \top$, then $\mathcal{Q}.(\phi_1 \wedge \phi_2) \Leftrightarrow \mathcal{Q}.(\mathrm{Kernel}(\mathcal{Q}, \phi_1, X) \wedge \phi_2)$.*

Proof. First, observe that $\mathcal{Q}.\phi_1 \Leftrightarrow \top$ implies that the innermost variable in every equation is always existential, otherwise the universal player can falsify an equation solely using the innermost variable. Consider the first step in Definition 4.1. If x is the innermost variable in an equation $(A_i, b_i) \in \phi_1$, then x is existential and private in $\mathrm{pivot}(\phi_1, x, i)$. Moreover, if $x \notin X$, then the value assigned to it does not affect ϕ_2, so the existential player can use it to satisfy (A_i, b_i), hence the equation can be safely removed.

The second step in Definition 4.1 is pivoting, which preserves the set of satisfying assignments. To verify correctness of the third step, let ϕ_1' be the formula obtained after steps 1 and 2. Consider an equation $(A, b) \in \phi_1'$ with $A \setminus X \neq \emptyset$. Let x be the innermost variable in $A \setminus X$. Note that by the second step, x is private. Let $A' = (A \cap X) \cup \{x\}$ and let ϕ_1'' be the formula obtained by replacing (A, b) with (A', b). We claim that $\mathcal{Q}.(\phi_1' \wedge \phi_2) \Leftrightarrow \mathcal{Q}.(\phi_1'' \wedge \phi_2)$, more precisely, the player that controls x wins on the left-hand side formula if and only if they win on the right-hand side formula. Indeed, to obtain a strategy for one formula, a player that controls x can modify their strategy for the other formula by replacing the value of x with $\bigoplus_{y \in A \setminus X} \tau(y)$, where τ is the partial assignment up to and including x. Since x is a private variable, this choice does not affect any other equation, and since $x \notin X$, it does not affect ϕ_2. □

We additionally want to prove that Definition 4.1 lets us construct an actual kernel in polynomial time.

Lemma 4.3. *There is a polynomial-time algorithm that takes a formula $\mathcal{Q}.(\phi_1 \wedge \phi_2)$ in* CNF+AFF *with $\phi_1 \in$* AFF *and $|V(\phi_2)| = k$, and returns an instance of the same problem on at most $2k$ variables.*

Proof. First, check if $\mathcal{Q}.\phi_1 \Leftrightarrow \top$, and if not, output a trivially false formula, e.g. $(\bot)$. Otherwise, let $X = V(\phi_2)$ and let $\phi_1' = \mathrm{Kernel}(\mathcal{Q}, \phi_1, X)$. As everything needed to do this is pivoting, substituting and removing variables, it can be constructed in polynomial time. Observe that there are at most k equations in ϕ_1', each of them containing at most $k + 1$ variables, at most one of which is from $V(\phi) \setminus X$. Hence, $\phi_1' \wedge \phi_2$ contains at most $2k$ variables and we can construct a new prefix $\mathcal{Q}'$ by taking $\mathcal{Q}$ and removing all variables not in $\phi_1' \wedge \phi_2$. As by Theorem 4.2 we have $\mathcal{Q}.(\phi_1 \wedge \phi_2) \Leftrightarrow \mathcal{Q}'.(\phi_1' \wedge \phi_2)$. □

Observe that the formula we obtain in Theorem 4.3 contains at most k equations and an arbitrary CNF on k variables, which can have $\mathcal{O}(3^k)$ clauses. The size of the formula cannot be reduced to polynomial in k unless coNP is included in NP/poly (which implies that the polynomial hierarchy collapses) (Dell and Van Melkebeek 2014).

For the purposes of an FPT algorithm, we could stop here and just branch in $\mathcal{O}^*(2^{2k})$ time. However, we can obtain the SETH-optimal running time $\mathcal{O}^*(2^k)$ by noting how the affine part of the instance forces assignments to certain variables. This gives us the main theorem for the section.

Theorem 4.4. QBF(CNF+AFF) *is solvable in $\mathcal{O}^*(2^k)$ time, where k is the size of a CC-backdoor to* AFF.

Proof. By Theorem 4.3 we can construct our kernel $\mathcal{Q}'.(\phi_1' \wedge \phi_2)$ such that $\mathcal{Q}.(\phi_1 \wedge \phi_2) \Leftrightarrow \mathcal{Q}'.(\phi_1' \wedge \phi_2)$ in polynomial time. By pivoting on the innermost variable of each equation in ϕ_1', we make the said variables private. Now, for any equation $(A, b) \in \phi_1'$ with innermost variable x, observe that x is existential since otherwise $\mathcal{Q}.\phi_1 \Leftrightarrow \bot$ and the kernel is a trivially false instance. Thus, once every variable in $A \setminus \{x\}$ has been assigned by some partial assignment τ, we can directly assign a value to $\tau(x)$ so that $\bigoplus_{y \in A} \tau(y) = b$. Hence, $|\phi_1'|$ variables can be assigned directly in each branch, and as $|V(\phi_1' \wedge \phi_2)| \leq |\phi_1'| + k$, we only need to branch on at most k variables, giving $\mathcal{O}^*(2^k)$ running time. □

Hence, we have proven that QBF(CNF+AFF) is in FPT when parameterized by the size k of the affine CC-backdoor. Again, since the backdoor can be an arbitrary QBF instance, this upper bound is tight under SETH. Note that if we do not restrict ourselves to a resulting formula strictly in CNF+AFF, we could from Theorem 4.3 in polynomial time construct a formula of size $|\phi_2|$ with only k variables by taking private variables from ϕ_1' and substituting. I.e. for any equation $(A, b) \in \phi_1'$ with innermost private variable $x \in A$, replace x by $b \oplus \bigoplus_{y \in A \setminus x} y$. However, the $2k$ variable kernel is in the worst case a more compact representation.

5 Intractability: Backdoors to Horn

The last classical class of QBF that we consider is HORN where we, in contrast to the positive FPT results thus far, prove that CC-backdoor evaluation is not FPT unless FPT = W[1]. The starting point of our reductions is the well-known W[1]-hard MULTIPARTITE INDEPENDENT SET (MIS) problem (see e.g. Section 13.2 in (Cygan et al. 2015)): an instance is a graph with the vertex set partitioned into k parts, where k is the parameter, and the computational question is whether the graph contains an independent set with one vertex from each part, i.e., whether one can choose one vertex from each part so that no pair of chosen vertices is adjacent. Our reduction is inspired by Lemma 14 in (Eriksson et al. 2024) but modified to produce a formula with a CC-backdoor to HORN.

Lemma 5.1. *There is a polynomial-time reduction that takes an instance (G, k) of MIS and produces a QBF(CNF) formula $\forall Y \exists X.\phi$ with CC-backdoor to* HORN *of size k such that (G, k) is a yes-instance if and only if $\mathcal{Q}.\phi \Leftrightarrow \bot$.*

Proof. Let G be a graph with vertex set $V_1 \uplus \cdots \uplus V_k$. For convenience, we refer to vertex j in part V_i as (i, j).

We will construct out formula $\forall Y \exists X.\phi$ on variables $Y = \{y_v : v \in V(G)\}$ and $X = \{x_1, \ldots, x_k\}$ that is false if and only if (G, k) has an independent set with exactly one vertex from every part. For every vertex $v = (i, j) \in V(G')$, add a clause C_v to ϕ defined as

$$C_v = \{y_v\} \cup \{\neg y_u : u \in N_G(v)\} \cup \{\neg x_i\}$$

where $N_G(v)$ is the set of neighbors to v. Finally, add the clause $C_\exists = (x_1 \vee \ldots \vee x_k)$ to ϕ. Clearly the conjunction of all clauses C_v is in HORN, and $C_\exists$ contains k variables. Hence, $\forall Y \exists X.\phi$ is a QBF(CNF) formula with a CC-backdoor to HORN of size k. We only give a sketch of the correctness proof since it is similar the proof of Lemma 5.4. First observe that $\forall Y \exists X.\phi$ is false if and only if there is $\alpha \colon Y \to \{0, 1\}$ to Y such that for every $i \in [k]$ there is a vertex $v \in V_i$ whose clause C_v is falsified except for $\neg x_i$. Such α naturally corresponds to choosing exactly one vertex from each part V_i. The construction of the clauses then ensures that the chosen vertices are pairwise non-adjacent, since otherwise α would have to assign both y_u and $\neg y_u$ to 0 for some vertex u. Then, such an α exists if and only if G contains an independent set selecting exactly one vertex from each part. □

To reduce from HORN to 3-HORN one can apply a standard transformation to every Horn clause with more than 3 literals until no such clause remains.

Lemma 5.2. *Let $\Psi = \mathcal{Q}.(\phi \wedge C)$ be a* QBF(CNF) *formula where $C = \{x_1, \neg x_2, \ldots, \neg x_r\}$ for $r > 3$. Then Ψ is equivalent to $\Psi' = \mathcal{Q}\exists v.(\phi \wedge C_1 \wedge C_2)$, where $C_1 = \{x_1, \neg x_2, \neg v\}$ and $C_2 = \{v, \neg x_3, \ldots, \neg x_r\}$.*

Observe that the clauses C and C_2 in the lemma above are in HORN, C_1 is 3-HORN and C_2 contains at least one fewer variable than C. Hence, we may first run the algorithm from Theorem 5.1, obtaining a formula from the MIS instance, then apply Theorem 5.2 to that formula until every HORN clause has size at most 3. Thus obtaining an FPT-reduction from MIS to the CC-backdoor evaluation problem to 3-HORN.

Theorem 5.3. QBF(CNF) *is* W[1]*-hard parameterized by the CC-backdoor size to* 3-HORN *even on formulas with quantifier prefix of the form $\forall \ldots \forall \exists \ldots \exists$.*

Having established W[1]-hardness for CC-backdoors to HORN, it is natural to ask whether the hardness also holds for its subclasses. To this end, recall that IHSB$_+$ (and its dual IHSB$_-$) is the class of formulas that allows positive (negative) clauses of unbounded arity, implications and negative (positive) unit clauses.

Lemma 5.4. QBF(CNF) *is* W[1]*-hard parameterized by the CC-backdoor size to* IHSB$_+$ *or* IHSB$_-$ *even on formulas with quantifier prefix of the form $\forall \ldots \forall \exists \ldots \exists$.*

Proof. We show the proof for IHSB$_-$ and the argument for IHSB$_+$ is analogous. The hardness follows by a modification of the proof in Theorem 5.1. We start with an instance (G, k) of MIS and in polynomial time construct a formula $\Phi = \forall Y \exists X.\phi_1 \wedge \phi_2$ such that $\phi_1 \in$ IHSB$_-$, $V(\phi_2) = X = \{x_1, \ldots, x_k, z_1, \ldots, z_k\}$ and $\Phi \Leftrightarrow \bot$ if and only if (G, k) is a yes-instance.

Let $V_1 \uplus \cdots \uplus V_k$ be the vertex set of G. We let $Y = \{y_v : v \in V(G)\}$. For a variable $v \in V_i$, define the clause

$$C_v = \{\neg y_w : w \in V_i \setminus \{v\}\} \cup \{\neg y_u : u \in N_G(v)\} \cup \{\neg x_i\}.$$

Observe that C_v is negative. For a color class $i \in [k]$, let

$$\psi_i = \bigwedge_{v \in V_i} (z_i \to y_v).$$

Finally, let $\phi_1 = \bigwedge_{v \in V(G)} C_v \wedge \bigwedge_{i=1}^{k} \psi_i$ and ϕ_2 contain a single clause $\{x_1, \ldots, x_k, z_1, \ldots, z_k\}$.

Observe that $\Phi \Leftrightarrow \bot$ if and only if there is an assignment $\alpha : Y \to \{0, 1\}$ such that $\Phi[\alpha]$ is unsatisfiable. The latter holds if and only if $\Phi[\alpha]$ implies $(\neg x_i)$ and $(\neg z_i)$ for all $i \in [k]$. If $\alpha(y_v) = 1$, we will say that α *selects* v. To ensure that $\Phi[\alpha]$ implies $(\neg z_i)$, by the definition of ψ_i, is necessary and sufficient for α to select at least one vertex in V_i. To ensure that $\Phi[\alpha]$ implies $(\neg x_i)$, it is necessary and sufficient for there to be a clause C_v with $v \in V_i$ such that α falsifies all literals in $C_v \setminus \{\neg x_i\}$. Thus, if α selects some $v \in V_i$, then to make sure that $\Phi[\alpha]$ implies $(\neg x_i)$, α must not select any other vertex $w \in V_i \setminus \{v\}$ or any neighbor u

of v in G. Thus, the set of vertices selected by α must form an independent set in G with at least one vertex from each color class. □

We note that the reduction from Theorem 5.2 does not apply to IHSB$_+$, so the hardness for bounded-arity formulas remains open, which we discuss in greater detail in Section 6. To complement this hardness result, we show that disallowing implications makes the problem fixed-parameter tractable.

Lemma 5.5. *Let $\mathcal{C} \subseteq$ IHSB$_+$ be the class of formulas with positive clauses (of any size) and negative unit clauses. Then* QBF(CNF) *is* FPT *parameterized by the CC-backdoor size to $\mathcal{C}$.*

Proof. First, we may remove unit clauses: if there is a unit clause with a universal variable, then we have a no-instance; if there is a unit clause with an existential variable, then we have assign that variable to the value satisfying the unit clause. Observe that when no unit clauses remain, all clauses outside the backdoor are positive. Now we remove non-backdoor variables: if a non-backdoor variable is existential, assign it 1, and if a non-backdoor variable is universal, assign it 0. Clearly, these choices are safe for positive clauses. Finally, we are left with a formula with only backdoor variables, and it can be evaluated in time $\mathcal{O}^*(2^k)$, where k is the number of variables. □

Naturally, the result extends to the corresponding subclass of IHSB$_-$ without implication.

6 Towards a Full CC-Classification

In this section we analyse to which extent the results so far completely describe the complexity of the CC-evaluation problem. As we will prove, there is only one case (or two, if we also count the dual) missing for a full dichotomy, namely, bounded arity d-IHSB$_+$ (and d-IHSB$_-$) for $d \geq 3$. To study this in a general setting we first define a *Boolean constraint language* Γ as a set of Boolean relations. Let $\mathcal{C}_\Gamma$ be the class of conjunctive formulas with atoms only using relations from Γ. It is common to refer to the problem QBF($\mathcal{C}_\Gamma$) as QBF(Γ) or QCSP(Γ). We are thus interested in the complexity of conjunctive QBF parameterized by clause covering backdoor size to $\mathcal{C}_\Gamma$ for every Γ. We let Γ-EV be the problem of deciding QBF(CNF + Γ) when given a CC-backdoor B into $\mathcal{C}_\Gamma$ (where CNF + Γ is a shorthand for CNF$+\mathcal{C}_\Gamma$). Naturally, we are then interested in Γ-EV when parameterized by $|B|$. As we will prove, we are surprisingly close to a full parameterized complexity dichotomy.

To avoid exhaustive case analyses we apply the *algebraic approach* where the basic idea is to equip each set of relations Γ with an algebraic object Pol(Γ) (*polymorphisms*) such that Pol(Γ) abstracts away from many irrelevant details, but still decides the (classical) complexity of QBF(Γ). Concretely, an operation $f\colon \{0,1\}^d \to \{0,1\}$ is a *polymorphism* of Γ if the following holds for all relations $R \in \Gamma$: for every $(d \times r)$-matrix of $A = (a_{ij})$ such that all rows $(a_{i1}, \ldots, a_{ir})$ belong to R, the tuple $(b_1, \ldots, b_r)$ obtained by applying f to A column-wise, i.e., $b_j = f(a_{1j}, a_{2j}, \ldots, a_{dj})$, also belongs to R. We let Pol(Γ) be the set of all polymorphisms of Γ.

For a Boolean function $f\colon \{0,1\}^d \to \{0,1\}$ we define its *dual* operation f^{dual} as $f^{\text{dual}}(\overline{x_1}, \ldots, \overline{x_d}) = \overline{f(x_1, \ldots, x_d)}$ for all $x_1, \ldots, x_d \in \{0,1\}$, where $\overline{x_i}$ is Boolean complement. Similarly, define $R^{\text{dual}} = \{(\overline{x_1}, \ldots, \overline{x_n}) \mid (x_1, \ldots, x_n) \in R\}$ for an n-ary relation R, and $\Gamma^{\text{dual}} = \{R^{\text{dual}} \mid R \in \Gamma\}$ for a set of relations Γ. It is then well-known that $\{f^{\text{dual}} \mid f \in \text{Pol}(\Gamma)\} = \text{Pol}(\Gamma^{\text{dual}})$. Last, say that a function f is *idempotent* if $f(x, \ldots, x) = x$ for all $x \in \{0,1\}$. Then, it is well-known and easy to see that each function in Pol(Γ^+), where $\Gamma^+ = \Gamma \cup \{\{(0)\}, \{(1)\}\}$, is idempotent. We then have the following basic relationships.

Theorem 6.1. *Let Γ be a (not necessarily finite) constraint language and let Δ be a finite constraint language. The following statements are true.*

1. *If* Pol(Γ) $\subseteq$ Pol(Δ) *then* Δ-EV $\leq$ Γ-EV*,*
2. Γ-EV *and* Γ^{dual}-EV *are FPT-equivalent,*
3. Γ^+-EV *and* Γ-EV *are FPT-equivalent.*

Proof. We prove each item in turn. The first claim implies (Chen 2006) that each relation in Δ is logically equivalent to a conjunctive formula (possibly containing fresh existentially quantified variables) over Γ (where each atom either uses a relation from Γ or is an equality constraint). Importantly, since Δ is finite, we can precompute all such definitions and store them in a table of constant size. Given $\mathcal{Q}.(\phi_1 \wedge \phi_2)$, where $\phi_1 \in \mathcal{C}_\Delta$, we then replace each constraint in ϕ_1 by the corresponding set of constraints over Γ, and treat the fresh variables as existentially quantified, added last in the prefix. We then gradually remove equality constraints by either identifying the two variables in question, or trivially detecting that the formula is false (if one variable in the equality constraint is universally quantified and is preceded by the other one). See (Chen 2006) for additional details. Since this construction does not affect the backdoor constraints in ϕ_2 the parameter $|V(\phi_2)|$ does not increase.

For the second claim, given $\mathcal{Q}.(\phi_1 \wedge \phi_2)$, where $\phi_1 \in \mathcal{C}_\Gamma$, replace each atom $R(\cdot)$ in ϕ_1 by $R^{\text{dual}}(\cdot)$, replace each clause $(l_1 \vee \ldots \vee l_r)$ in ϕ_2 by $(\neg l_1 \vee \cdots \vee \neg l_r)$, and keep the quantifier prefix $\mathcal{Q}$ intact. Any tree witnessing a yes-instance of one instance can be converted into a tree for the dual formula by complementing each value, since this preserves the truth of ϕ_1 (by definition of the dual atoms) and of ϕ_2 (since the literals of every clause have been negated).

For the third claim, let $\mathcal{Q}.(\phi_1 \wedge \phi_2)$ be an instance of Γ^+-EV. We introduce two fresh variables X_0 and X_1 and for each variable $x \in V(\phi_1)$ where $\{(0)\}(x) \in \phi_1$ we identify it with X_0. Dually, if $\{(1)\}(x) \in \phi_1$ we identify it with X_1, and if we detect that we have both $\{(0)\}(x), \{(1)\}(x) \in \phi_1$ then we have a trivial no-instance. Next, since $\{(0)\}(X_0)$ can be viewed as a unary clause $(\overline{X_0})$, and $\{(1)\}(X_1)$ as a unary clause (X_1), we delete them from ϕ_1 and instead take $\phi_2 \cup \{(X_1), (\overline{X_0})\}$ as the new backdoor. This only increases the parameter by 2 and the reduction is indeed FPT. □

To have any hope of obtaining a CC-backdoor complexity classification we first need to firmly establish the tractable cases of QCSP. The following classification appeared without proof as Theorem 6.1 in Schaefer (Schaefer 1978), restated in modern terminology using polymorphisms (see e.g. (Chen 2006)). First, define the following operations:

- binary $\min(x, y) := x \wedge y$,
- binary $\max(x, y) := x \vee y$,
- ternary $\text{majority}(x, y, z) := (x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$,
- ternary $\text{minority}(x, y, z) = x \oplus y \oplus z$, where $\oplus$ is the binary exclusive OR operation.

Theorem 6.2. *Let Γ be a Boolean constraint language. Then* $\text{QBF}(\Gamma)$ *is in P if* min, max, majority *or* minority *is a polymorphism of Γ, otherwise it is PSPACE-hard.*

Moreover, there is a neat connection between syntactically defined classes of formulas (such as 2-CNF, HORN and AFF) and tractable classes of QBF defined by polymorphisms (see, e.g., (Böhler et al. 2003)).

Lemma 6.3. *Let Γ be a constraint language. If*

1. $\text{majority} \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{2-CNF}) \subseteq \text{Pol}(\Gamma)$,
2. $\text{minority} \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{AFF}) \subseteq \text{Pol}(\Gamma)$,
3. $\min \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{HORN}) \subseteq \text{Pol}(\Gamma)$,
4. $\max \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{DualHORN}) \subseteq \text{Pol}(\Gamma)$,
5. $x \wedge (y \vee z) \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{IHSB}_-) \subseteq \text{Pol}(\Gamma)$,
6. $x \vee (y \wedge z) \in \text{Pol}(\Gamma)$ *then* $\text{Pol}(\text{IHSB}_+) \subseteq \text{Pol}(\Gamma)$.

Additionally, for $d \geq 3$ define the d-ary *threshold function* (t_d) as

$$t_d(x_1, \ldots, x_d) = \begin{cases} 1 & \text{if } x_1 + \ldots + x_d < d, \\ 0 & \text{otherwise.} \end{cases}$$

To simplify the forthcoming statements, say that Γ is an IHSB_+-language (IHSB_--language) if $\text{Pol}(\text{IHSB}_+) \subseteq \text{Pol}(\Gamma)$ ($\text{Pol}(\text{IHSB}_-) \subseteq \text{Pol}(\Gamma)$). We remark that Γ is an IHSB_+-language if and only if each $R \in \Gamma$ can be described as CNF formula where each clause is in IHSB_+ (Creignou, Kolaitis, and Zanuttini 2008).

Additionally, the threshold function t_d provides a precise description of definability of $(x_1 \vee \ldots \vee x_d)$ in the sense that if $t_d \in \text{Pol}(\Gamma)$ then Γ does not contain (nor can it define) $(x_1 \vee \ldots \vee x_k)$ for any $k \geq d$ (Böhler et al. 2003).

Theorem 6.4. *Let Γ be a finite IHSB_+-language. Then* Γ-EV *when parameterized by the CC-backdoor size is*

1. *FPT if $x \vee (y \wedge \bar{z}) \in \text{Pol}(\Gamma)$ or $t_3 \in \text{Pol}(\Gamma)$,*
2. *FPT-equivalent to d-IHSB_+-EV for some $d \geq 3$ depending on the arity of of Γ.*

Proof. We consider each case in turn. First, we observe that $f(x, y, z) = x \vee (y \wedge \bar{z})$ does not preserve implication $\{(0,0), (0,1), (1,1)\}$ since $f((0,0), (1,1), (0,1)) = (f(0,1,0), f(0,1,1)) = (1,0) \notin (0,0), (0,1), (1,1)$. It follows that each relation in Γ can be expressed as a CNF-formula where each clause is either positive, or a negative unit clause, and FPT follows from Lemma 5.5. Second, if $t_3 \in \text{Pol}(\Gamma)$ then we first observe that t_3 does not preserve any positive clause of arity more than 2 (applying t_3 to three tuples with Hamming weight > 1 produces the constant tuple of Hamming weight 0). It follows that each relation in Γ can be expressed as 2-CNF, and FPT follows from Theorem 3.3. For the second statement, the condition implies that $t_{d+1} \in \text{Pol}(\Gamma)$ for some $d \geq 3$, which is known to imply that each relation in Γ can be defined as a CNF-formula in d-IHSB_+ and that $\text{Pol}(\Gamma) = \text{Pol}(d\text{-IHSB}_+)$ (Creignou, Kolaitis, and Zanuttini 2008), which allows us to apply Theorem 6.1. □

An analogous statement can be made for IHSB_--languages. For languages that are not IHSB_+ or IHSB_-, however, we obtain a complete classification.

Theorem 6.5. *Let Γ be a finite Boolean constraint language that is not an* IHSB_+- *or* IHSB_-*-language. Then* Γ-EV *parameterized by the CC-backdoor size is*

- *FPT if* majority *or* minority *are polymorphisms of Γ,*
- *W[1]-hard if* min *or* max *are polymorphisms of Γ but* majority *is not (even on formulas with a single quantifier alternation), and*
- *paraPSPACE-hard otherwise.*

Proof. For the tractable cases we (via Lemma 6.3) know that $\text{Pol}(\text{2-CNF})$ or $\text{Pol}(\text{AFF})$ is contained in $\text{Pol}(\Gamma)$. Note that the evaluation problem for each such language is FPT by combining Theorem 3.3, Theorem 4.4, and the first item of Theorem 6.1.

For W[1]-hardness, we first apply Lemma 6.3 and Theorem 5.3 and the second item of Theorem 6.1 (to cover both HORN and DualHORN). Last, by an application of the first item of Theorem 6.1, W[1]-hardness extends to any $\text{Pol}(\Gamma) \subseteq \text{Pol}(\text{HORN})$ or $\text{Pol}(\Gamma) \subseteq \text{Pol}(\text{DualHORN})$ (note that the fresh existential variables are added last, so we do not increase the quantifier alternations).

Finally, if the two first condition do not hold for a finite Γ then it is not preserved by majority, minority, min, max, and Theorem 6.2 implies that the problem is PSPACE-hard even with a backdoor of size 0. □

Put together, Γ-EV is thus either (1) FPT, (2) W[1]-hard, or (3) FPT-equivalent with d-IHSB_+-EV for some $d \geq 3$. It might also be interesting to note that the positive FPT cases hold regardless of the number of quantifier alternations.

7 Conclusion

In this paper we introduced a variant of the well-known backdoor notion, CC-backdoors, with the goal of obtaining new FPT algorithms for QBF with unbounded quantifier alternations. We began the paper with two positive FPT results for 2-CNF and AFF but then quickly discovered that HORN and IHSB_- (and their duals) gave W[1]-hardness. We then managed to prove that the d-IHSB_- and d-IHSB_+ ($d \geq 3$) are the only obstructions for a complete parameterized complexity classification. Hence, resolving the CC-evaluation problem for d-IHSB_+ is an interesting, but likely challenging, open problem, and the techniques for 2-CNF and AFF do not seem to be applicable. Nevertheless, there are also other unresolved questions, which we now discuss.

Hardness of Horn and d-IHSB$_+$. While $\forall\exists$3-Horn being $\mathrm{W}[1]$-hard is an interesting result, it is quite far from the complete picture and a multitude of follow-up questions now arise: is $\forall\exists$3-Horn in XP? Could it be harder than $\mathrm{W}[1]$? By taking into consideration the definition of the polynomial hierarchy it then also becomes interesting to ask if $\forall\exists\forall\exists$3-Horn (i.e. the prefix is $\forall\ldots\forall\exists\ldots\exists\forall\ldots\forall\exists\ldots\exists$) is $\mathrm{W}[2]$-hard? How about other types of fixed prefixes? And then finally, is 3-Horn with no prefix restrictions para-NP-hard when parameterized by a CC-backdoor?

Extensions to quantified constraint satisfaction. The QCSP problem can be defined over arbitrary finite domain. Can our Boolean results be generalized to this (significantly more general) setting? This seems difficult to answer in full generality since the classical complexity of such QCSPs has not been determined, but there is nothing preventing us from studying specific languages. For example, QCSPs over linear equations are tractable and could form suitable backdoor classes, while the algebraic extension of 2-CNF to arbitrary finite domains via *semilattice operations* is not always tractable (Chen 2004). We also mention the 0/1/all-constraints (Cooper, Cohen, and Jeavons 1994) as a promising generalization of 2-CNF that supports propagation. Also note that Boolean QCSPs are always either tractable or PSPACE-complete, but finite-domain QCSPs exhibit a much wider range in complexity and can e.g. also be NP-complete, coNP-complete, DP-complete, or Π_2^P-complete; however, curiously, there is nothing between Π_2^P and PSPACE (Zhuk 2024). Attacking these restricted QCSP classes with a backdoor approach is tempting since, from an algorithmic standpoint, they are likely easier to handle than PSPACE-complete QCSPs.

Acknowledgements

The second author was partially supported by the Swedish Research Council (VR) under grant 2021-04371 and 2025-04487. The fourth author was supported by VR under grant 2024-00274.

AI Declaration

The authors have not employed any generative AI tools.

References

Abrahamsen, M.; Kleist, L.; and Miltzow, T. 2021. Training neural networks is er-complete. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 18293–18306. Curran Associates, Inc.

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

Biere, A.; Heule, M.; and van Maaren, H. v. 2021. *Handbook of Satisfiability: Second Edition*. Frontiers in Artificial Intelligence and Applications. IOS Press.

Böhler, E.; Creignou, N.; Reith, S.; and Vollmer, H. 2003. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *SIGACT News* 34.

Calabro, C.; Impagliazzo, R.; and Paturi, R. 2009. The complexity of satisfiability of small depth circuits. In Chen, J., and Fomin, F. V., eds., *Parameterized and Exact Computation*, volume 5917 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 75–85.

Calabro, C.; Impagliazzo, R.; and Paturi, R. 2013. On the exact complexity of evaluating quantified *k*-CNF. *Algorithmica* 65(4):817–827.

Chen, H. 2004. Quantified constraint satisfaction and 2-semilattice polymorphisms. In Wallace, M., ed., *Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 - October 1, 2004, Proceedings*, volume 3258 of *Lecture Notes in Computer Science*, 168–181. Springer.

Chen, H. 2006. A rendezvous of logic, complexity, and algebra. *ACM SIGACT News* 37(4):85–114.

Cooper, M. C.; Cohen, D. A.; and Jeavons, P. 1994. Characterising tractable constraints. *Artificial Intelligence* 65(2):347–361.

Creignou, N.; Kolaitis, P.; and Zanuttini, B. 2008. Structure identification of boolean relations and plain bases for co-clones. *Journal of Computer and System Sciences* 74(7):1103–1115.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition.

Dell, H., and Van Melkebeek, D. 2014. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM* 61(4).

Downey, R. G.; Fellows, M. R.; et al. 2013. *Fundamentals of parameterized complexity*, volume 4. Springer.

Eriksson, L.; Lagerkvist, V.; Ordyniak, S.; Osipov, G.; Panolan, F.; and Rychlicki, M. 2024. Solving quantified Boolean formulas with few existential variables. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-2024)*, 1889–1897. ijcai.org.

Fichte, J. K.; Ganian, R.; Hecher, M.; Slivovsky, F.; and Ordyniak, S. 2023. Structure-aware lower bounds and broadening the horizon of tractability for QBF. In *Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS-2023)*, 1–14.

Fomin, F. V.; Lokshtanov, D.; Saurabh, S.; and Zehavi, M. 2019. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press.

Nunn, P.; Sälzer, M.; Schwarzentruber, F.; and Troquard, N. 2024. A logic for reasoning about aggregate-combine graph neural networks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

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

Schaefer, T. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory Of Computing (STOC-1978)*, 216–226. ACM Press.

Sälzer, M.; Schwarzentruber, F.; and Troquard, N. 2025. Verifying quantized graph neural networks is PSPACE-complete. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-2025)*, 4660–4668. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhuk, D. 2024. $\prod_2^{\mathrm{p}}$ vs PSpace dichotomy for the quantified constraint satisfaction problem. In *Proceedings of the 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS-2024)*, 560–572. IEEE.