

Counting Linear Extensions: Parameterizations by Treewidth*

Eduard Eiben¹, Robert Ganian^{†2}, Kustaa Kangas³, and Sebastian Ordyniak⁴

1 Algorithms and Complexity Group, TU Wien, Vienna, Austria

2 Algorithms and Complexity Group, TU Wien, Vienna, Austria

3 Helsinki Institute for Information Technology, Department of Computer Science, University of Helsinki, Finland

4 Algorithms and Complexity Group, TU Wien, Vienna, Austria

Abstract

We consider the $\#P$ -complete problem of counting the number of linear extensions of a poset ($\#LE$); a fundamental problem in order theory with applications in a variety of distinct areas. In particular, we study the complexity of $\#LE$ parameterized by the well-known decompositional parameter treewidth for two natural graphical representations of the input poset, i.e., the cover and the incomparability graph. Our main result shows that $\#LE$ is fixed-parameter intractable parameterized by the treewidth of the cover graph. This resolves an open problem recently posed in the Dagstuhl seminar on Exact Algorithms. On the positive side we show that $\#LE$ becomes fixed-parameter tractable parameterized by the treewidth of the incomparability graph.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, G.2.1 Combinatorics

Keywords and phrases Partially ordered sets, Linear extensions, Parameterized Complexity, Structural parameters, Treewidth

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.39

1 Introduction

Counting the number of linear extensions of a poset is a fundamental problem of order theory that has applications in a variety of distinct areas such as sorting [30], sequence analysis [25], convex rank tests [27], sampling schemes of Bayesian networks [28], and preference reasoning [24]. Determining the exact number of linear extensions of a given poset is known to be $\#P$ -complete [6] already for posets of height at least 3. Informally, $\#P$ -complete problems are as hard as counting the number of accepting paths of any polynomial time nondeterministic Turing machine, implying that such problems are not tractable unless $P = NP$. The currently fastest known method for counting linear extensions of a general n -element poset is by dynamic programming over the lattice of downsets and runs in time $\mathcal{O}(2^n \cdot n)$ [10]. Polynomial time algorithms have been found for various special cases such as series-parallel posets [26] and posets whose cover graph is a (poly)tree [2]. Fully polynomial time randomized approximation schemes are known for estimating the number of linear extensions [13, 7].

Due to the inherent difficulty of the problem, it is natural to study whether it can be solved efficiently by exploiting the structure of the input poset. In this respect, the

* The authors acknowledge support by the Austrian Science Fund (FWF, projects P26696 and W1255-N23).

† Robert Ganian is also affiliated with FI MU, Brno, Czech Republic.



parameterized complexity framework [12, 9] allows a refined view of the interactions between various forms of structure in the input and the running time of algorithms. The idea of the framework is to measure the complexity of problems not only in terms of input sizes, but also with respect to an additional numerical parameter. The goal is then to develop so-called *fpt algorithms*, which are algorithms that run in time $f(k)n^{\mathcal{O}(1)}$ where n is the input size and f is a computable function depending only on the parameter k . A less favorable outcome is a so-called XP algorithm, which runs in time $n^{f(k)}$; the existence of such algorithms then gives rise to the respective complexity classes FPT (*fixed-parameter tractable*) and XP.

The first steps in this general direction have been taken, e.g., in [19], using the decomposition diameter as a parameter, in [15] using a parameter called activity for N-free posets, and very recently in [22], where the treewidth of the so-called cover graph was considered as a parameter. Also the exact dynamic programming algorithm [10] can be shown to run in time $\mathcal{O}(n^w \cdot w)$ for a poset with n elements and width w (the size of the largest anti-chain). Interestingly, none of these efforts has so far led to an fpt algorithm.

We believe that this uncertainty about the exact complexity status of counting linear extensions with respect to these various parameterizations is at least partly due to the fact that we deal with a counting problem whose decision version is trivial, i.e., every poset has at least one linear extension. This fact makes it considerably harder to show that the problem is fixed-parameter intractable; in particular, the usual approach for counting problems based on parsimonious reductions (i.e., polynomial time one-one reductions) fails. On the other hand, the same predicament makes studying the complexity of counting linear extensions significantly more interesting, as noted also by Flum and Grohe [16]:

The theory gets interesting with those counting problems that are harder than their corresponding decision versions.

1.1 Results

In this paper we study the complexity of counting linear extensions when the parameter is the treewidth – a fundamental graph parameter which has already found a plethora applications in many areas of computer science [18, 17, 29]. In particular, we settle the fixed-parameter (in)tractability of the problem when parameterizing by the treewidth of two of the most prominent graphical representations of posets, the cover graph (also called the Hasse diagram) and the incomparability graph.

Our main result then provides the first evidence that the problem does not allow for an fpt algorithm parameterized by the treewidth of the cover graph unless $\text{FPT} = \text{W}[1]$. We remark that this complements the XP algorithm of [22] and resolves an open problem recently posed in the Dagstuhl seminar on Exact Algorithms [21]. The result is based on a so-called *fpt turing reduction* from EQUITABLE COLORING parameterized by treewidth [14], and combines a counting argument with a fine-tuned construction to link the number of linear extensions with the existence of an equitable coloring. To the best of our knowledge, this is the first time this technique has been used to show fixed-parameter intractability of a counting problem.

We complement this negative result by obtaining an fpt algorithm for the problem when the parameter is the treewidth of the incomparability graph of the poset. To this end, we use the so-called *combined graph* (also called the cover-incomparability graph [5]) of the poset, which is obtained from the cover graph by adding the edges of the incomparability graph. We employ a special normalization procedure on a decomposition of the incomparability graph to show that the treewidth of the combined graph must be bounded by the treewidth of the

incomparability graph. Once this is established, the result follows by giving a formulation of the problem in *Monadic Second Order Logic* and applying an extension of Courcelle's Theorem for counting.

The paper is organized as follows. Section 2 introduces the required preliminaries and notation. Section 3 is then dedicated to proving the fixed-parameter intractability of the problem when parameterized by the treewidth of the cover graph, and the subsequent Section 4 presents our positive results for the problem. Concluding notes are then provided in Section 5.

2 Preliminaries

For standard terminology in graph theory, such as the notions of a graph, digraph, path, etc. we refer readers to [11]. Given a graph G , we let $V(G)$ denote its vertex set and $E(G)$ its edge set. The (open) neighborhood of a vertex $x \in V(G)$ is the set $\{y \in V(G) : (x, y) \in E(G)\}$ and is denoted by $N(x)$. The closed neighborhood $N[v]$ of x is defined as $N(v) \cup \{v\}$. A path between two disjoint vertex sets $A, B \subseteq V(G)$ is a path with one endpoint in A , one endpoint in B , and all internal vertices disjoint from $A \cup B$. A set $X \subseteq V(G)$ is a separator in G if $G - X$ contains at least two connected components.

We use $[i]$ to denote the set $\{0, 1, \dots, i\}$. The following fact about prime numbers will also be useful later.

► **Fact 1** ([6]). *For any $n \geq 4$, the product of primes strictly between n and n^2 is at least $n!2^n$.*

2.1 Treewidth

A *tree-decomposition* of a graph G is a pair $(T, \mathcal{X} = \{X_t\}_{t \in V(T)})$, where T is a rooted tree whose every vertex t is assigned a vertex subset $X_t \subseteq V(G)$, called a *bag*, such that the following properties hold:

(T1) $\cup_{t \in V(T)} X_t = V(G)$,

(T2) for every $u \in V(G)$, the set $T_u = \{t \in V(T) : u \in X_t\}$ induces a connected subtree of T (*monotonicity*), and

(T3) for each $uv \in E(G)$ there exists $t \in V(T)$ such that $u, v \in X_t$.

To distinguish between the vertices of the tree T and the vertices of the graph G , we will refer to the vertices of T as *nodes*. The *width* of the tree-decomposition \mathcal{T} is $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of G , $\text{tw}(G)$, is the minimum width over all tree-decompositions of G .

A path-decomposition is a tree-decomposition where each node of T has degree at most 2, and the notion of *pathwidth* is then defined analogously to treewidth. A tree-decomposition $\mathcal{T} = (T, \mathcal{X})$ is *nice* if T contains a root r , the root and all leaves have empty bags, and each non-leaf node belongs to one of three categories: **Introduce**, **Forget**, **Join** (see, e.g., [9]). A nice tree-decomposition (path-decomposition) can be obtained from a tree-decomposition (path-decomposition) of the same width in polynomial time [23]. Observe that any path-decomposition can be fully characterized by the order of appearance of its bags along T , and hence we will consider succinct representations of path-decompositions in the form $\mathcal{Q} = (Q_1, \dots, Q_d)$, where Q_i is the i -th bag in \mathcal{Q} .

We list some useful facts about treewidth and pathwidth.

► **Fact 2** ([3, 4]). *There exists an algorithm which, given a graph G and an integer k , runs in time $\mathcal{O}(k^{\mathcal{O}(k^3)}n)$ and either outputs a tree-decomposition of G of width at most k or correctly*

identifies that $\text{tw}(G) > k$. Furthermore, there exists an algorithm which, given a graph G and an integer k , runs in time $\mathcal{O}(k^{\mathcal{O}(k^3)}n)$ and either outputs a path-decomposition of G of width at most k or correctly identifies that $\text{pw}(G) > k$.

► **Fact 3** (Folklore). Let \mathcal{T} be a tree-decomposition of G and $t \in V(T)$. Then each connected component of $G - X_t$ lies in a single subtree of $T - t$. In particular, for each connected component C of $G - X_t$ there exists a subtree T' of $T - t$ such that for each vertex $a \in C$ there exists $t_a \in V(T')$ such that $a \in X_{t_a}$.

We note that if G is a directed graph, then $\text{tw}(G)$ and a tree-decomposition of G refers to the treewidth and a tree-decomposition of the underlying undirected graph of G , i.e., the undirected graph obtained by replacing each directed edge with an edge.

2.2 Monadic Second Order Logic

We consider *Monadic Second Order* (MSO) logic on (edge-)labeled directed graphs in terms of their incidence structure whose universe contains vertices and edges; the incidence between vertices and edges is represented by a binary relation. We assume an infinite supply of *individual variables* x, x_1, x_2, \dots and of *set variables* X, X_1, X_2, \dots . The *atomic formulas* are Vx (“ x is a vertex”), Ey (“ y is an edge”), Ixy (“vertex x is incident with edge y ”), Hxy (“vertex x is the head of the edge y ”), Txy (“vertex x is the tail of the edge y ”), $x = y$ (equality), $x \neq y$ (inequality), P_ax (“vertex or edge x has label a ”), and Xx (“vertex or edge x is an element of set X ”). *MSO formulas* are built up from atomic formulas using the usual Boolean connectives ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), quantification over individual variables ($\forall x, \exists x$), and quantification over set variables ($\forall X, \exists X$).

Let $\Phi(X)$ be an MSO formula with a free set variable X . For a labeled graph $G = (V, E)$ and a set $S \subseteq E$ we write $G \models \Phi(S)$ if the formula Φ holds true on G whenever X is instantiated with S .

The following result (an extension of the well-known Courcelle’s Theorem [8]) shows that if G has bounded treewidth then we can count the number of sets S with $G \models \Phi(S)$.

► **Fact 4** ([1]). Let $\Phi(X)$ be an MSO formula with a free set variable X and w a constant. Then there is a linear-time algorithm that, given a labeled directed graph $G = (V, E)$ of treewidth at most w , outputs the number of sets $S \subseteq E$ such that $G \models \Phi(S)$.

2.3 Posets

A *partially ordered set* (poset) \mathcal{P} is a pair (P, \leq^P) where P is a set and \leq^P is a reflexive, antisymmetric, and transitive binary relation over P . The *size* of a poset $\mathcal{P} = (P, \leq^P)$ is $|\mathcal{P}| := |P|$. We say that p *covers* p' for $p, p' \in P$, denoted by $p' \triangleleft^P p$, if $p' \leq^P p$, $p \neq p'$, and for every p'' with $p' \leq^P p'' \leq^P p$ it holds that $p'' \in \{p, p'\}$. We say that p and p' are *incomparable* (in \mathcal{P}), denoted $p \parallel^P p'$, if neither $p \leq^P p'$ nor $p' \leq^P p$.

A *chain* C of \mathcal{P} is a subset of P such that $x \leq^P y$ or $y \leq^P x$ for every $x, y \in C$. An *antichain* A of \mathcal{P} is a subset of P such that for all $x, y \in A$ it is true that $x \parallel^P y$. A family C_1, \dots, C_ℓ of pairwise disjoint subsets of P forms a *total order* if for each $i, j \in [\ell]$ and each $a \in C_i, b \in C_j$, it holds that $a \leq b$ iff $i < j$. Furthermore, for each $i \in [\ell - 1]$ we say that C_i and C_{i+1} are *consecutive*. We call a poset \mathcal{P} such that every two elements of \mathcal{P} are comparable a *linear order*. A *linear extension* of a poset $\mathcal{P} = (P, \leq^P)$ is a reflexive, antisymmetric, and transitive binary relation \preceq over P such that $x \preceq y$ whenever $x \leq^P y$ and a poset $\mathcal{P}^* = (P, \preceq)$ is a linear order.

We denote the number of linear extensions of \mathcal{P} by $e(\mathcal{P})$. For completeness, we provide a formal definition of the problem of counting the number of linear extensions below.

#LE
Instance: A poset \mathcal{P} .
Task: Compute $e(\mathcal{P})$.

We consider the following graph representations of a poset $\mathcal{P} = (P, \leq^{\mathcal{P}})$. The *cover graph* of \mathcal{P} , denoted $C(\mathcal{P})$, is the directed graph with vertex set P and edge set $\{(a, b) \mid a \triangleleft b\}$. The *incomparability graph* of \mathcal{P} , denoted $I(\mathcal{P})$, is the undirected graph with vertex set P and edge set $\{\{a, b\} \mid a \parallel b\}$. The *combined graph* of \mathcal{P} , denoted $I_C(\mathcal{P})$, is the directed graph with vertex set P and edge set $\{(a, b) \mid (a \triangleleft b) \vee (a \parallel b)\}$; observe that $I_C(\mathcal{P})$ can be obtained by taking disjoint union of the edge sets of $C(\mathcal{P})$ and $I(\mathcal{P})$ and then replacing undirected edges by two directed ones. Finally, the *poset graph* of \mathcal{P} , denoted $P_G(\mathcal{P})$, is the directed graph with vertex set P and edge set $\{(a, b) \mid a \leq b\}$. We will use the following known fact about tree-decompositions and path-decompositions of incomparability graphs.

► **Fact 5** ([20, Theorem 2.1]). *Let \mathcal{P} be a poset. Then $\text{tw}(I(\mathcal{P})) = \text{pw}(I(\mathcal{P}))$.*

► **Corollary 6** (of Fact 2 and 5). *Let \mathcal{P} be a poset and $k = \text{tw}(I(\mathcal{P}))$. Then it is possible to compute a nice path-decomposition \mathcal{Q} of $I(\mathcal{P})$ of width at most k in time $\mathcal{O}(k^{\mathcal{O}(k^3)}n)$.*

2.4 Parameterized Complexity

We refer the reader to [12, 9, 16] for an in-depth introduction to parameterized complexity. In particular, we will need the notions of *parameterized (decision) problem*, the complexity classes $W[1]$ and *FPT*, *fpt algorithm*, and *fpt turing reduction*. Informally, recall that an fpt turing reduction from problem \mathcal{A} to problem \mathcal{B} is an fpt-algorithm that solves \mathcal{A} using an oracle for \mathcal{B} . A *parameterized counting problem* \mathcal{P} is a function $\Sigma^* \times \mathbb{N} \rightarrow \mathbb{N}$ for some finite alphabet Σ . We call a parameterized counting problem \mathcal{P} *fixed-parameter tractable (FPT)* if \mathcal{P} can be computed in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ where f is an arbitrary computable function and (x, k) is the instance. To avoid confusion, we remark that there also exists the complexity class $\#W[1]$ which is an analog of $\#P$ for parameterized counting problems. Our main negative result is based on an fpt turing reduction from the following fairly well-known $W[1]$ -hard decision problem [14].

EQUITABLE COLORING[tw]
Instance: A graph G and an integer r .
Parameter: $\text{tw}(G) + r$.
Question: Does G admit a proper r -coloring such that the number of vertices in any two color classes differ by at most one?

We denote by $\#\text{EC}(G, r)$ the number of equitable colorings of graph G with r colors.

3 Fixed-Parameter Intractability of Counting Linear Extensions

The goal of this section is to prove Theorem 7, stated below.

► **Theorem 7.** *#LE parameterized by the treewidth of the cover graph of the input poset does not admit an fpt algorithm unless $W[1]=\text{FPT}$.*

We begin by giving a brief overview of the proof, whose general outline follows the #P-hardness proof of the problem [6]. However, since our parameter is treewidth, we needed to reduce from a problem that is not fixed-parameter tractable parameterized by treewidth. Consequently, instead of reducing from SAT, we will use EQUITABLE COLORING. This made the reduction considerably more complicated and required the introduction of novel gadgets, which allow us to encode the problem without increasing the treewidth too much.

The proof is based on solving an instance (G, r) of EQUITABLE COLORING[tw] in FPT time using an oracle that solves #LE in FPT time parameterized by the treewidth of the cover graph (i.e., an fpt turing reduction). The first step is the construction of an auxiliary poset $\mathcal{P}(G, r)$ of size $2(r-1)|V(G)| + (r^2-1)|E(G)|$. Then, for a given sufficiently large (polynomially larger than $|V(G)|$) prime number p , we show how to construct a poset $\mathcal{P}(G, r, p)$ such that $e(\mathcal{P}(G, r, p)) \equiv e(\mathcal{P}(G, r)) \cdot \#EC(G, r) \cdot A_p \pmod{p}$, where A_p is a constant that depends on p and is not divisible by p . Therefore, if we choose a prime p that does not divide $e(\mathcal{P}(G, r)) \cdot \#EC(G, r)$, then $e(\mathcal{P}(G, r, p))$ will not be divisible by p . Using Fact 1 we show that if $\#EC(G, r) \neq 0$, then there always exists a prime p within a specified polynomial range of $|V(G)|$ such that p does not divide $e(\mathcal{P}(G, r)) \cdot \#EC(G, r)$.

From the above, it follows that there exists an equitable coloring of G with r colors if and only if, for at least one prime p within a specified (polynomial) number range, the number of linear extensions of $\mathcal{P}(G, r, p)$ is not divisible by p . Moreover, we show that all inputs for the oracle will have size polynomial in the size of G and treewidth bounded by polynomial in $\text{tw}(G) + r$. Before proceeding to a formal proof of Theorem 7, we state two auxiliary lemmas which will be useful for counting linear extensions later in the proof.

► **Lemma 8.** *If a poset \mathcal{P} is a disjoint union of posets $\mathcal{P}_1, \dots, \mathcal{P}_k$ for some positive integer k , then*

$$e(\mathcal{P}) = \frac{(\sum_{i=1}^k |\mathcal{P}_i|)!}{\prod_{i=1}^k |\mathcal{P}_i|!} \prod_{i=1}^k e(\mathcal{P}_i).$$

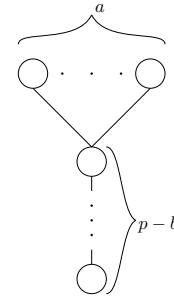
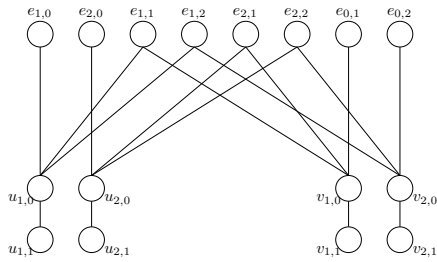
► **Lemma 9.** *Let p be a prime number and \mathcal{Q} be a connected component of poset \mathcal{P} such that $|\mathcal{Q}| = p - 1$. If the number of linear extensions of \mathcal{P} is not divisible by p , then the number of elements in each connected component of \mathcal{P} other than \mathcal{Q} is divisible by p .*

We now proceed to the proof of the theorem.

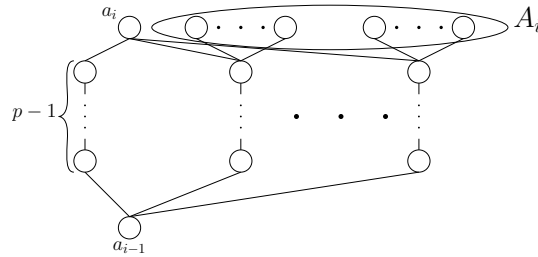
Proof of Theorem 7. The proof is structured into the construction of $\mathcal{P}(G, r)$, the construction of $\mathcal{P}(G, r, p)$, establishing the desired properties of $\mathcal{P}(G, r, p)$ and $\mathcal{P}(G, r)$, and the conclusion.

Construction of $\mathcal{P}(G, r)$ and the main gadget

Let (G, r) be an instance of EQUITABLE COLORING[tw] such that $|V(G)|$ is divisible by r (if this is not the case, then this can be enforced by padding the instance with isolated vertices, see also [14]). We begin by constructing the poset $\mathcal{P}(G, r)$, which will play an important role later on. For every vertex v of $V(G)$ we create $2(r-1)$ elements denoted $v_{i,j}$, where $1 \leq i \leq r-1$ and $j \in \{0, 1\}$, such that the only dependencies in the poset between these elements are $v_{i,1} \leq v_{i,0}$ for all $v \in V(G)$, for all $i \in \{1, \dots, r-1\}$. For every edge $e = uv \in E(G)$ we create $r^2 - 1$ pairwise-incomparable elements $e_{i,j}$, such that $(i, j) \in (\{0, \dots, r-1\}^2 \setminus \{(0, 0)\})$. The dependencies of $e_{i,j}$ are: if $i > 0$ then $v_{i,0} \leq e_{i,j}$, and if $j > 0$ then $v_{j,0} \leq e_{i,j}$ (see also Fig. 1).



■ **Figure 1** The cover graph for an edge $e = uv$ in $\mathcal{P}(G, 3)$. ■ **Figure 2** An (a, b) -flower.



■ **Figure 3** Each level consists of a chain of length $p - 1$ and a few flowers. The set of petals associated with level L_i is denoted by A_i .

Let us now fix a prime number p such that p does not divide $e(\mathcal{P}(G, r))$ and $p > 2r|V(G)| + r^2|E(G)|$. The main gadget in our reduction is a so-called (a, b) -flower, which consists of an antichain of a vertices (called the *petals*) covering a chain of $p - b$ elements (called the *stalk*); see Fig. 2. Due to Lemma 9, (a, b) -flowers will later allow us to force a choice of exactly b vertices out of a .

Construction of $\mathcal{P}(G, r)$

Let G be a graph, r be an integer and p be a prime number as above. Recall that $|V(G)|$ is divisible by r and let $s = \frac{|V(G)|}{r}$ (note that this implies that each color in an equitable coloring of G must occur precisely s times in G). We proceed with a description of the poset $\mathcal{P}(G, r, p)$. The poset $\mathcal{P}(G, r, p)$ is split into $r + 3$ “levels” L_1, \dots, L_{r+3} by linearly ordered elements $a_0 \leq a_1 \leq \dots \leq a_{r+2} \leq a_{r+3}$, called the *anchors*. Each of these levels, besides L_{r+3} , will consist of some flowers and a chain of $p - 1$ elements which we call a *stick*; each of these flowers and the stick will always be pairwise incomparable. The anchors a_0 and a_{r+3} are the unique minimum and maximum elements, respectively. The stick and all the stalks of flowers in level L_i will always lie between two consecutive elements a_{i-1} and a_i , and the petals of these flowers will be incomparable with a_i as well as some anchors above that (as defined later). Observe that while the relative position of any stalk and any anchor is fixed in every linear extension, petals can be placed above a_i .

We say that a flower (or its stalk, petals, or elements) is *associated* with the level in which it is constructed, i.e., with the level L_i such that $a_{i-1} \leq c \leq a_i$ for stalk elements c and $a_{i-1} \leq d$ and $d \parallel a_i$ for petals d . We denote the set of all petals associated with level L_i as A_i (see Fig. 3).

For the construction, it will be useful to keep in mind the following intended goal: whenever an (a, b) -flower is placed in level i , it will force the selection of precisely b petals (from its total of a petals), where selected elements remain on level i (i.e., between a_{i-1} and a_i) in the linear extension and unselected elements are moved to level $r + 2$ (i.e., between

a_{r+2} and a_{r+3}) in the linear extension. We will later show that the total number of linear extensions which violate this goal must be divisible by p , and hence such extensions can all be disregarded modulo p .

The first r levels are so-called *color class* levels, each representing one color class. We use these levels to make sure that every color class contains exactly s vertices. Aside from the stick, each such level contains a single $(|V(G)|, s)$ -flower. Recall that the stalk and the stick on level $1 \leq i \leq r$ both lie between anchors a_{i-1} and a_i , and that the stick and the flower are incomparable. We associate each petal of the flower at level L_i with a unique vertex $v \in V(G)$ and denote the petal v_i . Each petal v_i will be incomparable with all anchors above a_{i-1} up to a_{r+3} , i.e., $v_i \parallel a_j$ for $i \leq j \leq r+2$ and $v_i \leq a_{r+3}$. Intuitively, the flower in each color class level will later force a choice of s vertices to be assigned the given color.

Level L_{r+1} is called the *vertex* level and consists of one stick and $|V(G)|$ -many $(r, 1)$ -flowers; the purpose of this level is to ensure that every vertex is assigned exactly one color. Each flower is associated with one vertex $v \in V(G)$ and we denote the petals of the flower associated with vertex v as v^i for $1 \leq i \leq r$. We set $v_i \leq v^i$ for all $v \in V(G)$ and $1 \leq i \leq r$.

Level L_{r+2} is called the *edge* level, and its purpose is to ensure that the endpoints of every edge have a different color. It consists of a stick and $|E(G)|$ -many $(r^2, 1)$ -flowers. Each flower is associated with one edge $e = uv \in E(G)$ and we denote the petals of the flower associated with e as $e_{i,j}$ for $1 \leq i \leq r$ and $1 \leq j \leq r$. Moreover, for edge $e = uv$ we set $u^i \leq e_{i,j}$, $v^j \leq e_{i,j}$, and we set $a_{r+2} \leq e_{i,j}$ whenever $i = j$. Observe that this forces any petal $e_{i,i}$ to lie between a_{r+2} and a_{r+3} in every linear extension (i.e., prevents $e_{i,i}$ from being “selected”).

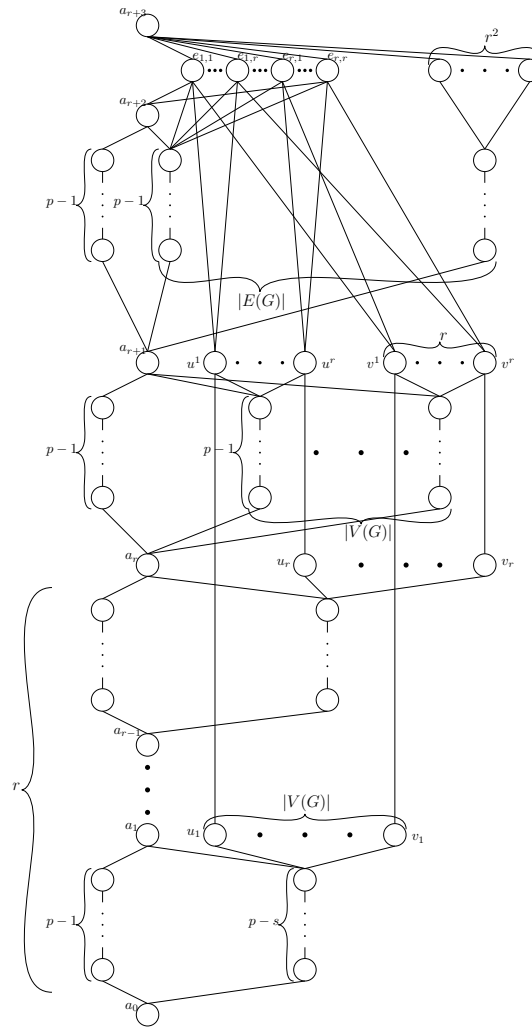
Level L_{r+3} is called the *trash level*. It does not contain any new elements in the poset, but it plays an important role in the reduction: we will later show that any petals which are interpreted as “not selected” must be located between a_{r+2} and a_{r+3} in any linear extension that is not automatically “canceled out” due to counting modulo p .

A high-level overview of the whole constructed poset $\mathcal{P}(G, r, p)$ is presented in Fig. 4.

Establishing the desired properties of $\mathcal{P}(G, r, p)$ and $\mathcal{P}(G, r)$

We begin by formalizing the notion of selection. Let a *configuration* be a partition ϕ of petals of all flowers into $r+3$ sets $L_1^\phi, \dots, L_{r+3}^\phi$. Let Φ denote a set of all configurations. We say that a linear extension \preceq of $\mathcal{P}(G, r, p)$ *respects* the configuration ϕ if $L_1^\phi \preceq a_1 \preceq L_2^\phi \preceq a_2 \preceq \dots \preceq a_{r+2} \preceq L_{r+3}^\phi$ and we denote the set of all linear extensions of $\mathcal{P}(G, r, p)$ that respects ϕ by \mathcal{L}^ϕ . We say that a configuration ϕ is *consistent* if \mathcal{L}^ϕ is non-empty; this merely means that $L_1^\phi \leq a_1 \leq L_2^\phi \leq a_2 \leq \dots \leq a_{r+2} \leq L_{r+3}^\phi$ does not violate any inequalities in $\mathcal{P}(G, r, p)$. Observe that if ϕ is consistent, then \mathcal{L}^ϕ is exactly the set of linear extension of the partial order $\mathcal{P}^\phi(G, r, p)$, where $\mathcal{P}^\phi(G, r, p)$ is obtained by enriching $\mathcal{P}(G, r, p)$ with the relations $L_1^\phi \leq a_1 \leq L_2^\phi \leq a_2 \leq \dots \leq a_{r+2} \leq L_{r+3}^\phi$ and performing transitive closure (in other words, $\mathcal{P}^\phi(G, r, p)$ is obtained by enforcing ϕ onto $\mathcal{P}(G, r, p)$).

Since every linear extension of $\mathcal{P}(G, r, p)$ respects exactly one configuration, it is easy to see that $e(\mathcal{P}(G, r, p)) = \sum_{\phi \in \Phi} |\mathcal{L}^\phi| = \sum_{\phi \in \Phi} e(\mathcal{P}^\phi(G, r, p))$. Intuitively, a configuration ϕ *contributes* to the above sum modulo p if $e(\mathcal{P}^\phi(G, r, p))$ is not divisible by p . We shall prove that the only configurations which contribute to this sum modulo p are those where from every (a, b) -flower there are exactly b petals in the same level as the stalk, and the remaining $a - b$ petals are in the trash. Furthermore, in each configuration ϕ which contributes to the above sum modulo p , the petals in L_{r+1}^ϕ represent a proper equitable coloring of G with r colors, and each such configuration is respected by the same number of linear extensions.



■ **Figure 4** The cover graph of $\mathcal{P}(G, r, p)$. The edge e is the edge in G between vertices u and v .

Let us first remark that for any configuration ϕ , the anchors a_0, a_1, \dots, a_{r+3} are comparable to all elements of $\mathcal{P}^\phi(G, r, p)$. Now, let $\mathcal{P}_{L_i}^\phi$ be the poset induced by all elements $e \in \mathcal{P}^\phi(G, r, p)$ such that $a_{i-1} \leq e \leq a_i$. It is readily seen that $e(\mathcal{P}^\phi(G, r, p)) = \prod_{i=1}^{r+3} e(\mathcal{P}_{L_i}^\phi)$. We proceed by stating a series of claims about our construction.

► **Claim 10.** For each $i \in \{1, \dots, r\}$, it holds that either $e(\mathcal{P}_{L_i}^\phi) \equiv 0 \pmod p$, or $e(\mathcal{P}_{L_i}^\phi) = s! \binom{2p-1}{p}$ and L_i^ϕ contains exactly s petals of A_i and no other petals.

Proof of the claim. Assume that $e(\mathcal{P}_{L_i}^\phi) \not\equiv 0 \pmod p$ and recall that level L_i contains a stick, which is a chain of $p - 1$ elements that is incomparable with all elements of $\mathcal{P}_{L_i}^\phi$ in every configuration ϕ . By Lemma 9 this implies that every connected component of $\mathcal{P}_{L_i}^\phi$ has size divisible by p . Clearly, L_i^ϕ contains only those stalks that are associated with the level L_i , and it contains all such stalks. It is readily seen from the construction that any petal in $\cup_{j < i} A_j$ would necessarily form a component of size one in $\mathcal{P}_{L_i}^\phi$. Hence, $\mathcal{P}_{L_i}^\phi$ contains only elements associated with level L_i , namely elements of the chain with $p - 1$ vertices and elements of a $(|V(G)|, s)$ -flower. Moreover, by Lemma 9 and the fact that $|V(G)| + p - s < 2p$,

each such flower has exactly p elements in level $\mathcal{P}_{L_i}^\phi$. Since the $p - s$ elements of the stalk must be in $\mathcal{P}_{L_i}^\phi$, the poset $\mathcal{P}_{L_i}^\phi$ contains exactly s elements of A_i . Clearly, the number of linear extensions of the petals of the $(|V(G)|, s)$ -flower in $\mathcal{P}_{L_i}^\phi$ is $s!$ and hence by Lemma 8 $e(\mathcal{P}_{L_i}^\phi) = s! \binom{2p-1}{p}$, which concludes the proof. \blacktriangleleft

► **Claim 11.** *Either $e(\mathcal{P}_{L_{r+1}}^\phi) \equiv 0 \pmod{p}$, or $e(\mathcal{P}_{L_{r+1}}^\phi) = \frac{(|V(G)|p+p-1)!}{(p-1)!} (p!)^{|V(G)|}$ and L_{r+1}^ϕ contains exactly $|V(G)|$ elements of A_{r+1} , specifically one petal for each $(r, 1)$ -flower on level L_{r+1} .*

Proof of the claim. Assume $e(\mathcal{P}_{L_{r+1}}^\phi) \not\equiv 0 \pmod{p}$, and let us first examine elements that are not associated with level L_{r+1} . Clearly, no element associated with level L_{r+2} can appear in $\mathcal{P}_{L_{r+1}}^\phi$ and the only elements associated with any level $i < r + 1$ that can end up in $\mathcal{P}_{L_{r+1}}^\phi$ are petals. Each of these elements is smaller than exactly one petal at level L_{r+1} and independent to all other elements associated with this level. It is easy to see that largest possible size of a connected component of $\mathcal{P}_{L_{r+1}}^\phi$ is $p - 1 + 2r < 2p$. By Lemma 9, every connected component in $\mathcal{P}_{L_{r+1}}^\phi$ (except for the stick) will have size p , and therefore $\mathcal{P}_{L_{r+1}}^\phi$ will contain exactly one element for every antichain associated with L_{r+1} and no other elements. Hence, $\mathcal{P}_{L_{r+1}}^\phi$ consists of $|V(G)|$ chains of length p and one chain of length $p - 1$. Then $e(\mathcal{P}_{L_{r+1}}^\phi) = \frac{(|V(G)|p+p-1)!}{(p-1)!} (p!)^{|V(G)|}$ follows from Lemma 8. \blacktriangleleft

► **Claim 12.** *Either $e(\mathcal{P}_{L_{r+2}}^\phi) \equiv 0 \pmod{p}$, or $e(\mathcal{P}_{L_{r+2}}^\phi) = \frac{(|E(G)|p+p-1)!}{(p-1)!} (p!)^{|E(G)|}$ and L_{r+2}^ϕ contains exactly $|E(G)|$ elements of A_{r+2} , specifically one petal for each $(r^2, 1)$ -flower on level L_{r+2} .*

Proof of the claim. The idea of the proof is similar to the proof of the previous claim, with one additional obstacle: that several flowers can be connected with petals from lower levels into one connected component on level L_{r+2} through the petals of flowers on level L_{r+1} . So, assume $e(\mathcal{P}_{L_{r+2}}^\phi)$ contains a connected component C which contains at least a single stalk. For each stalk in C , there must be at least one petal in the same flower (otherwise the stalk cannot be connected to the rest of C); in other words, the intersection of each flower and C contains at least p vertices. Let a denote the number of flowers which intersect C , b_2 denote $|A_{r+2} \cap C|$, b_1 denote $|A_{r+1} \cap C|$ and b_0 denote $\sum_{i=1}^r |A_r \cap C|$. Then it follows that $|C| = p \cdot a + (b_2 - a) + b_1 + b_0 \leq p \cdot a + r^2 |E(G)| + r |V(G)| + r |V(G)|$, and recall that $r^2 |E(G)| + r |V(G)| + r |V(G)| < p$. Furthermore, if $b_1 > 0$ (and at least one petal from A_{r+1} is required unless C contains only a single flower), we have $a \cdot p < |C| < (a + 1) \cdot p$. Hence any such C cannot have size divisible by p and by Lemma 9 we have $e(\mathcal{P}_{L_{r+2}}^\phi) \equiv 0 \pmod{p}$. Otherwise, if no two flowers are connected through a petal of a flower associated with level L_{r+1} , then every connected component of $\mathcal{P}_{L_{r+2}}^\phi$ of size p must consist of a stalk and exactly one petal and the claim follows analogously as the proof of Claim 11. \blacktriangleleft

► **Claim 13.** *If ϕ is a consistent configuration and for all $i \in \{1, \dots, r + 2\}$ it holds that $e(\mathcal{P}_{L_i}^\phi) \not\equiv 0 \pmod{p}$, then the petals in L_{r+1}^ϕ encode a proper equitable coloring of $V(G)$ where vertex v receives color i iff the petal v_i lies in L_i^ϕ and $\mathcal{P}_{L_{r+3}}^\phi$ is isomorphic with $\mathcal{P}(G, r)$.*

Proof of the claim. From Claims 10, 11 and 12 together with the assumption that $e(\mathcal{P}_{L_i}^\phi) \not\equiv 0 \pmod{p}$, it follows that each of the levels $L_1^\phi, \dots, L_r^\phi$ contains exactly s petals associated with the corresponding level, level L_{r+1}^ϕ contains exactly one petal for each vertex of G and level L_{r+2}^ϕ contains exactly one petal for each edge of G .

For the first part of this claim, we observe that each pair of petals in $L_1^\phi, \dots, L_r^\phi$ are associated with distinct vertices of G . If this were not the case, then since $|V(G)| = rs$ there would exist a vertex v such that no element of $L_1^\phi, \dots, L_r^\phi$ is associated with v . But due to the construction at level $r+1$ there exists some $i \in 1, \dots, r$ such that $v^i \in L_{r+1}^\phi$. Then, since $v_i \leq v^i$ and v_i can only occur either in level L_i^ϕ or L_{r+3}^ϕ (the latter of which lies above v^i in the linear extension due to the configuration ϕ), this would lead to a contradiction. In particular, we conclude that there is a matching between the petals in level $r+1$ (encoding the color for each vertex) and the union of petals in levels $1, 2, \dots, r$ (encoding the vertices assigned to each color class), and by Claim 10 it follows that there are exactly s petals in L_{r+1} associated with each color class.

We now argue that the coloring is proper. Observe that by the same argument as above, if an edge $e = uv$ satisfies $e_{i,j} \in L_{r+2}^\phi$, then $u_i \in L_{r+1}^\phi$ and $v_j \in L_{r+1}^\phi$. From the construction of $\mathcal{P}(G, r, p)$ it follows that if $i = j$, then $e_{i,j} \notin L_{r+2}$. Combining these two facts we get that the coloring encoded in L_{r+1}^ϕ is indeed proper.

Now let us take a look at level L_{r+3}^ϕ . To prove the claim, we will construct an isomorphism f from elements of $\mathcal{P}_{L_{r+3}^\phi}^\phi$ to elements of $\mathcal{P}(G, r)$. For every vertex $v \in V(G)$, precisely one element $v^i \in L_{r+1}^\phi$ and precisely one of the first r levels contains an element associated with v ; to be precise, $v_i \in L_i^\phi$ and $v_j \in L_{r+3}^\phi$ and hence also $v^j \in L_{r+3}^\phi$ for all $j \neq i$. We set $f(v^j) = v_{j,0}$ and $f(v_j) = v_{j,1}$, whenever $j \neq i$ and $j < r$. For the last remaining elements, we set $f(v^r) = v_{i,0}$ and $f(v_r) = v_{i,1}$. Next, for every edge $e = uv$ there is exactly one $e_{a,b} \in L_{r+2}^\phi$. Moreover, if $e_{a,b} \in L_{r+2}^\phi$ then $u_a \in L_{r+1}^\phi$ and $v_b \in L_{r+1}^\phi$, and all other petals for this edge e are in L_{r+3}^ϕ . Let $g_i(r) = i$, $g_i(i) = 0$, and $g_i(k) = k$ otherwise. Then we set $f(e_{i,j}) = e_{g_a(i), g_b(j)}$. Observe that, since $e_{a,b}$ does not lie in L_{r+3}^ϕ , no edge is mapped to the non-existent element $e_{0,0}$ in $\mathcal{P}(G, r)$. It is straightforward to verify that f is really bijective mapping between elements of $\mathcal{P}_{L_{r+3}^\phi}^\phi$ and $\mathcal{P}(G, r)$. Moreover, $f(u) \leq f(v)$ in $\mathcal{P}(G, r)$ if and only if $u \leq v$ in $\mathcal{P}_{L_{r+3}^\phi}^\phi$. Therefore, $\mathcal{P}_{L_{r+3}^\phi}^\phi$ is isomorphic with $\mathcal{P}(G, r)$ and the claim holds. ◀

► **Claim 14.** $e(\mathcal{P}(G, r, p)) \not\equiv 0 \pmod{p}$ if and only if $e(\mathcal{P}(G, r)) \cdot \#\text{EC}(G, r) \not\equiv 0 \pmod{p}$.

► **Claim 15.** If $\#\text{EC}(G, r) \neq 0$, then there is a prime number p greater than $2r|V(G)| + r^2|E(G)|$ and smaller than $(2r|V(G)| + r^2|E(G)|)^2$ such that p does not divide $e(\mathcal{P}(G, r)) \cdot \#\text{EC}(G, r)$.

► **Claim 16.** $\text{tw}(C(\mathcal{P}(G, r, p))) \leq r \cdot (\text{tw}(G) + 3) + 6$.

Concluding the proof

Let us summarize the fpt turing reduction used to prove Theorem 7. Given an instance (G, r) of EQUITABLE COLORING[tw], we loop over all primes p such that $2r|V(G)| + r^2|E(G)| < p < (2r|V(G)| + r^2|E(G)|)^2$, and for each such prime we construct the poset $\mathcal{P}(G, r, p)$; from Claim 15 it follows that if $\#\text{EC}(G, r) \neq 0$, then at least one such prime will not divide $e(\mathcal{P}(G, r)) \cdot \#\text{EC}(G, r)$, and by Claim 16 each of the constructed posets $\mathcal{P}(G, r, p)$ has bounded treewidth of the cover graph. For each such poset $\mathcal{P}(G, r, p)$, we compute $e(\mathcal{P}(G, r, p))$ by the black-box procedure provided as part of the reduction. If for any prime p we get $e(\mathcal{P}(G, r, p)) \not\equiv 0 \pmod{p}$, then we conclude that (G, r) is a yes-instance, and otherwise we reject (G, r) , and this is correct by Claim 14. ◀

We remark that the above construction can be extended to also compute the exact number of equitable colorings. However, because EQUITABLE COLORING[tw] is not known to be

$\#W[1]$ -hard, this does not immediately imply $\#W[1]$ -hardness for counting the number of linear extensions.

4 Fixed-Parameter Tractability of Counting Linear Extensions

This section is dedicated to proving our algorithmic result, stated below.

► **Theorem 17.** *$\#LE$ is fixed-parameter tractable parameterized by the treewidth of the incomparability graph of the input poset.*

The proof of Theorem 17 is divided into two steps. First, we apply a transformation process to a path-decomposition \mathcal{Q} of small width (the existence of which is guaranteed by Corollary 6) of $I(\mathcal{P})$ which results in a tree-decomposition \mathcal{T} of $I(\mathcal{P})$ satisfying certain special properties. We call these “blocked tree-decompositions” and the construction is given in Lemma 21. The properties of \mathcal{T} are then used to prove that $I_C(\mathcal{P})$ has treewidth bounded by the treewidth of $I(\mathcal{P})$ (Corollary 26). In the second step, we construct an MSO formulation which enumerates all the linear extensions of \mathcal{P} using $I_C(\mathcal{P})$, and apply Fact 4.

4.1 The Treewidth of Combined Graphs

We begin by arguing a useful property of separators in incomparability graphs.

► **Lemma 18.** *Let $S \subseteq V(I(\mathcal{P}))$. Then for each pair of distinct connected components C_1, C_2 in $I(\mathcal{P}) - S$, it holds that for any $a_1, b_1 \in C_1$ and any $a_2, b_2 \in C_2$ we have $a_1 \leq a_2$ iff $b_1 \leq b_2$. Namely, the poset contains a total order of all connected components in $I(\mathcal{P}) - S$.*

Proof. We begin by proving the following claim.

► **Claim 19.** *Let a, b, c be three distinct elements of \mathcal{P} such that $a \parallel b$ and both pairs a, c and b, c are comparable. Then $a \leq c$ iff $b \leq c$.*

Proof of the claim. Suppose that, w.l.o.g., $a \leq c$ and $c \leq b$. Then by the transitivity of \leq , we get $a \leq b$ which contradicts our assumption that $a \parallel b$. ◀

Now to prove Lemma 18, assume for a contradiction that, w.l.o.g., there exist $a_1, b_1 \in C_1$ and $a_2, b_2 \in C_2$ such that $a_1 \leq b_1$ and $b_2 \leq a_2$. Let Q_1 be an a_1 - a_2 path in $I[C_1]$. By Claim 19, $a_1 \leq b_1$ implies that every element q on Q_1 satisfies $q \leq b_1$, and in particular $a_2 \leq b_1$. Next, let Q_2 be a b_1 - b_2 path in $I[C_2]$. Then Claim 19 also implies that each element q' on Q_2 satisfies $a_2 \leq q'$. Since b_2 lies on Q_2 , this would imply that $a_2 \leq b_2$, a contradiction. ◀

To proceed further, we will need some notation. Let $\mathcal{T} = (T, \mathcal{X})$ be a rooted tree-decomposition and $t \in V(T)$. We denote by $L(t)$ the set of all vertices which occur in the “branch” of $T - t$ containing the root r ; formally, $L(t) = \{v \in X_{t'} \setminus X_t \mid t' \text{ lies in the same connected component as } r \text{ in } T - t\}$. We then set $R(t) = V(G) \setminus (L(t) \cup X_t)$ (the intuition behind L and R is that they represent “left” and “right”). We also let T_t^r denote the connected component of $T - t$ which contains the root r .

Next, recall that each connected component of the graph obtained after deleting X_t must lie in a subtree of $T - t$ (Fact 3). A *block* of a bag X_t in a rooted tree-decomposition $\mathcal{T} = (T, \mathcal{X})$ is a sequence of consecutive connected components in $(I(\mathcal{P}) - X_t) \cap R(t)$. We say that a node $t \in V(T)$ has z blocks if there exist z distinct blocks of X_t . Blocks will

play an important role in the tree-decomposition we wish to obtain from our initial path-decomposition of $I(\mathcal{P})$. The following lemma captures the operation we will use to alter our path-decomposition.

► **Lemma 20.** *Let $\mathcal{T} = (T, \mathcal{X})$ be a rooted tree-decomposition of a graph G and let $t \in V(T)$ be such that there are z blocks of X_t . Then there is a tree-decomposition $\mathcal{T}'(T', \mathcal{X}')$ satisfying:*

1. *The width of \mathcal{T}' is at most the width of \mathcal{T} .*
2. *The tree T' contains T_t^r as a subtree which is separated from the rest of T' by t .*
3. *The degree of t in T' is $z + 1$.*
4. *There exists a bijection α between the z blocks of X_t and the z trees in $T' - t$ other than T_t^r such that for each block B of X_t , we have $\bigcup_{s \in \alpha(B)} X'_s \setminus X_t = B$.*
5. *For each $t' \in N[t] \setminus V(T_t^r)$, we have $X_{t'} = X_t$.*

We proceed by showing how Lemma 20 is applied to transform a given path-decomposition.

► **Lemma 21.** *Let \mathcal{Q} be a nice path-decomposition of $I(\mathcal{P})$. Then there is a rooted tree-decomposition $\mathcal{T} = (T, \mathcal{X})$ of $I(\mathcal{P})$ with the following properties. \mathcal{T} is rooted at a leaf r and $X_r = \emptyset$, the width of \mathcal{T} is at most the width of \mathcal{Q} , and for any node $t \in V(T)$ with $z > 1$ blocks:*

1. *The degree of t in T is $z + 1$.*
2. *There exists a bijection α between the z blocks of X_t and the z trees in $T' - t$ other than T_t^r such that for each block B of X_t , we have $\bigcup_{s \in \alpha(B)} X_s \setminus X_t = B$.*
3. *For $t' \in N(t) \cap V(T_t^r)$ there exists a vertex v such that $X_{t'} = X_t \setminus \{v\}$, and furthermore t' has degree 2 and 1 block.*
4. *For each pair of neighbors $t, t' \in V(T)$, it holds that $|X_t \setminus X_{t'}| + |X_{t'} \setminus X_t| \leq 1$.*

We call a tree-decomposition rooted at a leaf with $X_r = \emptyset$ which satisfies the properties of Lemma 21 a *blocked tree-decomposition*. The next ingredient we will need for proving that $I_C(\mathcal{P})$ has small treewidth is the notion of *cover-guards*.

Let $\mathcal{T} = (T, \mathcal{X})$ be a tree-decomposition of $I(\mathcal{P})$ rooted at r and let $t \neq r$. Then the *cover-guard* of t , denoted \mathcal{A}_t , is the set of vertices in $L(t)$ which are incident to a cover edge whose other endpoint lies in $R(t)$; formally, $\mathcal{A}_t = \{v \in L(t) \mid \exists u \in R(t) : (uv \in E(C(\mathcal{P})) \vee vu \in E(C(\mathcal{P})))\}$. For a vertex $v \in I(\mathcal{P})$, we let $\mathcal{A}^v = \{t \in V(T) \mid v \in \mathcal{A}_t\}$ and $X^v = \{t \in V(T) \mid v \in X_t\}$.

Our next aim is to add all the cover-guards into each bag. The following lemma will allow us to argue that the result is still a tree-decomposition; it is worth noting that the assumption that the decomposition is blocked is essential for the lemma to hold.

► **Lemma 22.** *Let $\mathcal{T} = (T, \mathcal{X})$ be a blocked tree-decomposition of $I(\mathcal{P})$ rooted at r and let $v \in I(\mathcal{P})$. Then $T[\mathcal{A}^v \cup X^v]$ is a tree.*

Next we show that the cover-guards in blocked tree-decompositions are never too large.

► **Lemma 23.** *Let $\mathcal{T} = (T, \mathcal{X})$ be a blocked tree-decomposition of $I(\mathcal{P})$ of width k . Then for each $t \in V(T)$ it holds that $|\mathcal{A}_t| \leq 2k + 2$.*

Proof. First, observe that if a node $t \in V(T)$ has 0 blocks, then $R(t) = \mathcal{A}_t = \emptyset$. So, consider a node t which has exactly 1 block consisting of connected components (D_1, \dots, D_j) in $(I(\mathcal{P}) - X_t) \cap R(t)$.

► **Claim 24.** $|\mathcal{A}_t| \leq 2k + 2$.

Proof of the claim. Assume for a contradiction that $|\mathcal{A}_t| > 2k + 2$. By Lemma 18 we have that (D_1, \dots, D_j) are consecutive connected components in a total order of connected components in $I(\mathcal{P}) - X_t$. Hence any edge in $C(\mathcal{P}) - X_t$ between $R(t)$ and $L(t)$ must necessarily have one endpoint in $D_1 \cup D_j$. Furthermore, an element in \mathcal{A}_t cannot be adjacent to both D_1 and D_j in $C(\mathcal{P}) - X_t$ due to transitivity and acyclicity. So, we may partition \mathcal{A}_t into $\mathcal{A}_t^1 = \{v \in \mathcal{A}_t \mid \exists u \in D_1 : v \triangleleft^{\mathcal{P}} u\}$ and $\mathcal{A}_t^2 = \{v \in \mathcal{A}_t \mid \exists u \in D_j : u \triangleleft^{\mathcal{P}} v\}$.

By Lemma 18, it also follows that \mathcal{A}_t^1 and \mathcal{A}_t^2 must each lie in separate connected components of $I(\mathcal{P}) - X_t$, say C^1 and C^2 respectively. Furthermore, each element in \mathcal{A}_t^1 is maximal in C^1 and each element in \mathcal{A}_t^2 is minimal in C^2 . In particular, each of \mathcal{A}_t^1 , \mathcal{A}_t^2 forms a clique in $I(\mathcal{P})$. But by our assumption on the size of \mathcal{A}_t , at least one of \mathcal{A}_t^2 and \mathcal{A}_t^1 must have size greater than $k + 1$, which implies that $I(\mathcal{P})$ contains a clique of size at least $k + 2$. It is well-known that each clique must be completely contained in at least one bag of a tree-decomposition, and so we arrive at a contradiction with $\text{tw}(I(\mathcal{P})) \leq k$. Hence we conclude that $|\mathcal{A}_t| \leq 2k + 2$ and the claim holds. \blacktriangleleft

Finally, consider a node t which has at least 2 blocks. By Property 3 of Lemma 21, it holds that t has a neighbor t' in T_t^r such that $X_{t'} = X_t \setminus \{v\}$ and t' has 1 block. By Claim 24 we know that $|\mathcal{A}_{t'}| \leq 2k + 2$. Since $L(t) = L(t')$ and $R(t) \subseteq R(t')$, it follows that $\mathcal{A}_t \subseteq \mathcal{A}_{t'}$, and in particular $|\mathcal{A}_t| \leq |\mathcal{A}_{t'}|$. We have now proved the desired bound for all nodes in \mathcal{T} , and so the lemma holds. \blacktriangleleft

With Lemma 22 and Lemma 23, we have the tools necessary for arguing that there exists a tree-decomposition of the combined graph of small width.

► Lemma 25. *Let $\mathcal{T} = (T, \mathcal{X})$ be a blocked tree-decomposition of $I(\mathcal{P})$ such that $\text{tw}(\mathcal{T}) \leq k$. Then there exists a tree-decomposition \mathcal{T}' of $I_C(\mathcal{P})$ of width at most $3k + 2$.*

Proof. Consider the tree-decomposition $\mathcal{T}' = (T, \mathcal{X}')$ where $\mathcal{X}' = \{X'_t \mid t \in V(T)\}$ is defined as follows. For each $t \in V(T)$ such that its unique neighbor s in T_t^r satisfies $|X_t \setminus X_s| = 1$, we set $X'_t = X_t \cup \mathcal{A}_s$; it will be useful to observe that $\mathcal{A}_s \supseteq \mathcal{A}_t$. For all other nodes $t \in V(T)$, we then set $X'_t = X_t \cup \mathcal{A}_t$. We call nodes of the first type *non-standard* and nodes of the second type *standard*.

First, we note that the size of each bag in \mathcal{T}' is at most $3k + 2$, since every node $t \in V(T)$ satisfies $|\mathcal{A}_t| \leq 2k + 2$ by Lemma 23. Furthermore, \mathcal{T}' satisfies condition (T1) because \mathcal{T} was a tree-decomposition of $I(\mathcal{P})$. \mathcal{T}' also satisfies condition (T2); indeed, for each $v \in \mathcal{P}$ it holds that X'^v restricted to standard nodes is a connected tree by Lemma 22, and by construction every non-standard node t such that $v \in X'_t \setminus X_t$ is adjacent to a standard node containing v . So, it only remains to argue condition (T3).

Obviously, condition (T3) holds for any edge of $I(\mathcal{P})$. So, consider two elements u, v of \mathcal{P} such that $u \triangleleft^{\mathcal{P}} v$ or $v \triangleleft^{\mathcal{P}} u$. If there exists a node $t \in V(T)$ such that $u, v \in X_t$, then $u, v \in X'_t$ and the condition also holds for this edge in $I_C(\mathcal{P})$. So, assume that X^v and X^u are disjoint and let Q be the unique X^v - X^u path in T . By Property 4, the X^v - X^u path Q in T must contain at least one internal node.

Consider the case where one of these subtrees, say w.l.o.g. X^v , lies in the connected component T_t^r of $T - Q$. Then for each internal node $q \in Q$, it holds that $v \in L(q)$ and $u \in R(q)$, which in turn implies that $v \in \mathcal{A}_q$. Let q_u be the endpoint of Q in X^u and let q_0 be the neighbor of q_u in Q . By Property 4 we have $X_{q_u} \setminus X_{q_0} = \{u\}$, which implies that q_u is a non-standard node and in particular $\mathcal{A}_{q_0} \subseteq \mathcal{X}'_{q_u}$. Since q_0 is an internal node of Q , it follows that $v \in X'_{q_u}$ which means that condition (T3) also holds for any edge uv in this case.

Finally, consider the case where there exists a node $q \in Q$ of degree at least 3 such that each of X^u , X^v and r occur in different components of $T - q$. Then we reach a contradiction similarly as in the proof of Lemma 22. In particular, since $u, v \in R(q)$ due to the location of the root and there is a cover edge between them, it follows that either u, v occur in the same connected component of X_q or in two consecutive ones, but in either case u, v must lie in the same block of q , say block B . But since $u, v \notin X_q$, this contradicts Property 2 in Lemma 21; indeed, each tree in $T - q$ contains at most one of v, u in its bags, and hence there exists no tree T' in $T - q$ satisfying $\bigcup_{t' \in V(T')} X_{t'} \setminus X_q = B$. Hence this case in fact violates our assumptions and cannot occur.

Summarizing the above arguments, we conclude that each bag in \mathcal{T}' has size at most $3k + 2$ and that \mathcal{T}' satisfies all of the conditions of a tree-decomposition. \blacktriangleleft

► **Corollary 26.** *Let \mathcal{P} be a poset such that $\text{tw}(I(\mathcal{P})) \leq k$. Then $\text{tw}(I_C(\mathcal{P})) \leq 3k + 2$.*

Proof. By Corollary 6 we know that there exists a nice path-decomposition of $I(\mathcal{P})$ of width at most k . By Lemma 21, it follows that there exists a blocked tree-decomposition of $I(\mathcal{P})$ of width at most k . The corollary then follows by Lemma 25. \blacktriangleleft

4.2 MSO Formulation

In this subsection, we use Fact 4 to prove the following result, which forms the second ingredient required for our proof of Theorem 17.

► **Lemma 27.** *#LE is fixed-parameter tractable parameterized by the treewidth of the combined graph of the input poset.*

Sketch of the Proof. Let $\mathcal{P} := (P, \preceq^P)$ be a poset. Let G be the (edge-)labeled directed graph obtained from $I_C(\mathcal{P})$ by directing every bidirectional edge of $I_C(\mathcal{P})$, i.e., every edge of $I(\mathcal{P})$, in an arbitrary way and labeling it with the label \parallel .

For a set of edges $E \subseteq E(G)$ with label \parallel , let $G[E]$ be the graph obtained from G after reversing every edge in E . Moreover, for a linear extension \preceq of \mathcal{P} let $E_G(\preceq)$ be the set of edges (u, v) of G such that $v \preceq u$. Note that because every linear extension of \mathcal{P} has to respect the direction of the edges in G given by C , it holds that every edge in $E_G(\preceq)$ has label \parallel .

► **Claim 28.** *$E_G(\preceq)$ defines a bijection between the set of linear extensions of \mathcal{P} and the set of subsets E of edges of G with label \parallel such that $G[E]$ is acyclic.*

Proof of the claim. Let \preceq be a linear extension of \mathcal{P} . Then, as observed above, $E_G(\preceq)$ is a set of edges of G with label \parallel . Moreover, because $G[E_G(\preceq)]$ is a subgraph of $P_G(\preceq)$ and $P_G(\preceq)$ is acyclic so is $G[E_G(\preceq)]$. Hence, $E_G(\preceq)$ is a function from the set of linear extensions of \mathcal{P} to the set of subsets E of edges of G with label \parallel such that $G[E]$ is acyclic. Towards showing that $E_G(\preceq)$ is injection assume for a contradiction that this is not the case, i.e., there are two distinct linear extensions \preceq_1 and \preceq_2 of \mathcal{P} such that $E_G(\preceq_1) = E_G(\preceq_2)$ and let u and v be two elements of \mathcal{P} ordered differently by \preceq_1 and \preceq_2 . Then $\{u, v\} \in I(\mathcal{P})$ and hence either $(u, v) \in G$ or $(v, u) \in G$ the label of (u, v) or (v, u) respectively is \parallel . W.l.o.g. assume that $(u, v) \in G$ with label \parallel . But then, because \preceq_1 and \preceq_2 differ on u and v , either $(u, v) \in E_G(\preceq_1)$ but not $(u, v) \in E_G(\preceq_2)$ or $(u, v) \in E_G(\preceq_2)$ but not $(u, v) \in E_G(\preceq_1)$. In both cases we get a contradiction to our assumption that $E_G(\preceq_1) = E_G(\preceq_2)$. It remains to show that $E_G(\preceq)$ is surjective. To see this let E be a subsets of the edges of G with label \parallel such that $G[E]$ is acyclic. Because $G[E]$ is acyclic it has a topological ordering, say \preceq , of its

vertices. Because $G[E]$ contains $C(\mathcal{P})$ as a subgraph and any topological ordering of $C(\mathcal{P})$ is a linear extension of \mathcal{P} , we obtain that \preceq is a linear extension and also $E = E_G(\preceq)$. ◀

It follows from the above that instead of counting the number of linear extensions of \mathcal{P} directly, we can count the number of subsets E of the edges of G with label \parallel such that $G[E]$ is acyclic. It can be shown that there exists an MSO formula Φ (with length independent of G) such that $G \models \Phi(X)$ if and only if X is a subset of the edges of G with label \parallel such that $G[X]$ is acyclic. Because of Fact 4, this implies that $\#LE$ is fixed-parameter tractable when parameterized by $\text{tw}(G)$ and hence also when parameterized by $\text{tw}(I_C(\mathcal{P}))$, concluding the proof of the lemma. Generally speaking, $\Phi(X)$ only needs to check that X is a set of edges of G with label \parallel and there is no non-empty set of edges C of $G[X]$ that forms a cycle. ◀

We conclude this section by stating the proof of Theorem 17.

Proof of Theorem 17. Let \mathcal{P} be the input poset and let $k = \text{tw}(I(\mathcal{P}))$. Then $\text{tw}(I_C(\mathcal{P})) \leq 3k + 2$ by Corollary 26, and the theorem follows by Lemma 27. ◀

5 Conclusions and Future Work

We have given the first parameterized intractability result for counting linear extensions. We hope that the employed techniques will inspire similar results and expand our knowledge about the parameterized complexity of counting problems. In particular, even for $\#LE$ there remain many open questions concerning other very natural parameterizations such as the width of the poset or the treewidth of the poset graph. Moreover, our intractability result for the treewidth of the cover graph poses the question whether there are stronger parameterizations under which $\#LE$ becomes tractable, e.g., the treewidth of the poset graph, the treedepth or even vertex cover number of the poset- or cover graph, as well as combinations of these parameters with parameters such as the width, the dimension, or the height of the poset. These numerous examples illustrate that the parameterized complexity of $\#LE$ is still largely unexplored. As a side note it would also be interesting to establish whether our hardness result for $\#LE$ can be sharpened to $\#W[1]$ -hardness and to obtain matching membership results.

References

- 1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.
- 2 Mike D Atkinson. On computing the number of linear extensions of a tree. *Order*, 7(1):23–25, 1990.
- 3 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.
- 4 Hans L. Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996.
- 5 Bostjan Bresar, Manoj Changat, Sandi Klavzar, Matjaz Kovse, Joseph Mathews, and Antony Mathews. Cover-incomparability graphs of posets. *Order*, 25(4):335–347, 2008.
- 6 Graham Brightwell and Peter Winkler. Counting linear extensions is $\#P$ -complete. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC’91, pages 175–181, 1991.
- 7 Russ Bublely and Martin Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, 201(1–3):81–88, 1999.

- 8 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
- 9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 10 Karel De Loof, Hans De Meyer, and Bernard De Baets. Exploiting the lattice of ideals representation of a poset. *Fundam. Inf.*, 71(2,3):309–321, February 2006. URL: <http://dl.acm.org/citation.cfm?id=1227505.1227514>.
- 11 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 12 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 13 Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, January 1991. doi:10.1145/102782.102783.
- 14 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshantov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Inf. Comput.*, 209(2):143–153, 2011.
- 15 Stefan Felsner and Thibault Manneville. Linear extensions of N-free orders. *Order*, 32(2):147–155, 2014. doi:10.1007/s11083-014-9321-0.
- 16 Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004.
- 17 Georg Gottlob, Reinhard Pichler, and Fang Wei. Bounded treewidth as a key to tractability of knowledge representation and reasoning. *Artificial Intelligence*, 174(1):105–132, 2010. doi:10.1016/j.artint.2009.10.003.
- 18 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1), 2007.
- 19 M. Habib and R.H. Möhring. On some complexity properties of N-free posets and posets with bounded decomposition diameter. *Discrete Mathematics*, 63(2):157–182, 1987. doi:10.1016/0012-365X(87)90006-9.
- 20 Michel Habib and Rolf H Möhring. Treewidth of cocomparability graphs and a new order-theoretic parameter. *Order*, 11(1):47–60, 1994.
- 21 Thore Husfeldt, Ramamohan Paturi, Gregory B. Sorkin, and Ryan Williams. Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time (Dagstuhl Seminar 13331). *Dagstuhl Reports*, 3(8):40–72, 2013. doi:10.4230/DagRep.3.8.40.
- 22 Kustaa Kangas, Teemu Hankala, Teppo Niinimäki, and Mikko Koivisto. Counting linear extensions of sparse posets. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016, New York City, USA*, 2016. to appear.
- 23 T. Kloks. *Treewidth: Computations and Approximations*. Springer Verlag, Berlin, 1994.
- 24 Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Probabilistic preference logic networks. In Torsten Schaub, Gerhard Friedrich, and Barry O’Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 561–566. IOS Press, 2014.
- 25 Heikki Mannila and Christopher Meek. Global partial orders from sequential data. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’00*, pages 161–168, New York, NY, USA, 2000. ACM. doi:10.1145/347090.347122.
- 26 Rolf H. Möhring. *Algorithms and Order*, chapter Computationally Tractable Classes of Ordered Sets, pages 105–193. Springer Netherlands, 1989.

- 27 Jason Morton, Lior Pachter, Anne Shiu, Bernd Sturmfels, and Oliver Wienand. Convex rank tests and semigraphoids. *SIAM Journal on Discrete Mathematics*, 23(3):1117–1134, 2009. doi:10.1137/080715822.
- 28 Teppo Mikael Niinimäki and Mikko Koivisto. Annealed importance sampling for structure learning in Bayesian networks. In *IJCAI. IJCAI/AAAI*, 2013.
- 29 Daniel Paulusma, Friedrich Slivovsky, and Stefan Szeider. Model counting for CNF formulas of bounded modular treewidth. *Algorithmica*, pages 1–27, 2015. doi:10.1007/s00453-015-0030-x.
- 30 Marcin Peczarski. New results in minimum-comparison sorting. *Algorithmica*, 40(2):133–145, July 2004. doi:10.1007/s00453-004-1100-7.