Algorithmica (2016) 74:540–557
DOI 10.1007/s00453-014-9958-5

Backdoors to q-Horn

Serge Gaspers · Sebastian Ordyniak · M. S. Ramanujan · Saket Saurabh · Stefan Szeider

Received: 7 April 2013 / Accepted: 11 November 2014 / Published online: 26 November 2014

Abstract The class q-Horn, introduced by Boros, Crama and Hammer in 1990, is one of the largest known classes of propositional CNF formulas for which satisfiability can be decided in polynomial time. This class properly contains the fundamental classes of Horn and 2-CNF formulas as well as the class of renamable (or disguised) Horn formulas. In this paper we extend this class so that its favorable algorithmic properties can be made accessible to formulas that are outside but "close" to this class. We show that deciding satisfiability is fixed-parameter tractable parameterized by the distance of the given formula from q-Horn. The distance is measured by the smallest number of variables that we need to delete from the formula in order to get a q-Horn formula,

A preliminary version of this paper appeared in the proceedings of STACS 2013.

S. Gaspers
The University of New South Wales, Sydney, Australia
e-mail: sergeg@cse.unsw.edu.au

S. Gaspers
National ICT Australia, Sydney, Australia

S. Ordyniak
Masaryk University, Brno, Czech Republic
e-mail: sordyniak@gmail.com

M. S. Ramanujan (✉) · S. Saurabh
The Institute of Mathematical Sciences, Chennai, India
e-mail: msramanujan@imsc.res.in

S. Saurabh
e-mail: saket@imsc.res.in

S. Szeider
Institute of Information Systems, Vienna University of Technology, Vienna, Austria
e-mail: stefan@szeider.net

i.e., the size of a smallest deletion backdoor set into the class q-Horn. This result generalizes known fixed-parameter tractability results for satisfiability decision with respect to the parameters distance from Horn, 2-CNF, and renamable Horn.

Keywords Algorithms and data structures · Backdoor sets · Satisfiability · Fixed parameter tractability

1 Introduction

The satisfiability problem (SAT) is a well-known fundamental problem in Computer Science [4]. Many hard combinatorial problems including problems from the domains of hardware and software verification, Artificial Intelligence, planning and scheduling can be encoded as SAT instances [3,5,18,20,27]. However, the problem is known to be NP-hard and thus we cannot hope to solve it in polynomial time [8]. In spite of this, over the last two decades, SAT-solvers have become quite successful in solving formulas with hundreds of thousands of variables that encode problems arising from various application areas (see, e.g., [17]), but theoretical performance guarantees are far from explaining this empirically observed efficiency. In fact, there is an enormous gap between theory and practice.

The discrepancy between theory and practice can be potentially explained by the presence of a certain “hidden structure” in real-world problem instances. One such “hidden structure” in real-world instances of SAT is the presence of *small backdoor sets* [28]. There are three variants of backdoor sets with respect to a particular base class $\mathcal{C}$ of polynomial-time decidable CNF formulas: strong $\mathcal{C}$-backdoor sets, where for each truth assignment to the backdoor variables, the reduced formula belongs to $\mathcal{C}$; deletion $\mathcal{C}$-backdoor sets, where deleting all backdoor variables and their negations from the formula moves the formula into the base class $\mathcal{C}$; and weak backdoor sets where, for at least one truth assignment to the backdoor variables, the reduced formula belongs to $\mathcal{C}$ and is satisfiable. Given a backdoor set of a formula with respect to a particular tractable base class $\mathcal{C}$, the satisfiability of the formula can be decided by guessing an assignment to the variables in the backdoor set and deciding the satisfiability of the reduced formula, which is guaranteed to be in $\mathcal{C}$, using a sub-solver for $\mathcal{C}$. An equivalent view of this is to consider the size of the backdoor set to be the “distance” of the formula from the class $\mathcal{C}$. The objective is to extend the favorable algorithmic properties of the class $\mathcal{C}$ to formulas which are “close” to this class. Ideally, we would want the class $\mathcal{C}$ to be as large as possible.

In a 1990 paper [6], Boros, Crama and Hammer introduced an interesting class of CNF formulas, later called q-Horn [7], with favorable algorithmic properties: both recognition as well as deciding satisfiability of q-Horn formulas can be performed in linear time [6,7]. The class q-Horn properly contains the fundamental classes of Horn and 2-CNF formulas [26], and the class of renamable (or disguised) Horn formulas [2,19]:

$$\begin{tikzcd}
\text{Horn} \arrow[r, phantom, "\subsetneq"] & \text{renamable Horn} \arrow[dr, phantom, sloped, "\subsetneq"] & \\
& & \text{q-Horn} \\
& \text{2-CNF} \arrow[ur, phantom, sloped, "\subsetneq"] &
\end{tikzcd}$$

The fact that the class q-Horn is large enough to contain these well-known classes also serves as an additional motivation for choosing it as our base class of interest. In this paper, we study the problem of finding small backdoor sets with respect to the class of q-Horn formulas and obtain algorithmic as well as hardness results.

Contribution The main contribution of this paper is an algorithm that, given a CNF formula F of length ℓ with n variables and an integer $k \geq 0$, runs in time $O(6^k k\ell n)$, and either returns a deletion q-Horn-backdoor set for F of size at most k^2+k, or concludes correctly that no such set of size at most k exists. As a consequence, we obtain the result that SAT is fixed-parameter tractable with the size of the smallest deletion q-Horn-backdoor set as the parameter, as we can use this algorithm to reduce the satisfiability problem of a CNF formula F of distance k from being q-Horn to testing the satisfiability of $2^{O(k^2)}$ q-Horn formulas. Our result simultaneously generalizes known fixed-parameter tractability results for SAT parameterized by the deletion distance from the class of renamable Horn formulas [24] and from the class of 2-CNF formulas [22].

A high level description of our algorithm is as follows. We first design a sub-routine to compute a bounded number of variables whose deletion from the given instance results in an instance whose smallest deletion q-Horn-backdoor set is strictly smaller than that of the original instance. By repeatedly computing such a set of variables and deleting it, we obtain an approximate solution. The main technical part of the paper is the sub-routine to compute the bounded set of variables with the required properties. This algorithm relies on a characterization of q-Horn formulas in terms of their *quadratic cover* by Boros et al. [7]. We use this characterization to model the problem of finding a small deletion q-Horn-backdoor set as a problem of *hitting* certain types of paths in an auxiliary digraph related to the formula. Using this characterization, we show that if we are guaranteed that an optimal solution hits all paths between a carefully chosen pair of vertices in this digraph, then we can compute in polynomial time a set of variables whose size is bounded by $f(k)$ (for some computable function f) such that (a) there is a minimal (though not necessarily optimal) solution containing these variables and (b) the deletion of these variables results in a formula whose solution is strictly smaller than the solution for the formula we started with. A standout feature of our algorithm is that at its core, it reduces to computing flows in a directed graph whose size is linear in the input size. As a result, our algorithm is quite efficient not only with respect to the dependence of the running time on the parameter, but also with respect to the dependence on the input size, along with having only a small hidden constant factor in the asymptotic running time. Finally, towards the end of the paper we also provide parameterized complexity results regarding the detection of weak and strong backdoor sets with respect to the class q-Horn.

Related Work The parameterized complexity of finding small backdoor sets was initiated by Nishimura et al. [22] who showed that for the base classes of Horn formulas and 2-CNF formulas, the detection of strong backdoor sets (referred to as STRONG HORN- BDS or STRONG 2- CNF- BDS) is fixed-parameter tractable (see Table 1). Their algorithms exploit the fact that for these two base classes, strong and deletion backdoor

Table 1 The table gives the previous fixed parameter (in)tractability results obtained for this problem as well as the ones obtained in this paper

Base Class $\mathcal{C}$	Weak $\mathcal{C}$-BDS	Strong $\mathcal{C}$-BDS	Deletion $\mathcal{C}$-BDS
Horn	W[2]-Hard [22]	FPT [22]	FPT [22]
2-CNF	W[2]-Hard [22]	FPT [22]	FPT [22]
Renamable Horn	W[2]-Hard [16]	W[2]-Hard [16]	FPT [24]
q-Horn	W[2]-Hard (Theorem 3)	W[2]-Hard (Theorem 2)	FPT-approximation (Theorem 1)

The formal definition of an FPT-approximation is given in the preliminaries

sets coincide, and that deletion backdoor sets with respect to Horn and 2-CNF can be characterized in terms of vertex covers and hitting sets of certain graphs and 3-uniform hypergraphs associated with the input formula, respectively. For base classes other than Horn and 2-CNF, strong backdoor sets can be much smaller than deletion backdoor sets, and their detection is more difficult. In particular, for the base classes of renamable Horn and q-Horn, there are formulas that have a strong backdoor set of size 1 but require an arbitrarily large deletion backdoor set. In fact, it follows from the result of Razgon and O'Sullivan [24] that the detection of deletion backdoor sets with respect to the base class renamable Horn is fixed-parameter tractable although the detection of strong backdoor sets is W[2]- hard [16]. For more recent results, the reader is referred to a survey on the parameterized complexity of backdoor sets [16].

2 Preliminaries

Formulas A *literal* is a propositional variable x or a negated variable $\bar{x}$; if $y = x$ or $y = \bar{x}$ is a literal for some variable x, then we write $\bar{y}$ to denote $\bar{x}$ or x, respectively. For a set S of literals we put $\bar{S} = \{\,\bar{x} : x \in S\,\}$; S is *consistent* if $S \cap \bar{S} = \emptyset$. A *clause* is a finite consistent set of literals; we consider a clause as a disjunction of its literals. A finite set of clauses is a *CNF formula* (or *formula*, for short); we consider a formula to be the conjunction of its clauses. A formula is *Horn* if each of its clauses contains at most one positive literal, and a formula is *2CNF* (or *quadratic*) if each clause contains at most two literals. A variable x *occurs* in a clause C if $x \in C \cup \bar{C}$; $\operatorname{var}(C)$ denotes the set of variables which occur in C. For a set X of variables, $\operatorname{lit}(X)$ denotes the set of literals of the variables in X, that is, $\operatorname{lit}(X) = X \cup \bar{X}$ and for a set L of literals, $\operatorname{var}(L)$ denotes the set of variables whose literals are in L, that is, $\operatorname{var}(L) = \{\,x : x \in L \text{ or } \bar{x} \in L\,\}$. A variable x *occurs* in a formula F if it occurs in at least one of its clauses, and we let $\operatorname{var}(F) = \bigcup_{C \in F} \operatorname{var}(C)$ and $\operatorname{lit}(F) = \operatorname{var}(F) \cup \overline{\operatorname{var}(F)}$. The *length* of a CNF formula F, denoted by $\|F\|$, is defined as $\sum_{C \in F} |C|$. If F is a formula and X a set of variables, then we denote by $F - X$ the formula obtained from F after removing all literals in $\operatorname{lit}(X)$ from the clauses in F. If $X = \{x\}$ we simply write $F - x$ instead of $F - \{x\}$.

Let F be a formula and $X \subseteq \operatorname{var}(F)$. A *truth assignment* is a mapping $\tau : X \to \{0, 1\}$ defined on some set X of variables; we write $\operatorname{var}(\tau) = X$. For $x \in \operatorname{var}(\tau)$ we define $\tau(\bar{x}) = 1 - \tau(x)$. For a truth assignment τ and a formula F, we define $F[\tau] = \{\,C \setminus \tau^{-1}(0) : C \in F,\ C \cap \tau^{-1}(1) = \emptyset\,\}$, i.e., $F[\tau]$ denotes the result of instantiating variables according to τ and applying the usual simplifications, i.e., removing clauses that are satisfied by τ and removing unsatisfied literals from clauses. A truth assignment τ *satisfies* a clause C if C contains some literal x with $\tau(x) = 1$; τ satisfies a formula F if it satisfies all clauses of F. A formula is *satisfiable* if it is satisfied by some truth assignment; otherwise it is *unsatisfiable*.

The SATISFIABILITY (SAT) problem asks whether a given CNF formula is satisfiable.

Parameterized Complexity An instance of a parameterized problem is a pair (I, k) where I is the *main part* and k is the *parameter*; the latter is usually a non-negative integer. A parameterized problem is *fixed-parameter tractable* if there exist a computable function f and a constant c such that instances (I, k) can be solved in time $O(f(k)\|I\|^c)$ where $\|I\|$ denotes the size of I. FPT is the class of all fixed-parameter tractable decision problems and algorithms which run in the time specified above are called FPT algorithms.

An FPT-*reduction* is a many-one reduction where the parameter for one problem maps into the parameter for the other. More specifically, given two parameterized decision problems L and L', problem L reduces to problem L' if there is a mapping R from instances of L to instances of L' such that (i) (I, k) is a yes-instance of L if and only if $(I', k') = R(I, k)$ is a yes-instance of L', (ii) $k' \leq g(k)$ for a computable function g, and (iii) R can be computed in time $O(f(k)\|I\|^c)$ where f is a computable function and c is a constant.

The *Weft Hierarchy* consists of parameterized complexity classes $\mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots$ which are defined as the closure of certain parameterized problems under FPT-reductions (see [10,13] for definitions). There is strong theoretical evidence that parameterized problems that are hard for classes $\mathrm{W}[i]$ are unlikely to be fixed-parameter tractable. For example FPT $=$ W[1] implies that the Exponential Time Hypothesis (ETH) fails; that is, FPT $=$ W[1] implies the existence of a $2^{o(n)}$ algorithm for n-variable 3-SAT [13].

An *FPT-approximation algorithm* with ratio ρ for a minimization problem P is an FPT algorithm that, given an instance x of P and a positive integer k, either determines that there is no solution of size at most k or computes a solution of size at most $k\rho(k)$ (see, e.g., [11]). The definition can be adapted to maximization problems. Note that the approximation ratio ρ is a function of k and not the input size: intuitively, if k is small, then $k\rho(k)$ can be still considered small. We say that a problem is *FPT-approximable* if it has an FPT-approximation algorithm for some function ρ.

Backdoors Here, we introduce the basic terminology for backdoors and the class of q-Horn formulas. For further information on backdoors and other tractable base classes of SATISFIABILITY we refer the reader to [16].

Backdoors are defined with respect to a fixed class $\mathcal{C}$ of CNF formulas, the *base class* (or *target class*, or more figuratively, *island of tractability*). We say a class $\mathcal{C}$ of formulas is *clause-induced* if it is closed under subsets, i.e. , if $F \in \mathcal{C}$ implies $F' \in \mathcal{C}$ for each $F' \subseteq F$.

A *strong* $\mathcal{C}$*-backdoor set* of a CNF formula F is a set B of variables such that $F[\tau] \in \mathcal{C}$ for each assignment $\tau : B \to \{0, 1\}$. A *weak* $\mathcal{C}$*-backdoor set* of F is a set

B of variables such that $F[\tau]$ is satisfiable and $F[\tau] \in \mathcal{C}$ holds for some assignment $\tau : B \to \{0, 1\}$. A *deletion $\mathcal{C}$-backdoor set* of F is a set B of variables such that $F - B \in \mathcal{C}$. Backdoor sets were independently introduced by Crama et al. [9] and by Williams et al. [28], with the latter set of authors coining the term "backdoor".

If we know a strong $\mathcal{C}$-backdoor set of F of size k, we can reduce the satisfiability of F to the satisfiability of 2^k formulas in $\mathcal{C}$. Thus SAT becomes fixed-parameter tractable with k as the parameter. If we know a weak $\mathcal{C}$-backdoor set of F, then F is clearly satisfiable, and we can verify it by trying for each $\tau \in \{0, 1\}^k$ whether $F[\tau]$ is in $\mathcal{C}$ and satisfiable. If $\mathcal{C}$ is clause-induced, every deletion $\mathcal{C}$-backdoor set of F is a strong $\mathcal{C}$-backdoor set of F. For several base classes, deletion backdoor sets are of interest because they are easier to detect than strong backdoor sets. The challenging problem is to find a strong, weak, or deletion $\mathcal{C}$-backdoor set of size at most k if it exists. For each class $\mathcal{C}$ of CNF formulas, the deletion backdoor detection problem is defined as follows with other variants having analogous definitions.

DELETION $\mathcal{C}$-BACKDOOR SET DETECTION **Parameter:** k
Input: A CNF formula F and a positive integer k
Question: Does F have a deletion $\mathcal{C}$-backdoor set of size at most k?

q-Horn Formulas In this paper we are mainly interested in the class of q-Horn formulas [6,7]. A CNF formula F is in this class if there is a *certifying function* $\beta : \text{var}(F) \cup \overline{\text{var}(F)} \to \{0, \frac{1}{2}, 1\}$ with $\beta(x) = 1 - \beta(\bar{x})$ for every $x \in \text{var}(F)$ such that $\sum_{l \in C} \beta(l) \leq 1$ for every clause C of F.

In the following sense, strong q-Horn-backdoor sets are more general than deletion q-Horn-backdoor sets: For every positive integer n there is a formula F_n such that F_n has a strong q-Horn-backdoor set of size 1 but every deletion q-Horn-backdoor set of F has size at least n. To see this, take for instance $F = \bigcup_{1 \leq i \leq n} \{\{x_i, y_i, z_i, a\}, \{\bar{x}_i, \bar{y}_i, \bar{z}_i, \bar{a}\}\}$. Evidently, $\{a\}$ is a strong q-Horn-backdoor set of F. However, every deletion q-Horn-backdoor set of F must contain at least one variable x_i, y_i, or z_i for every $1 \leq i \leq n$.

3 FPT-Approximation for DELETION q-Horn BACKDOOR SET DETECTION

In this section we prove our main result:

Theorem 1 *There is an algorithm that, given an instance (F, k) of* DELETION *q-Horn* BACKDOOR SET DETECTION*, runs in time $O(6^k k\ell n)$ and either correctly concludes that F has no deletion q-Horn-backdoor set of size at most k or returns a deletion q-Horn-backdoor set of F of size at most $k^2 + k$, where ℓ is the length of F and n is the number of variables in F.*

3.1 Quadratic Covers, Implication Graphs and Separators

In this subsection we give some definitions regarding quadratic covers, implication graphs and separators in implication graphs, which will be required for the description

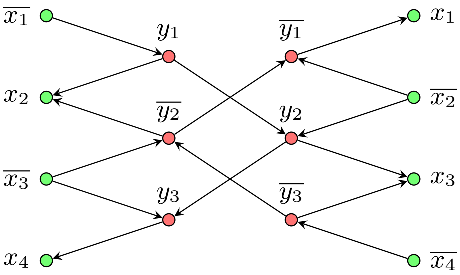

Fig. 1 An illustration of the implication graph ($D(F_2)$) for the formula F consisting of the clause $\{x_1, x_2, x_3, x_4\}$

of our algorithm. The following definition of the quadratic cover of a CNF formula was used Boros et al. [7] to give a linear time algorithm to recognize q-Horn formulas.

Definition 1 Given a CNF formula F, the quadratic cover of F, is a 2-CNF formula denoted by F_2 and is defined as follows. Let $x_1, \ldots, x_n$ be the variables of F. For every clause C, we have $|C| - 1$ new variables $y_1^C, \ldots, y_{|C|-1}^C$. We order the literals in each clause according to their variables, that is, a literal of x_i will occur before a literal of x_j if $i < j$. Let $l_1^C, \ldots, l_{|C|}^C$ be the literals of the clause C in this order. The quadratic cover is defined as

$$F_2 = \bigcup_{C \in F} \bigcup_{1 \le i \le |C|-1} \{\{l_i^C, y_i^C\}, \{\bar{y}_i^C, l_{i+1}^C\}\}$$
$$\cup \quad \bigcup_{C \in F} \bigcup_{1 \le i \le |C|-2} \{\{\bar{y}_i^C, y_{i+1}^C\}\}.$$

Definition 2 Given a CNF formula F, the implication graph of F_2 is denoted by $\mathrm{D}(F_2)$ and defined as follows. The vertex set of the graph is the set of literals of F_2 and for every clause $\{l_1, l_2\}$ in F_2, we have arcs $(\bar{l}_1, l_2)$ and $(\bar{l}_2, l_1)$. We refer to the vertices of the implication graph as literals since there is a one to one correspondence between the two. Given a set $X \subseteq var(F)$ of variables, we define the graph $\mathrm{D}(F_2){-}X$ as the graph obtained from $\mathrm{D}(F_2)$ by deleting $\mathrm{lit}(X)$.

An illustration for Definition 2 is shown in Fig. 1.

The following observations are direct consequences of the definition of an implication graph.

Observation 1 *Let F be a CNF formula of length ℓ.*

(a) If there is a path from l_1 to l_2 in $\mathrm{D}(F_2)$, then there is also a path from $\bar{l}_2$ to $\bar{l}_1$ in $\mathrm{D}(F_2)$.
(b) The number of arcs in $\mathrm{D}(F_2)$ is $O(\ell)$.
(c) Let $C = \{l_1, \ldots, l_r\}$ be a clause of F. Then, for any $1 \le i < j \le r$, $\mathrm{D}(F_2)$ contains a path from $\bar{l}_i$ to l_j and from $\bar{l}_j$ to l_i whose internal vertices are all disjoint from $\mathrm{lit}(F)$.
(d) Let $X \subseteq var(F)$ and $F' = F - X$. Then, for any literal $l \in \mathrm{lit}(F) \setminus \mathrm{lit}(X)$, there is a path from l to $\bar{l}$ in $\mathrm{D}(F'_2)$ if and only if there is a path from l to $\bar{l}$ in $\mathrm{D}(F_2){-}X$.

Proof (a) This follows from the fact that for every arc (l_i, l_j) in $\mathrm{D}(F_2)$, there is also an arc $(\bar{l}_j, \bar{l}_i)$.

(b) Observe that a clause $C \in F$ contributes at most $3|C|$ clauses to F_2. Hence, the number of clauses in F_2 is bounded by 3ℓ. Since each clause of F_2 contributes 2 arcs to $\mathrm{D}(F_2)$, the number of arcs in $\mathrm{D}(F_2)$ is bounded by 6ℓ.

(c) From the definition of the formula F_2 and the implication graph $\mathrm{D}(F_2)$, we know that $\mathrm{D}(F_2)$ contains a directed path $P_1 = y_1^C, y_2^C, \ldots, y_{r-1}^C$. Furthermore, there are arcs $(\bar{l}_i, y_i^C)$ and (y_{j-1}^C, l_j) in $\mathrm{D}(F_2)$. Along with the path P_1, these arcs imply that there is a path from $\bar{l}_i$ to l_j and by (a), a path from $\bar{l}_j$ to l_i. Clearly, in both these paths, the internal vertices are disjoint from $\mathrm{lit}(F)$.

(d) In order to prove the statement it is sufficient to show that for any two literals l_1 and l_2 such that $\mathrm{var}(l_1) \neq \mathrm{var}(l_2)$, there is a path from l_1 to l_2 in $\mathrm{D}(F_2')$ if and only if there is a path from l_1 to l_2 in $\mathrm{D}(F_2)-X$. Furthermore, in order to prove this statement, it suffices to show that this statement is true for all paths where the internal vertices are not in $\mathrm{lit}(F)$. The statement will then follow by induction on the number of literals of F along the paths.

Consider a path from l_1 to l_2 in $\mathrm{D}(F_2')$ where the internal vertices are all disjoint from $\mathrm{lit}(F)$. We claim that $\bar{l}_1$ and l_2 occur in a clause in F'. This is so since the definitions of F_2' and $\mathrm{D}(F_2')$ imply that any literal not in $\mathrm{lit}(F)$ corresponds to a single clause of F' and has arcs only to or from other literals of this type which also correspond to this clause or to a literal in $\mathrm{lit}(F)$ which occurs positively in the same clause or from a literal in $\mathrm{lit}(F)$ which occurs negatively in the same clause. Therefore, $\bar{l}_1$ and l_2 also occur in a clause in F. This implies that there is a path from l_1 to l_2 in $\mathrm{D}(F_2)$ where the internal vertices are disjoint from $\mathrm{lit}(F)$ (by (c)). Therefore, removing $\mathrm{lit}(X)$ from $\mathrm{D}(F_2)$ does not affect this path.

Conversely, consider a path from l_1 to l_2 in $\mathrm{D}(F_2)-X$ such that the internal vertices are disjoint from $\mathrm{lit}(F)$. Since this is also a path in $\mathrm{D}(F_2)$, by our previous argument, it implies that $\bar{l}_1$ and l_2 occur in a clause in F and hence in F' and by (c), there is a path from l_1 to l_2 in $\mathrm{D}(F_2')$ where the internal vertices are disjoint from $\mathrm{lit}(F)$.

Definition 3 Given a CNF formula F and a set L of literals of F, we denote by $N_F^+(L)$ the set of literals in $\mathrm{lit}(F) \setminus L$ which can be reached from L in $\mathrm{D}(F_2)$ via a path whose internal vertices are disjoint from $\mathrm{lit}(F)$.

The following definition and lemma in Boros et al. [7] give a notion of a canonical certificate for a q-Horn formula.

Definition 4 ([7]) Given a CNF formula F, define a canonical function $\hat{\beta} : \mathrm{lit}(F) \rightarrow \{0, \frac{1}{2}, 1\}$ as follows. Consider a topological ordering π of the strongly connected components of $\mathrm{D}(F_2)$. For every literal $l \in \mathrm{lit}(F)$ such that the strongly connected component containing l appears before the one containing $\bar{l}$ in π, set $\hat{\beta}(l) = 1$ and for every literal l such that the strongly connected component containing l also contains $\bar{l}$, set $\hat{\beta}(l) = \frac{1}{2}$.

Lemma 1 ([7]) *A CNF formula F is q-Horn if and only if the function $\hat{\beta}$ defined above is a certifying function for F.*

The above lemma can be intuitively understood as follows. Consider a formula F. If there is an l_1-l_2 path in the graph $\mathrm{D}(F_2)$ where the internal vertices are not in $\mathrm{lit}(F)$, then there is a clause containing the literals $\bar{l}_1$ and l_2 (by the definition of implication graphs), and hence, for any certifying function β such that $\beta(\bar{l}_1) = 0$, it must be the case that $\beta(l_2) = 0$. Similarly, for any certifying function β such that $\beta(l_2) = 1$, it must be the case that $\beta(\bar{l}_1) = 1$. The same argument can be extended to l_1-l_2 paths with internal vertices from $\mathrm{lit}(F)$. This implies that, if there is a closed walk containing l_1 and $\bar{l}_1$ in $\mathrm{D}(F_2)$, then for any certifying function β, $\beta(l_1) = \beta(\bar{l}_1) = \frac{1}{2}$.

Definition 5 A clause C of a given CNF formula is called a violating clause if $\sum_{l \in C} \hat{\beta}(l) > 1$. Any three literals l_1, l_2, l_3 of a violating clause such that $\sum_{i=1}^{3} \hat{\beta}(l_i) > 1$ form a violating triple.

Lemma 2 *Let F be a CNF formula of length ℓ and suppose that F is not a q-Horn formula. Any violating clause of F has a violating triple lying entirely inside a strongly connected component of* $\mathrm{D}(F_2)$ *and we can compute such a violating triple in time $O(\ell)$.*

Proof Consider a violating clause $C = \{l_1, \ldots, l_r\}$. We first show that for any l_i, $\hat{\beta}(l_i) \leq \frac{1}{2}$. Suppose this is not the case and there is an l_i such that $\hat{\beta}(l_i) = 1$. Since C is a violating clause, there is also a literal l_j such that $\hat{\beta}(l_j) > 0$. It follows that l_j and $\bar{l}_j$ either occur in the same strongly connected component of $\mathrm{D}(F_2)$ or the strongly connected component containing l_j occurs before that containing $\bar{l}_j$. However, we already know that l_i is in a strongly connected component occurring before $\bar{l}_i$. But, there is an $\bar{l}_i$-l_j path and an $\bar{l}_j$-l_i path in $\mathrm{D}(F_2)$ [(Observation 1(c)], implying that $\bar{l}_i$ is in a strongly connected component occurring before the strongly connected component containing l_i, a contradiction. Therefore for any l_i, $\hat{\beta}(l_i) \leq \frac{1}{2}$.

Since C is a violating clause, there must be at least three literals l_i, l_j, l_m such that $\hat{\beta}(l_i) = \hat{\beta}(l_j) = \hat{\beta}(l_m) = \frac{1}{2}$. By the definition of $\hat{\beta}$, each pair of complementary literals appear in the same strongly connected component of $\mathrm{D}(F_2)$. However, there are paths from $\bar{l}_i$ to l_j and l_m and from $\bar{l}_j$ and $\bar{l}_m$ to l_i [Observation 1(c)]. This implies that the literals l_i, l_j, l_m and their complements all appear in the same strongly connected component of $\mathrm{D}(F_2)$. Observe that to compute the violating triple, it is sufficient to construct $\mathrm{D}(F_2)$ and compute a violating clause, which can be done in time $O(\ell)$ [7].

We now move on to some definitions on separators in implication graphs which will be required in the description of our algorithm.

Definition 6 Let F be a CNF formula and $L \subseteq \mathrm{lit}(F)$ be a consistent set of literals. We say that a set $J \subseteq \mathrm{lit}(F)$ is an L-$\bar{L}$ separator if J is disjoint from L and $\bar{L}$ and there is no path from L to $\bar{L}$ in the graph $\mathrm{D}(F_2) - J$. We say that J is a minimal L-$\bar{L}$ separator if no proper subset of J is an L-$\bar{L}$ separator.

Definition 7 Let F be a CNF formula, $L \subseteq \mathrm{lit}(F)$ be a consistent set of literals and let X be a set of variables of F. We call X an L-$\bar{L}$ variable separator if $\mathrm{lit}(X)$ is an L-$\bar{L}$ separator. We call X a minimal L-$\bar{L}$ variable separator if no proper subset of X is an L-$\bar{L}$ variable separator. We drop the word *variable* if it is clear from the context that the set we are dealing with is a set of variables.

Definition 8 Let F be a CNF formula, $L \subseteq \text{lit}(F)$ be a consistent set of literals and X be an L-$\bar{L}$ variable separator. We denote by $R(L, X)$ the set of literals of F that can be reached from L via directed paths in $\text{D}(F_2){-}X$, and we denote by $\bar{R}(L, X)$ the set of literals of F which have a directed path to L in $\text{D}(F_2){-}X$.

The following observation is implicit in the proof of Observation 1(d). We state it explicitly in the form in which it will be used as part of the description of our algorithm.

Observation 2 *Let F be a CNF formula, $L \subseteq \text{lit}(F)$ be a consistent set of literals and X be an L-$\bar{L}$ variable separator. Then, the sets $R(L, X)$ and $\bar{R}(\bar{L}, X)$ are also consistent and in fact complements of each other.*

3.2 The Algorithm

We first prove a lemma (Lemma 3) which when combined with Lemma 2, allows us to compute in linear time, a set of three literals such that for every solution X one of the three corresponding variables is part of X or for at least one of these literals, say l, there is a path from l to $\bar{l}$ in $\text{D}(F_2)$ and X is an l-$\bar{l}$ variable separator in $\text{D}(F_2)$.

Lemma 3 *Let (F, k) be an instance of* Deletion *q-Horn* Backdoor Set Detection. *Let (l_1, l_2, l_3) be a violating triple in a strongly connected component of* $\text{D}(F_2)$ *and X be a solution for the given instance disjoint from $\{var(l_1), \text{var}(l_2), \text{var}(l_3)\}$. Then, for some $1 \leq i \leq 3$, X is an l_i-$\bar{l}_i$ separator in* $\text{D}(F_2)$.

Proof Let $\hat{\beta}'$ be the canonical certifying function for $F' = F - X$ obtained from the graph $\text{D}(F'_2)$. We claim that there is an $1 \leq i \leq 3$ such that $\hat{\beta}'(l_i) = 0$. This is true since F' contains a clause with all three literals l_1, l_2 and l_3 and it cannot be the case that any certifying function sets non-zero values to all three. By definition of $\hat{\beta}'$, $\hat{\beta}'(l_i) = 0$ implies that there is no path from l_i to $\bar{l}_i$ in the graph $\text{D}(F'_2)$. If X were not an l_i-$\bar{l}_i$ separator in $\text{D}(F_2)$, then $\text{D}(F'_2)$ would also contain an l_i-$\bar{l}_i$ path [by Observation 1(d)], a contradiction. This completes the proof of the lemma.

The structural properties guaranteed by the above lemma imply a starting point for our branching steps. More precisely, we branch on the three variables in a violating triple and in each branch, we either find a variable which is part of a solution or find a literal l_i such that some solution is an l_i-$\bar{l}_i$ separator in $\text{D}(F_2)$. The following lemma gives us a sufficient condition for when we can perform a *cut and paste* operation on a deletion q-Horn-backdoor set.

Lemma 4 *Let (F, k) be an instance of* Deletion *q-Horn* Backdoor Set Detection *and X be a solution such that it is disjoint from* $\text{var}(l)$ *and is an l-$\bar{l}$ separator for some literal $l \in \text{lit}(F)$. Consider an l-$\bar{l}$ variable separator X'. Let X'' be the set of variables of X with a literal in $R(l, X')$. Then, the set $\tilde{X} = (X \setminus X'') \cup X'$ is also a deletion q-Horn-backdoor set for the given instance.*

Proof Let $F' = F - X$ and $\tilde{F} = F - \tilde{X}$. If $\tilde{X}$ were not a deletion q-Horn-backdoor set, then there is a violating clause in $\tilde{F}$ and by Lemma 2, there is a violating triple

(l_1, l_2, l_3) in a strongly connected component of $\mathrm{D}(\tilde{F}_2)$. This implies the presence of a closed walk in $\mathrm{D}(\tilde{F}_2)$ containing all the literals of the violating triple and their complements (by Lemma 2). Since X was a solution, this closed walk could not have survived in $\mathrm{D}(F_2')$ and hence must contain a literal of a variable in $X \setminus \tilde{X}$. Recall that the only variables of X that are not in $\tilde{X}$ are those in X''. Let p be a literal on this closed walk which corresponds to such a variable, that is, $\mathrm{var}(p) \in X''$. On the other hand, by definition, the literals of the variables in X'' can either reach $\bar{l}$ or be reached from l in $\mathrm{D}(\tilde{F}_2)$, that is, they must lie in $R(l, \tilde{X})$ or $\bar{R}(\bar{l}, \tilde{X})$. This implies that either p and $\bar{p}$ are both reachable from l in $\mathrm{D}(\tilde{F}_2)$ or both these vertices have paths to $\bar{l}$ in $\mathrm{D}(\tilde{F}_2)$. We first consider the case when p and $\bar{p}$ are reachable from l in $\mathrm{D}(\tilde{F}_2)$. Since $\mathrm{D}(\tilde{F}_2)$ is an implication graph, a path from l to $\bar{p}$ implies a path from p to $\bar{l}$, which implies that there is a path from l to $\bar{l}$ via p in $\mathrm{D}(\tilde{F}_2)$. Similarly, the case when p and $\bar{p}$ have paths to $\bar{l}$ in $\mathrm{D}(\tilde{F}_2)$ also leads to the presence of a path from l to $\bar{l}$ in $\mathrm{D}(\tilde{F}_2)$. However, by construction, $\tilde{X}$ is also an l-$\bar{l}$ separator in $\mathrm{D}(F_2)$. Observation 1(d) implies that this is a contradiction. This completes the proof of the lemma.

Note that the above lemma says nothing about the optimality of the deletion q-Horn-backdoor sets under consideration. The next lemma gives an algorithm that either concludes that there is no set of the required kind, or returns a set which can be used in the cut and paste operation described in the previous lemma.

Lemma 5 *Let (F, k) be an instance of* Deletion *q-Horn* Backdoor Set Detection *where F is a CNF formula of length ℓ, with n variables. Let X be a solution to the given instance and let l be a literal of F such that there is an l-$\bar{l}$ path in $\mathrm{D}(F_2)$. Furthermore, suppose that X is an l-$\bar{l}$ variable separator. Then, there is an algorithm that, given F, k and l, runs in time $O(k\ell n)$ and either concludes correctly that there is no k-sized l-$\bar{l}$ variable separator in $\mathrm{D}(F_2)$ or returns an l-$\bar{l}$ variable separator X' of size at most $2k$ such that $(X' \cup \mathrm{var}(R(l, X'))) \cap X$ is non-empty.*

Proof We show that Algorithm 3.1 has the stated properties. The algorithm computes an l-$\bar{l}$ variable separator X' which essentially maximizes the set of literals of $\mathrm{D}(F_2)$ reachable from l after removing X'. We will then show that such a separator indeed has the required properties.

If the algorithm returns No in Line 4, then $\mathrm{D}(F_2)$ has no l-$\bar{l}$ variable separator of size at most k. Let S be the minimal separator in $\mathrm{D}(F_2)$ which was computed in the penultimate iteration of the while loop. We claim that $X' = \mathrm{var}(S)$ satisfies the conditions in the statement of the lemma. Clearly, it must be the case that for some choice of a literal l' in $\mathrm{lit}(\mathrm{var}(S)) \cap N_F^+(L)$, the next iteration of the loop could not find an $L \cup \{l'\}$-$\bar{L} \cup \{\bar{l}'\}$ separator of size at most $2k$. We now show that $(\mathrm{var}(S) \cup \mathrm{var}(R(l, \mathrm{var}(S)))) \cap X$ is non-empty.

Suppose that this is not the case. Recall that when the procedure stops, $L = R(l, \mathrm{var}(S))$. Furthermore, since there is at least one path from l to $\bar{l}$ in $\mathrm{D}(F_2)$ it must be the case that $\mathrm{lit}(\mathrm{var}(S)) \cap N_F^+(L)$ is non-empty. Since X is an l-$\bar{l}$ separator and disjoint from L, X is also an L-$\bar{L}$ separator. Since X is also disjoint from $\mathrm{var}(S)$, for any $l' \in \mathrm{lit}(\mathrm{var}(S)) \cap N_F^+(L)$, X intersects all paths from $L \cup \{l'\}$ to $\bar{L} \cup \{\bar{l}'\}$. Hence, $\mathrm{lit}(X)$ is a set of size at most $2k$ which intersects all $L \cup \{l'\}$-$\bar{L} \cup \{\bar{l}'\}$ paths, which is

```
Input  : A tuple $(F, k, l)$ where $F$ is a CNF formula, $k$ a positive integer and $l$ a literal of $F$
Output: NO provided that $\mathrm{D}(F_2)$ has no $l$-$\bar{l}$ variable separator of size at most $k$, or an $l$-$\bar{l}$ variable
        separator $S$ of size at most $2k$ such that $(S \cup \mathrm{var}(R(l, S)))$ has non-empty intersection with
        any minimum deletion q-Horn-backdoor set
if there is an $l$-$\bar{l}$ separator of size at most $2k$ in $\mathrm{D}(F_2)$ then
    $S \leftarrow$ such a separator
end
else return NO
$L \leftarrow R(l, \mathrm{var}(S))$ // $L$ is consistent by Observation 2
while there is an $L \cup \{l'\}$-$\bar{L} \cup \{\bar{l'}\}$ separator of size at most $2k$ where $l' \in (\mathrm{lit}(\mathrm{var}(S)) \cap N^+_F(L))$ is
      an arbitrarily chosen such literal do
    $S \leftarrow$ such a separator
    $L \leftarrow R(L, \mathrm{var}(S))$
end
return $\mathrm{var}(S)$
```

Algorithm 3.1: Algorithm COMPUTE-SEPARATOR

a contradiction. Therefore, the set $(\mathrm{var}(S) \cup \mathrm{var}(R(l, \mathrm{var}(S)))) \cap X$ is non-empty for any l-$\bar{l}$ variable separator X of size at most k.

To bound the running time, observe that in each iteration, we only need to test if there is an L-$\bar{L}$ separator of size at most $2k$. Hence, it suffices for us to run the Ford-Fulkerson algorithm [14] for at most $2k$ steps on the graph $\mathrm{D}(F_2)$ and the number of iterations is bounded by the number of variables in the formula since in each iteration, we add a literal to L. Since the number of arcs in $\mathrm{D}(F_2)$ is $O(\ell)$ (Observation 1(b)), the claimed time bound follows. This completes the proof of the lemma.

The previous lemma (Lemma 5) and the next lemma (Lemma 6) will give us an algorithm to compute a bounded set of variables whose deletion from the formula results in an instance that has a solution which is strictly smaller than any solution of the input instance.

Lemma 6 *Let (F, k) be an instance of* DELETION *q-Horn* BACKDOOR SET DETECTION *and let l be a literal of F disjoint from a solution X and suppose that X is an l-$\bar{l}$ variable separator in* $\mathrm{D}(F_2)$. *Consider an l-$\bar{l}$ variable separator in* $\mathrm{D}(F_2)$, *X', such that $(X' \cup \mathrm{var}(R(l, X'))) \cap X$ is non-empty. Then, the instance $F - X'$ has a deletion q-Horn-backdoor set of size at most $|X| - 1$.*

Proof By Lemma 4, we know that the set $\hat{X} = (X \setminus X'') \cup X'$ is a deletion q-Horn-backdoor set. Hence, $X \setminus (X'' \cup X')$ is indeed a deletion q-Horn-backdoor set for the instance $F - X'$. Since $(X' \cup X'') \cap X$ is non-empty, the size of $X \setminus (X'' \cup X')$ is at most $|X| - 1$. This completes the proof of the lemma.

This completes the formalization of our ideas and we are now ready to prove Theorem 1 by describing our algorithm for DELETION q-Horn BACKDOOR SET DETECTION.

Description of the Algorithm Algorithm 3.2 checks whether there is a violating triple and if so, computes one and in the first 3 branches, it adds the variable corresponding to each of the literals of the violating triple to the solution, deletes it from the formula

```
    Input  : A CNF formula $F$ of length $\ell$ with $n$ variables, integer $k$
    Output: Either no solution of size at most $k$ or a solution of size at most $k^2 + k$ for the instance
            $(F, k)$ of DELETION q-Horn BACKDOOR SET DETECTION
 1  if $k < 0$ then return NO
 2  check for a violating clause by computing $\mathrm{D}(F_2)$ and a topological ordering of $\mathrm{D}(F_2)$
 3  if there is no violating clause then return $\emptyset$
 4  Compute a violating triple $(l_1, l_2, l_3)$
 5  for $l = l_1, l_2, l_3$ do
 6      $S_1 \leftarrow$ DELETION-QHORN-BSD($F - \{var(l)\}, k - 1$)
 7      if $S_1$ is not NO then return $S_1 \cup \{var(l)\}$
 8  end
 9  for $l = l_1, l_2, l_3$ do
10      $S \leftarrow$ COMPUTE-SEPARATOR($F, k, l$)
11      if $S$ is NO then return NO else
12          $S_1 \leftarrow$ DELETION-QHORN-BSD($F - \{S\}, k - 1$)
13      end
14      if $S_1$ is not NO then return $S_1 \cup \{S\}$
15  end
16  return NO
```

Algorithm 3.2: Algorithm DELETION-QHORN-BSD

and recurses on the resulting instance with a budget of $k - 1$. In each of the next 3 branches, it picks a literal of the violating triple and continues by assuming that this literal is assigned 0 by a certifying function of $F - X$ where X is a solution. We know that there must be at least one such literal (see the proof of Lemma 3) in the violating triple. This implies that X is an l-$\bar{l}$ separator for the literal l in the violating triple which is assigned 0 by a certifying function of $F - X$. Finally, Lemma 5 is used to either conclude that there is no l-$\bar{l}$ variable separator of size at most k in which case the algorithm returns NO, or to compute an l-$\bar{l}$ variable separator of size at most $2k$ with the required properties. The variables in X' are added to our proposed approximate solution and deleted from the formula, and the algorithm recurses on the resulting instance with a budget of $k - 1$.

Analysis Since Steps 2, 4, and 10 at any node of the search tree take time $O(k\ell n)$ and we have a 6-way branching at each node of the search tree with the budget k dropping by 1 in each branch, the algorithm clearly runs in the claimed time bound. Therefore, it only remains for us to prove the correctness of the algorithm. Let X be a solution for the given instance and let β be a certifying function for $F - X$. We prove the correctness of the algorithm by induction on k.

In the base case, when $k = 0$, the algorithm is correct by Lemma 1. We assume as induction hypothesis that the algorithm is correct for all values of k up to some $k' - 1$ where $k' - 1 > 0$. That is, any solution returned by the algorithm for any value of $k \leq k' - 1$ is bounded by $k^2 + k$. We now consider the case when $k = k'$.

In Lines 5–8, we consider the case when X has a non-empty intersection with the set $\{\mathrm{var}(l_1), \mathrm{var}(l_2), \mathrm{var}(l_3)\}$ and branch accordingly. Since the budget in the recursive call is $k - 1$, the induction hypothesis gives us that the size of any returned solution in a subsequent recursive call is at most $(k - 1)^2 + (k - 1)$. Hence, the size of a solution returned here is bounded by $1 + (k - 1)^2 + (k - 1) \leq k^2 + k$.

In Lines 9–15, we consider the case when X is disjoint from the set of variables corresponding to l_1, l_2 and l_3. Since l_1, l_2, l_3 lie in the same clause and none of their corresponding variables are in X, by Lemma 3, X is an l_i-$\bar{l}_i$ separator for at least one of the literals l_i. Let us assume that this literal is l_1. In Line 10, we apply Lemma 5 to compute an l_1-$\bar{l}_1$ separator S of size at most $2k$ and add it to the solution we are constructing. By Lemma 6, we know that there is a solution for the instance $F - S$ of size at most $|X| - 1$. Hence, by the induction hypothesis, we obtain a solution of size at most $(k-1)^2 + (k-1)$ from the subsequent recursive call and adding to it the set S of size at most $2k$ results in a solution of size at most $k^2 + k$, which proves the correctness of the algorithm, completing the proof of Theorem 1.

In order to test the satisfiability of a given CNF formula F, it suffices to first compute a smallest deletion q-Horn-backdoor set of F and for each assignment to this set, test the satisfiability of the reduced formula which is q-Horn. Since testing satisfiability of a q-Horn formula can be done in linear time [6], Theorem 1 has the following corollary.

Corollary 1 *There is an algorithm that, given a formula F of length ℓ with n variables, runs in time $2^{O(k^2)}\ell n$ and decides the satisfiability of F, where k is the size of the smallest deletion q-Horn-backdoor set of F.*

4 Hardness

In this section we show that there is no FPT algorithm for Strong q-Horn-backdoor Set Detection or Weak q-Horn-backdoor Set Detection unless FPT=W[2]. In order to show this, we begin from the following problem, which is well-known to be W[2]-complete [10].

Hitting Set **Parameter:** k
Input: A set E of elements, a family $\mathcal{S}$ of finite subsets of E, and an integer $k > 0$.
Question: Does $\mathcal{S}$ have a hitting set, i.e., a subset H of E such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{S}$, of size at most k?

Theorem 2 Strong *q-Horn*-backdoor Set Detection *is W[2]-hard.*

Proof We prove the theorem via an FPT-reduction from Hitting Set. Let $(E, \mathcal{S}, k)$ be an instance of Hitting Set. We construct a formula F that has a strong q-Horn-backdoor set of size at most k if and only if $\mathcal{S}$ has a hitting set of size at most k. The formula F has two clauses $P_S^i = S \cup \{x_i, y_i, z_i\}$ and $N_S^i = \bar{E} \cup \{\bar{x}_i, \bar{y}_i, \bar{z}_i\}$ for every $S \in \mathcal{S}$ and $1 \leq i \leq k+1$. Note that $\text{var}(F) = E \cup \{x_i, y_i, z_i : 1 \leq i \leq k+1\}$. Furthermore, for any S and for any $1 \leq i \leq k+1$, the formula comprising the two clauses P_S^i and N_S^i is clearly not q-Horn. Let H be a hitting set of $\mathcal{S}$ of size at most k. We show that $B = H$ is a strong q-Horn-backdoor set for F. Let $\tau : B \to \{0, 1\}$ be an assignment of the variables in B. If τ sets at least one variable in B to 0 then $F[\tau]$ contains only positive clauses and hence $F[\tau]$ is trivially q-Horn. On the other hand if τ sets all variables in B to 1 then because H is a hitting set of $\mathcal{S}$ the formula $F[\tau]$ contains only negative clauses and hence $F[\tau]$ is again trivially q-Horn. Consequently, B is a strong q-Horn-backdoor set for F.

For the reverse direction suppose that $B \subseteq \mathrm{var}(F)$ is a strong q-Horn-backdoor set for F of size at most k. We show that $H = B \cap E$ is a hitting set of $\mathcal{S}$ of size at most k. Suppose for a contradiction that H is not a hitting set for $\mathcal{S}$ and let $S \in \mathcal{S}$ be such that $H \cap S = \emptyset$. Furthermore, let τ be the assignment of the variables in B that sets all variables to 1. It follows that $F[\tau]$ contains two clauses P_S^i and N_S^i for some $1 \leq i \leq k+1$ and hence $F[\tau]$ is not q-Horn, a contradiction to the assumption that B is a strong q-Horn-backdoor set for F.

Theorem 3 WEAK *q-Horn*-BACKDOOR SET DETECTION *is W[2]-hard, even for 3-CNF formulas.*

Proof We prove the theorem via an FPT-reduction from HITTING SET. Let $(E, \mathcal{S}, k)$ be an instance of HITTING SET. We construct a 3-CNF formula F that has a weak q-Horn-backdoor set of size at most k if and only if $\mathcal{S}$ has a hitting set of size at most k. For every $S \in \mathcal{S}$ with $S = \{s_1, \ldots, s_{|S|}\}$, every $1 \leq i \leq |S|$, and every $1 \leq j \leq k+1$ the formula F contains the clauses $\{z_i^j(S), \bar{s}_i, \bar{z}_{i+1}^j(S)\}$, $\{\bar{z}_1^j(S), z_{|S|+1}^j(S)\}$, $\{\bar{z}_1^j(S), \bar{z}_{|S|+1}^j(S)\}$, $\{z_1^j(S), z_{|S|+1}^j(S)\}$, $\{z_{|S|+1}^j(S), a^j(S), b^j(S)\}$, and finally $\{\bar{a}^j(S), \bar{b}^j(S)\}$. Note that $\mathrm{var}(F) = E \cup \{\, z_i^j(S) : S \in \mathcal{S}$ and $1 \leq i \leq |S|+1$ and $1 \leq j \leq k+1 \,\} \cup \{\, a^j(S), b^j(S) : S \in \mathcal{S}$ and $1 \leq j \leq k+1 \,\}$. Note furthermore that F is satisfiable by the assignment τ_{SAT} that sets the variables in $\{\, z_{|S|+1}^j(S), a^j(S) : S \in \mathcal{S}$ and $1 \leq j \leq k+1 \,\}$ to 1 and all other variables to 0. Let H be a hitting set of $\mathcal{S}$ of size at most k. We show that $B = H$ is a weak q-Horn-backdoor set for F. Let τ be the assignment of the variables in B that sets all variables to 0. Then $F[\tau]$ is satisfiable because τ assigns the same values to the variables in B as τ_{SAT}. To show that $F[\tau]$ is q-Horn we define the certifying function $\beta : \mathrm{lit}(F[\tau]) \rightarrow \{0, \frac{1}{2}, 1\}$ as follows. We set $\beta(e) = 1$ for every $e \in E \setminus H$ and for every $S \in \mathcal{S}$ with $S = \{s_1, \ldots, s_{|S|}\}$, every $1 \leq j \leq k+1$, and some $1 \leq h \leq |S|$ such that $s_h \in S \cap H$, $\beta(z_i^j) = 1$ for every $1 \leq i \leq h$, and $\beta(z_i^j) = 0$ for every $h < i \leq |S|+1$. Finally, we set $\beta(a^j(S)) = 1$, $\beta(b^j(S)) = 0$ for all S and j. We claim that β certifies that F is q-Horn formula. Consider the clauses of the form $\{\bar{z}_1^j(S), z_{|S|+1}^j(S)\}$, $\{\bar{z}_1^j(S), \bar{z}_{|S|+1}^j(S)\}$, $\{z_1^j(S), z_{|S|+1}^j(S)\}$. Since H is a hitting set, we have that there exists a $1 \leq h \leq |S|$ such that $s_h \in S \cap H$ and hence, $\beta(z_{|S|+1}^j(S)) = 0$ and $\beta(z_1^j(S)) = 1$. Therefore, β clearly certifies these clauses. Now, consider clauses of the form $\{z_{|S|+1}^j(S), a^j(S), b^j(S)\}$, and $\{\bar{a}^j(S), \bar{b}^j(S)\}$. Since $\beta(z_{|S|+1}^j(S)) = 0$ and $\beta(a^j(S)) = 1$ and $\beta(b^j(S)) = 0$, β also certifies such clauses. Finally, consider clauses of the form $\{z_i^j(S), \bar{s}_i, \bar{z}_{i+1}^j(S)\}$ where s_i corresponds to an element not in H (recall that we have already assigned 0 to the variables corresponding to elements in H, and hence removed the satisfied clauses). Since s_i corresponds to an element not in the hitting set, any index h such that $s_h \in H \cap S$ is such that $h > i$ or $h < i$. In the former case, we have that $\beta(z_i^j(S)) = \beta(z_{i+1}^j(S)) = 1$ and in the latter case, we have that $\beta(z_i^j(S)) = \beta(z_{i+1}^j(S)) = 0$. Since $\beta(s_i) = 1$, observe that in either case, β certifies these clauses. This completes the proof for the forward direction.

For the reverse direction suppose that $B \subseteq \mathrm{var}(F)$ is a weak q-Horn-backdoor set for F of size at most k and let τ be an assignment for the variables in B witnessing this.

We show that $H = B \cap E$ is a hitting set of $\mathcal{S}$ of size at most k. Suppose for a contradiction that H is not a hitting set for $\mathcal{S}$ and let $S \in \mathcal{S}$ be such that $H \cap S = \emptyset$. It follows that there is a $1 \leq j \leq k+1$ such that $F[\tau]$ contains the clauses $\{z_i^j(S), \bar{s}_i, \bar{z}_{i+1}^j(S)\}$, $\{\bar{z}_1^j(S), z_{|S|+1}^j(S)\}$, $\{\bar{z}_1^j(S), \bar{z}_{|S|+1}^j(S)\}$, $\{z_1^j(S), z_{|S|+1}^j(S)\}$, $\{z_{|S|+1}^j(S), a^j(S), b^j(S)\}$, and $\{\bar{a}^j(S), \bar{b}^j(S)\}$ for every $1 \leq i \leq |S|$. We claim that the subformula of $F[\tau]$ induced on the above clauses, and hence $F[\tau]$, is not a q-Horn formula, contradicting our assumption that B is a weak q-Horn-backdoor set for F. Consider the clauses $\{z_i^j(S), \bar{s}_i, \bar{z}_{i+1}^j(S)\}$, $1 \leq i \leq |S|$ in the formula $F[\tau]$. These clauses imply that $z_1^j(S)$ and $z_{|S|+1}^j(S)$ are both in the same strongly connected component of $\mathrm{D}(F_2)$ and furthermore, the same strongly connected component contains the complements of these two literals. Therefore, if there is a certifying function for this formula, then there is also a certifying function β such that $\beta(z_1^j(S)) = \beta(z_{|S|+1}^j(S)) = \frac{1}{2}$. This implies that $\beta(a^j(S)) + \beta(b^j(S)) \leq \frac{1}{2}$ due to the presence of the clause $\{z_{|S|+1}^j(S), a^j(S), b^j(S)\}$. But this implies that $\beta(\bar{a}^j(S)) + \beta(\bar{b}^j(S)) \geq \frac{3}{2}$, in which case, β cannot certify the clause $\{\bar{a}^j(S), \bar{b}^j(S)\}$. Therefore, we conclude that β is not a certifying function, proving that $F[\tau]$ is not a q-Horn formula. This completes the proof of the theorem.

It remains an open problem whether the Strong q-Horn-backdoor Set Detection or Weak q-Horn-backdoor Set Detection problems are FPT-approximable. However we note that since the reductions used in the above theorems are parameter preserving, an FPT-approximation algorithm for either of these problems would imply the existence of an FPT-approximation algorithm for Hitting Set, which is an open problem [21].

5 Conclusions

In this paper we have developed an FPT-approximation algorithm for the detection of deletion q-Horn-backdoor sets (Theorem 1). This renders SAT, parameterized by the *deletion distance* from the class of q-Horn- formulas (i.e., the size of a smallest deletion q-Horn-backdoor set) fixed-parameter tractable (Corollary 1). Our result simultaneously generalizes the known fixed-parameter tractability results for SAT parameterized by the deletion distance from the class of renamable Horn formulas [24] and from the class of 2-CNF formulas [22]. We would like to point out that our FPT-approximation algorithm is quite efficient, and its asymptotic running time does not include large hidden factors.

The deletion distance from q-Horn is incomparable with parameters for SAT based on width measures such as the branch-width of the formula's hypergraph [1], treewidth of the formula's primal, dual, or incidence graph [12,25], or the rank-width of the formula's signed incidence graph [15]. This can be easily verified, since one can define q-Horn formulas where all of these width parameters are arbitrarily large. Conversely, by adding to a formula variable-disjoint copies of itself, we can make the deletion distance from q-Horn arbitrarily large; the width however does not increase.

Finally, our hardness results contribute additional attention and significance to the problem of whether the parameterized Hitting Set problem has an FPT-approximation algorithm [21]. Very recently, a subset of the authors of this paper have shown that the DELETION q-Horn-BACKDOOR SET DETECTION problem is FPT [23].

Acknowledgments The authors acknowledge support from the OeAD (Austrian Indian collaboration grant, IN13/2011). Serge Gaspers, Sebastian Ordyniak, and Stefan Szeider acknowledge support from the European Research Council (COMPLEX REASON, 239962) and Serge Gaspers acknowledges support from the Australian Research Council (DE120101761).

References

1. Alekhnovich, M., Razborov, A.A.: Satisfiability, branch-width and Tseitin tautologies. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02), pp. 593–603 (2002)
2. Aspvall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified Boolean formulas. Inf. Process. Lett. **8**(3), 121–123 (1979)
3. Biere, A.: Bounded model checking. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, volume 185 of Frontiers in Artificial Intelligence and Applications, pp. 457–481. IOS Press (2009)
4. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability, volume 185 of Frontiers in Artificial Intelligence and Applications. IOS Press (2009)
5. Bjesse, P., Leonard, T., Mokkedem, A.: Finding bugs in an alpha microprocessor using satisfiability solvers. In: Berry, G., Comon, H., Finkel, A. (eds.) Computer Aided Verification: 13th International Conference, CAV 2001, Paris, France, July 18–22, 2001, Proceedings, pp. 454–464 (2001)
6. Boros, E., Crama, Y., Hammer, P.L.: Polynomial-time inference of all valid implications for horn and related formulae. Ann. Math. Artif. Intell. **1**, 21–32 (1990)
7. Boros, E., Hammer, P.L., Sun, X.: Recognition of q-Horn formulae in linear time. Discret. Appl. Math. **55**(1), 1–13 (1994)
8. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the 3rd Annual Symposium on Theory of Computing, pp. 151–158. Shaker Heights, Ohio (1971)
9. Crama, Y., Ekin, O., Hammer, P.L.: Variable and term removal from Boolean formulae. Discret. Appl. Math. **75**(3), 217–230 (1997)
10. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
11. Downey, R.G., Fellows, M.R., McCartin, C.: Parameterized approximation problems. In: Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, volume 4169 of Lecture Notes in Computer Science, pp. 121–129. Springer, Berlin (2006)
12. Fischer, E., Makowsky, J.A., Ravve, E.R.: Counting truth assignments of formulas of bounded treewidth or clique-width. Discret. Appl. Math. **156**(4), 511–529 (2008)
13. Flum, J., Grohe, M.: Parameterized Complexity Theory, volume XIV of Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
14. Ford Jr, L.R., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399–404 (1956)
15. Ganian, R., Hlinený, P., Obdrzálek, J.: Better algorithms for satisfiability problems for formulas of bounded rank-width. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15–18, 2010, Chennai, India, volume 8 of LIPIcs, pp. 73–83 (2010)
16. Gaspers, S., Szeider, S.: Backdoors to satisfaction. In: Bodlaender, H.L., Downey, R., Fomin, F.V., 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, pp. 287–317. Springer, Berlin (2012)
17. Gomes, C.P., Kautz, H., Sabharwal, A., Selman, B.: Satisfiability solvers. In: Handbook of Knowledge Representation, volume 3 of Foundations of Artificial Intelligence, pp. 89–134. Elsevier, Amsterdam (2008)
18. Kautz, H.A., Selman, B.: Planning as satisfiability. In: Proceedings ECAI, pp. 359–363 (1992)
19. Lewis, H.R.: Renaming a set of clauses as a Horn set. J. ACM **25**(1), 134–135 (1978)

20. Prasad, A.G.M., Biere, A.: A survey of recent advances in SAT-based formal verification. Softw. Tools Technol. Transf. **7**(2), 156–173 (2005)
21. Marx, D.: Can you beat treewidth? Theory Comput. **6**, 85–112 (2010)
22. Nishimura, N., Ragde, P., Szeider, S.: Detecting backdoor sets with respect to Horn and binary clauses. In: Proceedings of SAT 2004 (Seventh International Conference on Theory and Applications of Satisfiability Testing, 10–13 May, 2004, Vancouver, BC, Canada), pp. 96–103 (2004)
23. Ramanujan, M.S., Saurabh, S.: Linear time parameterized algorithms via skew-symmetric multicuts. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014, pp. 1739–1748 (2014)
24. Razgon, I., O'Sullivan, B.: Almost 2-SAT is fixed parameter tractable. J. Comput. Syst. Sci. **75**(8), 435–450 (2009)
25. Samer, M., Szeider, S.: Algorithms for propositional model counting. J. Discret. Algorithms **8**(1), 50–64 (2010)
26. Schaefer, T.J.: The complexity of satisfiability problems. In: Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978), pp. 216–226. ACM (1978)
27. Velev, M.N., Bryant, R.E.: Effective use of Boolean satisfiability procedures in the formal verification of superscalar and VLIW microprocessors. J. Symb. Comput. **35**(2), 73–106 (2003)
28. Williams, R., Gomes, C., Selman, B.: Backdoors to typical case complexity. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI, 2003, pp. 1173–1178 (2003)