Strong Backdoors to Nested Satisfiability *

Serge Gaspers Stefan Szeider

Institute of Information Systems
Vienna University of Technology
Vienna, Austria.
gaspers@kr.tuwien.ac.at
stefan@szeider.net

Abstract

Knuth (1990) introduced the class of nested formulas and showed that their satisfiability can be decided in polynomial time. We show that, parameterized by the size of a smallest strong backdoor set to the base class of nested formulas, checking the satisfiability of any CNF formula is fixed-parameter tractable. Thus, for any $k > 0$, the satisfiability problem can be solved in polynomial time for any formula F for which there exists a variable set B of size at most k such that for every truth assignment τ to B, the formula $F[\tau]$ is nested; moreover, the degree of the polynomial is independent of k.

Our algorithm uses the grid-minor theorem of Robertson and Seymour (1986) to either find that the incidence graph of the formula has bounded treewidth—a case that is solved using model checking for monadic second order logic—or to find many vertex-disjoint obstructions in the incidence graph. For the latter case, new combinatorial arguments are used to find a small backdoor set. Combining both cases leads to an approximation algorithm producing a strong backdoor set whose size is upper bounded by a function of the optimum. Going through all assignments to this set of variables and using Knuth's algorithm, the satisfiability of the input formula is decided.

1 Introduction

In a 1990 paper [17] Knuth introduced the class of nested CNF formulas and showed that their satisfiability can be decided in polynomial time. A CNF formula is *nested* if its variables can be linearly ordered such that there is no pair of clauses that *straddle* each other; a clause c straddles a clause c' if there are variables $x, y \in \mathsf{var}(c)$ and $z \in \mathsf{var}(c')$ such that $x < z < y$ in the linear ordering under consideration. Nested denotes the class of nested CNF formulas. For an example see Figure 1. Since nested formulas have incidence graphs of bounded treewidth [2], one can use treewidth-based algorithms [9, 31] to even compute the number of satisfying truth assignments of nested formulas in polynomial time (incidence graphs are defined in Section 2). Hence the problems SAT and #SAT are polynomial for nested formulas.

The aim of this paper is to extend the nice computational properties of nested formulas to formulas that are not nested but are of small distance from being nested. We measure the distance of a CNF formula F from being nested as the size of a smallest set B of variables, such that for all partial truth assignments τ to B, the reduced formula $F[\tau]$ is nested. Such a set B is called a *strong backdoor set* with respect to the class of nested formulas [34], or strong Nested-backdoor

[*] The authors acknowledge support from the European Research Council (COMPLEX REASON, 239962).

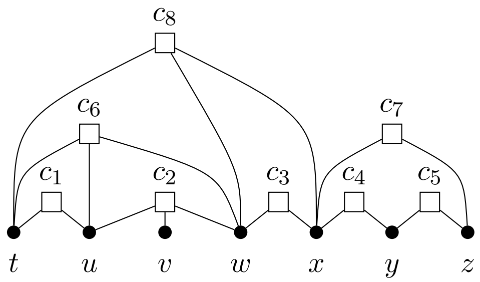

Figure 1: Incidence graph of the nested formula $F = \bigwedge_{i=1}^{8} c_i$ with $c_1 = t \vee \neg u$, $c_2 = u \vee v \vee w$, $c_3 = w \vee x$, $c_4 = x \vee \neg y$, $c_5 = y \vee \neg z$, $c_6 = t \vee u \vee \neg w$, $c_7 = \neg x \vee z$, $c_8 = \neg t \vee w \vee x$.

set, for short. Once we have found such a backdoor set of size k, we can decide the satisfiability of F by checking the satisfiability of 2^k nested formulas, or for model counting, we can take the sum of the number of models of the 2^k nested formulas. Thus the problems SAT and #SAT can be solved in time $O(2^k|F|^c)$ where $|F|$ denotes the length of F and k denotes the size of the given strong Nested-backdoor set; c is a small constant. In other words, the problems SAT and #SAT are *fixed-parameter tractable* for parameter k (for background on fixed-parameter tractability see Section 2). However, in order to use the backdoor set we must find it first. Is the detection of strong Nested-backdoor sets fixed-parameter tractable as well?

Let $\mathbf{sb_N}(F)$ denote the size of a smallest strong Nested-backdoor set of a CNF formula F. To find a strong backdoor set of size $k = \mathbf{sb_N}(F)$ one can try all possible sets of variables of size at most k, and check for each set whether it is a strong backdoor set. However, for a formula with n variables we have to check $\binom{n}{k} = \Omega(n^k)$ such sets. Thus, this brute-force approach scales poorly in k and does not provide fixed-parameter tractability, as the order of the polynomial increases with k.

In this paper we show that one can overcome this limitation with a more sophisticated algorithm. *We show that the problems SAT and #SAT are fixed-parameter tractable when parameterized by* $\mathbf{sb_N}$*, the size of a smallest strong* Nested*-backdoor set, even when the backdoor set is not provided as an input.*

Our algorithm is constructive and uses the Grid Minor Theorem of Robertson and Seymour [29] to either find that the incidence graph of the formula has bounded treewidth—a case that is solved using model checking for monadic second order logic [1]—or to find many vertex-disjoint obstructions in the incidence graph. For the latter case, new combinatorial arguments are used to find a small strong backdoor set. Combining both cases leads to an algorithm producing a strong backdoor set of a given formula F of size at most 2^k for $k = \mathbf{sb_N}(F)$. Solving all the 2^{2^k} resulting nested formulas provides a solution to F.

Our result provides a new parameter $\mathbf{sb_N}$ that makes SAT and #SAT fixed-parameter tractable. The parameter $\mathbf{sb_N}$ is *incomparable* with other known parameters that make SAT and #SAT fixed-parameter tractable. Take for instance the treewidth of the incidence graph of a CNF formula F, denoted $\mathbf{tw}^*(F)$. As mentioned above, SAT and #SAT are fixed-parameter tractable for parameter $\mathbf{tw}^*$ [9, 31], and $\mathbf{tw}^*(F) \leq 3$ holds if $\mathbf{sb_N}(F) = 0$ (i.e., if $F \in$ Nested) [2]. However, by allowing only $\mathbf{sb_N}(F) = 1$ we already get formulas with arbitrarily large $\mathbf{tw}^*(F)$. This can be seen as follows. Take a CNF formula F_n whose incidence graph is an $n \times n$ square grid, with vertices $v_{i,j}$, $1 \leq i, j \leq n$. Assume the clauses correspond to vertices $v_{i,j}$ where $i + j$ is even, and call a clause even or odd according to whether for the corresponding vertex $v_{i,j}$, the sum $i + j$ is a multiple of 4 or not, respectively. It is well known that the $n \times n$ grid, $n \geq 2$, has treewidth n (folklore). Hence we have $\mathbf{tw}^*(F) = n$. Now take a new variable x and add it positively to all odd clauses and negatively to all even clauses. Let F_n^x denote the new formula. Since the incidence graph of F_n is a subgraph of the incidence graph of F_n^x, we have $\mathbf{tw}^*(F_n^x) \geq \mathbf{tw}^*(F_n) = n$. However, setting x to true removes all odd clauses and thus yields a formula whose incidence graph is a disjoint union of paths, which is easily seen to be nested. Similarly, setting x to false yields a nested formula as well. Hence $\{x\}$ forms a strong Nested-backdoor set, and so $\mathbf{sb_N}(F) = 1$. One can also construct formulas where $\mathbf{sb_N}$ is large and $\mathbf{tw}^*$ is small, for example by taking the variable-disjoint union F of formulas $F_i = (x_i \vee y_i \vee z_i) \wedge (\neg x_i \vee y_i \vee z_i)$ with $\mathbf{sb_N}(F_i) = 1$ and $\mathbf{tw}^*(F_i) = 2$, $1 \leq i \leq n$. Then $\mathbf{tw}^*(F) = \mathbf{tw}^*(F_i) = 2$, but $\mathbf{sb_N}(F) = \sum_{i=1}^n \mathbf{sb_N}(F_i) = n$.

One can also define *deletion backdoor sets* of a CNF formula F with respect to a base class of formulas by requiring that deleting all literals $x, \neg x$ with $x \in B$ from F produces a formula that belongs to the base class [26]. For many base classes it holds that every deletion backdoor set is a strong backdoor set, but in most cases, including the base class Nested, the reverse is not true. In fact, it is easy to see that if a CNF formula F has a Nested-deletion backdoor set of size k, then $\mathbf{tw}^*(F) \leq k + 3$. In other words, the parameter "size of a smallest deletion Nested-backdoor set" is dominated by the parameter incidence treewidth and therefore of limited interest. We note in passing, that one can use the algorithm from [21] to show that the detection of deletion Nested-backdoor sets is fixed-parameter tractable.

Related Work. Williams *et al.* [34] introduced the notion of backdoor sets to explain favorable running times and the heavy-tailed behavior of SAT and CSP solvers on practical instances. The parameterized complexity of finding small backdoor sets was initiated by Nishimura *et al.* [25] who showed that with respect to the classes of Horn formulas and of 2CNF formulas, the detection of strong backdoor sets is fixed-parameter tractable. Their algorithms exploit the fact that for these two base classes strong and deletion backdoor sets coincide. For other base classes, deleting literals is a less powerful operation than applying partial truth assignments. This is the case for the class Nested but also for the class RHorn of renamable Horn formulas. In fact, finding a deletion RHorn-backdoor set is fixed-parameter tractable [27], but it is open whether this is the case for the detection of strong RHorn-backdoor sets. For clustering formulas the situation is similar: detection of deletion backdoor sets is fixed-parameter tractable, detection of strong backdoor sets is most probably not [26]. Very recently, the authors of the present paper showed that for the base class of formulas whose incidence graph is acyclic there is a fixed-parameter approximation algorithm for strong backdoor sets. That is, the following problem is fixed-parameter tractable: find a strong backdoor set of size at most k or decide that there is no strong backdoor set of size at most 2^k [12]. The present paper extends the ideas from [12] to the significantly more involved case with Nested as the base class.

We conclude this section by referring to a recent survey on the parameterized complexity of backdoor sets [13].

2 Preliminaries

Parameterized Complexity. Parameterized Complexity [7, 10, 24] is a two-dimensional framework to classify the complexity of problems based on their input size n and some additional parameter k. It distinguishes between running times of the form $f(k)n^{g(k)}$ where the degree of the polynomial depends on k and running times of the form $f(k)n^{O(1)}$ where the exponential part of the running time is independent of n.

A parameterized problem is *fixed-parameter tractable* (FPT) if there exists an algorithm that solves an input of size n and parameter k in time bounded by $f(k)n^{O(1)}$. In this case we say that the *parameter dependence* of the algorithm is f and we call it an FPT *algorithm*.

Parameterized Complexity has a hardness theory, similar to the theory of NP-completeness to show that certain problems have no FPT algorithm under complexity-theoretic assumptions.

Graphs. Let $G = (V, E)$ be a simple, finite graph. Let $S \subseteq V$ be a subset of its vertices and $v \in V$ be a vertex. We denote by $G - S$ the graph obtained from G by removing all vertices in S and all edges incident to vertices in S. We denote by $G[S]$ the graph $G - (V \setminus S)$. The *(open) neighborhood* of v is $N(v) = \{u \in V : uv \in E\}$, the *(open) neighborhood* of S is $N(S) = \bigcup_{u \in S} N(u) \setminus S$, and their *closed neighborhoods* are $N[v] = N(v) \cup \{v\}$ and $N[S] = N(S) \cup S$, respectively. A v_1–v_k *path* P of length k in G is a sequence of k pairwise distinct vertices $(v_1, v_2, \cdots, v_k)$ such that $v_i v_{i+1} \in E$ for each $i \in \{1, \ldots, k-1\}$. The vertices v_1 and v_k are the *endpoints* of P and all other vertices from P are *internal*. An edge is *internal* to P if it is incident to two internal vertices from P. Two or more paths are independent if none of them contains an inner vertex of another.

A *tree decomposition* of G is a pair $(\{X_i : i \in I\}, T)$ where $X_i \subseteq V$, $i \in I$, and T is a tree with elements of I as nodes such that:

1. $\bigcup_{i \in I} X_i = V$;
2. $\forall uv \in E$, $\exists i \in I$ such that $\{u, v\} \subseteq X_i$;
3. $\forall i, j, k \in I$, if j is on the path from i to k in T then $X_i \cap X_k \subseteq X_j$.

The *width* of a tree decomposition is $\max_{i \in I} |X_i| - 1$. The *treewidth* [28] of G is the minimum width taken over all tree decompositions of G and it is denoted by $\mathbf{tw}(G)$. A graph is *planar* if it can be drawn in the plane with no crossing edges. For other standard graph-theoretic notions not defined here, we refer to [6].

CNF Formulas and Satisfiability. We consider propositional formulas in conjunctive normal form (CNF) where no clause contains a complementary pair of literals. For a clause c, we write $\mathsf{lit}(c)$ and $\mathsf{var}(c)$ for the sets of literals and variables occurring in c, respectively. For a CNF formula F we write $\mathsf{cla}(F)$ for its set of clauses, $\mathsf{lit}(F) = \bigcup_{c \in \mathsf{cla}(F)} \mathsf{lit}(c)$ for its set of literals, and $\mathsf{var}(F) = \bigcup_{c \in \mathsf{cla}(F)} \mathsf{var}(c)$ for its set of variables.

For a set $X \subseteq \mathsf{var}(F)$ we denote by 2^X the set of all mappings $\tau : X \to \{0, 1\}$, the *truth assignments* on X. A truth assignment on X can be extended to the literals over X by setting $\tau(\neg x) = 1 - \tau(x)$ for all $x \in X$. Given a CNF formula F and a truth assignment $\tau \in 2^X$ we define $F[\tau]$ to be the formula obtained from F by removing all clauses c such that τ sets a literal of c to 1, and removing the literals set to 0 from all remaining clauses.

A CNF formula F is *satisfiable* if there is some $\tau \in 2^{\mathsf{var}(F)}$ with $F[\tau] = \emptyset$. SAT is the NP-complete problem of deciding whether a given CNF formula is satisfiable [4, 19]. #SAT is the #P-complete problem of determining the number of distinct $\tau \in 2^{\mathsf{var}(F)}$ with $F[\tau] = \emptyset$ [32].

Nested Formulas. Consider a linear order $<$ of the variables of a CNF formula F. A clause c *straddles* a clause c' if there are variables $x, y \in \mathsf{var}(c)$ and $z \in \mathsf{var}(c')$ such that $x < z < y$. Two clauses *overlap* if they straddle each other. A CNF formula F is *nested* if there exists a linear ordering $<$ of $\mathsf{var}(F)$ in which no two clauses of F overlap [17]. The satisfiability of a nested CNF formula can be determined in polynomial time [17].

The *incidence graph* of a CNF formula F is the bipartite graph $\mathsf{inc}(F) = (V, E)$ with $V = \mathsf{var}(F) \cup \mathsf{cla}(F)$ and for a variable $x \in \mathsf{var}(F)$ and a clause $c \in \mathsf{cla}(F)$ we have $xc \in E$ if $x \in \mathsf{var}(c)$. The *sign* of the edge xc is *positive* if $x \in \mathsf{lit}(c)$ and *negative* if $\neg x \in \mathsf{lit}(c)$.

The graph $\mathsf{inc{+}u}(F)$ is $\mathsf{inc}(\mathsf{univ}(F))$, where $\mathsf{univ}(F)$ is obtained from F by adding a *universal* clause c^* containing all variables of F. By a result of Kratochvíl and Křivánek [18], F is nested if and only if $\mathsf{inc}{+}\mathsf{u}(F)$ is planar. Since $\mathsf{inc}(F)$ has treewidth at most 3 if F is nested [2], the number of satisfying assignments of F can also be counted in polynomial time [9, 31].

Backdoors. Backdoor sets are defined with respect to a fixed class $\mathcal{C}$ of CNF formulas, the *base class*. Let B be a set of propositional variables and F be a CNF formula. B is a *strong $\mathcal{C}$-backdoor set* of F if $F[\tau] \in \mathcal{C}$ for each $\tau \in 2^B$. B is a *deletion $\mathcal{C}$-backdoor set* of F if $F - B \in \mathcal{C}$, where $F - B = \{C \setminus \{x, \neg x : x \in B\} : C \in F\}$.

If we are given 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 FPT in k. If $\mathcal{C}$ is clause-induced (i.e., $F \in \mathcal{C}$ implies $F' \in \mathcal{C}$ for every $F' \subseteq F$), any deletion $\mathcal{C}$-backdoor set of F is a strong $\mathcal{C}$-backdoor set of F. The interest in deletion backdoor sets is motivated for base classes where they are easier to detect than strong backdoor sets. The challenging problem is to find a strong or deletion $\mathcal{C}$-backdoor set of size at most k if it exists. Denote by $\mathbf{sb_N}(F)$ the size of a smallest strong NESTED-backdoor set.

Minors and Grids. The *r-grid* is the graph $L_r = (V, E)$ with vertex set $V = \{(i, j) : 1 \leq i \leq r, 1 \leq j \leq r\}$ in which two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. We say that a vertex $(i, j) \in V$ has horizontal index i and vertical index j.

A graph H is a *minor* of a graph G if H can be obtained from a subgraph of G by contracting edges. The *contraction* of an edge uv makes u adjacent to all vertices in $N(v) \setminus \{u\}$ and removes v.

If H is a minor of G, then one can find a model of H in G. A *model* of H in G is a set of vertex-disjoint connected subgraphs of G, one subgraph C_u for each vertex u of H, such that if uv is an edge of H, then there is an edge of G with one endpoint in C_u and the other in C_v.

By Wagner's theorem [33], a graph is planar if and only if it has no $K_{3,3}$ and no K_5 as a minor. Here, K_5 denotes the complete graph on 5 vertices and $K_{3,3}$ the complete bipartite graph with 3 vertices in both independent sets of the bipartition.

We will use Robertson and Seymour's grid-minor theorem.

Theorem 1 ([29])**.** *For every positive integer r, there exists a constant $f(r)$ such that if a graph G has treewidth at least $f(r)$, then G contains an r-grid as a minor.*

By [30], $f(r) \leq 20^{2r^5}$. A linear FPT algorithm (parameterized by k) by Bodlaender [3] finds a tree decomposition of width at most k of a graph G if $\mathbf{tw}(G) \leq k$. A quadratic FPT algorithm (parameterized by r) by Kawarabayashi *et al.* [14] finds an r-grid minor in a graph G if G contains an r-grid as a minor.

3 Detection of Strong Nested-Backdoor Sets

Our overall approach to find strong NESTED-backdoor sets resembles the approach from [12] to find strong FOREST-backdoor sets. Looking more closely at both algorithms, the reader will see significant differences in how the two main cases are handled.

Let F be a CNF formula and k be an integer. Our FPT algorithm will decide the satisfiability of F if F has a strong NESTED-backdoor set of size at most k.

The first step of the algorithm is to find a good approximation for a smallest strong NESTED-backdoor set. Specifically, it will either determine that F has no strong NESTED-backdoor set of size at most k, or it will compute a strong NESTED-backdoor set of size at most 2^k. In case F has no strong NESTED-backdoor set of size at most k, the algorithm stops, and if it finds a strong NESTED-backdoor set B of size at most 2^k, it uses Knuth's algorithm [17] to check for every assignment $\tau \in 2^B$ whether $F[\tau]$ is satisfiable and answers YES if at least one such assignment reduced F to a satisfiable formula and NO otherwise. Since $\mathbf{tw}^*(F[\tau]) \leq 3$ [2] for every truth assignment τ to B, a

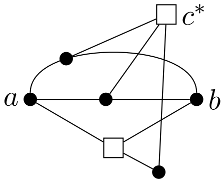

Figure 2: A Nested-obstruction leading to a $K_{3,3}$-minor with the universal clause c^*.

tree decomposition of $\mathsf{inc}(F[\tau])$ can be computed in linear time [3], and treewidth-based dynamic programming algorithms can be used to compute the number of satisfying assignments of $F[\tau]$ in polynomial time [9, 31].

We will arrive at our main theorem.

Theorem 2. *The problems SAT and #SAT are fixed-parameter tractable parameterized by* $\mathbf{sb_N}(F)$.

It only remains to find a strong Nested-backdoor set with an FPT algorithm. In the remainder of this section we present an FPT algorithm that either determines that F has no strong Nested-backdoor set of size at most k, or computes one of size at most 2^k. An algorithm of that kind is called an FPT-*approximation algorithm* [20], as it is an FPT algorithm that computes a solution that approximates the optimum with an error bounded by a function of the parameter.

Consider the incidence graph $G = (V, E) = \mathsf{inc}(F)$ of F. By [30], it either has treewidth at most $\mathsf{tw}(k)$, or it has a $\mathsf{grid}(k)$-grid as a minor. Here,

$$\begin{aligned}\mathsf{tw}(k) &:= 20^{2\mathsf{grid}(k)^5},\\ \mathsf{grid}(k) &:= 4 \cdot \sqrt{\mathsf{obs}(k)+1},\\ \mathsf{obs}(k) &:= 2^k \cdot \mathsf{same}(k) + k, \text{ and}\\ \mathsf{same}(k) &:= 15 \cdot 2^{2k+2}.\end{aligned}$$

3.1 Large Grid Minor

The goal of this subsection is to design an FPT algorithm that, given a $\mathsf{grid}(k)$-grid as a minor in G, computes a set S^* of $2^{O(k^{10})}$ variables from $\mathsf{var}(F)$ such that every strong Nested-backdoor set of size at most k contains a variable from S^*. Suppose G has a $\mathsf{grid}(k)$-grid as a minor.

Definition 1. *An a–b* Nested-obstruction *is a subgraph of* $\mathsf{inc}(F)$ *consisting of*

- *five distinct vertices a, b, p_1, p_2, p_3, such that p_1, p_2, p_3 are variables,*
- *three independent a–b paths P_1, P_2, P_3, and*
- *an edge between p_i and a vertex from P_i for each $i \in \{1, 2, 3\}$.*

In particular, if a path P_i has a variable v as an interior vertex, we can take $p_i := v$. See Figure 2.

Lemma 1. *If F' is a CNF formula such that* $\mathsf{inc}(F')$ *contains a* Nested*-obstruction, then $F' \notin$* Nested.

Proof. Suppose $\mathsf{inc}(F')$ contains a Nested-obstruction. We will exhibit a $K_{3,3}$-minor in $\mathsf{inc{+}u}(F')$. A model of a $K_{3,3}$ can be obtained by taking the subgraphs consisting of the singleton vertices a, b, and the universal clause c^* for one side of the bipartition, and the three subgraphs induced by

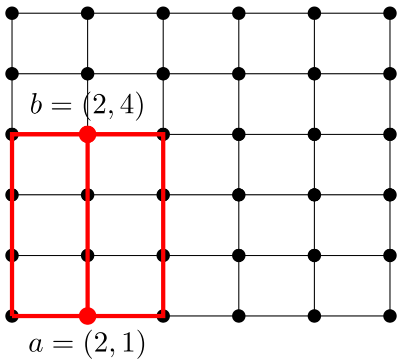

Figure 3: The 6-grid and a highlighted NESTED-obstruction.

$(P_i \cup \{p_i\}) \setminus \{a, b\}, 1 \leq i \leq 3$, for the other side. Since, by Wagner's Theorem [33], no planar graph has a $K_{3,3}$ as a minor, and by a result of Kratochvíl and Křivánek [18], F' is nested if and only if $\mathsf{inc{+}u}(F')$ is planar, we conclude that $F' \notin$ NESTED. □

By Lemma 1, we have that for each assignment to the variables of a strong NESTED-backdoor set, at least one variable from each NESTED-obstruction vanishes in the reduced formula. Using the r-grid, we now find a set $\mathcal{O}$ of $\mathsf{obs}(k)$ vertex-disjoint NESTED-obstructions in G.

Lemma 2. *Given a $\mathsf{grid}(k)$-grid minor of $G = \mathsf{inc}(F)$, a set of $\mathsf{obs}(k)$ vertex-disjoint* NESTED*-obstructions can be found in polynomial time.*

Proof. Let i and j be two integers with $1 \leq i \leq \mathsf{grid}(k)/3$ and $1 \leq j \leq \mathsf{grid}(k)/4$. In the $\mathsf{grid}(k)$-grid, consider the two vertices a and b, with $a = (3i - 1, 4j - 3)$ and $b = (3i - 1, 4j)$, and 3 independent a–b paths containing only vertices with horizontal index between $3i - 2$ and $3i$ and vertical index between $4j - 3$ and $4j$. See Figure 3. Denote this subgraph of the grid by Q.

In the r-grid model, each vertex v from Q corresponds to a connected subgraph C_v. For each edge uv from Q, select one representative edge with one endpoint in C_u and the other endpoint in C_v. For each vertex v from Q, compute a spanning tree of C_v. Consider the set N_v of vertices from C_v that are incident to representative edges. Note that $2 \leq |N_v| \leq 3$. Select one representative vertex r_v from C_v such that the spanning tree contains independent paths from r_v to each vertex in N_v. If $|N_v| = 2$, r_v is any vertex from the unique subpath of the spanning tree connecting the two vertices from N_v. If $|N_v| = 3$, r_v is the unique vertex that is on all subpaths of the spanning tree pairwise connecting vertices from N_v. A NESTED-obstruction in G is now formed by connecting each representative vertex to their representative edges by independent paths chosen as subpaths of the spanning trees. Note that an r_a–r_b NESTED-obstruction in $\mathcal{O}$ has 3 independent r_a–r_b paths with length at least 4, and thus, each of them contains at least one variable. Moreover, the NESTED-obstruction contains only vertices from the connected subgraphs of G corresponding to vertices from the grid with horizontal index between $3i - 2$ and $3i$ and vertical index between $4j - 3$ and $4j$. Thus, any two NESTED-obstructions defined from distinct (i, j) are vertex-disjoint.

The number of NESTED-obstructions defined this way is $\lfloor \frac{\mathsf{grid}(k)}{3} \rfloor \cdot \lfloor \frac{\mathsf{grid}(k)}{4} \rfloor \geq (\frac{\mathsf{grid}(k)}{3} - 1) \cdot (\frac{\mathsf{grid}(k)}{4} - 1)$. Since $\mathsf{grid}(k) \geq 24$ if $k \geq 1$, we have that $\frac{\mathsf{grid}(k)}{3} - 1 \geq \frac{\mathsf{grid}(k)}{4} + 1$. Thus, the number of NESTED-obstructions is at least $(\frac{\mathsf{grid}(k)}{4})^2 - 1 \geq \mathsf{obs}(k)$. □

Denote by $\mathcal{O}$ a set of $\mathsf{obs}(k)$ vertex-disjoint NESTED-obstructions obtained via Lemma 2. A backdoor variable can destroy a NESTED-obstruction either because it participates in the NESTED-obstruction, or because every setting of the variable satisfies a clause that participates in the NESTED-obstruction.

Definition 2. *Let x be a variable and O a* Nested*-obstruction in G. We say that x* kills *O if neither $\mathsf{inc}(F[x=1])$ nor $\mathsf{inc}(F[x=0])$ contains O as a subgraph. We say that x kills O* internally *if $x \in \mathsf{var}(O)$, and that x kills O* externally *if x kills O but does not kill it internally. In the latter case, O contains a clause c containing x and a clause c' containing $\neg x$ and we say that x kills O (externally)* in *c and c'.*

Our algorithm will make a series of $O(1)$ guesses about the strong Nested-backdoor set, where each guess is made out of a number of choices that is upper bounded by a function of k. At any stage of the algorithm, a *valid* strong Nested-backdoor set is one that conforms to the guesses that have been made. For a fixed series of guesses, the algorithm will compute a set $S \subseteq \mathsf{var}(F)$ such that every valid strong Nested-backdoor set of size at most k contains a variable from S. The union of all such S, taken over all possible series of guesses, forms a set S^* and each strong Nested-backdoor set of size at most k contains a variable from S^*. Bounding the size of each S by a function of k enables us to bound $|S^*|$ by a function of k, and S^* can then be used in a bounded search tree algorithm (see Subsection 3.3).

For any strong Nested-backdoor set of size at most k, at most k Nested-obstructions from $\mathcal{O}$ are killed internally since they are vertex-disjoint. The algorithm guesses k Nested-obstructions from $\mathcal{O}$ that may be killed internally. Let $\mathcal{O}'$ denote the set of the remaining Nested-obstructions, which need to be killed externally.

Suppose F has a strong Nested-backdoor set B of size k killing no Nested-obstruction from $\mathcal{O}'$ internally. Then, B defines a partition of $\mathcal{O}'$ into 2^k parts where for each part, the Nested-obstructions contained in this part are killed externally by the same set of variables from B. Since $|\mathcal{O}'| = \mathsf{obs}(k) - k = 2^k \cdot \mathsf{same}(k)$, at least one of these parts contains at least $\mathsf{same}(k)$ Nested-obstructions from $\mathcal{O}'$. The algorithm guesses a subset $\mathcal{O}_s \subseteq \mathcal{O}'$ of $\mathsf{same}(k)$ Nested-obstruction from this part and it guesses how many variables from the strong Nested-backdoor set kill the obstructions in this part externally.

Suppose each Nested-obstruction in $\mathcal{O}_s$ is killed externally by the same set of ℓ backdoor variables, and no other backdoor variable kills any Nested-obstruction from $\mathcal{O}_s$. Clearly, $1 \le \ell \le k$. Compute the set of external killers for each Nested-obstruction in $\mathcal{O}_s$. Denote by Z the common external killers of the Nested-obstruction in $\mathcal{O}_s$. The presumed backdoor set contains exactly ℓ variables from Z and no other variable from the backdoor set kills any Nested-obstruction from $\mathcal{O}_s$.

We will define three rules for the construction of S, and the algorithm will execute the first applicable rule.

Rule 1 (Few Common Killers)**.** *If $|Z| < |\mathcal{O}_s|$, then set $S := Z$.*

The correctness of this rule follows since any valid strong Nested-backdoor set contains ℓ variables from Z and $\ell \ge 1$.

For each $O \in \mathcal{O}_s$ we define an auxiliary graph $G_O = (Z, E_O)$ whose edge set is initially empty. As long as G_O has a vertex v with degree 0 such that v and some other vertex in Z have a common neighbor from O in G, select a vertex u of minimum degree in G_O such that u and v have a common neighbor from O in G and add the edge uv to E_O. As long as G_O has a vertex v with degree 0, select a vertex u of minimum degree in G_O such that v has a neighbor $v' \in V(O)$ in G and u has a neighbor $u' \in V(O)$ in G and there is a v'–u' path in O in which no internal vertex is adjacent to a vertex from $Z \setminus \{v\}$; add the edge uv to E_O.

By the construction of G_O, we have the following property on the degree of all vertices.

Fact 1. *For each $O \in \mathcal{O}_s$, the graph G_O has minimum degree at least 1.*

Recall that no clause contains complimentary literals. Consider two variables $u, v \in Z$ that share an edge in G_O. By the construction of G_O, there is a u–v path P in G whose internal edges are in O, such that for each variable $z \in Z$, all edges incident to z and a clause from P have the same sign. Moreover, since no variable from a valid strong NESTED-backdoor set kills O externally, unless it is in Z, for each potential backdoor variable $x \in \mathsf{var}(F) \setminus Z$, all edges incident to x and a clause from P have the same sign. Thus, we have the following fact.

Fact 2. *If $u, v \in Z$ share an edge in G_O, then for every valid strong* NESTED*-backdoor set that does not contain u and v, there is a truth assignment τ to B such that $\mathsf{inc}(F[\tau])$ contains a u–v path whose internal edges are in O.*

Consider the multigraph $G_m(\mathcal{O}_s) = (Z, \biguplus_{O \in \mathcal{O}_s} E_O)$, i.e., the union of all G_O over all $O \in \mathcal{O}_s$, where the multiplicity of an edge is the number of distinct sets E_O where it appears, $O \in \mathcal{O}_s$.

Rule 2 (Multiple Edges)**.** *If there are two vertices $u, v \in Z$ such that $G_m(\mathcal{O}_s)$ has a u–v edge with multiplicity at least $2 \cdot 2^k + 1$, then set $S := \{u, v\}$.*

Consider any valid strong NESTED-backdoor set B of size k. Then, by Fact 2, for each u–v edge there is some truth assignment τ to B such that $\mathsf{inc}(F[\tau])$ contains a u–v path in G. Moreover, since each u–v edge comes from a different $O \in \mathcal{O}_s$, all these u–v paths are independent. Since there are 2^k truth assignments to B but at least $2 \cdot 2^k + 1$ u–v edges, for at least one truth assignment τ to B, there are 3 independent u–v paths P_1, P_2, P_3 in $\mathsf{inc}(F[\tau])$. We obtain a u–v NESTED-obstruction choosing as $p_i, 1 \le i \le 3$, a variable from P_i or a variable neighboring a clause from P_i and belonging to the same NESTED-obstruction in $\mathcal{O}_s$. Thus, any valid strong NESTED-backdoor set contains u or v.

Now, consider the graph $G(\mathcal{O}_s)$ obtained from the multigraph $G_m(\mathcal{O}_s)$ by merging multiple edges, i.e., we retain each edge only once.

Rule 3 (No Multiple Edges)**.** *Set S to be the $2k$ vertices of highest degree in $G(\mathcal{O}_s)$ (ties are broken arbitrarily).*

For the sake of contradiction, suppose F has a valid strong NESTED-backdoor set B of size k with $B \cap S = \emptyset$. First, we show a lower bound on the number of edges in $G(\mathcal{O}_s) - B$. Since $G_m(\mathcal{O}_s)$ has at least $\frac{|Z|}{2}\mathsf{same}(k)$ edges and each edge has multiplicity at most 2^{k+1}, the graph $G(\mathcal{O}_s)$ has at least $\frac{|Z|\mathsf{same}(k)}{2 \cdot 2^{k+1}} = 3 \cdot 5 \cdot 2^k \cdot |Z|$ edges. Let d be the sum of the degrees in $G(\mathcal{O}_s)$ of the vertices in $B \cap Z$. Now, the sum of degrees of vertices in S is at least $2d$ in $G(\mathcal{O}_s)$, and at least d in $G(\mathcal{O}_s) - B$. Therefore, $G(\mathcal{O}_s) - B$ has at least $d/2$ edges. On the other hand, the number of edges deleted to obtain $G(\mathcal{O}_s) - B$ from $G(\mathcal{O}_s)$ is at most d. It follows that the number of edges in $G(\mathcal{O}_s) - B$ is at least a third the number of edges in $G(\mathcal{O}_s)$, and thus at least $5 \cdot 2^k \cdot |Z|$.

Now, we iteratively build a truth assignment τ for B. Set $H := G(\mathcal{O}_s) - B$. Order the variables of B as $b_1, \ldots, b_k$. For increasing i, we set $\tau(b_i) = 0$ if in G, the vertex $v \in B$ is adjacent with a positive edge to more paths that correspond to an edge in H than with a negative edge and set $\tau(b_i) = 1$ otherwise; if $\tau(b_i) = 0$, then remove each edge from H that corresponds to a path in G that is adjacent with a negative edge to b_i, otherwise remove each edge from H that corresponds to a path in G that is adjacent with a positive edge to b_i.

Observe that for a variable $v \in B$ and a path P in G that corresponds to an edge in $G(\mathcal{O}_s) - B$, v is not adjacent with a positive and a negative edge to P. If $v \in Z$ this follows by the construction of G_O, and if $v \notin Z$, this follows since v does not kill any NESTED-obstruction from $\mathcal{O}_s$. Therefore, each of the k iterations building the truth assignment τ has removed at most half the edges of H. In the end, H has at least $5|Z|$ edges.

Next, we use the following theorem of Kirousis *et al.* [16].

Theorem 3 ([16])**.** *If a graph has n vertices and $m > 0$ edges, then it has an induced subgraph that is $\lceil \frac{m+n}{2n} \rceil$-vertex-connected.*

We conclude that H has an induced subgraph H' that is 3-vertex-connected. Let $x, y \in V(H')$. We use Menger's theorem [22].

Theorem 4 ([22]). *Let $G = (V, E)$ be a graph and $x, y \in V$. Then the size of a minimum x, y-vertex-cut in G is equal to the maximum number of independent x–y paths in G.*

Since the minimum size of an x, y-vertex cut is at least 3 in H', there are 3 independent x–y paths in H'. Replacing each edge by its corresponding path in G, gives rise to 3 walks from x to y in G. Shortcutting cycles, we obtain three x–y paths P_1, P_2, P_3 in G. By construction, each edge of these paths is incident to a vertex from a Nested-obstruction in $\mathcal{O}_s$. We assume that P_1, P_2, P_3 are edge-disjoint. Indeed, by the construction of the G_O, $O \in \mathcal{O}_s$, they can only share the first and last edges. In case P_1 shares the first edge with P_2, replace x by its neighbor on P_1, remove the first edge from P_1 and P_2, and replace P_3 by its symmetric difference with this edge. Act symmetrically for the other combinations of paths sharing the first or last edge.

Lemma 3 ([11]). *Let $G = (V, E)$ be a graph. If there are two vertices $x, y \in V$ with 3 edge-disjoint x–y paths in G, then there are two vertices $x', y' \in V$ with 3 independent x'–y' paths in G.*

Proof. Let P_1, P_2, P_3 denote 3 edge-disjoint x–y paths, and let $S = \{s_1, s_2, s_3\}$, where s_i neighbors x on P_i. Consider the connected component G' of $G - \{x\}$ containing y. Then, G' contains all vertices from S. Let T be a spanning tree of G'. Select y' to be the vertex belonging to every subpath of T that has two vertices from S as endpoints. Set $x' := x$, and obtain 3 independent x'–y' paths in G by moving from x' to s_i, and then along the s_i–y' subpath of T to y', $1 \leq i \leq 3$. □

By Lemma 3 we obtain two vertices x', y' in G with 3 independent x'–y' paths P'_1, P'_2, P'_3 in G. Since the lemma does not presuppose any other edges in G besides those from the edge-disjoint x–y paths, P'_1, P'_2, P'_3 use only edges from the paths P_1, P_2, P_3. Thus, each edge of P'_1, P'_2, P'_3 is incident to a vertex from a Nested-obstruction in $\mathcal{O}_s$. Thus, we obtain a x'–y' Nested-obstruction with the paths P'_1, P'_2, P'_3, and for each path P'_i, we choose a variable from this path or a variable from $\mathcal{O}_s$ neighboring a clause from this path. We arrive at a contradiction for B being a valid strong Nested-backdoor set. This proves the correctness of Rule 3.

The number of possible guesses the algorithm makes is upper bounded by $\binom{\mathsf{obs}(k)}{k} \cdot \binom{\mathsf{obs}(k)-k}{\mathsf{same}(k)} \cdot k = 2^{O(k^8)}$, and each series of guesses leads to a set S of at most $\mathsf{same}(k)$ variables. Thus, the set S^*, the union of all such S, contains at most $2^{O(k^8)} \cdot \mathsf{same}(k) = 2^{O(k^{10})}$ variables. Finally, we have shown the following lemma in this subsection.

Lemma 4. *There is an* FPT *algorithm that, given a CNF formula F, a positive integer parameter k, and a $\mathsf{grid}(k)$-grid as a minor in $\mathsf{inc}(F)$, computes a set $S^* \subseteq \mathsf{var}(F)$ of size $2^{O(k^{10})}$ such that every strong* Nested*-backdoor set of size at most k contains a variable from S^*.*

3.2 Small Treewidth

The goal of this subsection is to design an FPT algorithm that, given a tree decomposition of G of width at most $\mathsf{tw}(k)$, finds a strong Nested-backdoor set of F of size k or determines that F has no such strong Nested-backdoor set.

Our algorithm uses Arnborg *et al.*'s extension [1] of Courcelle's Theorem [5]. It gives, amongst others, an FPT algorithm that takes as input a graph $\mathcal{A}$ with labeled vertices and edges and a Monadic Second Order (MSO) sentence $\varphi(X)$, and computes a minimum-sized set of vertices X such that $\varphi(X)$ is true in $\mathcal{A}$. Here, the parameter is $|\varphi| + \mathbf{tw}(\mathcal{A})$.

We will define a labeled graph whose treewidth is upper bounded by a function of $\mathbf{tw}(\mathsf{inc}(F))$, and an MSO-sentence of constant length such that the graph models the MSO-sentence whenever its argument is a strong Nested-backdoor set of F.

Lemma 5. *There is an* FPT *algorithm that takes as input a CNF formula F, a positive integer parameter k, and a tree decomposition of G of width at most* $\mathsf{tw}(k)$*, and finds a strong* Nested-*backdoor set of F of size k if one exists.*

Proof. First, we define the labeled graph A_F for F. The set of vertices of A_F is $\mathrm{LIT} \cup \mathrm{CLA}$, with $\mathrm{LIT} = \mathsf{lit}(\mathsf{univ}(F))$ and $\mathrm{CLA} = \mathsf{cla}(\mathsf{univ}(F))$. They are labeled by LIT and CLA, respectively. The vertices from $\mathsf{var}(F)$ are additionally labeled by VAR. The subset of edges $\{x\neg x : x \in \mathsf{var}(F)\}$ is labeled NEG and the subset of edges $\{xc : x \in \mathrm{LIT}, c \in \mathrm{CLA},\ x \in \mathsf{lit}(c)\}$ is labeled IN.

Since a tree decomposition for A_F may be obtained from a tree decomposition for $\mathsf{inc}(F)$ by replacing each variable by both its literals and adding the universal clause c^* to each bag of the tree decomposition, we have that $\mathbf{tw}(A_F) \le 2 \cdot \mathbf{tw}(\mathsf{inc}(F)) + 2$.

The goal is to find a subset X of variables such that for each truth assignment τ to X the incidence graph of $\mathsf{univ}(F[\tau])$ is planar. By Wagner's theorem [33], a graph is planar if and only if it has no K_5 and no $K_{3,3}$ as a minor.

We break up our MSO sentence into several simpler sentences and we use the notation of [10].

The following sentence checks whether X is a subset of variables.

$$\mathrm{var}(X) = \forall x(Xx \to \mathrm{VAR}x)$$

An assignment to X is a subset Y of LIT containing no complementary literals such that every selected literal is a variable from X or its negation, and for every variable x from X, x or $\neg x$ is in Y. The following sentence checks whether Y is an assignment to X.

$$\begin{aligned}\mathrm{ass}(X,Y) = \forall y(Yy \to ((Xy \vee (\exists z(Xz \wedge \mathrm{NEG}yz)))\\ \wedge (\forall z(Yz \to \neg\mathrm{NEG}yz))))\\ \wedge \forall x(Xx \to (Yx \vee \exists y(Yy \wedge \mathrm{NEG}xy)))\end{aligned}$$

To test whether a graph has a K_5-minor, we will check whether it contains five disjoint sets of vertices, such that each such set induces a connected subgraph and all 5 sets are pairwise connected by an edge. Deleting all vertices that are in none of the five sets, and contracting each of the five sets into one vertex, one obtains a K_5. The following sentence checks whether A is disjoint from B.

$$\mathrm{disjoint}(A,B) = \neg\exists x(Ax \wedge Bx)$$

To check whether A is connected with respect to the edges labeled IN, we check that there is no set B that is a proper nonempty subset of A such that B is closed under taking neighbors in A.

$$\begin{aligned}\mathrm{connected}(A) = \neg\exists B(\exists x(Ax \wedge \neg Bx) \wedge \exists x(Bx) \wedge \forall x(Bx \to Ax)\\ \forall x,y((Bx \wedge Ay \wedge \mathrm{IN}xy) \to By))\end{aligned}$$

The following sentence checks whether some vertex from A and some vertex from B have a common edge labeled IN.

$$\mathrm{edge}(A,B) = \exists x,y(Ax \wedge Bx \wedge \mathrm{IN}xy)$$

An assignment removes from the incidence graph all variables that are assigned and all clauses that are assigned correctly. Therefore, the minors we seek must not contain any variable that is assigned nor any clause that is assigned correctly. The following sentence checks whether all vertices from a set A survive when assigning Y to X.

$$\mathrm{survives}(A,X,Y) = \neg\exists x(Ax \wedge (Xx \vee \exists y(Yy \wedge \mathrm{IN}yx)))$$

Testing whether a K_5-minor survives in the incidence graph is now done as follows.

$$K_5\text{-minor}(X,Y) = \exists A_1,\ldots,A_5(\bigwedge_{i=1}^{5}\text{survives}(A_i) \wedge \bigwedge_{1\leq i\neq j\leq 5}\text{disjoint}(A_i,A_j)\wedge$$
$$\bigwedge_{i=1}^{5}\text{connected}(A_i) \wedge \bigwedge_{1\leq i\neq j\leq 5}\text{edge}(A_i,A_j))$$

Similarly, testing whether a $K_{3,3}$-minor survives in the incidence graph is done as follows.

$$K_{3,3}\text{-minor}(X,Y) = \exists A_1,A_2,A_3,B_1,B_2,B_3(\bigwedge_{i=1}^{3}(\text{survives}(A_i)\wedge\text{survives}(B_i))\wedge$$
$$\bigwedge_{1\leq i,j\leq 3}\text{disjoint}(A_i,B_j) \wedge \bigwedge_{1\leq i\neq j\leq 3}(\text{disjoint}(A_i,A_j)\wedge\text{disjoint}(B_i,B_j))\wedge$$
$$\bigwedge_{i=1}^{3}(\text{connected}(A_i)\wedge\text{connected}(B_i)) \wedge \bigwedge_{1\leq i,j\leq 3}\text{edge}(A_i,B_j))$$

Our final sentence checks whether X is a strong Nested-backdoor set of F.

$$\text{SNB}(X) = \text{var}(X) \wedge \forall Y(\text{ass}(X,Y) \rightarrow \neg(K_5\text{-minor}(X,Y) \vee K_{3,3}\text{-minor}(X,Y))))$$

Since $|\text{SNB}| = O(1)$ and $\mathbf{tw}(A_F) \leq 2\cdot\mathsf{tw}(k)+2$, by [1] there is an FPT algorithm finding a strong Nested-backdoor set of minimum size, where the parameter is k. □

We note that in the case where $\mathbf{tw}(G)$ is bounded, one could immediately solve the satisfiability problem for F [9, 31]. However, finding a strong Nested-backdoor set enables us to give an FPT approximation algorithm for the backdoor detection problem.

3.3 The FPT algorithm

Our FPT-approximation algorithm combines the results from the previous two subsections. In case G has treewidth at most $\mathsf{tw}(k)$, Lemma 5 is used to find a solution of size k if one exists. Otherwise, Lemma 4 provides a set S^* of $2^{O(k^{10})}$ variables such that any solution of size at most k contains a variable from S^*. For each $x \in S^*$, the algorithm recurses on both formulas $F[x=0]$ and $F[x=1]$. If both recursive calls return strong Nested-backdoor sets $B_{\neg x}$ and B_x, then $\{x\}\cup B_x\cup B_{\neg x}$ is a strong Nested-backdoor set of F, otherwise, no strong Nested-backdoor set of F of size at most k contains x.

Since $B_{\neg x}$ could be disjoint from B_x in the worst case, while $F[x=0]$ and $F[x=1]$ could have strong Nested-backdoor sets $B'_{\neg x}$ and B'_x of size $k-1$ with $B'_{\neg x} = B'_x$, our approach approximates the optimum with a factor of $2^k/k$.

Theorem 5. *There is an* FPT *algorithm, which, for a CNF formula F and a positive integer parameter k, either concludes that F has no strong* Nested*-backdoor set of size at most k or finds a strong* Nested*-backdoor set of F of size at most 2^k.*

Proof. If $k \leq 1$, our algorithm solves the problem exactly in polynomial time. Otherwise, it runs Bodlaender's FPT algorithm [3] with input G and parameter $\mathsf{tw}(k)$ to either find a tree decomposition of G of width at most $\mathsf{tw}(k)$ or to determine that $\mathbf{tw}(G) > \mathsf{tw}(k)$. In case a tree decomposition of width at most $\mathsf{tw}(k)$ is found, the algorithm uses Lemma 5 to compute a strong Nested-backdoor set of F of size k if one exists, and it returns the answer.

In case Bodlaender's algorithm determines that $\mathsf{tw}(G) > \mathsf{tw}(k)$, by [30] we know that G has a $\mathsf{grid}(k)$-grid as a minor. Such a $\mathsf{grid}(k)$-grid is found by running the FPT algorithm of Kawarabayashi *et al.* [14] with input G and parameter $\mathsf{grid}(k)$. The algorithm now executes the procedure from Lemma 4 to find a set S^* of $2^{O(k^{10})}$ variables from $\mathsf{var}(F)$ such that every strong Nested-backdoor set of size at most k contains a variable from S^*. The algorithm considers all possibilities that the backdoor set contains some $x \in S^*$; there are $2^{O(k^{10})}$ choices for x. For each such choice, recurse on $F[x = 1]$ and $F[x = 0]$ with parameter $k - 1$. If, for some $x \in S^*$, both recursive calls return backdoor sets B_x and $B_{\neg x}$, then return $B_x \cup B_{\neg x} \cup \{x\}$, otherwise, return No. As $2^k - 1 = 2 \cdot (2^{k-1} - 1) + 1$, the solution size is upper bounded by $2^k - 1$. On the other hand, if at least one recursive call returns No for every $x \in S^*$, then F has no strong Nested-backdoor set of size at most k. □

In particular, this proves Theorem 2.

4 Conclusion

We have classified the problems SAT and #SAT as fixed-parameter tractable when parameterized by the size of a smallest strong backdoor set with respect to the base class of nested formulas. As argued in the introduction, this parameter is incomparable with incidence treewidth.

The parameter dependence makes our algorithm impractical. However, we would like to note that the class of fixed-parameter tractable problems has proven to be quite robust: Once a problem is shown to belong to this class, one can start to develop faster and more practical algorithms. For many cases in the past this was successful. For instance, the problem of recognizing graphs of genus k was originally shown to be fixed-parameter tractable by means of non-constructive tools from graph minor theory [8]. Later a linear-time algorithm with doubly exponential parameter dependence was found [23], and more recently, the algorithm could be improved to a single exponential parameter dependence [15].

It would be interesting to see whether a similar improvement is possible for finding or FPT-approximating strong backdoor sets with respect to nested formulas.

References

[1] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[2] Therese Biedl and Philip Henderson. Nested SAT graphs have treewidth three. Technical Report CS-2004-70, University of Waterloo, 2004.

[3] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[4] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158, 1971.

[5] Bruno Courcelle. Graph rewriting: an algebraic and logic approach. In *Handbook of theoretical computer science, Vol. B*, pages 193–242. Elsevier Science Publishers, North-Holland, Amsterdam, 1990.

[6] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2nd edition, 2000.

[7] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer Verlag, New York, 1999.

[8] Michael R. Fellows and Michael A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35(3):727–739, 1988.

[9] E. Fischer, J. A. Makowsky, and E. R. Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discr. Appl. Math.*, 156(4):511–529, 2008.

[10] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series.* Springer Verlag, Berlin, 2006.

[11] Serge Gaspers. From edge-disjoint paths to independent paths. In preparation.

[12] Serge Gaspers and Stefan Szeider. Backdoors to acyclic SAT. Technical Report 1110.6384, arXiv, 2011.

[13] Serge Gaspers and Stefan Szeider. Backdoors to satisfaction. Technical Report 1110.6387, arXiv, 2011.

[14] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012.

[15] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 771–780, 2008.

[16] Lefteris M. Kirousis, Maria J. Serna, and Paul G. Spirakis. Parallel complexity of the connected subgraph problem. *SIAM J. Comput.*, 22(3):573–586, 1993.

[17] Donald E. Knuth. Nested satisfiability. *Acta Informatica*, 28(1):1–6, 1990.

[18] Jan Kratochvíl and Mirko Křivánek. Satisfiability and co-nested formulas. *Acta Informatica*, 30:397–403, 1993.

[19] Leonid Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[20] Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

[21] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.

[22] Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[23] Bojan Mohar. Embedding graphs in an arbitrary surface in linear time. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC 1996)*, pages 392–397, 1996.

[24] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.

[25] Naomi Nishimura, Prabhakar Ragde, and Stefan Szeider. Detecting backdoor sets with respect to Horn and binary clauses. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, pages 96–103, 2004.

[26] Naomi Nishimura, Prabhakar Ragde, and Stefan Szeider. Solving #SAT using vertex covers. *Acta Informatica*, 44(7-8):509–523, 2007.

[27] Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.

[28] Neil Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[29] Neil Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986.

[30] Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Combin. Theory Ser. B*, 62(2):323–348, 1994.

[31] Marko Samer and Stefan Szeider. Algorithms for propositional model counting. *J. Discrete Algorithms*, 8(1):50–64, 2010.

[32] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[33] Klaus Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[34] Ryan Williams, Carla Gomes, and Bart Selman. Backdoors to typical case complexity. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1173–1178, 2003.