# Detecting Backdoor Sets with Respect to Horn and Binary Clauses

Naomi Nishimura[1,⋆], Prabhakar Ragde[1,⋆⋆], and Stefan Szeider[2,⋆⋆⋆]

[1] School of Computer Science, University of Waterloo,
Waterloo, Ontario, N2L 3G1, Canada
nishi,plragde@uwaterloo.ca
[2] Department of Computer Science, University of Toronto,
Toronto, Ontario, M5S 3G4, Canada
szeider@cs.toronto.edu

**Abstract.** We study the parameterized complexity of detecting backdoor sets for instances of the propositional satisfiability problem (SAT) with respect to the polynomially solvable classes HORN and 2-CNF. A backdoor set is a subset of variables; for a strong backdoor set, the simplified formulas resulting from any setting of these variables is in a polynomially solvable class, and for a weak backdoor set, there exists one setting which puts the satisfiable simplified formula in the class. We show that with respect to both HORN and 2-CNF classes, the detection of a strong backdoor set is fixed-parameter tractable (the existence of a set of size $k$ for a formula of length $N$ can be decided in time $f(k)N^{\mathcal{O}(1)}$), but that the detection of a weak backdoor set is W[2]-hard, implying that this problem is not fixed-parameter tractable.

## 1 Introduction

The propositional satisfiability problem (SAT) asks whether a given propositional formula in conjunctive normal form (CNF) has a satisfying assignment. Even though SAT is NP-complete in general [4], applications often impose on formulas a hidden structure that can be used for an efficient solution.

One example of such hidden structure is a *backdoor set of variables*, a concept recently introduced by Williams, Gomes, and Selman [10, 11]. A *weak backdoor set* of a formula $F$ is a subset $B$ of the variables of $F$ such that if one assigns to the variables in $B$ certain truth values, then the simplified instance is satisfiable and belongs to a class $\mathcal{C}$ of instances that can be solved in polynomial time. The class $\mathcal{C}$ does not necessarily have a simple syntactic characterization and may be implicitly described by an (incomplete) polynomial-time algorithm.

In order to make the concept of backdoor sets applicable to unsatisfiable instances, we consider *strong backdoor sets*: a set $B$ of variables is a strong backdoor set of $F$ if for each possible truth assignment to the variables in $B$, the respective simplified formula belongs to the class $\mathcal{C}$.

In this paper we address the computational complexity of deciding whether a given formula has a weak/strong backdoor set of size at most $k$ for some integer $k$. We study this problem with respect to the two most fundamental classes of polynomial-time decidable formulas: the class of *Horn formulas* (each clause contains at most one positive literal) and the class of *2-CNF formulas* (each clause contains at most two literals). Satisfiability of Horn formulas can be decided in linear time by Dowling and Gallier's algorithm [5]; satisfiability of 2-CNF formulas can be decided in linear time by Aspvall, Plass, and Tarjan's algorithm [2].

As the backdoor-set approach for SAT only makes sense for instances that allow small backdoor sets, it is reasonable to consider $k$ as a fixed small integer, say $k \leq 20$, whereas the size $N$ of the instance can be arbitrarily large. By exhaustive search, we can find a weak/strong backdoor set (if one exists) by considering all subsets $B$ of variables of the given instance with $|B| \leq k$, and checking whether one (all) of the $2^{|B|}$ assignments to the variables in $B$ yields a formula that belongs to the class $\mathcal{C}$ under consideration.

⋆ Supported by the Natural Science and Engineering Research Council of Canada.
⋆⋆ Supported by the Natural Science and Engineering Research Council of Canada.
⋆⋆⋆ Supported by the Austrian Science Funds (FWF) Project J2295.

However, such a trivial approach becomes impractical for large $N$ even if the parameter $k$, the maximum size of a backdoor set, is chosen to be small. In this paper, we tackle the question of whether a backdoor set can be found in a more efficient manner.

Parameterized complexity [6] provides an excellent framework for studying this question. A parameterized problem is a set $L \subseteq \Sigma^* \times \Sigma^*$ for some fixed alphabet $\Sigma$. For a problem instance $(x, k) \in L$, we refer to $x$ as the main part, and to $k$ as the parameter. Typically (and for all problems considered in the sequel), the parameter is a non-negative integer (presented in unary). XP denotes the class of parameterized problems that can be solved in polynomial time whenever the parameter is considered as a fixed constant. The above discussion of exhaustive search shows that the detection of a backdoor set is certainly in XP, but we wish to do better than this.

If a parameterized problem $L$ can be solved in time $\mathcal{O}(f(k)n^c)$ where $f$ is any function of the parameter and $c$ is a constant (independent of $k$), then $L$ is called *fixed-parameter tractable*; FPT denotes the class of all fixed-parameter tractable problems. The class XP contains a hierarchy of parameterized complexity classes

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\mathrm{P}] \subseteq \mathrm{XP}.$$

All inclusions are assumed to be proper (FPT $\neq$ XP is known). The higher a problem is located in this hierarchy, the more unlikely it is to be fixed-parameter tractable (analogous to an NP-complete problem being unlikely to be in P). If a problem in W[2] turns out to be fixed-parameter tractable, then the satisfiability of CNF formulas with $n$ variables could be solved in time $2^{o(n)}$ [1]. Parameterized versions of Cook's Theorem provide further evidence to assume that FPT $\neq$ W[2] as discussed by Cesati [3] and Downey and Fellows [6].

The classes above FPT are defined in terms of complete problems with respect to parameterized reductions [6]. A parameterized reduction is a straightforward extension of a polynomial-time many-one reduction that ensures a parameter for one problem maps into a parameter for another. More specifically, language $L$ reduces to $L'$ if there are functions $k \mapsto k'$, $k \mapsto k''$, and $\langle x, k\rangle \mapsto x'$ such that $\langle x, k\rangle \mapsto x'$ is computable in time $k''|x|^{O(1)}$ and $\langle x, k\rangle \in L$ if and only if $\langle x', k'\rangle \in L'$.

We show that for the classes HORN and 2-CNF, detection of strong backdoor sets is fixed-parameter tractable. We develop algorithms with time complexity $\mathcal{O}(2^k N)$ and $\mathcal{O}(3^k N)$, respectively.

On the other hand, we show that detecting weak backdoor sets for the classes HORN and 2-CNF is unlikely to be fixed-parameter tractable, by showing that these problems are W[2]-hard. In both cases we demonstrate parameterized reductions from a modified version of the problem HITTING SET, which is known to be W[2]-complete [6].

The remainder of the paper is organized as follows. Related work and definitions are given in Section 2. In Section 3, we present the W[2]-hardness results for weak backdoor sets. Next, we present algorithms for strong backdoor sets in Section 4, followed by NP-hardness results for the non-parameterized versions of both problems in Section 5. Finally, in Section 6 we summarize the work and with end a few concluding remarks.

## 2 Background

### Related Work

Szeider [9] studies the parameterized problem of detecting weak backdoor sets with respect to classes that can be decided by subsolvers of the classic Davis-Logemann-Loveland (DLL) Procedure. That is, classes of formulas that can be decided by Unit Propagation (UP), by Pure-Literal Elimination (PL), and by the combination of Unit Propagation and Pure-Literal Elimination (UP+PL). Backdoor sets with respect to algorithmically defined classes which do not admit a purely syntactic characterization are explicitly included in the definitions in [10, 11].

It turns out that for all three classes, detection of weak backdoor sets is W[P]-complete, and (by small modification of the arguments given in the paper) that the detection of strong backdoor sets is W[P]-hard.

We observe that HORN is a proper subset of the class of formulas decidable by Unit Propagation.

### Notation

We assume an infinite supply of propositional *variables*. A *literal* is a variable $x$ with an assigned parity $\epsilon \in \{0,1\}$, denoted by $x^\epsilon$; if $\epsilon = 1$ then $x^\epsilon$ is a *positive* literal, otherwise it is a *negative* literal. We also write $x = x^1$ and $\overline{x} = x^0$. A set $S$ is *tautological* if it contains both $x$ and $\overline{x}$ for some variable $x$. A *clause* is a finite non-tautological set of literals. A finite set of clauses is a *CNF formula* (or *formula*, for short).

The *width* of a clause is its cardinality. The *length* $N$ of a formula $F$ is the sum of the widths of its clauses, i.e., $N = \sum_{C \in F} |C|$.

A clause is called *Horn* if it contains at most one positive literal, and it is called *binary* if it contains at most two literals. A formula is called Horn (resp., binary) if all its clauses are Horn (binary); the class of Horn (binary) formulas is denoted by HORN (2-CNF).

For a formula $F$ we denote by $var(F)$ the set of variables $x$ such that $x^0$ or $x^1$ is contained in some clause of $F$. A literal $x^\epsilon$ is a *pure literal* of a formula $F$ if $x \in var(F)$ and no clause of $F$ contains $x^{1-\epsilon}$. A clause is *monotone* if it contains only positive literals; a formula is monotone if all its clauses are monotone. For a formula $F$ and a variable $x$ we put

$$F - x := \{\, C \setminus \{x^0, x^1\} : C \in F \,\}.$$

A *truth assignment* is a map $\tau : X \to \{0,1\}$ defined on some set $X$ of variables. For $x \in X$ we define $\tau(x^1) = \tau(x)$ and $\tau(x^0) = 1 - \tau(x)$. For a truth assignment $\tau$ and a formula $F$, $F[\tau]$ denotes the result of removing all clauses from $F$ which contain a literal $x$ with $\tau(x) = 1$ and removing literals $y$ with $\tau(y) = 0$ from the remaining clauses.

A truth assignment $\tau$ *satisfies* a formula $F$ if $F[\tau] = \emptyset$. A formula is *satisfiable* if it is satisfied by some truth assignment; otherwise it is *unsatisfiable*.

### Backdoor Sets

Backdoor sets are defined with respect to some class $\mathcal{C}$ of formulas (we think of $\mathcal{C}$ as a class which can be recognized in polynomial time, and for which satisfiability can be decided in polynomial time as well).

Consider a formula $F$ and a set $B$ of variables of $F$. $B \subseteq var(F)$ is a *weak backdoor set* of $F$ with respect to $\mathcal{C}$ (or *weak $\mathcal{C}$-backdoor set*, for short) if there is a truth assignment $\tau : B \to \{0,1\}$ such that $F[\tau]$ is satisfiable and belongs to $\mathcal{C}$. $B$ is a *strong backdoor set* of $F$ with respect to $\mathcal{C}$ (or *strong $\mathcal{C}$-backdoor set*, for short) if $B \subseteq var(F)$ and for every truth assignment $\tau : B \to \{0,1\}$ we have $F[\tau] \in \mathcal{C}$.

Taking the size of the backdoor set as a parameter, a class $\mathcal{C}$ gives rise to the following two parameterized decision problems.

WEAK $\mathcal{C}$-BACKDOOR.
*Input:* A formula $F$.
*Parameter:* A non-negative integer $k$.
*Question:* Does $F$ have a weak $\mathcal{C}$-backdoor set $B$ of size at most $k$?

STRONG $\mathcal{C}$-BACKDOOR.
*Input:* A formula $F$.
*Parameter:* A nonnegative integer $k$.
*Question:* Does $F$ have a strong $\mathcal{C}$-backdoor set $B$ of size at most $k$?

## 3 Detecting Weak Backdoor Sets is Hard

We will show below that the following problem can be reduced to WEAK $\mathcal{C}$-BACKDOOR.

$q$-HITTING SET
*Instance:* A family $\mathcal{S}$ of finite sets $S_1, \dots, S_m$, each containing at least $q$ elements.
*Parameter:* An integer $k \geq 0$.
*Question:* Is there a subset $R \subseteq \bigcup_{i=1}^m S_i$ of size at most $k$ such that $R \cap S_i \neq \emptyset$ for all $i = 1, \dots, m$? ($R$ is a *hitting set* of $\mathcal{S}$)

**Lemma 1** *$q$-HITTING SET is W[2]-complete for any $q \geq 0$.*

*Proof.* 0-HITTING SET is known to be W[2]-complete [6]. That this is also the case for $q$-HITTING SET, for any $q > 0$, can be seen by the following construction.

Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be an instance of 0-HITTING SET. We put $S_i^* := \{\, (x, j) : x \in S_i, j = 1, \ldots, q \,\}$ and consider the instance $\mathcal{S}^* = S_1^*, \ldots, S_m^*$ of $q$-HITTING SET. If $R$ is a hitting set of $\mathcal{S}$, then $R^* = \{\, (x, 1) : x \in R \,\}$ is evidently a hitting set of $\mathcal{S}^*$, and $|R^*| = |R|$. On the other hand, if $\mathcal{S}^*$ has a hitting set $R^*$, then $R = \{\, x : (x, j) \in R^* \text{ for some } j \,\}$ is a hitting set of $\mathcal{S}$, and $|R| \leq |R^*|$. Thus we have demonstrated a parameterized reduction of 0-HITTING SET to $q$-HITTING SET, and so the latter problem is W[2]-hard. Since $q$-HITTING SET is just a special case of 0-HITTING SET, it is therefore W[2]-complete. □

**Theorem 1** *For any $\mathcal{C} \in \{\text{HORN}, \text{2-CNF}\}$, the problem* WEAK $\mathcal{C}$-BACKDOOR *is W[2]-hard.*

*Proof.* The following proof holds for either choice of $\mathcal{C}$. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be an instance of 3-HITTING SET. We consider $\mathcal{S}$ as a monotone formula, assuming that the sets $S_i$ are composed of variables.

We claim that a set $B \subseteq \mathit{var}(\mathcal{S})$ is a hitting set of $\mathcal{S}$ if and only if $B$ is a weak $\mathcal{C}$-backdoor set of $\mathcal{S}$. Clearly, if $B$ is a hitting set, then the assignment $\tau : B \to \{1\}$ satisfies $\mathcal{S}$, and $\mathcal{S}[\tau] = \emptyset \in \mathcal{C}$ by trivial reasons. Conversely, assume that $B$ is a backdoor set for $\mathcal{S}$ with respect to $\mathcal{C}$; i.e., there is an assignment $\tau : B \to \{0, 1\}$ such that $\mathcal{S}[\tau] \in \mathcal{C}$. Since all clauses of $\mathcal{S}$ are monotone and contain at least three literals each, no clause is binary or Horn. Hence, in order to have $\mathcal{S}[\tau] \in \mathcal{C}$, $\tau$ must affect every clause $S_i$ of $\mathcal{S}$; i.e., $S_i \cap B \neq \emptyset$. Thus $B$ is a hitting set as claimed, and the theorem follows from Lemma 1. □

## 4 Parameterized Algorithms for Strong Backdoor Sets

Our first algorithm searches for a strong HORN-backdoor set $B$ for a formula $F$. If all clauses of $F$ are Horn clauses, then we take $B = \emptyset$ and we are done. Thus we can assume that $F$ contains at least one non-Horn clause. If $k = 0$ then we know that a strong HORN-backdoor set of size $k$ does not exist, and we are also done. Thus we can also assume $k \geq 1$.

Consider a non-Horn clause $C$ of $F$. By definition, $C$ contains at least two positive literals, say $p_1$ and $p_2$. We claim that either $p_1$ or $p_2$ must belong to any strong backdoor set $B$. Assume to the contrary that $p_1, p_2 \notin B$ and consider the assignment $\tau : B \to \{0, 1\}$ defined by

$$\tau(x) = \begin{cases} 1 - \epsilon & \text{if } x^\epsilon \in C; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Observe that $C$ is not satisfied by $\tau$ since $\tau$ is only defined for literals over variables in $B$, and by definition, $\tau(y^\epsilon) = 0$ whenever $y^\epsilon \in C$ and $y \in B$. We conclude that $C' := C \setminus \{\, x^0, x^1 : x \in B \,\}$ belongs to $F[\tau]$.

However, since $p_1, p_2 \in C'$, it follows that $F[\tau] \notin \text{HORN}$, contradicting the assumption that $B$ is a strong HORN-backdoor set. Thus any strong backdoor set contains either $p_1$ or $p_2$. Consequently, we can systematically search for a strong backdoor set by considering the two cases $p_1 \in B$ and $p_2 \in B$ separately. That is, we search for a strong backdoor set $B_i$ of size $k - 1$ for the formula $F_i := F - p_i$, $i = 1, 2$. If we find such a backdoor set $B_i$, then $B := B_i \cup \{p_i\}$ is a strong backdoor set for $F$. If, however, neither $F_1$ nor $F_2$ has a strong backdoor set of size $k - 1$, then $F$ has no strong backdoor set of size $k$. Thus, the problem of finding a strong backdoor set of size $k$ for $F$ reduces to two problems of finding strong backdoor sets of size $k - 1$ for $F_1$ or $F_2$, respectively.

Applying this reasoning recursively yields the algorithm **sb-horn** displayed in Figure 1. The outlined algorithm explores a binary search tree of height at most $k$. Since a non-Horn clause can be located in time linear in the length of the formula, and since the search tree has at most $2^k$ nodes, we have the following result.

**Theorem 2** *In time $\mathcal{O}(2^k N)$, we can either find a strong HORN-backdoor set of size $k$ for a formula of length $N$, or conclude that no such set exists. Therefore,* STRONG HORN-BACKDOOR *is in FPT.*

```
Procedure sb-horn($F, k$)
input: a CNF formula $F$ and a non-negative integer $k$;
output: either a strong HORN-backdoor set $B$ of $F$ of size at most $k$, or
"no" if such $B$ does not exist.

 1. If $F \in$ HORN, then return $\emptyset$.
 2. If $k = 0$, then return "no".
 3. Pick a non-Horn clause $C \in F$ and two positive literals $p_1, p_2 \in C$.
 4. Call sb-horn($F - p_1, k - 1$).
 5. If a set $B_1$ is returned, then return $B_1 \cup \{p_1\}$.
 6. Call sb-horn($F - p_2, k - 1$).
 7. If a set $B_2$ is returned, then return $B_2 \cup \{p_2\}$.
 8. Return "no".
```

**Fig. 1.** Algorithm for detecting strong HORN-backdoor sets

For the detection of strong 2-CNF-backdoor sets of size $k$ we can proceed in a similar fashion. We handle the trivial cases $F \in$ 2-CNF and $k = 0$ as above.

Now assume that $k \geq 1$, that we can pick a clause $C \in F$ that contains more than two literals, and let $q_1, q_2, q_3$ be three literals in $C$. We claim that any strong 2-CNF-backdoor set $B$ of $F$ must contain at least one of the literals $q_i$, $1 \leq i \leq 3$. As above, we suppose the contrary and define a truth assignment $\tau : B \to \{0, 1\}$ as in (1). We conclude that $C' := C \setminus \{\, x^0, x^1 : x \in B \,\}$ belongs to $F[\tau]$, and since $q_i \in C'$, $i \in \{1, 2, 3\}$, $|C'| \geq 3$ follows. Hence $F[\tau] \notin$ 2-CNF, contradicting our assumption that $B$ is strong 2-CNF-backdoor set.

Consequently, it suffices to consider three cases, searching for strong 2-CNF-backdoor sets $B_i$ of size $k - 1$ of $F_i := F - q_i$, $i \in \{1, 2, 3\}$. If such a $B_i$ is found, then $B = B_i \cup \{q_i\}$ is a strong 2-CNF-backdoor set of $F$.

Applying this reasoning recursively yields the algorithm **sb-2cnf** displayed in Figure 2. The algorithm implicitly explores a ternary search tree of height at most $k$.

```
Procedure sb-2cnf($F, k$)
input: a CNF formula $F$ and a non-negative integer $k$;
output: either a strong 2-CNF-backdoor set $B$ of $F$ of size at most $k$, or
"no" if such $B$ does not exist.

 1. If $F \in$ 2-CNF, then return $\emptyset$.
 2. If $k = 0$, then return "no".
 3. Pick a clause $C \in F$ with $|C| \geq 3$ and three literals $q_1, q_2, q_3 \in C$.
 4. Call sb-2cnf($F - p_1, k - 1$).
 5. If a set $B_1$ is returned, then return $B_1 \cup \{p_1\}$.
 6. Call sb-2cnf($F - p_2, k - 1$).
 7. If a set $B_2$ is returned, then return $B_2 \cup \{p_2\}$.
 8. Call sb-2cnf($F - p_3, k - 1$).
 9. If a set $B_3$ is returned, then return $B_3 \cup \{p_3\}$.
10. Return "no".
```

**Fig. 2.** Algorithm for detecting strong HORN-backdoor sets

**Theorem 3** *In time $\mathcal{O}(3^k N)$, we can either find a strong* 2-CNF *backdoor set of size $k$ for a formula of length $N$, or conclude that no such set exists. Therefore,* STRONG 2-CNF BACKDOOR is in FPT.

## 4.1 Deciding Satisfiability

The algorithms outlined above only search for strong backdoor sets but do not decide satisfiability. However, if a strong $\mathcal{C}$-backdoor set $B$ of a formula $F$ is found, then we only need to check satisfiability of $F[\tau]$ for all $2^{|B|} \leq 2^k$ possible assignments of $B$. By definition of a strong backdoor set, it is always the case that $B[\tau] \in \mathcal{C}$. For $\mathcal{C} \in \{\textsc{horn}, 2\text{-}\textsc{cnf}\}$, satisfiability of $F[\tau]$ can be decided in time linear in the length of $F[\tau]$ using classical linear-time algorithms [5, 2].

**Theorem 4** *Satisfiability of formulas with bounded size of strong $\mathcal{C}$-backdoor set is fixed-parameter tractable for $\mathcal{C} \in \{\textsc{horn}, 2\text{-}\textsc{cnf}\}$.*

Szeider [8] surveys other parameterizations of the SAT problem that allow fixed-parameter tractable SAT-decision.

## 5 NP-completeness of the non-parameterized versions of the considered problems

The problems WEAK/STRONG $\mathcal{C}$-BACKDOOR can be considered as traditional "non-parameterized" problems, by taking the parameter as part of the input. In this section we show that the non-parameterized problems are NP-complete, justifying our parameterized approach.

The reductions of the proofs of Lemma 1 and Theorem 1 can be considered as polynomial-time many-one reductions of HITTING SET ([7]) to WEAK HORN-BACKDOOR and WEAK 2-CNF-BACKDOOR, giving the following result.

**Theorem 5** *For any $\mathcal{C} \in \{\textsc{horn}, 2\text{-}\textsc{cnf}\}$, the non-parameterized problem* WEAK $\mathcal{C}$-BACKDOOR is NP-complete.

We will show that the analogous problems for strong backdoor sets are NP-complete. Membership in NP follows from the next lemma.

**Lemma 2** *Let $\mathcal{C} \in \{\textsc{horn}, 2\text{-}\textsc{cnf}\}$. A set $B$ of variables of a formula $F$ is a strong $\mathcal{C}$-backdoor set for $F$ if and only if $F - B \in \mathcal{C}$.*

*Proof.* Assume that $B$ is a strong $\mathcal{C}$-backdoor set for $F$ and choose a clause $C' \in F - B$ arbitrarily. Consequently, $C' = C \backslash \{ x^0, x^1 : x \in B \}$ for a clause $C \in F$. We define an assignment $\tau : B \to \{0, 1\}$ by setting

$$\tau(x) = \begin{cases} 0 & \text{if } x \in C; \\ 1 & \text{otherwise.} \end{cases}$$

We observe that $C' \in F[\tau]$. By assumption, $F[\tau] \in \mathcal{C}$, and since $C' \in F - B$ was chosen arbitrarily, $F - B \in \mathcal{C}$ follows.

Conversely, assume that $F - B \in \mathcal{C}$ and let $\tau : B \to \{0, 1\}$ be any assignment. We are going to show that $F[\tau] \in \mathcal{C}$. Choose $C' \in F[\tau]$ arbitrarily. By definition, there is some clause $C \in F$ such that $C' = C \setminus \{ x^{1-\tau(x)} : x \in B \}$ where $\tau$ does not assign 1 to any literal of $C$. Consequently, $C'$ is nothing but $C \setminus \{ x^0, x^1 : x \in B \}$. Thus $C' \in F - B \in \mathcal{C}$, and since $C'$ was chosen arbitrarily, $F[\tau] \in \mathcal{C}$. We can thus conclude that $B$ is a strong $\mathcal{C}$-backdoor set. □

**Theorem 6** *The non-parameterized problem* STRONG HORN-BACKDOOR is NP-complete.

*Proof.* By means of Lemma 2 we can verify in polynomial time whether a guessed set $B \subseteq \mathit{var}(F)$ is a strong HORN-backdoor set, thus the problem belongs to NP.

To show NP-hardness, we reduce VERTEX COVER [7] to STRONG HORN-BACKDOOR. Let $(G, k)$ be an instance of VERTEX COVER; that is, $G = (V, E)$ is a graph and $k$ is a non-negative integer. The question is whether there is a set $S$ of at most $k$ vertices of $G$ such that every edge of $G$ is incident with some vertex in $S$ (such a set $S$ is a *vertex cover* of $G$). Considering the vertices of $G$ as variables, every edge $uv$ of $G$ gives rise to a binary clause $\{u, v\}$; hence $G$ can be considered as a monotone formula. We claim that any set $S \subseteq V$ is a vertex cover if and only if it is a strong HORN-backdoor set.

Assume that $S$ is a vertex cover and choose an assignment $\tau : S \to \{0, 1\}$ arbitrarily. We suppose to the contrary that $F[\tau]$ is not Horn. Consequently, there is some clause $C = \{u, v\} \in F[\tau]$, i.e., $C \in F$ and so $uv \in E$. Since $S$ is a vertex cover, at least one of $u$ and $v$ belongs to $S$; we assume, without loss of generality, that $u \in B$. For any $\epsilon \in \{0,1\}$, $\tau(u) = \epsilon$ implies that $C \notin F[\tau]$, a contradiction. Hence $F[\tau] \in$ HORN and so $S$ is indeed a strong HORN-backdoor set.

Conversely, assume that $S$ is a strong HORN-backdoor set. Let $\tau : S \to \{0\}$ be the constant-0 assignment. Since $F[\tau]$ is Horn, $C \cap S \neq \emptyset$ for every $C \in F$. This, however, means that every edge of $G$ is incident with some vertex in $S$, i.e., $S$ is a vertex cover. □

**Theorem 7** *The non-parameterized problem* STRONG 2-CNF-BACKDOOR is NP-complete.

*Proof.* We proceed as in the proof of the previous theorem. Membership in NP follows again by Lemma 2. For showing NP-hardness we reduce an instance $(G, k)$, $G = (V, E)$, of VERTEX COVER. From $G$ we obtain a monotone formula $F$, taking for every edge $e = uv \in E$ a clause $\{u, v, w_e\}$ where $w_e$ is a new variable.

As above, it follows that a vertex cover $S \subseteq V$ of $G$ is also a strong 2-CNF-backdoor set of $F$. Conversely, assume that $S \subseteq var(F)$ is a strong 2-CNF-backdoor set of $F$. If $w_e \in S$ for some edge $e = uv \in E$, then we can replace $w_e$ by $u$ and still have a strong 2-CNF-backdoor set of $F$. Thus we can assume that $S \subseteq V$. In this case, however, $S$ is a vertex cover of $G$. □

## 6 Concluding Remarks

We have shown that with respect to Horn and binary clauses, the detection of strong backdoor sets is fixed-parameter tractable, but the detection of weak backdoor sets is W[2]-hard, hence very unlikely to be fixed-parameter tractable (FPT = W[2] implies the existence of a $2^{o(n)}$ SAT algorithm).

What makes the detection of weak backdoor sets hard and the detection of strong backdoor sets easy? The ultimate reason for this discrepancy seems to be that for strong backdoor sets we do not have to decide satisfiability, as we only have to insure that the chosen set of variables gives rise to a formula that meets the syntactic properties of the considered class $\mathcal{C}$. On the other hand, for weak backdoor sets, we not only have to achieve syntactic properties but also satisfiability of the simplified formula, a property that cannot be described by syntactic terms. Similarly, if we consider backdoor sets with respect to the class of formulas which can be decided by unit resolution (thus, an an algorithmically defined class which contains HORN as a proper subset), then backdoor set detection becomes fixed-parameter intractable as well [9].

Our positive results for HORN and 2-CNF give rise to several research questions. The FPT algorithms presented above certainly leave room for improvements; we think that by means of appropriate simplification rules a speed-up can be achieved. Empirical studies of how such algorithms perform in practice would be welcome. Finally, it would be interesting to identify other polynomial classes which allow fixed-parameter tractable backdoor detection, and to extend the approach to constraint satisfaction.

## References

1. K. A. Abrahamson, R. G. Downey, and M. R. Fellows. Fixed-parameter tractability and completeness. IV. On completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995.
2. B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.
3. M. Cesati. The Turing way to parameterized complexity. *J. of Computer and System Sciences*, 67:654–685, 2003.
4. S. A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd Annual Symp. on Theory of Computing*, pages 151–158, Shaker Heights, Ohio, 1971.
5. W. F. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming*, 1(3):267–284, 1984.
6. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer Verlag, 1999.
7. M. R. Garey and D. R. Johnson. *Computers and Intractability.* W. H. Freeman and Company, New York, 1979.

8. S. Szeider. On fixed-parameter tractable parameterizations of SAT. In E. Giunchiglia and A. Tacchella, editors, *Theory and Applications of Satisfiability, 6th International Conference, SAT 2003, Selected and Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 188–202. Springer Verlag, 2004.
9. S. Szeider. The parameterized complexity of SAT backdoors. In M. Atkinson, editor, *Computing: The Australasian Theory Symposium, CATS 2004,* Informal Proceedings, pages 252–261. University of Ontago, 2004.
10. R. Williams, C. Gomes, and B. Selman. Backdoors to typical case complexity. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003*, 2003. To appear.
11. R. Williams, C. Gomes, and B. Selman. On the connections between backdoors, restarts, and heavy-tailedness in combinatorial search. In *Sixth International Conference on Theory and Applications of Satisfiability Testing, SAT 2003,* Informal Proceedings, pages 222–230, 2003.