manuscript No.
(will be inserted by the editor)

Model Counting for CNF Formulas of Bounded Modular Treewidth

Daniel Paulusma · Friedrich Slivovsky · Stefan Szeider

Received: date / Accepted: date

Abstract We define the modular treewidth of a graph as its treewidth after contraction of modules. This parameter properly generalizes treewidth and is itself properly generalized by clique-width. We show that the number of satisfying assignments can be computed in polynomial time for CNF formulas whose incidence graphs have bounded modular treewidth. Our result generalizes known results for the treewidth of incidence graphs and is incomparable with known results for clique-width (or rank-width) of signed incidence graphs. The contraction of modules is an effective data reduction procedure. Our algorithm is the first one to harness this technique for #SAT. The order of the polynomial bounding the runtime of our algorithm depends on the modular treewidth of the input formula. We show that it is unlikely that this dependency can be avoided by proving that SAT is W[1]-hard when parameterized by the modular incidence treewidth of the given CNF formula.

Keywords Propositional Satisfiability · Model Counting · Algorithms

1 Introduction

A module in a graph is a set S of vertices such that for any vertex $v \notin S$, every vertex in S is a neighbor of v or every vertex in S is a non-neighbor of v. Contraction of modules, that is, removing from each module all but one vertex is an important preprocessing step for a wide range of combinatorial optimization problems [12]. The aim of this paper is to harness its power for propositional model counting (#SAT) that asks for the number of satisfying truth assignments of a given CNF formula, a well-studied problem with applications in artificial intelligence, such as probabilistic inference [1, 10, 21]. We illustrate the power of the module contraction operation with the following example. Given a set $\{x_1, \ldots, x_n\}$ of variables, consider a CNF formula $F_{m,n}$ that consists of m

Slivovsky and Szeider's research was supported by the ERC, grant reference 239962. Paulusma's research was supported by EPSRC, grant reference EP/G043434/1.

An extended abstract of this paper appeared in the Proceedings of STACS 2013.

D. Paulusma
School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, UK
E-mail: daniel.paulusma@durham.ac.uk

F. Slivovsky
Institute of Information Systems, Vienna University of Technology, A-1040 Vienna, Austria
E-mail: fslivovsky@gmail.com

S. Szeider
Institute of Information Systems, Vienna University of Technology, A-1040 Vienna, Austria
E-mail: stefan@szeider.net

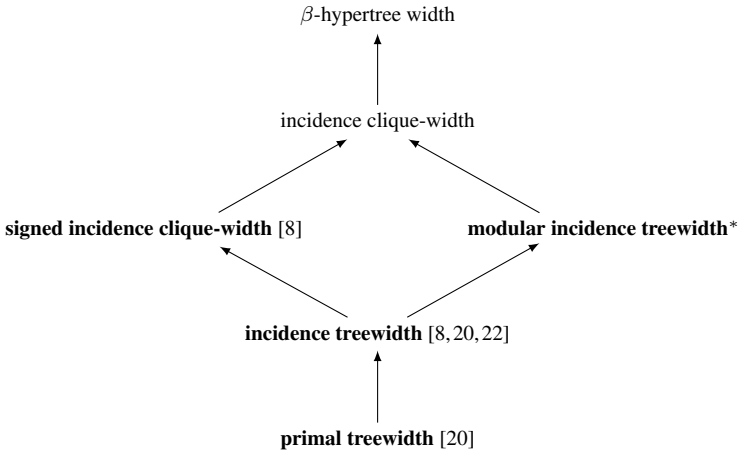

Fig. 1 A hierarchy of structural parameters. An arc from a parameter p to a parameter q reads as "q is bounded whenever p is bounded." Bold type is used to indicate parameters that render #SAT polynomial-time solvable when bounded. The result of this paper is indicated by a $*$-symbol.

distinct clauses each containing n literals over $\{x_1, \ldots, x_n\}$, so that every variable occurs in every clause. It is easy to see that $F_{m,n}$ has exactly $2^n - m$ satisfying truth assignments. The vertices of the incidence graph of $F_{m,n}$ (the bipartite graph whose vertex classes consist of variables and clauses, and a variable is exactly adjacent to those clauses it occurs in) can be partitioned into two modules. By contracting these modules, the incidence graph of $F_{m,n}$ reduces to a single edge. We call the treewidth of the graph obtained from the incidence graph of a CNF formula F after contracting all modules the *modular incidence treewidth* of F. We consider classes of CNF formulas that have bounded modular incidence treewidth and show that #SAT is polynomial-time tractable for such classes. More specifically, we prove the following theorem (where the length of a formula is equal to the sum of the number of literals in each of its clauses).

Theorem 1 #SAT *can be solved in time $O(\ell^{3k+4})$ on CNF formulas that have modular incidence treewidth at most k and length ℓ.*

This result characterizes a new hierarchy of tractable classes for a notoriously hard problem: #SAT is #P-complete in general [23] and remains #P-hard even for monotone 2CNF formulas and Horn 2CNF formulas [19]. It is NP-hard to approximate the number of satisfying truth assignments of a formula with n variables to within $2^{n^{1-\varepsilon}}$ for any $\varepsilon > 0$. Again, this hardness result even holds for monotone 2CNF formulas and Horn 2CNF formulas [19], which can be viewed as classes with some syntactic restriction. By contrast, modular incidence treewidth is a so-called structural parameter. Classes with some structural restriction are obtained by bounding parameters of (hyper)graphs associated with formulas. Figure 1 illustrates the relation of modular incidence treewidth to other structural parameters with respect to #SAT; in Section 1.1 we will discuss Figure 1 in detail.

After stating all necessary terminology and notation in Section 2, we prove Theorem 1 in Section 3 by dynamic programming on a tree decomposition of the modular incidence graph. This is the graph obtained from the incidence graph of a formula by contracting all modules. Vertices in the modular incidence graph represent modules that are either sets of variables or sets of clauses. As the formula $F_{m,n}$ given in our initial example shows, the sizes of these sets cannot be bounded in terms of the modular incidence treewidth alone. As a consequence, algorithms for solving #SAT on formulas of bounded treewidth, which typically rely on data structures indexed by the subsets of variables and clauses associated with a bag of the tree decomposition [20], do not yield tractability. It is a significant challenge to encode the information required to perform dynamic programming in space polynomially bounded by the input size. Our main technical contribution is the use of

projections for solving this task (the projection of a set $\mathcal{C}$ of clauses under some truth assignment τ is the subset of those clauses of $\mathcal{C}$ that are not satisfied by τ). We define an equivalence relation on truth assignments based on their projections for a particular formula. This formula is determined by the boundary of a subgraph induced by the tree decomposition. The resulting equivalence relation is sufficiently precise for the number of equivalence classes to still be polynomially bounded. This bound translates into a polynomial bound for the runtime of a dynamic programming algorithm, allowing us to establish Theorem 1. The order of the corresponding polynomial is a function in the modular incidence treewidth. The following result on the SATISFIABILITY problem (SAT) shows that under the widely believed assumption that FPT$\neq$W[1] this function cannot be replaced by a constant.

Theorem 2 SAT *is* W[1]*-hard, when parameterized by the modular incidence treewidth of the input formula.*

Theorem 2 can be proven by using the same W[1]-hardness reduction as the one in the proof of the result from Ordyniak et al. [15] that states that SAT is W[1]-hard when parameterized by the β-hypertree width.[1] In particular, Theorem 2 implies that Theorem 1 cannot be established by a direct application of the algorithmic meta-theorems of Courcelle, Makowsky, and Rotics [5,6], which can be used in the case of primal treewidth, incidence treewidth or signed incidence clique-width, the other three structural parameters of Figure 1 that render #SAT cubic-time solvable provided they are bounded (or even linear-time solvable in the case of primal treewidth and incidence treewidth as discussed in Section 1.1).

1.1 Structural Parameters Related to Modular Treewidth

In order to discuss the relationships between the structural parameters in Figure 1 in more detail we use the following terminology. For two structural parameters p and q of CNF formulas, we say that p *dominates* q if there is a function f such that $p(F) \leq f(q(F))$ for all formulas F. Parameters p and q are called *equivalent* if p dominates q and q dominates p. We say that p is *more general* than q if p dominates q but not the other way around, whereas p and q are called *incomparable* if neither p dominates q nor q dominates p. Let p be a structural parameter. We say that a class of CNF formulas has *bounded* p, whenever there is a constant c such that $p(F) \leq c$ for all formulas F in the class.

The primal graph of a given CNF formula F has as vertices the variables of F, and two variables are joined by an edge if and only if they occur together in at least one clause. The *primal treewidth* of a formula is the treewidth of its primal graph. It is known that #SAT linear-time tractable for classes of CNF formulas of bounded primal treewidth [20]. The same result has been shown for classes of formulas of bounded *branchwidth* [1], a parameter equivalent to primal treewidth [18] (also see [22]). The *incidence treewidth* of a formula is the treewidth of its incidence graph. This parameter is known to be more general than primal treewidth [11]. It is also known that #SAT is linear-time tractable for classes of CNF formulas of bounded incidence treewidth; see [8,20] for proofs of this result, which do not rely on the algorithmic meta-theorems of Courcelle, Makowsky and Rotics [5,6] that yield a cubic-time algorithm.

The clique-width of a graph is an invariant based on graph grammars [4], whereas the signed clique-width is a variant of clique-width for directed graphs [7]. The signed incidence graph is obtained from the incidence graph by orientating its edges so as to indicate positive or negative occurrences of variables. The *(signed) incidence clique-width* of a formula is the clique-width of its (signed) incidence graph. It is known that #SAT can be solved in cubic time for any class of CNF formulas of bounded signed incidence clique-width [8]. Incidence clique-width is more general than signed incidence clique-width [8] and signed incidence clique-width is more general than

[1] The proof of Theorem 2 can be found in Appendix A.

incidence treewidth [22]. However, the complexity status of #SAT for classes of CNF formulas of bounded incidence clique-width is still open. The β-*hypertree width* of a CNF formula is a parameter that is yet more general than incidence clique-width [11]. Also the complexity status of #SAT for classes of CNF formulas of bounded β-hypertree width remains to be settled.

Clique-width is closely related to a parameter known as rank-width [16]. The *signed incidence rank-width* of a formula is the rank-width of its signed incidence graph. This parameter is equivalent to the signed incidence clique-width [9]. Hence, it follows from the aforementioned result [8] for classes of CNF formulas of bounded signed incidence clique-width that #SAT can be solved in cubic time for classes of CNF formulas of bounded signed incidence rank-width. Ganian, Hlinený and Obdrzálek [9] improved the running time from single-exponential in the signed incidence clique-width to be single-exponential in the signed incidence rank-width.

As follows from our discussion, the equivalent structural parameters signed incidence clique-width and signed incidence rank-width are currently the most general structural parameters based on width measures for which #SAT is known to be polynomial-time tractable. We now explain the remaining three arcs in Figure 1. In particular, we will show that modular incidence treewidth is incomparable with incidence clique-width and more general than incidence treewidth. The following two examples will be useful for showing this; see Section 2 for the definition of treewidth.

Example 1 Let $x_1, \ldots, x_m$ be distinct variables. The formula φ_m is defined as the set of clauses $C_{i,j}$ for $1 \leq i, j \leq m$ and $i \neq j$, where $C_{i,j} = (\{x_1, \ldots, x_m\} \setminus \{x_i, x_j\}) \cup \{\neg x_i, \neg x_j\}$. The (unsigned) incidence graph corresponds to the complete bipartite graph $K_{n,m}$ for $n = \binom{m}{2}$, which has treewidth m. Hence, the incidence treewidth of φ_m is equal to m as well. The signed incidence clique-width of φ_m tends to infinity with m [8]. As in our initial example, contracting all modules reduces $K_{n,m}$ to a single edge. This means that the modular incidence treewidth of φ_m is 1 for arbitrary m.

Example 2 Let $x_1, \ldots, x_m, y_1, \ldots, y_m$ be $2m$ distinct variables. We let ψ_m consist of the clauses C_i for $1 \leq i \leq m$ where $C_i = \{y_i, x_1, \ldots, x_m\}$, along with m singleton clauses $\{x_1\}, \ldots, \{x_m\}$. The incidence graph $I(\psi_m)$ of ψ_m has no modules containing more than one vertex. Hence the modular incidence treewidth and the incidence treewidth of ψ_m coincide. Since $I(\psi_m)$ contains $K_{m,m}$ as a subgraph, its treewidth is at least m. Hence, the modular incidence treewidth of ψ_m is at least m. By contrast, it can be shown[2] that the signed incidence clique-width of ψ_m is at most 4 for arbitrary m. The incidence clique-width of any formula is at most twice its signed incidence clique-width [8], so the incidence clique-width of ψ_m is at most 8 for arbitrary m.

We are now ready to prove the following three propositions, each of which correspond to an arc in Figure 1.

Proposition 1 *Modular incidence treewidth and signed incidence clique-width are incomparable.*

Proof The statement of the proposition follows immediately from Examples 1 and 2. □

Proposition 2 *Modular incidence treewidth is more general than incidence treewidth.*

Proof Modular incidence treewidth dominates incidence treewidth: by contracting modules we obtain an induced subgraph of the incidence graph, and the treewidth of a graph is bounded from below by the treewidth of any of its subgraphs. Then the result follows from recalling that the class of formulas from Example 1 has unbounded treewidth and modular incidence treewidth 1. □

Proposition 3 *Incidence clique-width is more general than modular incidence treewidth.*

Proof It is well known that there is a function that provides an upper bound on the clique-width of any graph in terms of its treewidth, and that clique-width is invariant under contraction of modules [7]. Hence, incidence clique-width dominates modular incidence treewidth. Then the result follows from recalling that the class of formulas from Example 2 has unbounded modular incidence treewidth and bounded incidence clique-width. □

[2] The proof of this statement can be found in Appendix B.

2 Preliminaries

Let X and Y be sets, and let $f : X \to Y$ be a function. We write $f^{-1}(y) = \{x \in X \mid f(x) = y\}$. For a subset $X' \subseteq X$, we let $f|_{X'}$ denote the *restriction* of f to X', that is, $f|_{X'}(x) = f(x)$ for all $x \in X$. Similarly, a function h with domain $X^* \supseteq X$ is an *extension* of f if $h(x) = f(x)$ for all $x \in X$. If $Y = 2^Z$ for some set Z and $f(x) = \{z\}$ for some $z \in Z$, then we may write $f(x) = z$ instead. Let X^* and Y^* be sets, and let $g : X^* \to Y^*$ be a function that *agrees* with f on $X \cap X^*$, that is, $g(x) = f(x)$ for all $x \in X \cap X^*$. Then the function $f \cup g : X \cup X^* \to Y \cup Y^*$ is defined as $(f \cup g)(x) = f(x)$ if $x \in X$ and $(f \cup g)(x) = g(x)$ if $x \in X^* \setminus X$.

We assume an infinite supply of propositional *variables*. A *literal* is a variable x or a negated variable $\overline{x}$; we put $\mathrm{var}(x) = \mathrm{var}(\overline{x}) = x$; if $y = \overline{x}$ is a literal, then we write $\overline{y} = x$. For a set S of literals we write $\overline{S} = \{\, \overline{x} \mid x \in S \,\}$; S is *tautological* if $S \cap \overline{S} \neq \emptyset$. A *clause* is a finite non-tautological set of literals. A finite set of clauses is a *CNF formula* (or *formula*, for short). The *length* of a formula F is denoted by $\ell = \sum_{C \in F} |C|$. The *union* of two clauses C and D, denoted by CD, is the union of the literals of C and D. A variable x *occurs* in a clause C if $x \in C \cup \overline{C}$. We let $\mathrm{var}(C)$ denote the set of variables that occur in C. A variable x *occurs* in a formula F if it occurs in one of its clauses, and we let $\mathrm{var}(F) = \bigcup_{C \in F} \mathrm{var}(C)$.

Let F be a formula. The *incidence graph* of F is the bipartite graph $I(F)$ with vertex set $\mathrm{var}(F) \cup F$ and edge set $\{\, Cx \mid C \in F \text{ and } x \in \mathrm{var}(C) \,\}$. Two vertices are *twins* if they have the same neighbors in $I(F)$. The equivalence classes of the twin relation are called *modules*. By the definition of $I(F)$, twins either consist of two variables or of two clauses. If the vertices of a module correspond to clauses, then we call the module a *clause module*; otherwise we call it a *variable module*. Note that a clause module is a finite set of clauses and hence a formula. By definition, all clauses of any clause module $\mathcal{C}$ of F contain all variables of any variable module X of F if and only if one clause of $\mathcal{C}$ contains at least one variable from X. This implies that the set of variable modules of $\mathcal{C}$ is a subset of the set of variable modules of F. For a set of clause or variable modules $\mathcal{S}$, we let $\langle \mathcal{S} \rangle = \bigcup_{S \in \mathcal{S}} S$ denote the union of the elements of $\mathcal{S}$. The *modular incidence graph* $I^*(F)$ is the bipartite graph obtained from $I(F)$ after removing all but one vertices of each module.

Let F be a formula. A *truth assignment* is a mapping $\tau : X \to \{\, 0, 1 \,\}$ defined on some set of variables $X \subseteq \mathrm{var}(F)$. If $X = \mathrm{var}(F)$, then τ is a *total* truth assignment (of F). For $x \in X$, we define $\tau(\overline{x}) = 1 - \tau(x)$. A truth assignment τ *satisfies* a clause C if C contains some literal x with $\tau(x) = 1$. If τ satisfies all clauses of F, then τ *satisfies* F; in that case we call F satisfiable. The SATISFIABILITY (SAT) problem is that of testing whether a given formula is satisfiable. Propositional model counting (#SAT) is a generalization of SAT that asks for the number of satisfying total truth assignments of a given formula.

Let X be a set of variables. For a clause C, we let $C^X = \{\, x \in C \mid \mathrm{var}(x) \in X \,\}$; note that $C^X = \emptyset$ if and only if C contains no literals x or $\overline{x}$ with $x \in X$. For a formula F, we let $F^X = \{\, C^X \mid C \in F \,\} \setminus \{\emptyset\}$. For a truth assignment τ, we let $\mathbf{clause}(\tau) = \tau^{-1}(0) \cup \overline{\tau^{-1}(1)}$; note that $\mathbf{clause}(\tau) \neq \mathbf{clause}(\tau')$ for any two distinct truth assignments τ and τ'. The following lemma immediately follows from the above definitions.

Lemma 1 *Let $\tau : X \to \{0, 1\}$ be a truth assignment defined on some set of variables X. Let C be a clause. Then C is not satisfied by τ if and only if $C^X = \mathbf{clause}(\tau|_{X \cap \mathrm{var}(C)})$.*

We now adjust the notion of a projection as introduced by Kaski, Koivisto and Nederlof [13] for our purposes. This notion plays an important role in our paper. Let F be a formula and X be a set of variables. We refer to the set of clauses of F *not* satisfied by a truth assignment $\sigma : X \to \{0, 1\}$ as the *(negative) projection* of σ on F denoted $F(\sigma)$. We denote the set of all these projections by $\mathscr{P}_{(F,X)} = \{\, F(\sigma) \mid \sigma : X \to \{0, 1\} \,\}$. If $X \supseteq \mathrm{var}(F)$, then we may write $\mathscr{P}_F$ instead, as $\mathscr{P}_{(F,X)} = \mathscr{P}_{(F,\mathrm{var}(F))}$ holds in that case. Note that F is satisfiable if and only if the *empty* projection $\emptyset$ belongs to $\mathscr{P}_F$, and that the number of satisfying total truth assignments of F is equal

to $|\{\sigma : \mathrm{var}(F) \to \{0,1\} \mid F(\sigma) = \emptyset\}|$. The following lemma, which follows immediately from the definition of a projection, states a useful property of projections.

Lemma 2 *Let F be a formula and let X, Y be two sets of variables. Let $\sigma : X \to \{0,1\}$ and $\tau : Y \to \{0,1\}$ be two truth assignments that agree on $X \cap Y$.Then $F(\sigma \cup \tau) = F(\sigma) \cap F(\tau)$.*

For a clause C and a formula F we let $\mathbf{select}(F, C) = \{C' \in F \mid C \subseteq C'\}$. We will now prove two useful lemmas. The first lemma is for clause modules $\mathcal{C}$. It implies that every truth assignment on $\mathrm{var}(\mathcal{C})$ either satisfies $\mathcal{C}$ or does not satisfy a unique clause of $\mathcal{C}$. The second lemma is similar but with respect to variable modules X.

Lemma 3 *Let $\mathcal{C}$ be a clause module of a formula F, and let τ be a truth assignment defined on a set of variables X. Then $\mathcal{C}(\tau) = \mathbf{select}(\mathcal{C}, \mathbf{clause}(\tau|_{X \cap \mathrm{var}(\mathcal{C})}))$.*

Proof Let $C \in \mathcal{C}$. Because $\mathcal{C}$ is a clause module, $\mathrm{var}(C) = \mathrm{var}(\mathcal{C})$. Then, by using Lemma 1 and the definitions of **clause** and **select**, we find that $C \in \mathcal{C}(\tau)$ if and only if C is not satisfied by τ if and only if $C^X = \mathbf{clause}(\tau|_{X \cap \mathrm{var}(C)}) = \mathbf{clause}(\tau|_{X \cap \mathrm{var}(\mathcal{C})})$ if and only if $C \in \mathbf{select}(\mathcal{C}, \mathbf{clause}(\tau|_{X \cap \mathrm{var}(\mathcal{C})}))$. □

Lemma 4 *Let X be a variable module of a formula F, and let τ be a truth assignment defined on a superset of X. If $F^X(\tau) \neq \emptyset$, then $F^X(\tau) = \mathbf{clause}(\tau|_X)$.*

Proof Let $C \in F^X$. Because X is a variable module, $\mathrm{var}(C) = X$. Lemma 1 tells us that C is not satisfied by τ if and only if $C = \mathbf{clause}(\tau|_{X \cap \mathrm{var}(C)}) = \mathbf{clause}(\tau|_X)$. □

We also need the following lemma.

Lemma 5 *Let X be a variable module of a formula F. Let $E = \{\sigma : X \to \{0,1\} \mid F^X(\sigma) = \Pi\}$ for some $\Pi \in \mathscr{P}_{F^X}$. Then $|E| = 1$ if $\Pi \neq \emptyset$, and $|E| = 2^{|X|} - |F^X|$ if $\Pi = \emptyset$.*

Proof First suppose that $\Pi \neq \emptyset$. Recall that $\mathbf{clause}(\tau) \neq \mathbf{clause}(\tau')$ for any two distinct truth assignments τ and τ'. Then, by Lemma 4, there is only one truth assignment $\tau : X \to \{0,1\}$ with $F^X(\tau) = \Pi$, namely the truth assignment τ with $F^X(\tau) = \mathbf{clause}(\tau) = \Pi$. Hence, $|E| = 1$ in this case.

Now suppose that $\Pi = \emptyset$. The number of truth assignments defined on X is equal to $2^{|X|}$. By Lemma 4, each such truth assignment τ_X does not satisfy one unique clause with set of variables X, namely the clause $\mathbf{clause}(\tau_X)$. Then there are exactly $2^{|X|} - |F^X|$ truth assignments τ_X that do satisfy F^X, i.e., that have $F^X(\tau_X) = \emptyset = \Pi$. Hence, in this case, $|E| = 2^{|X|} - |F^X|$. □

We finish this section with some terminology on tree decompositions. Let $G = (V_G, E_G)$ be a finite, undirected graph with neither self-loops nor multiple edges. A *tree decomposition* of G is a triple (T, χ, r), where $T = (V_T, E_T)$ is a tree rooted at r and $\chi : V_T \to 2^{V_G}$ is a labeling of the vertices of T (called *nodes*) by subsets of V_G (called *bags*) such that the following three conditions hold:

1. $\bigcup_{t \in V_T} \chi(t) = V_G$,
2. for each edge $uv \in E_G$, there is a node $t \in V_T$ with $\{u, v\} \subseteq \chi(t)$,
3. for each vertex $x \in V_G$, the set of nodes t with $x \in \chi(t)$ forms a connected subtree of T.

The *width* of a tree decomposition (T, χ) is the size of a largest bag $\chi(t)$ minus 1. The *treewidth* of G is the minimum width over all possible tree decompositions of G. A tree decomposition (T, χ, r) is *nice* if T is a binary tree such that the nodes of T belong to one of the following four types:

A. a *leaf node* t is a leaf of T,
B. an *introduce node* t has one child t' and $\chi(t) \setminus \{v\} = \chi(t')$ for some vertex $v \in V_G$,
C. a *forget node* t has one child t' and $\chi(t') \setminus \{v\} = \chi(t)$ for some vertex $v \in V_G$,
D. a *join node* t has two children t_1, t_2 and $\chi(t) = \chi(t_1) = \chi(t_2)$.

Kloks [14] showed that every tree decomposition of a graph G can be converted in linear time to a nice tree decomposition, such that the size of the largest bag does not increase, and the corresponding tree has at most $4|V_G|$ nodes.

Let F be a formula. We call the treewidth of the modular incidence graph $I^*(F)$ the *modular incidence treewidth* of F. Let (T, χ, r) be a tree decomposition of $I^*(F)$. For $t \in V_T$, we write $\chi_c(t)$ and $\chi_v(t)$ to denote the sets of clause modules and variable modules in $\chi(t)$, respectively. Note that $\chi(t) = \chi_c(t) \cup \chi_v(t)$. Moreover, we let $\mathcal{X}_t$ and $\mathcal{F}_t$ denote the set of variable modules and the set of clause modules occurring in the subtree of T rooted at t, respectively. We write $X_t = \langle \mathcal{X}_t \rangle$ and $F_t = \langle \mathcal{F}_t \rangle$. Note that $X_r = \mathrm{var}(F)$ and $F_r = F$.

3 Solving #SAT for Formulas of Bounded Modular Treewidth

In this section, we present an algorithm for computing the number of satisfying total truth assignments of a formula F. Our algorithm runs in polynomial time provided that the modular incidence treewidth of F is fixed. Before discussing our algorithm in detail, we first explain the main ideas behind it.

Let F be a formula and X be a set of variables. We can partition truth assignments defined on X into equivalence classes with respect to a relation $\sim_{(F,X)}$, which is defined as follows. Let $\sigma, \tau : X \to \{0,1\}$ be two distinct truth assignments defined on X. Then $\sigma \sim_{(F,X)} \tau$ if and only if σ and τ satisfy exactly the same set of clauses of F, or equivalently, if and only if $F(\sigma) = F(\tau)$. Due to the latter equivalence, we can speak about *the* projection of an equivalence class of $\sim_{(F,X)}$ on F. Recall that the number of satisfying total truth assignments of F is equal to $|\{\sigma : \mathrm{var}(F) \to \{0,1\} \mid F(\sigma) = \emptyset\}|$, which is the size of the equivalence class of $\sim_{(F,\mathrm{var}(F))}$ corresponding to the empty projection.

Now let (T, χ, r) be a nice tree decomposition of $I^*(F)$. We will apply dynamic programming over (T, χ, r). As usual, we start in the leaves of the tree and, using the parent-child relation, move to nodes closer to the root, and we stop after having processed the root. For each node $t \in V_T$, we define the formula

$$F_t^* \;=\; \big\{F^X \mid X \in \chi_v(t)\big\} \,\cup\, F_t \;=\; \big\{F^X \mid X \in \chi_v(t)\big\} \,\cup\, \langle \chi_c(t) \rangle \,\cup\, F_t \setminus \langle \chi_c(t) \rangle,$$

and we compute the sizes of those equivalence classes $[\tau]$ of $\sim_{(F_t^*, X_t)}$ that consist of truth assignments τ with $(F_t \setminus \langle \chi_c(t) \rangle)(\tau) = \emptyset$; we call such equivalence classes *transferable*. Below we explain three reasons why we do this.

Reason 1. The union of the transferable equivalence classes of $\sim_{(F_r^*, X_r)} = \sim_{(F_r^*, \mathrm{var}(F))}$ that consist of truth assignments τ with $\langle \chi_c(r) \rangle(\tau) = \emptyset$ and $(F_r \setminus \langle \chi_c(t) \rangle)(\tau) = (F \setminus \langle \chi_c(t) \rangle)(\tau) = \emptyset$ contains exactly all satisfying total truth assignments of F. Note that satisfying truth assignments of F may not satisfy some formula F^X for some $X \in \chi_v(r)$, but in that case only clauses in $F^X \setminus F_r = F^X \setminus F$ are not satisfied, and these clauses are irrelevant for our output.

Reason 2. We do not have to compute the sizes of any non-transferable equivalence classes of $\sim_{(F_t^*, X_t)}$, because these equivalence classes only contain truth assignments τ that cannot be extended to satisfying total truth assignments of F. This can be seen as follows. Let $\tau : X_t \to \{0,1\}$ be a truth assignment that belongs to a non-transferable equivalence class of $\sim_{(F_t^*, X_t)}$. In order to obtain a contradiction, assume that τ can be extended to a satisfying total truth assignment of F. By definition of non-transferability, $F_t \setminus \langle \chi_c(t) \rangle$ contains a clause C that is not satisfied by τ. Then C must contain at least one variable $x \in X_r \setminus X_t$ in order to be satisfied by an extension of τ. Let $\mathcal{C}$ be the clause module that contains C. Let X be the variable module that contains x. Then $X\mathcal{C}$ is an edge in $I^*(F)$. Hence, by condition 2 of the definition of a tree decomposition, there exists a node $t' \in V_T$ with $\{X, \mathcal{C}\} \subseteq \chi(t')$. Because $x \in X_r \setminus X_t$, we find that $X \in \mathcal{X}_r \setminus \mathcal{X}_t$. Hence, t' is not a node of the subtree of T rooted at t. However, as $C \in F_t \setminus \langle \chi_c(t) \rangle$, we also have $\mathcal{C} \in \mathcal{F}_t \setminus \chi_c(t)$ besides $\mathcal{C} \in \chi(t')$. Because this violates condition 3 of the definition of a tree decomposition, we

obtain a contradiction. We conclude that non-transferable equivalence classes may be discarded during our dynamic programming.

Reason 3. We must keep track of how truth assignments in transferable equivalence classes that not yet satisfy F itself can be extended to truth assignments that do satisfy F in a later stage of the dynamic programming. In particular, such truth assignments may not yet satisfy clauses C of F that belong to clause modules in $\chi_c(t)$ or that contain variables from one or more variable modules in $\chi_v(t)$; in the latter case their restriction C^X is nonempty, and consequently belongs to F^X, for at least one set $X \in \chi_v(t)$. In order to do this bookkeeping we partition truth assignments that satisfy $F_t \setminus \langle \chi_c(t) \rangle$ into transferable equivalence classes, that is, classes of truth assignments that satisfy exactly the same clauses of any $\mathcal{C} \in \chi_c(t)$ and exactly the same clauses of any F^X with $X \in \chi_v(t)$. We now prove why this does not cause an exponential blow-up. For a clause module $\mathcal{C} \in \chi_c(t)$, the number of equivalence classes of $\sim_{(\mathcal{C},X_t)}$ is bounded by $|\mathcal{C}| + 1$. This can be seen as follows. Let σ and τ be two truth assignments defined on X_t. Then Lemma 3 tells us that $\mathcal{C}(\sigma) = \mathbf{select}(\mathcal{C}, \mathbf{clause}(\sigma|_{X_t \cap \mathrm{var}(\mathcal{C})}))$ and $\mathcal{C}(\tau) = \mathbf{select}(\mathcal{C}, \mathbf{clause}(\tau|_{X_t \cap \mathrm{var}(\mathcal{C})}))$. This means that $\mathcal{C}(\sigma)$ and $\mathcal{C}(\tau)$ have no common clauses if $\mathcal{C}(\sigma) \neq \mathcal{C}(\tau)$, that is, if σ and τ belong to two different equivalence classes of $\sim_{(\mathcal{C},X_t)}$. Hence, taking into account that the set $\{\sigma : X_t \to \{0,1\} \mid \mathcal{C}(\sigma) = \emptyset\}$ may correspond to an equivalence class as well, the number of equivalence classes of $\sim_{(\mathcal{C},X_t)}$ is at most $|\mathcal{C}| + 1$. Similarly, for a variable module $X \in \chi_v(t)$, the number of equivalence classes of $\sim_{(F^X,X_t)}$ is at most $|F| + 1$ due to Lemma 4. Hence, the total number of different transferable equivalence classes of $\sim_{(F_t^*,X_t)}$ is at most

$$\prod_{\mathcal{C} \in \chi_c(t)} (|\mathcal{C}| + 1) \cdot \prod_{X \in \chi_v(t)} (|F| + 1) \leq (|F| + 1)^{|\chi_c(t)| + |\chi_v(t)|} = (|F| + 1)^{|\chi(t)|} \leq (|F| + 1)^{k+1}, \quad (1)$$

where k denotes the treewidth of $I^*(F)$, that is, the modular incidence treewidth of F. We observe that this bound is polynomial if k is fixed.

In order to describe the transferable equivalence classes, we use terminology introduced by Ganian, Hlinený and Obdrzálek [9], which we adjust for our purposes. Let $t \in V_T$. A *shape* for t is a pair of mappings (α, θ) where α has domain $\chi_v(t)$ with $\alpha(X) \in \mathscr{P}_{F^X}$ for all $X \in \chi_v(t)$ and θ has domain $\chi_c(t)$ with $\theta(\mathcal{C}) \in \mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C} \in \chi_c(t)$. A truth assignment $\tau : X_t \to \{0,1\}$ is said to be *of shape* (α, θ) if it satisfies the following three conditions:

(a) $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$
(b) $\mathcal{C}(\tau) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t)$
(c) $(F_t \setminus \langle \chi_c(t) \rangle)(\tau) = \emptyset$.

In other words, the set of assignments τ that are of shape (α, θ) describes exactly one transferable equivalence class of $\sim_{(F_t^*,X_t)}$. From now we denote this class by $N_t(\alpha, \theta)$, and we write $n_t(\alpha, \theta) = |N_t(\alpha, \theta)|$. We denote the set of all shapes for t that correspond to a transferable equivalence class by $\mathscr{S}_t$. By (1), we have $|\mathscr{S}_t| \leq (|F| + 1)^{k+1}$ for all nodes $t \in V_T$. Also note that by definition any truth assignment $\tau : X_t \to \{0,1\}$ has a (unique) shape if and only if $(F_t \setminus \langle \chi_c(t) \rangle)(\tau) = \emptyset$. We sometimes denote the shape of such a truth assignment τ by $(\alpha_\tau^t, \theta_\tau^t)$, where $\alpha_\tau^t(X) = F^X(\tau)$ for all $X \in \chi_v(t)$ and $\theta_\tau^t(\mathcal{C}) = \mathcal{C}(\tau)$ for all $\mathcal{C} \in \chi_c(t)$. Because equivalence classes are nonempty by definition, not every pair (α, θ) with $\alpha(X) \in \mathscr{P}_{F^X}$ for all $X \in \chi_v(t)$ and $\theta(\mathcal{C}) \in \mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C} \in \chi_c(t)$ is a shape for a node $t \in V_T$. We make this more explicit in our next lemma; see condition (ii) in particular.

Lemma 6 *Let $(\alpha, \theta) \in \mathscr{S}_t$ with $t \in V_T$, and let $\chi_v^*(t) \subseteq \chi_v(t)$. Moreover, let $\tau : X_t \to \{0,1\}$ satisfy $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v^*(t)$. For all $\mathcal{C} \in \chi_c(t)$, the following three conditions hold:*

(i) If $\mathcal{C}$ has no variable modules in $\chi_v^*(t)$, then $\mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathcal{C}$.
(ii) If $\mathcal{C}$ has some variable module $X \in \chi_v^*(t)$ with $\alpha(X) = \emptyset$, then $\mathcal{C}(\tau|_X) = \mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathcal{C}(\tau) = \emptyset$, and moreover, $\theta(\mathcal{C}) = \emptyset$ should τ be in $N_t(\alpha, \theta)$.

(iii) If $\mathcal{C}$ has exactly $p \geq 1$ variable modules $X_1, \ldots, X_p$ in $\chi_v^*(t)$ and $\alpha(X_i) \neq \emptyset$ for $i = 1, \ldots, p$, then $\mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathbf{select}(\mathcal{C}, \alpha(X_1) \cdots \alpha(X_p))$.

Proof Let $\mathcal{C} \in \chi_c(t)$. We first show (i). Suppose that $\mathcal{C}$ has no variable modules in $\chi_v^*(t)$. Because $\tau|_{\langle \chi_v^*(t) \rangle}$ is defined on $\langle \chi_v^*(t) \rangle$, it can only satisfy clauses that contain at least one variable module from $\chi_v^*(t)$. Hence, no clause in $\mathcal{C}$ is satisfied, that is, $\mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathcal{C}$.

We now show (ii). Suppose that $\mathcal{C}$ has some variable module $X \in \chi_v^*(t)$ with $\alpha(X) = \emptyset$. Because $\alpha(X) = F^X(\tau) = \emptyset$, this means that all clauses in $\mathcal{C}^X \subseteq F^X$, and consequently, all clauses in $\mathcal{C}$ (as $\mathcal{C}^X \neq \emptyset$) are satisfied already by $\tau|_X$, and hence by its extensions $\tau|_{\langle \chi_v^*(t) \rangle}$ and τ. This means that $\mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathcal{C}(\tau) = \emptyset$, and consequently, $\theta(\mathcal{C}) = \mathcal{C}(\tau) = \emptyset$ if $\tau \in N_t(\alpha, \theta)$.

Finally we show (iii). Suppose that $\mathcal{C}$ has exactly $p \geq 1$ variable modules $X_1, \ldots, X_p$ in $\chi_v^*(t)$ and that $\alpha(X_i) \neq \emptyset$ for $i = 1, \ldots, p$. Then $\alpha(X_i) = F^{X_i}(\tau) = \mathbf{clause}(\tau|_{X_i})$ for $i = 1, \ldots, p$ due to Lemma 4. Lemma 3 tells us that $\mathcal{C}(\tau|_{\langle \chi_v^*(t) \rangle}) = \mathbf{select}(\mathcal{C}, \mathbf{clause}(\tau|_{\langle \chi_v^*(t) \rangle \cap \mathrm{var}(C)})) = \mathbf{select}(\mathcal{C}, \mathbf{clause}(\tau|_{X_1}) \cdots \mathbf{clause}(\tau|_{X_p})) = \mathbf{select}(\mathcal{C}, \alpha(X_1) \cdots \alpha(X_p))$. □

We will now give the exact details of our dynamic programming, that is, how we compute all sizes $n_t(\alpha, \theta)$ over all $t \in V_T$ in order to be able to compute the desired output

$$n = \sum_{(\alpha, \theta) \in \mathscr{S}_r^*} n_r(\alpha, \theta),$$

where $\mathscr{S}_r^*$ consists of those shapes $(\alpha, \theta) \in \mathscr{S}_r$ with $\theta(\mathcal{C}) = \emptyset$ for all $\mathcal{C} \in \chi_c(r)$. Note that $\mathscr{S}_r^* = \emptyset$ is possible; in that case F is not satisfiable and $n = 0$.

Recall that the nodes of a tree associated with a nice tree decomposition can be partitioned into four types of nodes. Our next four lemmas show how to compute the sizes $n_t(\alpha, \theta)$ for these four types, i.e., for leaf nodes, introduce nodes, forget nodes and join nodes t, respectively.

Lemma 7 *Let t be a leaf node and $(\alpha, \theta) \in \mathscr{S}_t$. Then $n_t(\alpha, \theta) = \prod_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|)$.*

Proof Let t be a leaf node and $(\alpha, \theta) \in \mathscr{S}_t$. We first prove that a truth assignment $\tau : X_t \to \{0, 1\}$ belongs to $N_t(\alpha, \theta)$ if and only if $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$. The forward implication holds by definition. To prove the backward implication, suppose that $\tau : X_t \to \{0, 1\}$ is a truth assignment with $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$. Because $(\alpha, \theta) \in \mathscr{S}_t$, we find that $N_t(\alpha, \theta) \neq \emptyset$ by definition. Let $\tau^* \in N_t(\alpha, \theta)$. Because t is a leaf, $\mathcal{X}_t = \chi_v(t)$. This means that $\tau = \tau|_{\langle \chi_v(t) \rangle}$ and $\tau^* = \tau^*|_{\langle \chi_v(t) \rangle}$. Applying Lemma 6 yields that $\mathcal{C}(\tau|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau^*|_{\langle \chi_v(t) \rangle})$ for all $\mathcal{C} \in \chi_c(t)$. Moreover, because $\tau^* \in N_t(\alpha, \theta)$, we have $\mathcal{C}(\tau^*) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t)$. By combining the above arguments we find that

$$\mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau^*|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau^*) = \theta(\mathcal{C})$$

for all $\mathcal{C} \in \chi_c(t)$. Hence, $\tau \in N_t(\alpha, \theta)$ as required.

Due to the above we are left to compute the number of truth assignments τ that satisfy $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$. Recall that $\mathcal{X}_t = \chi_v(t)$. Hence, any truth assignment τ defined on X_t can be decomposed as $\bigcup_{X \in \chi_v(t)} \tau_X$, where each τ_X has domain X. Because the domains $X \in \chi_v(t)$ are mutually disjoint, we may consider each $X \in \chi_v(t)$ separately. Let $X \in \chi_v(t)$. Because $(\alpha, \theta) \in \mathscr{S}_t$, we find that $N_t(\alpha, \theta) \neq \emptyset$ by definition. Hence, $\alpha(X) \in \mathscr{P}_{F^X}$. First suppose that $\alpha(X) \neq \emptyset$. By Lemma 5, there is exactly one truth assignment $\tau_X : X \to \{0, 1\}$ with $F^X(\tau_X) = \alpha(X)$. Now suppose that $\alpha(X) = \emptyset$. Then, by Lemma 5, there are exactly $2^{|X|} - |F^X|$ truth assignments $\tau_X : X \to \{0, 1\}$ with $F^X(\tau_X) = \emptyset = \alpha(X)$. By combining the above arguments, we obtain

$$n_t(\alpha, \theta) = \prod_{X \in \chi_v(t) \setminus \alpha^{-1}(\emptyset)} 1 \cdot \prod_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|) = \prod_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|).$$

Hence we have proven Lemma 7. □

Let $(\alpha,\theta) \in \mathscr{S}_t$ for some $t \in V_T$. To simplify the formulation of Lemma 8, we define a mapping g with domain $\chi_c(t) \times \chi_v(t)$ as follows. When $X \in \chi_v(t)$ is not a variable module of a clause $\mathcal{C} \in \chi_c(t)$, we let $g(\mathcal{C},X) = \mathcal{C}$. Otherwise, we let $g(\mathcal{C},X) = \emptyset$ if $\alpha(X) = \emptyset$, and $g(\mathcal{C},X) = \mathbf{select}(\mathcal{C},\alpha(X))$ if $\alpha(X) \neq \emptyset$.

Lemma 8 *Let $t \in V_T$ be an introduce node with child t', such that $\chi(t)\setminus\{S\} = \chi(t')$ for a module $S \in \chi(t)$. Let $(\alpha,\theta) \in \mathscr{S}_t$. Moreover, let $\alpha' = \alpha|_{\chi_v(t')}$ and $\theta' = \theta|_{\chi_c(t')}$.*

(i) If $S \in \chi_v(t)$, then $n_t(\alpha,\theta) = \begin{cases} \sum\limits_{\theta^* \in \mathscr{T}} n_{t'}(\alpha',\theta^*) & \text{if } \alpha(S) \neq \emptyset \\ (2^{|S|} - |F^S|) \sum\limits_{\theta^* \in \mathscr{T}} n_{t'}(\alpha',\theta^*) & \text{if } \alpha(S) = \emptyset, \end{cases}$

where $\mathscr{T} = \{\, \theta^* \mid (\alpha',\theta^*) \in \mathscr{S}_{t'}$ and $\theta^*(\mathcal{C}) \cap g(\mathcal{C},S) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t) \,\}$.

(ii) If $S \in \chi_c(t)$, then $n_t(\alpha,\theta) = n_{t'}(\alpha,\theta')$.

Proof We first prove (i). We write $X = S$. Note that $\chi_c(t) = \chi_c(t')$. Hence $\mathscr{T}$ is well defined. Let $E = \{\sigma : X \to \{0,1\} \mid F^X(\sigma) = \alpha(X)\}$, and let

$$M \;=\; E \;\times \bigcup_{\theta^* \in \mathscr{T}} N_{t'}(\alpha',\theta^*).$$

We will show that $f : \tau \mapsto (\tau|_X, \tau|_{X_{t'}})$ is a bijection from $N_t(\alpha,\theta)$ to M. Then afterward we are left to compute the size of M, which we will prove to be equal to the right hand side of the equation in (i). For proving that f is a bijection from $N_t(\alpha,\theta)$ to M we must show that f is into and bijective.

Claim 1. The mapping f is into.
We prove Claim 1 as follows. Let $\tau \in N_t(\alpha,\theta)$, and let $\tau' = \tau|_{X_{t'}}$. We will show that $\tau|_X \in E$ and that $\tau' \in N_{t'}(\alpha',\theta^*)$ for some $\theta^* \in \mathscr{T}$.

Because $\tau \in N_t(\alpha,\theta)$, we have $F^X(\tau) = \alpha(X)$. Because all clauses in F^X only contain variables from X, we find that $F^X(\tau|_X) = F^X(\tau)$. Hence, $F^X(\tau|_X) = \alpha(X)$, which implies that $\tau|_X \in E$.

We now show that $\tau' \in N_{t'}(\alpha',\theta^*)$ for some $\theta^* \in \mathscr{T}$. Let $\mathcal{C} \in \mathcal{F}_{t'}\setminus\chi_c(t')$. Then $\mathcal{C} \in \mathcal{F}_t\setminus\chi_c(t)$. Because $\tau \in N_t(\alpha,\theta)$, this means that τ satisfies $\mathcal{C}$. By condition 2 of the definition of a tree decomposition, $X \notin \mathrm{var}(\mathcal{C})$. Hence, τ' satisfies $\mathcal{C}$. This means that τ' is of shape $(\alpha^{t'}_{\tau'}, \theta^{t'}_{\tau'})$, so $\tau' \in N_{t'}(\alpha^{t'}_{\tau'}, \theta^{t'}_{\tau'})$. Because τ' is the restriction of τ on $X_{t'}$, we find that $\alpha^{t'}_{\tau'}(X') = F^{X'}(\tau') = F^{X'}(\tau) = \alpha'(X')$ for all $X' \in \chi_v(t')$. Consequently, $\alpha^{t'}_{\tau'} = \alpha'$.

We now show that $\theta^{t'}_{\tau'} \in \mathscr{T}$. In order to do this, we are left to prove that $\theta^{t'}_{\tau'}(\mathcal{C}) \cap g(\mathcal{C},X) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t)$. Let $\mathcal{C} \in \chi_c(t)$. Lemma 2 tells us that $\mathcal{C}(\tau) = \mathcal{C}(\tau' \cup \tau|_X) = \mathcal{C}(\tau') \cap \mathcal{C}(\tau|_X)$. Because $F^{X'}(\tau) = \alpha(X')$ for all $X' \in \chi_v(t)$, we can use Lemma 6. First suppose that X is not a variable module of $\mathcal{C}$. Then $\mathcal{C}(\tau|_X) = \mathcal{C}$ by Lemma 6 (i). Hence, $\theta^{t'}_{\tau'}(\mathcal{C}) \cap g(\mathcal{C},X) = \mathcal{C}(\tau') \cap \mathcal{C} = \mathcal{C}(\tau') \cap \mathcal{C}(\tau|_X) = \mathcal{C}(\tau) = \theta(\mathcal{C})$. Now suppose that X is a variable module of $\mathcal{C}$. If $\alpha(X) = \emptyset$, then $\theta(\mathcal{C}) = \emptyset$ by Lemma 6 (ii). Hence, $\theta^{t'}_{\tau'}(\mathcal{C}) \cap g(\mathcal{C},X) = \theta^{t'}_{\tau'}(\mathcal{C}) \cap \emptyset = \emptyset = \theta(\mathcal{C})$. If $\alpha(X) \neq \emptyset$, then $\mathcal{C}(\tau|_X) = \mathbf{select}(\mathcal{C},\alpha(X))$ by Lemma 6 (iii). Hence,

$$\theta^{t'}_{\tau'}(\mathcal{C}) \cap g(\mathcal{C},X) = \mathcal{C}(\tau') \cap \mathbf{select}(\mathcal{C},\alpha(X)) = \mathcal{C}(\tau') \cap \mathcal{C}(\tau|_X) = \mathcal{C}(\tau) = \theta(\tau).$$

We conclude that $\theta^{t'}_{\tau'} \in \mathscr{T}$. Because we already deduced that $\alpha^{t'}_{\tau'} = \alpha'$, we have found that $\tau' \in N_{t'}(\alpha',\theta^*)$ for some $\theta^* \in \mathscr{T}$, namely $\theta^* = \theta^{t'}_{\tau'}$. Because we already deduced that $\tau|_X \in E$, this means that $(\tau|_X, \tau') \in M$. Hence, f is into, which proves Claim 1.

Claim 2. The mapping f is bijective.
We prove Claim 2 as follows. By construction, f is injective. Below we show that f is surjective.

Let $(\tau_X,\tau') \in M$. Then $\tau_X \in E$ and $\tau' \in N_{t'}(\alpha',\theta^*)$ for some $\theta^* \in \mathscr{T}$. Because τ_X has domain X, and τ' has domain $X_{t'}$, and $X \cap X_{t'} = \emptyset$, we can well define $\tau = \tau_X \cup \tau'$. Note that

$f(\tau) = (\tau_X, \tau')$. We must show that $\tau \in N_t(\alpha, \theta)$. Let $\mathcal{C} \in \mathcal{F}_t \setminus \chi_c(t)$. Then $\mathcal{C} \in \mathcal{F}_{t'} \setminus \chi_c(t')$, as $\chi_c(t) = \chi_c(t')$. Because $\tau' \in N_{t'}(\alpha', \theta^*)$, we find that τ' satisfies $\mathcal{C}$. Because τ is an extension of τ', this means that τ satisfies $\mathcal{C}$. Hence, τ is of shape $(\alpha^t_\tau, \theta^t_\tau)$.

First we show that $\alpha^t_\tau = \alpha$. Because τ is an extension of τ', and τ' has domain $X_{t'} \supseteq \langle \chi_v(t') \rangle$, we find that τ and τ' agree on $\langle \chi_v(t') \rangle$. This means that $F^{X'}(\tau) = F^{X'}(\tau')$ for all $X' \in \chi_v(t')$. Hence $\alpha^t_\tau|_{\chi_v(t')} = \alpha'$. By using the definition of a shape and our assumption that $\tau_X \in E$, we find that $\alpha^t_\tau(X) = F^X(\tau) = F^X(\tau_X) = \alpha(X)$. We conclude that $\alpha^t_\tau = \alpha$.

Now we show that $\theta^t_\tau = \theta$. Let $\mathcal{C} \in \chi_c(t)$. Lemma 2 tells us that $\mathcal{C}(\tau) = \mathcal{C}(\tau' \cup \tau_X) = \mathcal{C}(\tau') \cap \mathcal{C}(\tau_X)$. Moreover, we may use Lemma 6, as $\tau \in N_t(\alpha^t_\tau, \theta^t_\tau)$ and $\alpha^t_\tau = \alpha$, and hence $F^X(\tau) = \alpha(X)$. According to Lemma 6, we have $\mathcal{C}(\tau|_X) = \mathcal{C}$ if X is not a variable module of $\mathcal{C}$, whereas otherwise $\mathcal{C}(\tau|_X) = \emptyset$ if $\alpha(X) = \emptyset$, and $\mathcal{C}(\tau|_X) = \mathbf{select}(\mathcal{C}, \alpha(X))$ if $\alpha(X) \neq \emptyset$. This means that $\mathcal{C}(\tau|_X) = g(\mathcal{C}, X)$. Then

$$\theta^t_\tau(\mathcal{C}) = \mathcal{C}(\tau) = \mathcal{C}(\tau') \cap \mathcal{C}(\tau_X) = \theta^*(\mathcal{C}) \cap \mathcal{C}(\tau|_X) = \theta^*(\mathcal{C}) \cap g(\mathcal{C}, X) = \theta(\mathcal{C}),$$

where the latter equality follows from our assumption that $\theta^* \in \mathscr{T}$. Hence, $\theta^t_\tau = \theta$. Because we already deduced that $\alpha^t_\tau = \alpha$, we have found that $\tau \in N_t(\alpha, \theta)$. This means that f is surjective, which completes the proof of Claim 2.

We are left to determine $|M|$. Because equivalence classes are mutually disjoint, we find that

$$|M| = |E| \ \times | \bigcup_{\theta^* \in \mathscr{T}} N_{t'}(\alpha', \theta^*)| = |E| \sum_{\theta^* \in \mathscr{T}} |N_{t'}(\alpha', \theta^*)| = |E| \sum_{\theta^* \in \mathscr{T}} n_{t'}(\alpha', \theta^*),$$

where $|E| = 1$ if $\alpha(X) \neq \emptyset$ and $|E| = 2^{|X|} - |F^X|$ otherwise, due to Lemma 5. This completes the proof of (i).

We now prove (ii). We write $\mathcal{C} = S$. Note that $\chi_v(t) = \chi_v(t')$. We claim that $\tau \in N_t(\alpha, \theta)$ if and only if $\tau \in N_{t'}(\alpha, \theta')$.

First suppose that $\tau \in N_t(\alpha, \theta)$. Because θ' is the restriction of θ to $\chi_c(t')$, we immediately find that $\tau \in N_{t'}(\alpha, \theta')$.

Now suppose that $\tau \in N_{t'}(\alpha, \theta')$. Then, as $\chi_v(t) = \chi_v(t')$, we find that $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$. Because $\tau \in N_{t'}(\alpha, \theta')$, we also find that τ satisfies $F_{t'} \setminus \langle \chi_c(t') \rangle$. The latter implies together with $\chi_c(t) = \chi_c(t') \cup \{\mathcal{C}\}$ that τ satisfies $F_t \setminus \langle \chi_c(t) \rangle$. Because $\tau \in N_{t'}(\alpha, \theta')$ and θ' is the restriction of θ to $\chi_c(t')$, we obtain $\mathcal{C}'(\tau) = \theta'(\mathcal{C}') = \theta(\mathcal{C}')$ for all $\mathcal{C}' \in \chi_c(t') = \chi_c(t) \setminus \{\mathcal{C}\}$. Hence we are left to show that $\mathcal{C}(\tau) = \theta(\mathcal{C})$.

Because $(\alpha, \theta) \in \mathscr{S}_t$, there exists a truth assignment $\tau^* \in N_t(\alpha, \theta)$. By condition 2 of the definition of a tree decomposition, all variable modules of $\mathcal{C}$ that are in $\mathcal{X}_t$ must belong to $\chi_v(t)$. Hence, $\mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle})$, and similarly, $\mathcal{C}(\tau^*) = \mathcal{C}(\tau^*|_{\langle \chi_v(t) \rangle})$. Because $\alpha(X) = F^X(\tau) = F^X(\tau^*)$ for all $X \in \chi_v(t)$, we may use Lemma 6. As a result of Lemma 6, we obtain that $\mathcal{C}(\tau|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau^*|_{\langle \chi_v(t) \rangle})$. This means that $\mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau^*|_{\langle \chi_v(t) \rangle}) = \mathcal{C}(\tau) = \theta(\mathcal{C})$. We conclude that $\tau \in N_t(\alpha, \theta)$. This completes the proof of Lemma 8 (ii). □

Lemma 9 *Let $t \in V_T$ be a forget node with child t', such that $\chi(t) = \chi(t') \setminus \{S\}$ for a module $S \in \chi(t')$. Let $(\alpha, \theta) \in \mathscr{S}_t$.*

(i) If $S \in \chi_v(t')$, then $n_t(\alpha, \theta) = \sum_{\Pi \in \mathscr{P}_{F^S}} n_{t'}(\alpha \cup \{(S, \Pi)\}, \theta)$.

(ii) If $S \in \chi_c(t')$, then $n_t(\alpha, \theta) = n_{t'}(\alpha, \theta \cup \{(S, \emptyset)\})$.

Proof We first prove (i). We write $X = S$. Let $M = \bigcup_{\Pi \in \mathscr{P}_{F^X}} N_{t'}(\alpha \cup \{(X, \Pi)\}, \theta)$. Because equivalence classes are mutually disjoint, we observe that

$$|M| = \sum_{\Pi \in \mathscr{P}_{F^X}} n_{t'}(\alpha \cup \{(X, \Pi)\}, \theta).$$

Hence, we are done after showing that $\tau \in N_t(\alpha, \theta)$ if and only if $\tau \in M$.

First suppose that $\tau \in N_t(\alpha, \theta)$. Let $\mathcal{C} \in \mathcal{F}_{t'} \setminus \chi_c(t')$. Because $\chi_c(t) = \chi_c(t')$, we find that $\mathcal{C} \in \mathcal{F}_t \setminus \chi_c(t)$. Hence τ satisfies $\mathcal{C}$ by the definition of a shape. As $X_{t'} = X_t$, this means that τ is has a shape with respect to t' as well, that is, $\tau \in N_{t'}(\alpha_\tau^{t'}, \theta_\tau^{t'})$. Let $\mathcal{C} \in \chi_c(t)$. Then $\theta_\tau^{t'}(\mathcal{C}) = \mathcal{C}(\tau) = \theta(\mathcal{C})$. As $\chi_c(t) = \chi_c(t')$, we find that $\theta_\tau^{t'} = \theta$. Let $X' \in \chi_v(t) = \chi_v(t') \setminus \{X\}$. Then $\alpha_\tau^{t'}(X') = F^{X'} = \alpha(X')$. Let $\Pi = \alpha_\tau^{t'}(X) = F^X(\tau) \in \mathscr{P}_{F^X}$. Hence, $\alpha_\tau^{t'} = \alpha \cup \{X, \Pi\}$. We conclude that $\tau \in N_t(\alpha \cup \{(X, \Pi)\}, \theta)$, and thus $\tau \in M$.

Now suppose that $\tau \in M$. Then $\tau \in N_{t'}(\alpha \cup \{(X, \Pi)\}, \theta)$ for some $\Pi \in \mathscr{P}_{F^X}$. This immediately implies that $\tau \in N_t(\alpha, \theta)$.

We now prove (ii). We write $S = \mathcal{C}$. We are done after showing that $\tau \in N_t(\alpha, \theta)$ if and only if $\tau \in N_{t'}(\alpha, \theta \cup \{(\mathcal{C}, \emptyset)\})$. First suppose that $\tau \in N_t(\alpha, \theta)$. Then τ satisfies $F_t \setminus \langle \chi_c(t) \rangle$, and hence, τ satisfies $\mathcal{C} \in \mathcal{F}_t \setminus \chi_c(t)$, that is, $\mathcal{C}(\tau) = \emptyset$. Combining this with $\tau \in N_t(\alpha, \theta)$ implies that $\tau \in N_{t'}(\alpha, \theta \cup \{(\mathcal{C}, \emptyset)\})$. Now suppose that $\tau \in N_{t'}(\alpha, \theta \cup \{(\mathcal{C}, \emptyset)\})$. Then $\tau \in N_t(\alpha, \theta)$ immediately follows. □

Lemma 10 *Let $t \in V_T$ be a join node with children t_1 and t_2. Let $(\alpha, \theta) \in \mathscr{S}_t$. Moreover, let $\mathscr{T}_{1,2} = \{ (\theta_1, \theta_2) \mid (\alpha, \theta_1) \in \mathscr{S}_{t_1}, (\alpha, \theta_2) \in \mathscr{S}_{t_2}$, and $\theta_1(\mathcal{C}) \cap \theta_2(\mathcal{C}) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t) \}$. Then the following equality holds:*

$$n_t(\alpha, \theta) = \frac{1}{\prod\limits_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|)} \sum_{(\theta_1, \theta_2) \in \mathscr{T}_{1,2}} n_{t_1}(\alpha, \theta_1) \cdot n_{t_2}(\alpha, \theta_2).$$

Proof By the definition of a nice tree decomposition, we have $\chi(t) = \chi(t_1) = \chi(t_2)$. Hence, $\chi_v(t) = \chi_v(t_1) = \chi_v(t_2)$ and thus $\mathcal{X}_t = \mathcal{X}_{t_1} = \mathcal{X}_{t_2}$, and moreover, $\chi_c(t) = \chi_c(t_1) = \chi_c(t_2)$ and thus $\mathcal{F}_t = \mathcal{F}_{t_1} = \mathcal{F}_{t_2}$. We will use these equations throughout the proof, but first we define two sets M' and $M \subseteq M'$ in the following way:

$$M' = \bigcup_{(\theta_1, \theta_2) \in \mathscr{S}_{1,2}} (N_{t_1}(\alpha, \theta_1) \times N_{t_2}(\alpha, \theta_2))$$

$$M = \{ (\tau_1, \tau_2) \in M' \mid \tau_1|_{\langle \chi_v(t) \rangle} = \tau_2|_{\langle \chi_v(t) \rangle} \}.$$

We will show that $f : \tau \mapsto (\tau|_{X_{t_1}}, \tau|_{X_{t_2}})$ is a bijection from $N_t(\alpha, \theta)$ to M. Then afterward we are left to compute the size of M, which we will prove to be equal to the right hand side of the equation in the statement of Lemma 10. For proving that f is a bijection from $N_t(\alpha, \theta)$ to M we must show that f is into and bijective.

Claim 1. The mapping f is into.
We prove Claim 1 as follows. Let $\tau \in N_t(\alpha, \theta)$. We write $\tau_1 = \tau|_{X_{t_1}}$ and $\tau_2 = \tau|_{X_{t_2}}$. Note that $\tau = \tau_1 \cup \tau_2$ and that $f(\tau) = (\tau_1, \tau_2)$.

We first show that τ_1 satisfies $F_{t_1} \setminus \langle \chi_c(t_1) \rangle$. Let $\mathcal{C}$ be a clause module in $\mathcal{F}_{t_1} \setminus \chi_c(t_1) = \mathcal{F}_t \setminus \chi_c(t)$ (as $\mathcal{F}_{t_1} = \mathcal{F}_t$ and $\chi_c(t_1) = \chi_c(t)$). Because $\tau \in N_t(\alpha, \theta)$ and $\mathcal{C} \in \mathcal{F}_t \setminus \chi_c(t)$, we find that $\mathcal{C}(\tau) = \emptyset$. By condition 3 of a tree decomposition, $\mathcal{C} \in \mathcal{F}_{t_1} \setminus \chi_c(t_1)$ may only occur in a bag $\chi(t')$ if t' belongs to the subtree of T rooted by t_1. By condition 2 of a tree decomposition, every variable module of $\mathcal{C}$ belongs to at least one bag that also contains $\mathcal{C}$. Hence all variable modules of $\mathcal{C}$ belong to $\mathcal{X}_{t_1}$. This means that $\mathcal{C}(\tau_1) = \mathcal{C}(\tau) = \emptyset$. Hence, τ_1 satisfies $F_{t_1} \setminus \langle \chi_c(t_1) \rangle$. By exactly the same arguments, we deduce that τ_2 satisfies $F_{t_2} \setminus \langle \chi_c(t_2) \rangle$.

Because τ_1 satisfies $F_{t_1} \setminus \langle \chi_c(t_1) \rangle$, we find that τ_1 is of shape $(\alpha_{\tau_1}^{t_1}, \theta_{\tau_1}^{t_1}) \in \mathscr{S}_{t_1}$. Similarly, because τ_2 satisfies $F_{t_2} \setminus \langle \chi_c(t_2) \rangle$, we find that τ_2 is of shape $(\alpha_{\tau_2}^{t_2}, \theta_{\tau_2}^{t_2}) \in \mathscr{S}_{t_2}$. Because τ_1, τ_2, and τ all agree on $\langle \chi_v(t) \rangle$, we deduce that $F^X(\tau_1) = F^X(\tau_2) = F^X(\tau)$ for all variable modules X in $\chi_v(t_1) = \chi_v(t_2) = \chi_v(t)$. Hence, $\alpha_{\tau_1}^{t_1} = \alpha_{\tau_2}^{t_2} = \alpha$.

We now show that $(\theta_{\tau_1}^{t_1}, \theta_{\tau_2}^{t_2}) \in \mathscr{T}_{1,2}$. Because $\alpha_{\tau_1}^{t_1} = \alpha_{\tau_2}^{t_2} = \alpha$, we have $(\alpha, \theta_{\tau_1}^{t_1}) \in \mathscr{S}_{t_1}$ and $(\alpha, \theta_{\tau_2}^{t_2}) \in \mathscr{S}_{t_2}$. Let $\mathcal{C} \in \chi_c(t)$. Because $\tau = \tau_1 \cup \tau_2$, we may apply Lemma 2 to deduce that

$\mathcal{C}(\tau) = \mathcal{C}(\tau_1) \cap \mathcal{C}(\tau_2)$. Using the definition of a shape, this yields that

$$\theta(\mathcal{C}) = \mathcal{C}(\tau) = \mathcal{C}(\tau_1) \cap \mathcal{C}(\tau_2) = \theta^{t_1}_{\tau_1}(\mathcal{C}) \cap \theta^{t_2}_{\tau_2}(\mathcal{C}).$$

Hence, $(\theta^{t_1}_{\tau_1}, \theta^{t_2}_{\tau_2}) \in \mathscr{T}_{1,2}$. We conclude that $(\tau_1, \tau_2) \in M'$. Because $\tau_1 = \tau|_{X_{t_1}}$ and $\tau_2 = \tau|_{X_{t_2}}$, and moreover, $\chi_v(t_1) = \chi_v(t_2) = \chi_v(t)$, we find that $\tau_1|_{\langle \chi_v(t) \rangle} = \tau_2|_{\langle \chi_v(t) \rangle}$. Hence $(\tau_1, \tau_2) \in M$, which means that f is into. This completes the proof of Claim 1.

Claim 2. The mapping f is bijective.
We prove Claim 2 as follows. By construction, f is injective. Below we show that f is surjective.

Let $(\tau_1, \tau_2) \in M$. Then τ_1 and τ_2 have domains X_{t_1} and X_{t_2}, respectively. Because $X_{t_1} \cap X_{t_2} = \langle \chi_v(t) \rangle$ on which τ_1 and τ_2 agree by the definition of M, we may define $\tau = \tau_1 \cup \tau_2$. Because $(\tau_1, \tau_2) \in M \subseteq M'$, we find that $\tau_1 \in N_{t_1}(\alpha, \theta_1)$ and $\tau_2 \in N_{t_2}(\alpha, \theta_2)$ for some $(\theta_1, \theta_2) \in \mathscr{T}_{1,2}$. Then, by definition, τ_1 satisfies $F_{t_1} \setminus \langle \chi_c(t_1) \rangle$ and τ_2 satisfies $F_{t_2} \setminus \langle \chi_c(t_2) \rangle$. Hence, $\tau = \tau_1 \cup \tau_2$ satisfies $(F_{t_1} \setminus \langle \chi_c(t_1) \rangle) \cup (F_{t_2} \setminus \langle \chi_c(t_2) \rangle) = F_t \setminus \langle \chi_c(t) \rangle$. Because $\mathcal{X}_t = \mathcal{X}_{t_1} = \mathcal{X}_{t_2}$, or equivalently, $X_t = X_{t_1} = X_{t_2}$, this means that τ is of shape $(\alpha^t_\tau, \theta^t_\tau) \in \mathscr{S}_t$. Because τ and τ_1 agree on $\langle \chi_v(t) \rangle$, we find that $\alpha^t_\tau = \alpha$. We now prove that $\theta^t_\tau = \theta$. Let $\mathcal{C} \in \chi_c(t)$. By using Lemma 2 and the definitions of a shape and the set $\mathscr{T}_{1,2}$, we find that

$$\theta^t_\tau(\mathcal{C}) = \mathcal{C}(\tau) = \mathcal{C}(\tau_1) \cap \mathcal{C}(\tau_2) = \theta_1(\mathcal{C}) \cap \theta_2(\mathcal{C}) = \theta(\mathcal{C}).$$

Hence, $\theta^t_\tau = \theta$. As we already deduced that $\alpha^t_\tau = \alpha$, we find that $\tau \in N_t(\alpha, \theta)$. This completes the proof of Claim 2.

We are left to determine the cardinality of M. For this purpose we first determine $|M'|$. Because $N_{t_1}(\alpha, \theta_1) \times N_{t_2}(\alpha, \theta_2)$ and $N_{t_1}(\alpha, \theta'_1) \times N_{t_2}(\alpha, \theta'_2)$ are disjoint for $\theta_1 \neq \theta'_1$ or $\theta_2 \neq \theta'_2$, we find that

$$|M'| = \sum_{(\theta_1, \theta_2) \in \mathscr{T}_{1,2}} n_{t_1}(\alpha, \theta_1)\, n_{t_2}(\alpha, \theta_2). \tag{2}$$

In order to determine $|M|$ we consider an equivalence relation $\sim$ defined on $2^{X_{t_1}} \times 2^{X_{t_2}}$ as follows. We let $(\tau_1, \tau_2) \sim (\tau'_1, \tau'_2)$ if each of the following three conditions are satisfied:

(i) $\tau_1 = \tau'_1$
(ii) $\tau_2|_{X_{t_2} \setminus \langle \chi_v(t_2) \rangle} = \tau'_2|_{X_{t_2} \setminus \langle \chi_v(t_2) \rangle}$
(iii) $F^X(\tau_2) = F^X(\tau'_2)$ for all $X \in \chi_v(t_2)$.

In particular, condition (ii) has the following crucial implication. If $(\tau_1, \tau_2) \sim (\tau_1, \tau'_2)$, then due to this condition τ'_2 can only differ from τ_2 on $\langle \chi_v(t_2) \rangle$. We will now compute the size of the equivalence classes $[(\tau_1, \tau_2)]$ of $\sim$ with $(\tau_1, \tau_2) \in M'$ and then show that in fact all pairs $(\tau_1, \tau'_2) \in [(\tau_1, \tau_2)]$ belong to M'. We then prove that exactly one pair (τ_1, τ'_2) of such an equivalence class belongs to M. We will use this information together with equality (2) to compute the size of $|M|$.

Let $(\tau_1, \tau_2) \in M'$, that is, $\tau_1 \in N_{t_1}(\alpha, \theta_1)$ and $\tau_2 \in N_{t_2}(\alpha, \theta_2)$ such that $(\theta_1, \theta_2) \in \mathscr{T}_{1,2}$ for some suitable θ_1 and θ_2. Consider any pair $(\tau_1, \tau'_2) \in [(\tau_1, \tau_2)]$. By definition, $(\tau_1, \tau_2) \sim (\tau_1, \tau'_2)$. Recall that τ'_2 can only differ from τ_2 on $\langle \chi_v(t) \rangle$. By the definitions of $\sim$ and of a shape, we have $F^X(\tau'_2) = F^X(\tau_2) = \alpha(X)$ for all $X \in \chi_v(t)$. By Lemma 5, the number of truth assignments $\tau_X : X \to \{0, 1\}$ with $F^X(\tau_X) = \alpha(X)$ is equal to 1 if $\alpha(X) \neq \emptyset$ and equal to $2^{|X|} - |F^X|$ otherwise. Hence $\alpha(X) \neq \emptyset$ implies that $\tau_2|_X = \tau'_2|_X$, whereas there exist $2^{|X|} - |F^X|$ ways to define τ'_2 on X if $\alpha(X) = \emptyset$; note that τ'_2 satisfies F^X if and only if $\tau'_2|_X$ satisfies F^X. Because the variable modules in $\chi_v(t)$ are mutually disjoint, we then find that

$$|[(\tau_1, \tau_2)]| = \prod_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|). \tag{3}$$

As noted, we now show that $[(\tau_1,\tau_2)]\cap M' = [(\tau_1,\tau_2)]$. Suppose that $(\tau_1,\tau_2)\sim(\tau_1,\tau_2')$. We claim that $(\tau_1,\tau_2')\in M'$. By assumption, $\tau_1\in N_{t_1}(\alpha,\theta_1)$ and $(\theta_1,\theta_2)\in\mathscr{T}_{1,2}$. Hence, in order to prove this claim, it suffices to prove that $\tau_2'\in N_{t_2}(\alpha,\theta_2)$.

Let $X\in\chi_v(t_2)$. Our assumption that $\tau_2\in N_{t_2}(\alpha,\theta_2)$ and condition (iii) of the definition of $\sim$ imply that $F^X(\tau_2')=F^X(\tau_2)=\alpha(X)$.

Now, let $\mathcal{C}\in\chi_c(t_2)$. By condition (ii) of the definition of $\sim$ we obtain $\mathcal{C}(\tau_2'|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle}) = \mathcal{C}(\tau_2|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle})$. Because $F^X(\tau_2')=F^X(\tau_2)=\alpha(X)$ for all $X\in\chi_v(t_2)$, we can use Lemma 6 to deduce that $\mathcal{C}(\tau_2'|_{\langle\chi_v(t_2)\rangle}) = \mathcal{C}(\tau_2|_{\langle\chi_v(t_2)\rangle})$. Then, by using Lemma 2 and the definition of a shape, we obtain

$$\mathcal{C}(\tau_2') = \mathcal{C}(\tau_2'|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle})\cap\mathcal{C}(\tau_2'|_{\langle\chi_v(t_2)\rangle}) = \mathcal{C}(\tau_2|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle})\cap\mathcal{C}(\tau_2|_{\langle\chi_v(t_2)\rangle}) = \mathcal{C}(\tau_2) = \theta_2(\mathcal{C}).$$

Finally, let $C\in F_{t_2}\setminus\langle\chi_c(t_2)\rangle$. In order to prove that $(F_{t_2}\setminus\langle\chi_c(t_2)\rangle)(\tau_2')=\emptyset$ we must show that τ_2' satisfies C. Our assumption that $\tau_2\in N_{t_2}(\alpha,\theta_2)$ implies that $(F_{t_2}\setminus\langle\chi_c(t_2)\rangle)(\tau_2)=\emptyset$. Hence τ_2 satisfies C, that is, C contains a literal x with $\tau_2(x)=1$. Let X be the variable module to which x belongs, and let $\mathcal{C}$ be the clause module to which C belongs. Then X and $\mathcal{C}$ are adjacent in $I^*(F)$. By condition 2 of a tree decomposition, X and $\mathcal{C}$ belong to some common bag $\chi(t')$. Because $C\in F_{t_2}\setminus\langle\chi_c(t_2)\rangle$, we have $\mathcal{C}\in\mathcal{F}_{t_2}\setminus\chi_c(t_2)$. Then, by condition 3 of a tree decomposition, t' must be a node not equal to t_2 in the subtree of T rooted at t_2. Hence, $X\in\mathcal{X}_{t_2}\setminus\chi_v(t_2)$, which means that $x\in X_{t_2}\setminus\langle x_v(t_2)\rangle$. Then $\tau_2'(x)=\tau_2(x)=1$ due to condition (ii) of the definition of $\sim$, which means that τ_2' satisfies C. Hence, we have proven that $(F_{t_2}\setminus\langle\chi_c(t_2)\rangle)(\tau_2')=\emptyset$. We conclude that $\tau_2'\in N_{t_2}(\alpha,\theta_2)$, and consequently, $(\tau_1,\tau_2')\in M'$.

As noted, our next step is to prove that exactly one pair from $[(\tau_1,\tau_2)]$ belongs to M. Recall that M consists of those truth assignment pairs of M' of which both truth assignments agree on $\chi_v(t)=\chi_v(t_1)=\chi_v(t_2)$. We first show that the pair (τ_1,τ_2') defined by $\tau_2'|_{\langle\chi_v(t_2)\rangle}=\tau_1|_{\langle\chi_v(t_2)\rangle}$ and $\tau_2'|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle}=\tau_2|_{X_{t_2}\setminus\langle\chi_v(t_2)\rangle}$ belongs to $[(\tau_1,\tau_2)]\cap M$. Because $\tau_1\in N_{t_1}(\alpha,\theta_1)$, we have $F^X(\tau_1)=\alpha(X)$ for all $X\in\chi_v(t_1)=\chi_v(t_2)$. Because $\tau_1|_{\langle\chi_v(t_2)\rangle}=\tau_2'|_{\langle\chi_v(t_2)\rangle}$, this means that $F^X(\tau_2')=\alpha(X)$ for all $X\in\chi_v(t_2)$. Similarly, because $\tau_2\in N_{t_2}(\alpha,\theta_2)$, we have $F^X(\tau_2)=\alpha(X)$ for all $X\in\chi_v(t_2)$. Hence, $F^X(\tau_2)=F^X(\tau_2')$ for all $X\in\chi_v(t_2)$, which is condition (iii) of the definition of $\sim$. As conditions (i) and (ii) of $\sim$ are satisfied by the way we constructed (τ_1,τ_2'), we find that $(\tau_1,\tau_2')\in[(\tau_1,\tau_2)]$. As we already showed that $[(\tau_1,\tau_2)]\cap M'=[(\tau_1,\tau_2)]$, this means that $(\tau_1,\tau_2')\in M'$. Consequently, $(\tau_1,\tau_2')\in M$, as $\tau_1|_{\langle\chi_v(t)\rangle}=\tau_2'|_{\langle\chi_v(t)\rangle}$ by construction. We conclude that $(\tau_1,\tau_2')\in[(\tau_1,\tau_2)]\cap M$. Hence, at least one pair from $[(\tau_1,\tau_2)]$ belongs to M.

We now show that at most one pair from $[(\tau_1,\tau_2)]$ belongs to M. Let (τ_1,τ_2') and (τ_1,τ_2'') be two pairs from M that are in $[(\tau_1,\tau_2)]$. Then, by the definition of $\sim$, truth assignments τ_2' and τ_2'' agree on $X_{t_2}\setminus\langle\chi_v(t_2)\rangle = X_t\setminus\langle\chi_v(t)\rangle$. By the definition of M, we have $\tau_1|_{\langle\chi_v(t)\rangle}=\tau_2'|_{\langle\chi_v(t)\rangle}$ and $\tau_1|_{\langle\chi_v(t)\rangle}=\tau_2''|_{\langle\chi_v(t)\rangle}$. Consequently, $\tau_2'|_{\langle\chi_v(t)\rangle}=\tau_2''|_{\langle\chi_v(t)\rangle}$. Hence we find that $\tau_2'=\tau_2''$. This proves that at most one pair from $[(\tau_1,\tau_2)]$ belongs to M.

We conclude that exactly one pair from $[(\tau_1,\tau_2)]$ belongs to M. Then the result follows from combining this fact with equations (2) and (3). □

As we will explain later in more detail, our algorithm is based on repeatedly applying Lemmas 7–10. In order to this, we must be able to check in polynomial time whether a pair (α,θ) belongs to $\mathscr{S}_t$ for some $t\in V_T$. The first statement of the next lemma deals with this. Recall that ℓ is the length of F and that k is the modular incidence treewidth of F.

Lemma 11 *Let $t\in V_T$. Let χ_t and $n_{t'}(\alpha',\theta')$ be given for all $(\alpha',\theta')\in\mathscr{S}_{t'}$ and all children t' of t. Then it is possible in $O(\ell^{2k+2})$ time to check whether a pair (α,θ) with $\alpha(X)\in\mathscr{P}_{F^X}$ for all $X\in\chi_v(t)$ and $\theta(\mathcal{C})\in\mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C}\in\chi_c(t)$ is in $\mathscr{S}_t$, and if so, to compute $n_t(\alpha,\theta)$.*

Proof Let $t\in V_T$, and let (α,θ) be a pair with $\alpha(X)\in\mathscr{P}_{F^X}$ for all $X\in\chi_v(t)$ and $\theta(\mathcal{C})\in\mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C}\in\chi_c(t)$. We distinguish four cases depending on whether t is a leaf, introduce, forget or join node of T. If t is an introduce or forget node, then we split the corresponding case

into two subcases. For all cases except for Case 1 and Case 2b, which both require some extra consideration, we show that we may simply compute the expression for $n_t(\alpha, \theta)$ as prescribed by a corresponding lemma (one of Lemmas 8–10) which either tells us that $(\alpha, \theta) \notin \mathscr{S}_t$ (if the expression has a value of 0) or gives us $n_t(\alpha, \theta)$ otherwise.

Case 1. t is a leaf node.
We claim that $(\alpha, \theta) \in \mathscr{S}_t$ if and only if the following four conditions hold:

(i) $2^{|X|} - |F^X| \geq 1$ for all $X \in \chi_v(t)$ with $\alpha(X) = \emptyset$.
(ii) $\theta(\mathcal{C}) = \mathcal{C}$ for all $\mathcal{C} \in \chi_c(t)$ with no variable modules in $\chi_v(t)$.
(iii) $\theta(\mathcal{C}) = \emptyset$ for all $\mathcal{C} \in \chi_c(t)$ with a variable module $X \in \chi_v(t)$ for which $\alpha(X) = \emptyset$.
(iv) $\theta(\mathcal{C}) = \mathbf{select}(\mathcal{C}, \alpha(X_1) \cdots \alpha(X_p))$ for all $p \geq 1$ and all $\mathcal{C} \in \chi_c(t)$ with exactly p variable modules $X_1, \ldots, X_p$ from $\chi_v(t)$, for which in addition $\alpha(X_i) \neq \emptyset$ for $i = 1, \ldots, p$.

In order to see this, first suppose that $(\alpha, \theta) \in \mathscr{S}_t$. Then there exists a truth assignment $\tau \in N_t(\alpha, \theta)$ by definition. Let $X \in \chi_v(t)$ with $\alpha(X) = \emptyset$. Then $F^X(\tau) = \alpha(X) = \emptyset$, which means that $2^{|X|} - |F^X| \geq 1$. Hence, condition (i) holds. Because t is a leaf node, $\chi_v(t) = \mathcal{X}_t$. Hence, $\tau = \tau|_{\langle \chi_v(t) \rangle}$, which implies that $\theta(\mathcal{C}) = \mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle})$ for all $\mathcal{C} \in \chi_c(t)$. Then, by Lemma 6, conditions (ii)–(iv) must hold as well.

Now suppose that conditions (i)–(iv) are satisfied. We choose a truth assignment $\tau : X_t \to \{0, 1\}$ with $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t) = \mathcal{X}_t$. This is possible due to the following reasons. First, $\alpha(X) \in \mathscr{P}_{F^X}$ either means that $\alpha(X) = \emptyset$ or $\alpha(X) = \mathbf{clause}(\tau_X)$ for some truth assignment $\tau_X : X \to \{0, 1\}$ due to Lemma 4. Hence, $\tau = \tau_X$ on those $X \in \chi_v(t)$ with $\alpha(X) \neq \emptyset$, whereas we can choose τ such that $F^X(\tau) = \emptyset$ on variable modules $X \in \chi_v(t)$ with $\alpha(X) = \emptyset$ due to condition (i). Because $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$, we may apply Lemma 6. This lemma together with conditions (ii)-(iv) and the fact that $\tau = \tau|_{\langle \chi_v(t) \rangle}$ implies that $\mathcal{C}(\tau) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t)$. We conclude that $\tau \in N_t(\alpha, \theta)$, and consequently, $(\alpha, \theta) \in \mathscr{S}_t$.

As we are given χ_t, we can check conditions (i)–(iv) in time $O(\ell^2) \subseteq O(\ell^{2k+2})$. Suppose that $(\alpha, \theta) \in \mathscr{S}_t$. If we have stored the values $2^{|X|} - |F^X|$ for all $X \in \chi_v(t)$ with $\alpha(X) = \emptyset$ while checking condition (i), then we can compute $n_t(\alpha, \theta)$ in time $O(\ell) \subseteq O(\ell^{2k+2})$ due to Lemma 7.

Case 2a. t is a node that introduces a variable module X.
Let t' be the (only) child of t. In Lemma 8 (i) we defined the set $\mathscr{T} = \{\, \theta^* \mid (\alpha', \theta^*) \in \mathscr{S}_{t'}$ and $\theta^*(\mathcal{C}) \cap g(\mathcal{C}, S) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t) \,\}$. In its proof we introduced the set $E = \{\sigma : X \to \{0, 1\} \mid F^X(\sigma) = \alpha(X)\}$. There, we also let $M = E \times \bigcup_{\theta^* \in \mathscr{T}} N_{t'}(\alpha', \theta^*)$ and showed that $n_t(\alpha, \theta) = |M|$ if $(\alpha, \theta) \in \mathscr{S}_t$. Moreover, if $(\tau_X, \tau') \in M$, then in Claim 2 of the proof of Lemma 8 (i) we showed that the truth assignment $\tau = \tau_X \cup \tau'$ belongs to $N_t(\alpha, \theta)$. Hence, in order to check whether $(\alpha, \theta) \in \mathscr{S}_t$ and if so to compute $n_t(\alpha, \theta)$, we only have to compute the size of M, which is given by the righthand side of the equality in Lemma 8 (i). By inequality (1) and because $\mathscr{T} \subseteq \mathscr{S}_{t'}$, we have that $|\mathscr{T}| \leq |\mathscr{S}_{t'}| \leq (|F| + 1)^{k+1}$. Then, because we are given $n_{t'}(\alpha', \theta')$ for all $(\alpha', \theta') \in \mathscr{S}_{t'}$, we can apply Lemma 8 (i) in time $O(\ell + (|F| + 1)^{k+1}) \subseteq O(\ell^{2k+2})$.

Case 2b. t is a node that introduces a clause module $\mathcal{C}$.
Let t' be the (only) child of t, and let θ' is the restriction of θ to $\chi_c(t')$. We claim that $(\alpha, \theta) \in \mathscr{S}_t$ if and only if the following four conditions hold:

(i) $(\alpha, \theta') \in \mathscr{S}_{t'}$.
(ii) $\theta(\mathcal{C}) = \mathcal{C}$ if $\mathcal{C}$ has no variable modules in $\chi_v(t)$.
(iii) $\theta(\mathcal{C}) = \emptyset$ if $\mathcal{C}$ has a variable module $X \in \chi_v(t)$ for which $\alpha(X) = \emptyset$.
(iv) $\theta(\mathcal{C}) = \mathbf{select}(\mathcal{C}, \alpha(X_1) \cdots \alpha(X_p))$ if $\mathcal{C}$ has exactly $p \geq 1$ variable modules $X_1, \ldots, X_p$ from $\chi_v(t)$, for which in addition $\alpha(X_i) \neq \emptyset$ for $i = 1, \ldots, p$.

In order to see this, first suppose that $(\alpha, \theta) \in \mathscr{S}_t$. Let τ be a truth assignment in $N_t(\alpha, \theta)$. Then $\tau \in N_{t'}(\alpha, \theta')$, as observed in the proof of Lemma 8 (ii) as well. Then, by the definition of a tree decomposition, $\mathcal{C}$ contains no variable modules from $\mathcal{X}_t \setminus \chi_v(t)$. This, together with our assumption that $(\alpha, \theta) \in \mathscr{S}_t$ means that $\theta(\mathcal{C}) = \mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle})$, whereas $(\alpha, \theta) \in \mathscr{S}_t$ also implies

that $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$. Hence, we may apply Lemma 6, which together with $\theta(\mathcal{C}) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle})$ gives us conditions (ii)–(iv).

Now suppose that conditions (i)-(iv) hold. By condition (i), there exists a truth assignment $\tau \in N_{t'}(\alpha, \theta')$. Note that τ is defined on $X_{t'} = X_t$. As shown in the proof of Lemma 8 (ii), we have that $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$ and $\mathcal{C}'(\tau) = \theta(\mathcal{C}')$ for all $\mathcal{C}' \in \chi_c(t) \setminus \{\mathcal{C}\}$, and moreover that τ satisfies $F_t \setminus \langle \chi_c(t) \rangle$. Recall that by the definition of a tree decomposition, $\mathcal{C}$ contains no variable modules from $\mathcal{X}_t \setminus \chi_v(t)$. Hence, $\mathcal{C}(\tau) = \mathcal{C}(\tau|_{\langle \chi_v(t) \rangle})$. Then, because $F^X(\tau) = \alpha(X)$ for all $X \in \chi_v(t)$, we may use Lemma 6 together with conditions (ii)–(iv) to obtain $\mathcal{C}(\tau) = \theta(\mathcal{C})$. We conclude that $\tau \in N_t(\alpha, \theta)$, and hence, $(\alpha, \theta) \in \mathscr{S}_t$.

We can check conditions (ii)-(iv) in time $O(\ell) \subseteq O(\ell^{2k+2})$. The same holds for condition (i), because we are given $n_{t'}(\alpha, \theta')$ in case $(\alpha, \theta') \in \mathscr{S}_{t'}$. Moreover, if we find that $(\alpha, \theta) \in \mathscr{S}_t$, then we only have to set $n_t(\alpha, \theta) = n_{t'}(\alpha, \theta')$ due to Lemma 8 (ii).

Case 3a. t is a node that forgets a variable module X.
Let t' be the (only) child of t. Recall that in the proof of Lemma 9 (i) we defined the set $M = \bigcup_{\Pi \in \mathscr{P}_{F^X}} N_{t'}(\alpha \cup \{(X, \Pi)\}, \theta)$ and showed that $|M| = \sum_{\Pi \in \mathscr{P}_{F^X}} n_{t'}(\alpha \cup \{(X, \Pi)\}, \theta)$. Moreover, we proved that $\tau \in N_t(\alpha, \theta)$ if and only if $\tau \in M$. Hence, in order to check whether $(\alpha, \theta) \in \mathscr{S}_t$ and if so to compute $n_t(\alpha, \theta)$, we only have to compute $|M|$. Because $|\mathscr{S}_{t'}| \leq (|F| + 1)^{k+1}$ due to equality (1), this can be done in time $O((|F| + 1)^{k+1}) = O(\ell^{2k+2})$.

Case 3b. t is a node that forgets a clause module $\mathcal{C}$.
Let t' be the (only) child of t. In the proof of Lemma 9 (ii) we showed that $\tau \in N_t(\alpha, \theta)$ if and only if $\tau \in N_{t'}(\alpha, \theta \cup \{(\mathcal{C}, \emptyset)\})$. Hence, in order to check whether $(\alpha, \theta) \in \mathscr{S}_t$ and if so to compute $n_t(\alpha, \theta)$, we only have to test if $n_{t'}(\alpha, \theta \cup \{(\mathcal{C}, \emptyset)\})$ is given to us. This can be done in constant time.

Case 4. t is a join node.
Let t_1 and t_2 be the two children of t. In the proof of Lemma 10 we defined the set $\mathscr{T}_{1,2} = \{(\theta_1, \theta_2) \mid (\alpha, \theta_1) \in \mathscr{S}_{t_1}, (\alpha, \theta_2) \in \mathscr{S}_{t_2}$, and $\theta_1(\mathcal{C}) \cap \theta_2(\mathcal{C}) = \theta(\mathcal{C})$ for all $\mathcal{C} \in \chi_c(t)\}$. We also defined the set $M' = \bigcup_{(\theta_1, \theta_2) \in \mathscr{S}_{1,2}} (N_{t_1}(\alpha, \theta_1) \times N_{t_2}(\alpha, \theta_2))$ as well as the set $M = \{(\tau_1, \tau_2) \in M' \mid \tau_1|_{\langle \chi_v(t) \rangle} = \tau_2|_{\langle \chi_v(t) \rangle}\}$ and proved that $n_t(\alpha, \theta) = |M|$ if $(\alpha, \theta) \in \mathscr{S}_t$. Moreover, if $(\tau_1, \tau_2) \in M$, then in Claim 2 of the proof of Lemma 10 we showed that the truth assignment $\tau = \tau_1 \cup \tau_2$ belongs to $N_t(\alpha, \theta)$. Hence, in order to check whether $(\alpha, \theta) \in \mathscr{S}_t$ and if so to compute $n_t(\alpha, \theta)$, we only have to compute the size of M, which was shown to be equal to $\frac{1}{\beta} \sum_{(\theta_1, \theta_2) \in \mathscr{T}_{1,2}} n_{t_1}(\alpha, \theta_1) \cdot n_{t_2}(\alpha, \theta_2)$, where $\beta = \prod_{X \in \alpha^{-1}(\emptyset)} (2^{|X|} - |F^X|)$. Because $|\mathscr{S}_{t_1}| \leq (|F| + 1)^{k+1}$ and $|\mathscr{S}_{t_2}| \leq (|F| + 1)^{k+1}$ due to equality (1) and β can be computed in $O(\ell^2)$ time, this takes time $O(\ell^2 + (|F| + 1)^{2k+2}) = O(\ell^{2k+2})$. □

We are now ready to present the proof of our main result, which we restate below.

Theorem 1 #SAT *can be solved in time $O(\ell^{3k+4})$ on CNF formulas that have modular incidence treewidth at most k and length ℓ.*

Proof (of Theorem 1) Let F be a formula with modular incidence treewidth at most k. We first construct $I(F)$ and then contract all its modules in order to obtain $I^*(F)$. This can be done in $O(\ell^2)$ time. By using Bodlaender's algorithm [2] we obtain in $O(\ell^2)$ time a tree decomposition of $I^*(F)$ of width at most k. Recall that Kloks [14] showed that such a tree decomposition can be converted in $O(\ell^2)$ time to a nice tree decomposition (T, χ, r) of width at most k that has at most $4|V_{I^*(F)}| \leq 4\ell$ nodes. Also recall that the desired output is

$$n = \sum_{(\alpha, \theta) \in \mathscr{S}_r^*} n_r(\alpha, \theta),$$

where $\mathscr{S}_r^*$ consists of those shapes $(\alpha, \theta) \in \mathscr{S}_r$ with $\theta(\mathcal{C}) = \emptyset$ for all $\mathcal{C} \in \chi_c(r)$. In order to compute n, we compute the sizes $n_t(\alpha, \theta)$ for all $t \in V_T$. Here we follow a bottom-up approach

starting at the leaves. This approach enables us to use Lemma 11, which we apply for each node $t \in V_T$ and each (α, θ) with $\alpha(X) \in \mathscr{P}_{F^X}$ for all $X \in \chi_v(t)$ and $\theta(\mathcal{C}) \in \mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C} \in \chi_c(t)$ and which takes $O(\ell^{2k+2})$ time per call. As soon as we are of distance 2 from a node t, we can forget the size $n_t(\alpha, \theta)$.

For all $t \in V_t$, equality (1) provides an upper bound of $(|F| + 1)^k$ on the number of pairs (α, θ) with $\alpha(X) \in \mathscr{P}_{F^X}$ for all $X \in \chi_v(t)$ and $\theta(\mathcal{C}) \in \mathscr{P}_{(\mathcal{C},X_t)}$ for all $\mathcal{C} \in \chi_c(t)$. This upper bound, together with the $O(\ell^{2k+2})$ time for each call to Lemma 11, means that it takes at most time $O((|F|+1)^{k+1}\ell^{2k+2}) \subseteq O(\ell^{3k+3})$ in total to compute the sizes $n_t(\alpha, \theta)$ for a node t. As the total number of nodes is at most 4ℓ, we find that the total running time is at most $O(4\ell \cdot \ell^{3k+3}) \subseteq O(\ell^{3k+4})$. □

4 Conclusion

We proved that #SAT is polynomial-time tractable for any class of formulas of bounded modular incidence treewidth. Modular incidence treewidth combines treewidth and module contraction, a powerful preprocessing technique widely used in combinatorial optimization. As shown, the resulting parameter is incomparable with the most general structural parameters for which #SAT is known to be tractable. With this result, we approach the frontier of tractability from a new direction. On the other side, one can find β-hypertree width and incidence clique-width. It remains open whether #SAT is polynomial-time for classes of formulas for which one of the latter two parameters is bounded. Graphs of small clique-width that do not contain large bipartite subgraphs are known to have small treewidth [3]. This gives us some reason to believe that our techniques may carry over to the case of bounded incidence clique-width.

References

1. Bacchus, F., Dalmao, S., Pitassi, T.: Algorithms and complexity results for #SAT and Bayesian inference. In: 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03), pp. 340–351 (2003)
2. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**(6), 1305–1317 (1996)
3. Courcelle, B.: The monadic second-order logic of graphs XIV: uniformly sparse graphs and edge set quantifications. Theoretical Computer Science **299**(13), 1 – 36 (2003)
4. Courcelle, B., Engelfriet, J., Rozenberg, G.: Handle-rewriting hypergraph grammars. J. of Computer and System Sciences **46**(2), 218–270 (1993)
5. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000)
6. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. Discr. Appl. Math. **108**(1-2), 23–52 (2001)
7. Courcelle, B., Olariu, S.: Upper bounds to the clique-width of graphs. Discr. Appl. Math. **101**(1-3), 77–114 (2000)
8. Fischer, E., Makowsky, J.A., Ravve, E.R.: Counting truth assignments of formulas of bounded tree-width or clique-width. Discr. Appl. Math. **156**(4), 511–529 (2008)
9. Ganian, R., Hlinený, P., Obdrzálek, J.: Better algorithms for satisfiability problems for formulas of bounded rank-width. Fund. Inform. **123**(1), 59–76 (2013)
10. Gomes, C.P., Sabharwal, A., Selman, B.: Model counting. In: A. Biere, M. Heule, H. van Maaren, T. Walsh (eds.) Handbook of Satisfiability, vol. 185, pp. 633–654. IOS Press (2009)
11. Gottlob, G., Pichler, R.: Hypergraphs in model checking: acyclicity and hypertree-width versus clique-width. SIAM J. Comput. **33**(2), 351–378 (2004)
12. Habib, M., Paul, C.: A survey of the algorithmic aspects of modular decomposition. Computer Science Review **4**(1), 41–59 (2010)
13. Kaski, P., Koivisto, M., Nederlof, J.: Homomorphic hashing for sparse coefficient extraction. In: 7th International Symposion on Parameterized and Exact Computation (IPEC 2012) (2012)
14. Kloks, T.: Treewidth: Computations and Approximations. Springer Verlag, Berlin (1994)
15. Ordyniak, S., Paulusma, D., Szeider, S.: Satisfiability of acyclic and almost acyclic CNF formulas. Theoretical Computer Science **481**, 85–99 (2013)
16. Oum, S., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theory Ser. B **96**(4), 514–528 (2006)
17. Pietrzak, K.: On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. J. of Computer and System Sciences **67**(4), 757–771 (2003)

18. Robertson, N., Seymour, P.D.: Graph minors X. Obstructions to tree-decomposition. J. Combin. Theory Ser. B **52**(2), 153–190 (1991)
19. Roth, D.: On the hardness of approximate reasoning. Artificial Intelligence **82**(1-2), 273–302 (1996)
20. Samer, M., Szeider, S.: Algorithms for propositional model counting. J. Discrete Algorithms **8**(1), 50–64 (2010)
21. Sang, T., Beame, P., Kautz, H.A.: Performing bayesian inference by weighted model counting. In: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA, pp. 475–482. AAAI Press / The MIT Press (2005)
22. Szeider, S.: On fixed-parameter tractable parameterizations of SAT. In: E. Giunchiglia, A. Tacchella (eds.) Theory and Applications of Satisfiability, 6th International Conference, SAT 2003, Selected and Revised Papers, *Lecture Notes in Computer Science*, vol. 2919, pp. 188–202. Springer Verlag (2004)
23. Valiant, L.G.: The complexity of computing the permanent. Theoretical Computer Science **8**(2), 189–201 (1979)

A The Proof of Theorem 2

Recall that we have put the proof of Theorem 2 in this appendix, because the W[1]-hardness reduction is exactly the same as the reduction in the proof of the result from [15] that states that SAT is W[1]-hard when parameterized by the β-hypertree width. To be more precise, only Claim 1 in the proof of Theorem 2 is new.

Theorem 2 SAT *is* W[1]*-hard, when parameterized by the modular incidence treewidth of the input formula.*

Proof A *clique* in a graph is a subset of vertices that are mutually adjacent. A k-partite graph is *balanced* if its k partition classes are of the same size. A *partitioned clique* of a balanced k-partite graph $G = (V_1, \ldots, V_k, E)$ is a clique K with $|K \cap V_i| = 1$ for $i = 1 \ldots, k$. We devise a parameterized reduction from the following problem, which is W[1]-complete [17].

PARTITIONED CLIQUE
Instance: A balanced k-partite graph $G = (V_1, \ldots, V_k, E)$.
Parameter: The integer k.
Question: Does G have a partitioned clique?

Before we describe the reduction we introduce some auxiliary concepts. For any three variables z, x_1, x_2, let $F(z, x_1, x_2)$ denote the formula consisting of the clauses

$$\{z, x_1, \overline{x_2}\}, \{z, \overline{x_1}, x_2\}, \{z, \overline{x_1}, \overline{x_2}\}, \{\overline{z}, x_1, x_2\}, \{\overline{z}, \overline{x_1}, \overline{x_2}\}.$$

This formula has exactly three satisfying assignments, corresponding to the vectors 000, 101, and 110. Hence each satisfying assignment sets at most one out of x_1 and x_2 to true, and if one of them is set to true, then z is set to true as well ("$z = x_1 + x_2$"). Taking several instances of this formula we can build a "selection gadget." Let $x_1, \ldots, x_m$ and $z_1, \ldots, z_{m-1}$ be variables. We define $F^{=1}(x_1, \ldots, x_m; z_1, \ldots, z_{m-1})$ as the union of $F(z_1, x_1, x_2)$, $\bigcup_{i=2}^{m-1} F(z_i, z_{i-1}, x_{i+1})$, and $\{\{z_{m-1}\}\}$. Now each satisfying assignment of this formula sets exactly one variable out of $\{x_1, \ldots, x_m\}$ to true, and, conversely, for each $1 \leq i \leq m$ there exists a satisfying assignment that sets exactly x_i to true and all other variables from $\{x_1, \ldots, x_m\}$ to false.

Now we describe the reduction. Let $G = (V_1, \ldots, V_k)$ be a balanced k-partite graph for $k \geq 2$. We write $V_i = \{v_1^i, \ldots, v_n^i\}$. We construct a formula F. As the variables of F we take the vertices of G plus new variables z_j^i for $1 \leq i \leq k$ and $1 \leq j \leq n-1$. We put $F = \bigcup_{i=0}^{k} F_i$ where the formulas F_i are defined as follows: F_0 contains for any $u \in V_i$ and $v \in V_j$ $(i \neq j)$ with $uv \notin E$ the clause $C_{u,v} = \{\,\overline{u}, \overline{v}\,\} \cup \{\,w \mid w \in (V_i \cup V_j) \setminus \{\,u, v\,\}\,\}$; for $i > 0$ we define $F_i = F^{=1}(v_1^i, \ldots, v_n^i; z_1^i, \ldots, z_{n-1}^i)$. To prove the theorem it suffices to show the following two claims.

Claim 1. The modular incidence treewidth of F is at most $\binom{k}{2} + 1$.

For $1 \leq i \leq k$, let $D_1^i, \ldots, D_n^i$ be the vertices in $I(F)$ with $N(D_1^i) = \{z_1^i, v_1^i, v_2^i\}$, $N(D_j^i) = \{z_j^i, z_{j-1}^i, v_{j+1}^i\}$ for $2 \leq j \leq n-1$ and $N(D_n^i) = \{z_{n-1}^i\}$. In $I(F)$, the set of vertices $C_{u,v}$ can be partitioned into modules $\mathcal{C}_1 \ldots, \mathcal{C}_m$, where $m \leq \binom{k}{2}$. By deleting these modules, we obtain a graph $I'(F)$ that consists of k connected components corresponding to the subgraphs of $I(F)$ induced by $\{v_1^i, \ldots, v_n^i, z_1^i, \ldots, z_{n-1}^i, D_1^i, \ldots, D_n^i\}$ for $1 \leq i \leq k$. Note that these components are trees, so the treewidth of $I'(F)$ is 1. Thus the graph obtained from $I'(F)$ by contracting modules has treewidth 1. We can turn the corresponding tree decomposition into a tree decomposition of $I^*(F)$ that has width $m + 1 \leq \binom{k}{2} + 1$ by simply adding the set of m clause modules $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ to each bag.

Claim 2. G has a partitioned clique if and only if F is satisfiable.

To prove Claim 2 we first suppose that G has a partitioned clique K. We define a partial truth assignment $\tau : V \to \{0, 1\}$ by setting $\tau(v) = 1$ for $v \in K$, and $\tau(v) = 0$ for $v \notin K$. This partial assignment satisfies F_0, and it is easy to extend τ to a satisfying truth assignment of F. Conversely, suppose that F has a satisfying truth assignment τ. Because of the formulas F_i, $1 \leq i \leq k$, τ sets exactly one variable $v_{j_i}^i \in V_i$ to true. Let $K = \{v_{j_1}^1, \ldots, v_{j_k}^k\}$. The clauses in F_0 ensure that $v_{j_i}^i$ and $v_{j_{i'}}^{i'}$ are adjacent in G for each pair $1 \leq i < i' \leq k$, hence K is a partitioned clique of G. This proves Claim 2. □

B The Missing Proof inside Example 2

Recall that in Example 2 we defined a formula ψ_m for $m \geq 1$ as follows. We let $x_1, \ldots, x_m, y_1, \ldots, y_m$ be $2m$ distinct variables. Then we let ψ_m consist of the clauses C_i for $1 \leq i \leq m$ where $C_i = \{y_i, x_1, \ldots, x_m\}$, along with m singleton clauses $\{x_1\}, \ldots, \{x_m\}$. In this appendix we show the following result; note that all necessary terminology not stated in Section 2 is defined inside the proof.

Proposition 4 *For all $m \geq 1$, the signed incidence clique-width of ψ_m is at most 4.*

Proof Let C be a finite set. A *C-labeled graph* is a pair (G, λ) where G is a directed graph and λ is a mapping $\lambda : V(G) \to C$, called a *C-labeling of G*. Let $\mathbf{G} = (G, \lambda)$ be a C-labeled graph and let $a, b \in C$ with $a \neq b$. We define the following two unary operations on labeled graphs:

- We let $\alpha_{a,b}(\mathbf{G}) = (G', \lambda)$, where G' is the graph obtained from G by introducing an arc (v, w) for any pair of vertices $v, w \in V(G)$ such that $\lambda(v) = a$ and $\lambda(w) = b$.
- We let $\rho_{a \to b}(\mathbf{G}) = (G, \lambda')$, where $\lambda'(v) = b$ if $\lambda(v) = a$ and $\lambda'(v) = \lambda(v)$ otherwise, for any $v \in V(G)$.

Moreover, for a set of vertices $U \subseteq V(G)$, we let $\mathbf{G}[U]$ denote the C-labeled graph $(G[U], \lambda|_U)$, where $\lambda|_U$ denotes the restriction of λ to U. The disjoint union $\mathbf{G_1} \oplus \mathbf{G_2}$ of two C-labeled graphs $\mathbf{G_1}$ and $\mathbf{G_2}$ is defined in the obvious way.

Let $\mathbf{G} = (G, \lambda)$ and $\mathbf{G'} = (G', \lambda')$ be C-labeled graphs. We write $\mathbf{G} \to \mathbf{G'}$ if $V(G') = V(G)$ and one of the following conditions holds:

1. $\mathbf{G} = \mathbf{G_1} \oplus \mathbf{G_2}, \mathbf{G'} = \mathbf{G_1} \oplus \rho_{a \to b}(\mathbf{G_2})$ or
2. $\mathbf{G} = \mathbf{G_1} \oplus \mathbf{G_2}, \mathbf{G'} = \mathbf{G_1} \oplus \alpha_{a,b}(\mathbf{G_2})$,

where $\mathbf{G_1}, \mathbf{G_2}$ are C-labeled graphs and $a, b \in C$ with $a \neq b$. A *C-construction* [7] of a labeled graph $\mathbf{G}$ is a sequence $\mathbf{G_0}, \mathbf{G_1}, \ldots, \mathbf{G_n}$ of C-labeled graphs satisfying the following properties:

1. $\mathbf{G_0} = (G_0, \lambda_0)$ such that $E(G_0) = \emptyset$;
2. $\mathbf{G_i} \to \mathbf{G_{i+1}}$ for $0 \leq i < n$;
3. $\mathbf{G_n} = \mathbf{G}$.

By Proposition 3.1 of [7] a directed labeled graph $\mathbf{G}$ has directed clique-width at most k if and only if $\mathbf{G}$ has a C-construction for some set C of cardinality k.

Note that the signed incidence graph of a formula is a directed graph and that the signed incidence clique-width of a formula is equal to the directed clique-width of the signed incidence graph. In line with [7], we view unlabeled (directed) graphs as labeled graphs where all vertices have the same label. We give a C-construction of (I_m, λ_1), where I_m denotes the signed incidence graph of ψ_m, λ_1 is a C-labeling of I_m such that $\lambda_1(v) = 1$ for each $v \in V(I_m)$, and $C = \{1, 2, 3, 4\}$. Recall that $\psi_m = \{ C_i, D_i \mid 1 \leq i \leq m \}$, where $C_i = \{y_i, x_1, \ldots, x_m\}$ and $D_i = \{x_i\}$. Accordingly, $V(I_m) = \bigcup_{i=1}^{m} \{x_i, y_i, C_i, D_i\}$ and $E(I_m) = \bigcup_{i=1}^{m} \{(x_i, D_i), (y_i, C_i)\} \cup \{ (x_i, C_j) \mid 1 \leq i, j \leq m \}$.

Let G_0 denote the (directed) graph with $V(G_0) = V(I_m) = V$ and $E(G_0) = \emptyset$, and let λ denote the C-labeling of G_0 given by

$$\lambda(D_i) = 1, \lambda(x_i) = 2, \lambda(C_i) = 3, \lambda(y_i) = 4,$$

for $1 \leq i \leq m$. The graph $\mathbf{G_0} = (G_0, \lambda)$ will be the initial graph in our C-construction. For $1 \leq i \leq m$, the graph $\mathbf{G_i}$ is simply the graph $\mathbf{G_{i-1}}$ with an arc from x_i to D_i. This can be expressed as

$$\mathbf{G_i} = \mathbf{G_{i-1}}[V \setminus \{x_i, D_i\}] \oplus \alpha_{2,1}(\mathbf{G_{i-1}}[\{x_i, D_i\}]). \tag{4}$$

Under the assumption that

$$\mathbf{G_{i-1}} = \mathbf{G_{i-1}}[V \setminus \{x_i, D_i\}] \oplus \mathbf{G_{i-1}}[\{x_i, D_i\}], \tag{5}$$

this implies that $\mathbf{G_{i-1}} \to \mathbf{G_i}$. Condition (5) is satisfied initially since G_0 does not contain any edges, and the definition in (4) ensures that it remains true for $1 \leq i \leq m$.

For $m + 1 \leq i \leq 2m$, we let $\mathbf{G_i}$ be $\mathbf{G_{i-1}}$ with an additional arc from y_i to C_i. That is,

$$\mathbf{G_i} = \mathbf{G_{i-1}}[V \setminus \{y_i, C_i\}] \oplus \alpha_{4,3}(\mathbf{G_{i-1}}[\{y_i, C_i\}]). \tag{6}$$

Again, this implies that $\mathbf{G_{i-1}} \to \mathbf{G_i}$ provided that

$$\mathbf{G_{i-1}} = \mathbf{G_{i-1}}[V \setminus \{y_i, C_i\}] \oplus \mathbf{G_{i-1}}[\{y_i, C_i\}]. \tag{7}$$

It follows from the construction of $\mathbf{G_m}$ and the definition in (6) that condition (7) must be satisfied for each $m + 1 \leq i \leq 2m$. We let $\mathbf{G_{2m+1}} = (G_{2m+1}, \lambda) = \alpha_{2,3}(\mathbf{G_{2m}})$, introducing all arcs (x_i, C_j) with $1 \leq i, j \leq m$. Since $G_{2m+1} = I_m$ and $\mathbf{G_i} \to \mathbf{G_{i+1}}$ for $0 \leq i \leq 2m$ we find that the sequence $\mathbf{G_0}, \mathbf{G_1}, \ldots, \mathbf{G_{2m+1}}$ is a C-construction of (I_m, λ). Finally, we replace redundant labels by setting $\mathbf{G_{2m+2}} = \rho_{2 \to 1}(\mathbf{G_{2m+1}})$, $\mathbf{G_{2m+3}} = \rho_{3 \to 1}(\mathbf{G_{2m+2}})$, and $\mathbf{G_{2m+4}} = \rho_{4 \to 1}(\mathbf{G_{2m+3}})$ to obtain a C-construction of (I_m, λ_1). We conclude that I_m has directed clique-width at most $|C| = 4$. Hence, the signed incidence clique-width of ψ_m (which is equal to the directed clique-width of I_m, as observed above) is at most $|C| = 4$ as well. □