Counting Equitable k-Colorings in Graphs of Bounded Clique-Width

Holger Dell
IT University of Copenhagen, Denmark

Thore Husfeldt
IT University of Copenhagen, Denmark

Amir Nikabadi
IT University of Copenhagen, Denmark

Abstract

For a graph G, a proper k-coloring of G is *equitable* if the sizes of any two color classes differ by at most one. The Equitable k-Coloring problem asks, for a given graph G and integer k, whether G admits an equitable k-coloring. Bodlaender and Fomin (Theoretical Computer Science 2005) showed that it is polynomial-time solvable on graphs of bounded treewidth, while it remains NP-hard on cographs, and thus on graphs of constant clique-width. Fellows et al. (Information and Computation 2011) showed that the problem becomes W[1]-hard when parameterized by tree-width (and hence clique-width) plus the number of colors k.

We first show that, there exists an algorithm, given an integer $k \geq 1$ and an n-vertex graph G together with a w-expression whose underlying unlabelled graph is G, computes the number of equitable k-colorings of G in time $2^{O(k \cdot w)} \cdot n^{O(k)}$. In particular, we show that for every fixed k, counting equitable k-colorings is polynomial-time solvable on graph classes of bounded clique-width, given a clique-width expression.

We then show that, under SETH, the dependence on clique-width in this algorithm is essentially optimal. As a consequence, our results provide a fairly tight picture of the complexity of Equitable k-Coloring with respect to the combined parameter k+clique-width in the following sense: For variable k, the problem is W[1]-hard, however for every fixed integer k, it is polynomial-time solvable on graphs of bounded clique-width given a clique-width expression, and this remains true even for the counting version.

Second, we refine our clique-width algorithm for the linear setting. We show that there exists an algorithm, given an integer $k \geq 1$ and an n-vertex graph G together with a linear w-expression constructing G, computes the number of equitable k-colorings of G in time $\max\{1, 2^k - 2\}^w \cdot n^{k+O(1)}$. Thus, for bounded linear clique-width, we obtain a significantly sharper dependence on the width parameter than in the general clique-width case.

Third, we consider a different structural restriction, namely the class of P_t-free graphs. A graph is called P_t-free if it does not contain the path on t vertices as an induced subgraph. This is a different setting from bounded clique-width; in particular, already P_5-free graphs have unbounded clique-width. Nevertheless, we show that for every P_t-free graph G, the number of equitable list 3-colorings of G can be computed in subexponential time.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Equitable coloring, Clique-width, P_t-free graphs

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.27

Funding *Thore Husfeldt*: Supported by the VILLUM Foundation grant 54451.
Amir Nikabadi: Supported by the Independent Research Fund Denmark (DFF), grant agreement number 2098-00012B.

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 27; pp. 27:1–27:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Let G be a graph. A proper k-coloring of G is *equitable* if the sizes of any two color classes differ by at most one. Equitable coloring has a long history. A celebrated result of Hajnal and Szemerédi [18] settled a conjecture of Erdős by showing that every graph G with maximum degree at most k admits an equitable $(k+1)$-coloring. In contrast to the classical setting, where Brooks' theorem states that every connected graph G with maximum degree Δ satisfies $\chi(G) \leq \Delta$, except when G is a complete graph or an odd cycle; the equitable analogue of this statement is still open. In particular, Chen, Lih, and Wu [4] conjectured that every connected graph G with maximum degree $\Delta \geq 2$ admits an equitable coloring with Δ colors, except when G is a complete graph, an odd cycle, or Δ is odd and $G = K_{\Delta,\Delta}$. This conjecture remains open in general, although it has been verified for several graph classes [4, 3, 5, 24].

Equitable k-Coloring

Input: A graph G and an integer k.
Task: Decide whether G admits an equitable k-coloring.

On the algorithmic side, the Equitable k-Coloring problem has been studied extensively. By a simple reduction from k-Coloring, it follows that Equitable k-Coloring is NP-hard. Bodlaender and Fomin [1] showed that the problem is solvable in polynomial time on graphs of bounded treewidth, while it remains hard on cographs. Fellows et al. [14] showed that the problem is W[1]-hard when parameterized by treewidth together with the number of colors. The parameterized complexity of Equitable k-Coloring has also been studied under structural parameterizations [15] and on restricted graph classes [16, 7].

In this paper, we first study Equitable k-Coloring with respect to the combined parameters clique-width and the number of colors. The *clique-width* (introduced by Courcelle et al. [9]) of a graph G, denoted by $\mathsf{cw}(G)$, is the minimum number of different labels needed to construct G using certain graph operations (see Section 2 for details). Clique-width has been studied extensively both in algorithmic and structural graph theory. It generalizes the classical notion of *treewidth*, denoted by tw, in the following sense: For every graph G, $\mathsf{cw}(G) \leq O(2^{\mathsf{tw}(G)})$ [8], whereas no converse bound holds, since there are graphs of arbitrarily large treewidth and bounded clique-width. The importance of clique-width stems in part from the fact that many problems, including Coloring and Hamilton Cycle [13, 21, 10, 23], that are NP-hard in general become polynomial-time solvable on any graph class, say $\mathcal{G}$, of bounded clique-width, that is, for which there exists a constant c, such that for every graph $G \in \mathcal{G}$, $\mathsf{cw}(G) \leq c$. It follows from the result of [14] that Equitable k-Coloring remains W[1]-hard when parameterized by the clique-width of the input graph together with the number of colors. We show, however, that counting equitable k-colorings can still be done in XP for these combined parameters. In particular, we prove the following.

▶ **Theorem 1.** *There exists an algorithm that, given an integer $k \geq 1$ and an n-vertex graph G together with a w-expression whose underlying unlabelled graph is G, computes the number of equitable k-colorings of G in time $2^{O(k \cdot w)} \cdot n^{O(k)}$. In particular, for every fixed k, counting equitable k-colorings is polynomial-time solvable on graph classes of bounded clique-width, given a clique-width expression.*

The proof of Theorem 1 is by dynamic programming over a given clique-width expression. Let us distinguish our approach from more standard dynamic programming algorithms over clique-width expressions. While the overall framework is the usual bottom-up dynamic programming over a given expression, the state representation is different. Ordinary coloring needs only local information about how colors interact with the current label classes. For equitable coloring, however, one must also keep track of the global sizes of the color classes. Accordingly, for each subexpression we use a signature that records, for every label, the set of colors appearing on that label class, and, for every color, the current size of its color class. It is this additional global information that makes it possible to enforce the equitable condition at the root.

We then present a complementary SETH-based lower bound for graphs of bounded clique-width. Although it does not fully match the running time in Theorem 1, it shows that for every fixed $k \geq 3$ one cannot hope for a running time of the form $O^*((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)})$. More precisely, we prove the following.

▶ **Theorem 2.** *Let G be a graph. For every fixed $k \geq 3$ and every $\varepsilon > 0$, unless SETH fails, EQUITABLE k-COLORING cannot be solved in time $O^*\left((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)}\right)$, where $\mathsf{cw}(G)$ denotes the clique-width of G.*

The proof of Theorem 2 is based on the SETH-based lower bound of Lampis [22] for k-COLORING parameterized by clique-width. We reduce from k-COLORING by padding the input graph with an independent set, in such a way that proper k-colorings of the original graph correspond exactly to equitable k-colorings of the new graph, while the clique-width increases by at most one.

We next refine our Theorem 1 by turning to linear clique-width, which plays a role analogous to that of pathwidth relative to treewidth. Because linear expressions have a prescribed left-to-right structure (we will get back to this in Section 4), they admit a more economical dynamic-programming state space. This yields a sharper running time than in the general clique-width setting.

▶ **Theorem 3.** *There exists an algorithm that, given an integer $k \geq 1$ and an n-vertex graph G together with a linear w-expression constructing G, computes the number of equitable k-colorings of G in time $\max\{1, 2^k - 2\}^w \cdot n^{k+O(1)}$. In particular, for every fixed integer k, counting equitable k-colorings is polynomial-time solvable on graph classes of bounded linear clique-width, given a linear clique-width expression.*

We finally turn to P_t-free graphs. Since graphs excluding a fixed induced subgraph form one of the most natural classes in structural graph theory, P_t-free graphs provide a particularly appealing test case: they are far beyond bounded treewidth, and even already P_5-free graphs have unbounded clique-width [12].

For ordinary k-COLORING (which asks whether a given graph is k-colorable) on P_t-free graphs, polynomial-time algorithms are known for $t \leq 5$ [19], for $(k, t) = (4, 6)$ [6], and for $(k, t) = (3, 7)$ [2], while Huang [20] showed that 4-COLORING is NP-complete for P_7-free graphs and 5-COLORING is NP-complete for P_6-free graphs; in particular, the complexity of 3-COLORING on P_t-free graphs remains open for $t \geq 8$. This makes P_t-free graphs a particularly natural setting in which to seek subexponential-time algorithms for equitable coloring. Groenland et al. [17] showed that the number of proper 3-colorings of a P_t-free graph can be computed in subexponential time. We extend this result to equitable colorings.

▶ **Theorem 4.** *Let $t \in \mathbb{N}$. There is an algorithm that, given a P_t-free graph G on n vertices and a list assignment $L\colon V(G) \to 2^{\{1,2,3\}}$, computes the number of equitable proper list 3-colorings of (G, L) in time $2^{O(\sqrt{n \log n})} \cdot \mathrm{poly}(n)$*[1]*.*

[1] Since t is fixed, the factor depending on t is absorbed in the $O(\cdot)$-notation.

The remainder of the paper is organized as follows. In Section 3 we prove Theorem 1 and Theorem 2. In Section 4 we prove Theorem 3. Finally, in Section 5 we prove Theorem 4. Proofs of statements marked with "♣" can be found in the attached full version.

2 Preliminaries

Sets and numbers. We use $\mathbb{N}$ to denote the set of positive integers. We let $[n] = \{1, \ldots, n\}$ for every $n \in \mathbb{N}$. For a set U, we denote by 2^U the set of all subsets of U. The identity map on U is denoted by id_U. We write $O^*(f(n))$ for a non-decreasing function $f(n)$ of the input size parameter n to suppress polynomial factors in n. For two sets A and B we denote by $A \uplus B$ the disjoint union of A and B. Let $f : D \to R$ be a function and let $A \subseteq D$. The *restriction of f to A*, denoted by $f|_A$, is the function $f|_A : A \to R$ defined by $f|_A(x) \coloneqq f(x)$ for all $x \in A$. For composable maps f and g, $g \circ f$ denotes their composition.

Graphs. Graphs in this paper have finite vertex sets and no loops or parallel edges. Let $G = (V, E)$ be a graph. A *clique* in G is a set of pairwise adjacent vertices. An *independent set* in G is a set of pairwise non-adjacent vertices. For $X \subseteq V(G)$, we denote the subgraph of G induced by X as $G[X]$, that is $G[X] = (X, \{uv\colon u, v \in X \text{ and } uv \in E\})$. We denote by $N(X)$ vertices of $G \setminus X$ with a neighbor in X, and $N[X] = N(X) \cup X$. For disjoint $X, Y \subseteq V(G)$, we say that X is *complete* to Y if every vertex in X is adjacent to every vertex in Y, and X is *anticomplete* to Y if there are no edges between X and Y. For given graphs G and H, we say that G is *H-free* if G does not contain H as an induced subgraph. We let P_n, C_n, and K_n denote the chordless path, chordless cycle, and the complete graph on n vertices. A *coloring* of G is a mapping $\varphi\colon V \to \{1, 2, \ldots\}$ that gives each vertex $u \in V$ a *color* $\varphi(u)$ in such a way that, for every two adjacent vertices u and v in G, we have that $\varphi(u) \neq \varphi(v)$. For $k \geq 1$, a coloring φ of G is a *k-coloring* if $\varphi(u) \in \{1, \ldots, k\}$ for every $u \in V$, and a graph is *k-colorable* if it admits a k-coloring. The *chromatic number* $\chi(G)$ is the smallest k for which G is k-colorable. A k-coloring of G is *equitable* if $||\varphi^{-1}(i)| - |\varphi^{-1}(j)|| \leq 1$ for every $i, j \in [k]$. Equivalently, G is equitably k-colorable if it is k-colorable and every color is used either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ times where $n = |V(G)|$. The *equitable chromatic number* $\chi_=(G)$ is the smallest k for which G is equitably k-colorable.

Clique-width and linear clique-width. For a fixed $w \geq 1$, a w-labelled graph is a graph G together with a labelling function $\mathsf{lab} : V(G) \to [w]$. Equivalently, it may be described by the partition $V(G) = C_1 \uplus \cdots \uplus C_w$, where $C_i = \mathsf{lab}^{-1}(i)$ is the set of vertices of label i. On w-labelled graphs we consider the following operations:

- for each $i \in [w]$, the introduce-operation $i(v)$, which constructs a single-vertex graph whose unique vertex v has label i;
- the union-operation $\mathsf{union}(\cdot, \cdot)$, which forms the disjoint union of two w-labelled graphs;
- for distinct $i, j \in [w]$, the relabel-operation $\mathsf{rename}_{i \to j}(\cdot)$, which changes the label of every vertex of label i to j;
- for distinct $i, j \in [w]$, the join-operation $\mathsf{join}_{i,j}(\cdot)$, which adds all edges with one endpoint of label i and the other of label j.

A *w-expression* is any valid term built from these operations. If μ is a w-expression, we write G_μ for the w-labelled graph constructed by μ, and lab_μ for its labelling function. The *clique-width* of an unlabelled graph G, denoted by $\mathsf{cw}(G)$, is the least integer w such that there exists a w-expression whose underlying unlabelled graph is G. Associated with every w-expression μ is its syntax tree T_μ in the natural way. For any node $t \in V(T_\mu)$, the subtree rooted at t induces a subexpression μ_t. Whenever μ is fixed, we write $G_t \coloneqq G_{\mu_t}, V_t \coloneqq V(G_t), E_t \coloneqq E(G_t), \mathsf{lab}_t \coloneqq \mathsf{lab}_{\mu_t}$. We use this notation to refer conveniently to the labelled graph associated with a node of the syntax tree of μ. A clique-expression μ is *linear* if in every union-operation, the second operand is a single-vertex w-labelled graph. The *linear clique-width* of an unlabelled graph G, denoted by $\mathsf{lcw}(G)$, is the least integer w such that G is the underlying unlabelled graph of some linear w-expression.

Complexity. For basics of parameterized complexity, see [11].

3 Graphs of bounded clique-width: The proofs of Theorem 1 and Theorem 2

We first prove Theorem 1. We use the clique-width operations introduced in Section 2. In particular, a w-expression is a term whose leaves are single-vertex w-labelled graphs and whose internal nodes apply one of the operations $\mathsf{union}(\cdot,\cdot)$, $\mathsf{join}_{i,j}(\cdot)$ $(i \neq j)$, and $\mathsf{rename}_{i\to j}(\cdot)$ $(i \neq j)$. We introduce the signature notation used by the dynamic programming algorithm.

▶ **Definition 5** (X-signature). *Let $k, w \in \mathbb{N}$. Let X be a subexpression of a given w-expression, and let G_X be the w-labelled graph constructed by X, with vertex set V_X and labelling function* $\mathsf{lab}_X : V_X \to [w]$. *An X-*signature *is a pair $\sigma = (\mathbf{S}, \mathbf{a})$ where $\mathbf{S} = (S_1, \ldots, S_w)$ with $S_\ell \subseteq [k]$ for all $\ell \in [w]$, and $\mathbf{a} = (a_1, \ldots, a_k) \in \{0, 1, \ldots, |V_X|\}^k$.*

We say that a proper k-coloring $\varphi : V_X \to [k]$ *is represented by* $\sigma = (\mathbf{S}, \mathbf{a})$ if

- $S_\ell = \{\, c \in [k] : \exists v \in V_X$ with $\mathsf{lab}_X(v) = \ell$ and $\varphi(v) = c\,\}$ for all $\ell \in [w]$, and
- $a_c = |\varphi^{-1}(c)|$ for all $c \in [k]$.

The key point is that this signature stores precisely the information needed by the clique-width operations: for join-operations it is enough to know which colors appear on each label class, while for equitable coloring one must additionally know the global sizes of the color classes.

▶ **Lemma 6** (♣). *Let $n = kq + r$ with $q = \lfloor n/k \rfloor$ and $0 \le r < k$. If $X^\star$ is a w-expression for G, then G admits an equitable k-coloring if and only if there exists an $X^\star$-signature $\sigma = (\mathbf{S}, \mathbf{a})$ represented by a proper k-coloring of G such that $a_c \in \{q, q+1\}$ for every $c \in [k]$, and $|\{c \in [k] : a_c = q+1\}| = r$.*

▶ **Definition 7.** *Let $X = \mathsf{rename}_{i\to j}(Y)$ with $i \neq j$. For a Y-signature $\sigma = (\mathbf{S}, \mathbf{a})$ and an X-signature $\sigma' = (\mathbf{S}', \mathbf{a}')$, we write $\sigma \mapsto_{i\to j} \sigma'$ if $\mathbf{a}' = \mathbf{a}$ and*

$$S'_\ell = \begin{cases} S_\ell & \text{if } \ell \notin \{i, j\}, \\ \emptyset & \text{if } \ell = i, \\ S_i \cup S_j & \text{if } \ell = j. \end{cases}$$

We can now prove the main result of this section.

▶ **Theorem 8.** *Fix an integer $k \ge 1$. There exists an algorithm that, given an n-vertex graph G together with a w-expression $\mathcal{E}$ of size m whose underlying unlabelled graph is G, computes the number of equitable k-colorings of G in time $2^{O(k\cdot w)} \cdot m \cdot n^{O(k)}$ (up to polynomial factors for arithmetic on integers of size at most k^n). In particular, for standard polynomial-size encodings of clique-width expressions, the running time is $2^{O(k\cdot w)} \cdot n^{O(k)}$.*

Proof. Let $\mathcal{E}$ be the given w-expression, and let $X^\star := \mathcal{E}$. For every subexpression X of $\mathcal{E}$, we denote by G_X the w-labelled graph constructed by X, with vertex set V_X and labelling function $\mathsf{lab}_X : V_X \to [w]$. For an X-signature $\sigma = (\mathbf{S}, \mathbf{a})$, we write $\mathbf{S} = (S_1, \dots, S_w)$ and $\mathbf{a} = (a_1, \dots, a_k)$. Recall that a proper k-coloring $\varphi : V_X \to [k]$ is *represented by* σ if

- $S_\ell = \{ c \in [k] : \exists v \in V_X \text{ with } \mathsf{lab}_X(v) = \ell \text{ and } \varphi(v) = c \}$ for all $\ell \in [w]$, and
- $a_c = |\varphi^{-1}(c)|$ for all $c \in [k]$.

For every subexpression X and every X-signature σ, define $\mathrm{count}[X, \sigma] \in \mathbb{Z}_{\geq 0}$ to be the number of proper k-colorings φ of G_X that are represented by σ. Equivalently, $\mathrm{count}[X, \sigma] = \bigl|\{\varphi : \varphi \text{ is a proper } k\text{-coloring of } G_X \text{ and is represented by } \sigma\}\bigr|$. We compute all values $\mathrm{count}[X, \sigma]$ by dynamic programming over the subexpressions X of $\mathcal{E}$, proceeding from smaller subexpressions to larger ones. We show, by induction on the structure of X, that for every subexpression X of $\mathcal{E}$ and every X-signature σ, the value $\mathrm{count}[X, \sigma]$ equals the number of proper k-colorings of G_X represented by σ. The base case is when X is a leaf. For the induction step, we assume the claim holds for the immediate proper subexpressions of X, and prove it for X according to whether the last operation of X is a union, join, or relabel operation.

Leaf. Suppose that X is a leaf subexpression creating a single vertex v with label $\ell \in [w]$. For $c \in [k]$, let $\mathbf{e}_c \in \{0,1\}^k$ be the vector whose cth coordinate is 1 and whose other coordinates are 0. For each color $c \in [k]$, define the signature $\sigma^{(\ell,c)} = (\mathbf{S}, \mathbf{a})$ by $S_\ell = \{c\}$, $S_{\ell'} = \emptyset$ for all $\ell' \in [w] \setminus \{\ell\}$, and $\mathbf{a} = \mathbf{e}_c$. We set

$$\mathrm{count}[X, \sigma] := \begin{cases} 1 & \text{if } \sigma = \sigma^{(\ell,c)} \text{ for some } c \in [k], \\ 0 & \text{otherwise.} \end{cases}$$

This is correct because G_X has a single vertex and no edges, so for each $c \in [k]$ there is exactly one proper coloring with $\varphi(v) = c$, and it represents exactly $\sigma^{(\ell,c)}$; no other signature can be represented.

Union. Assume $X = \mathsf{union}(Y, Z)$. By the semantics of union, the graphs G_Y and G_Z are vertex-disjoint induced subgraphs of G_X, there are no edges between V_Y and V_Z, and

$$V_X = V_Y \uplus V_Z, \quad E(G_X) = E(G_Y) \uplus E(G_Z), \quad \mathsf{lab}_X|_{V_Y} = \mathsf{lab}_Y, \mathsf{lab}_X|_{V_Z} = \mathsf{lab}_Z. \tag{1}$$

Fix an X-signature $\sigma = (\mathbf{S}, \mathbf{a})$. We define $\mathrm{count}[X, \sigma]$ by

$$\mathrm{count}[X, \sigma] := \sum_{\mathbf{a}^Y + \mathbf{a}^Z = \mathbf{a}} \ \sum_{\substack{\mathbf{S}^Y, \mathbf{S}^Z: \\ S_\ell = S_\ell^Y \cup S_\ell^Z \ \forall \ell \in [w]}} \mathrm{count}[Y, (\mathbf{S}^Y, \mathbf{a}^Y)] \cdot \mathrm{count}[Z, (\mathbf{S}^Z, \mathbf{a}^Z)]. \tag{2}$$

Let us justify (2). Let φ be any proper k-coloring of G_X. Define $\varphi_Y := \varphi|_{V_Y}$ and $\varphi_Z := \varphi|_{V_Z}$. Then φ_Y and φ_Z are proper colorings of G_Y and G_Z, respectively, because G_Y and G_Z are induced subgraphs of G_X by (1). Conversely, for any pair (ψ_Y, ψ_Z) where ψ_Y is a proper coloring of G_Y and ψ_Z is a proper coloring of G_Z, there is a unique coloring ψ of G_X obtained by combining them:

$$\psi(v) = \begin{cases} \psi_Y(v) & v \in V_Y, \\ \psi_Z(v) & v \in V_Z, \end{cases}$$

and ψ is proper since there are no edges between V_Y and V_Z. Now let $\sigma_Y = (\mathbf{S}^Y, \mathbf{a}^Y)$ and $\sigma_Z = (\mathbf{S}^Z, \mathbf{a}^Z)$ be the signatures represented by φ_Y and φ_Z, respectively. We claim that φ is represented by $\sigma = (\mathbf{S}, \mathbf{a})$ if and only if

$$\mathbf{a} = \mathbf{a}^Y + \mathbf{a}^Z \quad \text{and} \quad S_\ell = S_\ell^Y \cup S_\ell^Z \quad \text{for all } \ell \in [w]. \tag{3}$$

Indeed, fix any color $c \in [k]$. Since $V_X = V_Y \uplus V_Z$, we have $|\varphi^{-1}(c)| = |\varphi_Y^{-1}(c)| + |\varphi_Z^{-1}(c)|$, hence $a_c = a_c^Y + a_c^Z$, which is exactly the cth coordinate of $\mathbf{a} = \mathbf{a}^Y + \mathbf{a}^Z$.

Next fix any label $\ell \in [w]$. Using (1), the set of vertices of label ℓ in G_X is the disjoint union of the label-ℓ vertices in G_Y and in G_Z, so the set of colors used by φ on label ℓ is exactly the union of the sets of colors used by φ_Y and φ_Z on label ℓ, that is, $S_\ell = S_\ell^Y \cup S_\ell^Z$. This proves (3), and the converse direction is immediate from the definitions. Finally, for every pair (σ_Y, σ_Z) satisfying (3), the number of proper colorings φ of G_X represented by σ whose restrictions induce these particular signatures equals $\mathrm{count}[Y, \sigma_Y] \cdot \mathrm{count}[Z, \sigma_Z]$, since the choice of φ_Y and the choice of φ_Z are independent and determine φ uniquely. Summing over all such pairs yields (2).

Join. Assume $X = \mathsf{join}_{i,j}(Y)$ with $i \neq j$. Then $V_X = V_Y$, $\mathsf{lab}_X = \mathsf{lab}_Y$, and G_X is obtained from G_Y by adding every edge between a vertex of label i and a vertex of label j. Fix an X-signature $\sigma = (\mathbf{S}, \mathbf{a})$. We set

$$\mathrm{count}[X, \sigma] := \begin{cases} \mathrm{count}[Y, \sigma] & \text{if } S_i \cap S_j = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Let us justify (4). Let φ be a proper k-coloring of G_Y represented by σ. Then φ is also a coloring of G_X (same vertex set). If $S_i \cap S_j = \emptyset$, then no color c is used on both labels i and j by φ; hence for every new edge uv added by $\mathsf{join}_{i,j}$ (with $\mathsf{lab}_X(u) = i$ and $\mathsf{lab}_X(v) = j$) we have $\varphi(u) \neq \varphi(v)$. Thus φ remains proper in G_X. Conversely, if $S_i \cap S_j \neq \emptyset$, choose $c \in S_i \cap S_j$. By definition of S_i and S_j, there exist vertices u, v with $\mathsf{lab}_X(u) = i$ and $\mathsf{lab}_X(v) = j$ and $\varphi(u) = \varphi(v) = c$, and $\mathsf{join}_{i,j}$ adds the edge uv, contradicting properness in G_X. Hence no proper coloring of G_X can be represented by σ, so $\mathrm{count}[X, \sigma] = 0$. This proves (4).

Relabel. Assume $X = \mathsf{rename}_{i \to j}(Y)$ with $i \neq j$. Then $V_X = V_Y$, $E(G_X) = E(G_Y)$, and lab_X is obtained from lab_Y by changing every label-i vertex into label j.

Fix an X-signature $\sigma' = (\mathbf{S}', \mathbf{a})$. We set

$$\mathrm{count}[X, \sigma'] := \sum_{\sigma\colon\ \sigma \mapsto_{i \to j} \sigma'} \mathrm{count}[Y, \sigma], \tag{5}$$

where $\sigma \mapsto_{i \to j} \sigma'$ is the relation of Definition 7.

Let us justify (5). Since $E(G_X) = E(G_Y)$, a mapping $\varphi : V_X \to [k]$ is a proper coloring of G_X if and only if it is a proper coloring of G_Y. Thus relabeling does not change the set of proper colorings; it only changes which colors are recorded in which label-set S_ℓ. For a fixed proper coloring φ, let σ be its Y-signature and σ' its X-signature. Then necessarily $\mathbf{a}$ is the same in both signatures, and the label-sets satisfy exactly the update in Definition 7, i.e. $\sigma \mapsto_{i \to j} \sigma'$. Therefore, for each fixed σ', the set of proper colorings of G_X represented by σ' is the disjoint union over all Y-signatures σ mapping to σ' of the sets of proper colorings represented by σ in G_Y. Taking cardinalities yields (5). We compute $\mathrm{count}[X^\star, \sigma]$ for all signatures σ of $X^\star$. Let $n = kq + r$ with $0 \leq r < k$. By Lemma 6, a proper k-coloring of G

is equitable if and only if its size vector $\mathbf{a}$ satisfies $a_c \in \{q, q+1\}$ for all $c \in [k]$ and exactly r indices satisfy $a_c = q+1$. Hence the number of equitable k-colorings of G equals the sum of $\mathrm{count}[X^\star, \sigma]$ over all signatures $\sigma = (\mathbf{S}, \mathbf{a})$ of $X^\star$ with this property.

It remains to discuss the running time. For fixed k, represent each $S_\ell \subseteq [k]$ by a k-bit mask, so unions and intersections take constant time. The number of possible choices of $\mathbf{S}$ is $(2^k)^w$, and the number of possible vectors $\mathbf{a} \in \{0, \dots, n\}^k$ is $(n+1)^k$. Thus for each subexpression X the table contains at most $M = (2^k)^w \cdot (n+1)^k$ entries. For join nodes, each entry can be computed in constant time from the child table, hence these subexpressions can be processed in $O(M)$ time. For relabel nodes, fix a target signature $\sigma' = (\mathbf{S}', \mathbf{a})$. By Definition 7, a preimage under $\mapsto_{i \to j}$ can exist only if $S'_i = \emptyset$; otherwise $\mathrm{count}[X, \sigma'] = 0$. Assume therefore that $S'_i = \emptyset$. Then the only possible preimages are obtained by choosing, for every color in S'_j, whether it came from old label i, old label j, or both, while all other coordinates are forced. Hence the number of preimages is at most $3^{|S'_j|} \le 3^k$, which is constant for fixed k. Therefore relabel subexpressions also take $O(M)$ time. For a union subexpression, we may evaluate (2) by iterating over all pairs of child signatures (at most M^2 pairs), computing the unique combined signature, and adding the product of the two child counts; this takes $O(M^2)$ time. If m denotes the size of the given expression $\mathcal{E}$, then the number of subexpressions of $\mathcal{E}$ is $O(m)$. Therefore the arithmetic-operation count is $O(m \cdot M^2) = 2^{O(k \cdot w)} \cdot m \cdot n^{O(k)}$. Finally, $\mathrm{count}[X, \sigma] \le k^{|V_X|} \le k^n$, so for fixed k each table entry has $O(n)$ bits, and exact integer arithmetic contributes only a polynomial factor in n. This completes the proof of Theorem 8. ◀

We deduce the following.

▶ **Corollary 9.** *There exists an algorithm that, given an integer $k \ge 1$ and an n-vertex graph G together with a w-expression for G, decides whether G admits an equitable k-coloring in time $2^{O(k \cdot w)} \cdot n^{O(k)}$, assuming a standard polynomial-size encoding of the expression.*

Proof. Compute the number of equitable k-colorings using Theorem 8 and accept if and only if the count is nonzero. ◀

For the remainder of the section, we provide a SETH-based lower bound for EQUITABLE k-COLORING on graphs of bounded clique-width. We first need the following result of [22].

▶ **Theorem 10** (Lampis [22]). *For every fixed $k \ge 3$ and every $\varepsilon > 0$, if there exists an algorithm solving k-COLORING in time $O^*((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)})$ on an input graph G, then SETH is false.*

▶ **Theorem 11.** *Let G be a graph. For every fixed $k \ge 3$ and every $\varepsilon > 0$, unless SETH fails, EQUITABLE k-COLORING cannot be solved in time $O^*\big((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)}\big)$, where $\mathsf{cw}(G)$ denotes the clique-width of G.*

Proof. We proceed by a reduction from k-COLORING that increases clique-width by at most one. Suppose for a contradiction that there exists an algorithm $\mathcal{A}$ which, on input a graph H, decides EQUITABLE k-COLORING in time $O^*((2^k - 2 - \varepsilon)^{\mathsf{cw}(H)})$. Let G be an instance of k-COLORING and put $n \coloneqq |V(G)|$. Let I be an independent set on a fresh vertex set U with $|U| = (k-1)n$, and define $G' \coloneqq G \uplus I$. Note that $V(G') = V(G) \uplus U$ and $E(G') = E(G)$.

▷ Claim 12 (♣). G is properly k-colorable if and only if G' admits an equitable proper k-coloring.

▷ Claim 13 (♣). $\mathsf{cw}(G') \le \mathsf{cw}(G) + 1$.

We are now ready to complete the reduction. Given G, we construct G' in polynomial time and run $\mathcal{A}$ on G'. By Claim 12, $\mathcal{A}$ answers YES on G' if and only if G is k-colorable. Moreover, by Claim 13, $\mathsf{cw}(G') \leq \mathsf{cw}(G) + 1$. Hence the running time of $\mathcal{A}$ on G' is $O^*\big((2^k - 2 - \varepsilon)^{\mathsf{cw}(G')}\big) \leq O^*\big((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)+1}\big)$. Since for fixed k and $\varepsilon > 0$, the factor $(2^k - 2 - \varepsilon)$ is a constant absorbed by O^*, it follows that $\mathcal{A}$ would yield an algorithm for k-Coloring running in time $O^*\big((2^k - 2 - \varepsilon)^{\mathsf{cw}(G)}\big)$. This contradicts the result of Theorem 10, and hence completes the proof of Theorem 11. ◀

4 Graphs of bounded linear clique-width: The proof of Theorem 3

We prove Theorem 3 in this section. Since the dynamic programming algorithm follows a given linear w-expression from left to right, we work with the intermediate labelled graphs obtained after executing prefixes of the expression. Accordingly, we view a linear w-expression as a sequence of operations evaluated from left to right, starting from the empty w-labelled graph. To distinguish this from the term representation used earlier for clique-width expressions, we denote the creation of a new single vertex of label i by $\mathsf{add}(i)$, while the operations corresponding to $\mathsf{join}_{i,j}$ and $\mathsf{rename}_{i\to j}$ are written as $\eta(i,j)$ and $\rho(i \to j)$, respectively. We write such a sequence as $\Pi = \pi_1\pi_2\cdots\pi_L$.

Intermediate graphs. Let $\Pi = \pi_1\pi_2\cdots\pi_L$ be a fixed linear w-expression. For each $t \in \{0, 1, \ldots, L\}$, let $\mathbf{F}_t = (F_t; C_1^{(t)}, \ldots, C_w^{(t)})$ denote the intermediate labelled graph obtained after executing the prefix $\pi_1 \cdots \pi_t$, with $\mathbf{F}_0$ the empty w-labelled graph. Thus $\mathbf{F}_L$ is the w-labelled graph produced by the whole expression Π. We write $V_t \coloneqq V(F_t)$, $E_t \coloneqq E(F_t)$, and $n_t \coloneqq |V_t|$.

▶ **Definition 14.** *For distinct $p, q \in [w]$, let $r_{p\to q} : [w] \to [w]$ be the map defined by $r_{p\to q}(p) = q, r_{p\to q}(x) = x$ for all $x \neq p$. For $t < s$, define the map $\tau_{t,s} : [w] \to [w]$ by $\tau_{t,s} \coloneqq f_s \circ f_{s-1} \circ \cdots \circ f_{t+1}$, where for each $r \in \{t+1, \ldots, s\}$,*

$$f_r = \begin{cases} r_{p\to q} & \text{if } \pi_r = \rho(p \to q), \\ \mathrm{id}_{[w]} & \text{if } \pi_r = \mathsf{add}(\cdot) \text{ or } \pi_r = \eta(\cdot,\cdot). \end{cases}$$

In other words, $\tau_{t,s}(i)$ is the label at time s of a vertex that had label i at time t. We also let $\tau_{t,t} \coloneqq \mathrm{id}_{[w]}$.

▶ **Lemma 15 (♣).** *Fix $t < s$ and let $v \in V_t$ be any vertex. If $v \in C_i^{(t)}$, then in $\mathbf{F}_s$ the vertex v has label $\tau_{t,s}(i)$, that is, $v \in C_{\tau_{t,s}(i)}^{(s)}$.*

▶ **Definition 16** (Live labels). *Let $t \in \{0, \ldots, L\}$ and $i \in [w]$. We say that label i is* live at time t *if $C_i^{(t)} \neq \emptyset$ and there exists $s \in \{t+1, \ldots, L\}$ such that $\pi_s = \eta(p, q)$ with $p \neq q$ and either $\tau_{t,s-1}(i) = p, C_q^{(s-1)} \neq \emptyset$, or $\tau_{t,s-1}(i) = q, C_p^{(s-1)} \neq \emptyset$. We write $\mathsf{Live}(t) \subseteq [w]$ for the set of live labels at time t.*

Intuitively, a label i is live at time t precisely when the set of colors currently used on vertices of label i may still affect properness at some future join operation; labels that are not live can therefore be ignored by the dynamic programming algorithm.

▷ Claim 17 (♣). Given a linear w-expression $\Pi = \pi_1 \cdots \pi_L$, one can compute $\mathsf{Live}(t)$ for all $t \in \{0, \ldots, L\}$ in time $O(L^2 w^2)$.

▶ **Definition 18.** *A proper k-coloring $\varphi_t : V_t \to [k]$ of F_t is* extendible *if there exists a proper k-coloring $\varphi : V_L \to [k]$ of the graph F_L produced by the whole linear expression Π such that $\varphi|_{V_t} = \varphi_t$.*

▶ **Lemma 19 (♣).** *Let $t \in \{0, \dots, L\}$ and let φ_t be an extendible proper k-coloring of F_t. Then for every live label $i \in \mathsf{Live}(t)$ the set of colors used on $C_i^{(t)}$, $S_i(\varphi_t) \coloneqq \{\, c \in [k] : \exists v \in C_i^{(t)}$ with $\varphi_t(v) = c \,\}$, satisfies $S_i(\varphi_t) \neq \emptyset$ and $S_i(\varphi_t) \neq [k]$.*

▷ Claim 20 (♣). If $\pi_t = \mathsf{add}(i)$, then $\mathsf{Live}(t) \setminus \{i\} = \mathsf{Live}(t-1) \setminus \{i\}$.

▷ Claim 21 (♣). If $\pi_t = \mathsf{add}(i)$ and $i \in \mathsf{Live}(t)$ but $i \notin \mathsf{Live}(t-1)$, then $C_i^{(t-1)} = \emptyset$.

▷ Claim 22 (♣). If $\pi_t = \eta(p, q)$, then $\mathsf{Live}(t) \subseteq \mathsf{Live}(t-1)$.

▷ Claim 23 (♣). If $\pi_t = \rho(p \to q)$, then for every $\ell \in [w] \setminus \{p, q\}$ we have $\ell \in \mathsf{Live}(t)$ if and only if $\ell \in \mathsf{Live}(t-1)$.

▷ Claim 24 (♣). If $\pi_t = \rho(p \to q)$ and $q \in \mathsf{Live}(t)$ but $q \notin \mathsf{Live}(t-1)$, then $C_q^{(t-1)} = \emptyset$.

▷ Claim 25 (♣). If $\pi_t = \rho(p \to q)$, $q \in \mathsf{Live}(t)$, and $C_p^{(t-1)} \neq \emptyset$, then $p \in \mathsf{Live}(t-1)$.

We are now ready to prove the main result of this section.

▶ **Theorem 26.** *There exists an algorithm that, given an integer $k \geq 1$ and an n-vertex graph G together with a linear w-expression constructing G, computes the number of equitable k-colorings of G in time $\max\{1, 2^k - 2\}^w \cdot n^{k+O(1)}$. In particular, for every fixed integer k, counting equitable k-colorings is polynomial-time solvable on graph classes of bounded linear clique-width, given a linear clique-width expression.*

Proof. We first handle the case $k = 1$ separately. A proper 1-coloring of G exists if and only if G has no edges, and in that case it is unique. Thus the number of equitable 1-colorings is 1 if $E(G) = \emptyset$, and 0 otherwise. This can be checked in polynomial time from the given linear expression. In the remainder of the proof, assume that $k \geq 2$. Let $\Pi = \pi_1\pi_2 \cdots \pi_L$ be the given linear w-expression, where L is its length. Let $\mathbf{F}_t = (F_t; C_1^{(t)}, \dots, C_w^{(t)})$, for $t = 0, 1, \dots, L$, be the intermediate labelled graphs, with F_0 empty and $F_L = G$. Let $V_t \coloneqq V(F_t)$ and $n_t \coloneqq |V_t|$. By Claim 17, we may assume that $\mathsf{Live}(t)$ is known for all t. We use a distinguished symbol $\perp$ to denote that a label is not tracked. For each $t \in \{0, \dots, L\}$, a *state at time* t is a pair $\sigma = (\mathbf{S}, \mathbf{a})$ where $\mathbf{a} = (a_1, \dots, a_k) \in \{0, 1, \dots, n_t\}^k$, $\sum_{c=1}^{k} a_c = n_t$, and $\mathbf{S} = (S_1, \dots, S_w)$ satisfies

$$S_i = \begin{cases} \perp & \text{if } i \notin \mathsf{Live}(t), \\ \text{an element of } 2^{[k]} \setminus \{\emptyset, [k]\} & \text{if } i \in \mathsf{Live}(t). \end{cases}$$

For a proper k-coloring $\varphi_t : V_t \to [k]$ of F_t, its *projected signature at time* t is the pair $\sigma = (\mathbf{S}, \mathbf{a})$ defined by

$$a_c \coloneqq |\varphi_t^{-1}(c)| \quad (c \in [k]), \qquad S_i \coloneqq \begin{cases} \{c \in [k] : \exists v \in C_i^{(t)} \text{ with } \varphi_t(v) = c\} & \text{if } i \in \mathsf{Live}(t), \\ \perp & \text{if } i \notin \mathsf{Live}(t). \end{cases}$$

We now construct the DP table. For each t and each state σ at time t, let $\mathrm{DP}[t, \sigma] \in \mathbb{Z}_{\geq 0}$ be the number of proper k-colorings φ_t of F_t whose projected signature equals σ.

We do not store colorings whose projected signature is not a state at time t; equivalently, we discard colorings for which some live label uses all k colors. By Lemma 19, such colorings are not extendible to a proper coloring of F_L, and hence discarding them cannot affect the final count. Moreover, once a coloring is discarded for this reason, it can never later contribute to a valid state: if some live label uses all k colors at time t, then under subsequent add and relabel operations the corresponding live descendant label still uses all k colors, while when the witnessing future join is performed the coloring becomes improper.

At $t = 0$ the graph is empty, so $\mathsf{Live}(0) = \emptyset$. There is exactly one proper coloring, namely the empty map, whose projected signature is the unique state σ_0 with $\mathbf{a} = \mathbf{0}$ and $S_i = \perp$ for all i. Set $\mathrm{DP}[0, \sigma_0] = 1$ and $\mathrm{DP}[0, \sigma] = 0$ for all other states σ. Assume that $\mathrm{DP}[t-1, \cdot]$ has been computed. Before processing step t, initialize $\mathrm{DP}[t, \sigma] := 0$ for every state σ at time t. We now describe the DP update from time $t-1$ to time t, depending on the operation π_t.

Case 1: $\pi_t = \mathsf{add}(i)$. Let v be the new vertex, so $V_t = V_{t-1} \uplus \{v\}$ and $v \in C_i^{(t)}$. Fix a state $\sigma^- = (\mathbf{S}^-, \mathbf{a}^-)$ with $\mathrm{DP}[t-1, \sigma^-] > 0$. For each color $c \in [k]$, define $\mathbf{a}^+$ by $a_c^+ = a_c^- + 1$ and $a_{c'}^+ = a_{c'}^-$ for $c' \neq c$. Define $\mathbf{S}^+$ by:
- for every $\ell \in \mathsf{Live}(t) \setminus \{i\}$, set $S_\ell^+ := S_\ell^-$, which is well-defined by Claim 20;
- for every $\ell \notin \mathsf{Live}(t)$, set $S_\ell^+ := \perp$;
- for label i, set

$$S_i^+ := \begin{cases} S_i^- \cup \{c\} & \text{if } i \in \mathsf{Live}(t) \cap \mathsf{Live}(t-1), \\ \{c\} & \text{if } i \in \mathsf{Live}(t) \setminus \mathsf{Live}(t-1), \\ \perp & \text{if } i \notin \mathsf{Live}(t), \end{cases}$$

where the second case is valid by Claim 21.

If $i \in \mathsf{Live}(t)$ and $S_i^+ = [k]$, discard this update. Otherwise set $\mathrm{DP}[t, (\mathbf{S}^+, \mathbf{a}^+)] := \mathrm{DP}[t, (\mathbf{S}^+, \mathbf{a}^+)] + \mathrm{DP}[t-1, (\mathbf{S}^-, \mathbf{a}^-)]$. This is correct since proper colorings of F_t are in bijection with pairs consisting of a proper coloring of F_{t-1} and a color for the new isolated vertex v, and the projected signature changes exactly as above.

Case 2: $\pi_t = \eta(p, q)$ with $p \neq q$. If $C_p^{(t-1)} = \emptyset$ or $C_q^{(t-1)} = \emptyset$, then $F_t = F_{t-1}$. By Claim 22, we have $\mathsf{Live}(t) \subseteq \mathsf{Live}(t-1)$. For each state $\sigma^- = (\mathbf{S}^-, \mathbf{a})$, define $\sigma^+ = (\mathbf{S}^+, \mathbf{a})$ by $S_\ell^+ = S_\ell^-$ for $\ell \in \mathsf{Live}(t)$, and $S_\ell^+ = \perp$ otherwise, and update $\mathrm{DP}[t, \sigma^+] := \mathrm{DP}[t, \sigma^+] + \mathrm{DP}[t-1, \sigma^-]$. This simply projects the same proper coloring onto the smaller live set. Assume now that $C_p^{(t-1)} \neq \emptyset$ and $C_q^{(t-1)} \neq \emptyset$. Then $p, q \in \mathsf{Live}(t-1)$, because the current join step itself witnesses liveness at time $t-1$. Fix a state $\sigma^- = (\mathbf{S}^-, \mathbf{a})$ with $\mathrm{DP}[t-1, \sigma^-] > 0$. If $S_p^- \cap S_q^- \neq \emptyset$, add nothing. If $S_p^- \cap S_q^- = \emptyset$, define $\sigma^+ = (\mathbf{S}^+, \mathbf{a})$ by $S_\ell^+ = S_\ell^-$ for $\ell \in \mathsf{Live}(t)$, and $S_\ell^+ = \perp$ otherwise, and update $\mathrm{DP}[t, \sigma^+] := \mathrm{DP}[t, \sigma^+] + \mathrm{DP}[t-1, \sigma^-]$. Indeed, when both label classes are nonempty, a proper coloring survives the join if and only if no color is used on both sides, that is, if and only if $S_p^- \cap S_q^- = \emptyset$.

Case 3: $\pi_t = \rho(p \to q)$ with $p \neq q$. Fix a state $\sigma^- = (\mathbf{S}^-, \mathbf{a})$ with $\mathrm{DP}[t-1, \sigma^-] > 0$. The relabel operation does not change the graph F_{t-1}, only the labels, so the color-count vector remains $\mathbf{a}$. For every $\ell \in \mathsf{Live}(t) \setminus \{q\}$, set $S_\ell^+ := S_\ell^-$. This is well-defined for $\ell \notin \{p, q\}$ by Claim 23, and label p cannot belong to $\mathsf{Live}(t)$ because after the relabel its class is empty. For every $\ell \notin \mathsf{Live}(t)$, set $S_\ell^+ := \perp$.

For label q, define

$$S_q^+ := \begin{cases} \perp & \text{if } q \notin \mathsf{Live}(t), \\ S_q^- \cup S_p^- & \text{if } q \in \mathsf{Live}(t) \cap \mathsf{Live}(t-1) \text{ and } p \in \mathsf{Live}(t-1), \\ S_q^- & \text{if } q \in \mathsf{Live}(t) \cap \mathsf{Live}(t-1) \text{ and } p \notin \mathsf{Live}(t-1), \\ S_p^- & \text{if } q \in \mathsf{Live}(t) \setminus \mathsf{Live}(t-1). \end{cases}$$

These cases are exhaustive: if $q \notin \mathsf{Live}(t)$, we are in the first case; otherwise either $q \in \mathsf{Live}(t-1)$ or $q \notin \mathsf{Live}(t-1)$, and in the former situation either $p \in \mathsf{Live}(t-1)$ or $p \notin \mathsf{Live}(t-1)$. In the third case, if $C_p^{(t-1)} \neq \emptyset$, then Claim 25 would imply $p \in \mathsf{Live}(t-1)$, a contradiction. Hence $C_p^{(t-1)} = \emptyset$, so no vertices are moved from label p to label q, and therefore $S_q^+ = S_q^-$. In the last case, Claim 24 gives $C_q^{(t-1)} = \emptyset$. Since $q \in \mathsf{Live}(t)$, we have $C_q^{(t)} \neq \emptyset$ by definition of liveness. Because $\rho(p \to q)$ relabels all vertices of $C_p^{(t-1)}$ to label q, we have $C_q^{(t)} = C_q^{(t-1)} \cup C_p^{(t-1)} = C_p^{(t-1)}$, and therefore $C_p^{(t-1)} \neq \emptyset$. Now Claim 25 implies $p \in \mathsf{Live}(t-1)$, so S_p^- is defined. If $q \in \mathsf{Live}(t)$ and $S_q^+ = [k]$, discard this update. Otherwise set $\mathrm{DP}[t, (\mathbf{S}^+, \mathbf{a})] := \mathrm{DP}[t, (\mathbf{S}^+, \mathbf{a})] + \mathrm{DP}[t-1, (\mathbf{S}^-, \mathbf{a})]$. This is correct because relabeling induces a bijection between proper colorings before and after the operation, and the projected signature changes exactly as above. Any update yielding $S_q^+ = [k]$ can be discarded by Lemma 19, since such a coloring cannot extend to a proper coloring of F_L. This completes the case analysis for the DP update. By induction on t, for every state σ at time t, the table value $\mathrm{DP}[t, \sigma]$ equals the number of proper k-colorings of F_t whose projected signature equals σ. At time L, we have $\mathsf{Live}(L) = \emptyset$, so every state has the form $(\bot, \dots, \bot; \mathbf{a})$, and the projected signature records only the color-class sizes. For $n = kq + r$ with $0 \leq r < k$, the number of equitable proper k-colorings of $G = F_L$ is obtained by summing $\mathrm{DP}[L, \sigma]$ over all states whose vector $\mathbf{a}$ satisfies $a_c \in \{q, q+1\}$ with $c \in [k]$ and exactly r coordinates are equal to $q + 1$.

For the running time, first consider the case $k \geq 2$. At each time t, there are at most $(2^k - 2)^{|\mathsf{Live}(t)|} \leq (2^k - 2)^w$ choices for the tracked label-sets and at most $(n+1)^k$ choices for $\mathbf{a}$, so the number of states is at most $(2^k - 2)^w (n+1)^k$. Each state update requires only polynomially many operations on labels and on subsets of $[k]$. Representing subsets of $[k]$ by k-bit vectors, every union, intersection, equality test, and membership test can be carried out in time polynomial in k. Hence each transition can be processed in time polynomial in k and w. Since a linear w-expression that constructs an n-vertex graph uses at most n labels, we have $w \leq n$, and therefore every polynomial factor in w is absorbed into $n^{O(1)}$. Moreover, for $k \geq 2$ and $w \geq 1$, the factor $(2^k - 2)^w$ dominates every polynomial factor in k. Thus, up to polynomial factors in n, the total running time is $(2^k - 2)^w (n+1)^k = (2^k - 2)^w \cdot n^{k+O(1)}$. Together with the polynomial-time treatment of the case $k = 1$, this yields the claimed bound $\max\{1, 2^k - 2\}^w \cdot n^{k+O(1)}$. This completes the proof of Theorem 26. ◀

▶ **Corollary 27.** *For every fixed integer k, deciding whether a graph admits an equitable k-coloring is polynomial-time solvable on graph classes of bounded linear clique-width, given a linear clique-width expression.*

Proof. Run the algorithm of Theorem 26 to compute the number of equitable proper k-colorings of G and accept if and only if the computed number is nonzero. ◀

5 Graphs with no long induced paths: The proof of Theorem 4

In this section we prove Theorem 4. Recall that a list 3-coloring φ of (G, L) is *equitable* if, writing $|V(G)| = n = 3q + r$ with $r \in \{0, 1, 2\}$, the multiset $\{|\varphi^{-1}(1)|, |\varphi^{-1}(2)|, |\varphi^{-1}(3)|\}$ equals $\{q, q, q\}$ if $r = 0$, $\{q+1, q, q\}$ if $r = 1$, and $\{q+1, q+1, q\}$ if $r = 2$. Let H be a multigraph. A *graph homomorphism* from G to H is a map $f\colon V(G) \to V(H)$ such that $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$. Thus, a proper 3-coloring is precisely a graph homomorphism to K_3. A *list H-coloring instance* is a pair (G, L), where $L\colon V(G) \to 2^{V(H)}$.

A *list H-coloring* of (G, L) is a graph homomorphism f from G to H such that $f(v) \in L(v)$ for all $v \in V(G)$. Following [17], we denote the set of list H-colorings of (G, L) by $\mathcal{LC}((G, L), H)$. For a given multigraph H, we define the partition polynomial $p_{(G,L)\to H}(x)$ by

$$p_{(G,L)\to H}(x) \coloneqq \sum_{f\in\mathcal{LC}((G,L),H)} \prod_{v\in V(G)} x_{f(v)}. \tag{6}$$

We first note how equitable list 3-colorings can be recovered from the coefficients of a suitable partition polynomial. The strategy is to compute the polynomial $p_{(G,L)\to K_3}$ using Theorem 30 and then extract from it exactly those coefficients corresponding to equitable color-class sizes.

▶ **Lemma 28** (♣)**.** *Let $H = K_3$ and let $p \coloneqq p_{(G,L)\to K_3}(x_1, x_2, x_3)$ be as in* (6)*. For every triple $(a, b, c) \in \mathbb{Z}^3_{\geq 0}$ satisfying $a + b + c = n$, the coefficient of $x_1^a x_2^b x_3^c$ in p equals the number of list 3-colorings f of (G, L) such that $|f^{-1}(1)| = a$, $|f^{-1}(2)| = b$, and $|f^{-1}(3)| = c$.*

▶ **Lemma 29** (♣)**.** *Let $n = 3q + r$ with $r \in \{0, 1, 2\}$, and let $p \coloneqq p_{(G,L)\to K_3}(x_1, x_2, x_3)$ be as in* (6)*. Then the number of equitable proper list 3-colorings of (G, L), denoted by $\mathbf{\Lambda}(G, L)$, is*

$$\mathbf{\Lambda}(G, L) = \begin{cases} [x_1^q x_2^q x_3^q]\, p, & r = 0,\\ [x_1^{q+1} x_2^q x_3^q]\, p + [x_1^q x_2^{q+1} x_3^q]\, p + [x_1^q x_2^q x_3^{q+1}]\, p, & r = 1,\\ [x_1^{q+1} x_2^{q+1} x_3^q]\, p + [x_1^{q+1} x_2^q x_3^{q+1}]\, p + [x_1^q x_2^{q+1} x_3^{q+1}]\, p, & r = 2. \end{cases}$$

We need the following consequence of [17].

▶ **Theorem 30** (Groenland, Okrasa, Rzążewski, Scott, Seymour, and Spirkl [17])**.** *Let $t \geq 4$ and let H be a multigraph such that for all distinct $h, h' \in V(H)$, $|N_H(h) \cap N_H(h')| \leq 1$. Then for every P_t-free graph G on n vertices and every list assignment $L : V(G) \to 2^{V(H)}$, the polynomial $p_{(G,L)\to H}(x)$ can be computed in time $2^{O(\sqrt{n \log n})} \cdot \mathrm{poly}(n)$.*

We are now ready to prove Theorem 4.

Proof of Theorem 4. If $t = 1$ or $t = 2$, then G is edgeless. In this case every list 3-coloring is simply a choice of an allowed color from $L(v)$ for each vertex v, and the number of equitable list 3-colorings can be computed in polynomial time by a dynamic program whose state records the three current color-class sizes. If $t = 3$, then G is a disjoint union of cliques. A clique component of size at least 4 has no 3-coloring, so in that case the answer is 0. Otherwise, each component has size at most 3, and one can again use a polynomial-time dynamic programming algorithm over the components, where the state records the three current color-class sizes and the transitions respect the list assignment on the vertices of the next component. Hence we may assume that $t \geq 4$. We invoke Theorem 30 with $H = K_3$. Since for all distinct $h, h' \in V(K_3)$ we have $|N_{K_3}(h) \cap N_{K_3}(h')| = 1$, the theorem applies, and we can compute the polynomial $p \coloneqq p_{(G,L)\to K_3}(x_1, x_2, x_3)$ in time $2^{O(\sqrt{t \cdot n \log n})} \cdot \mathrm{poly}(n)$ (note that since t is fixed, the factor depending on t is absorbed in the $O(\cdot)$-notation). Write $n = 3q + r$ with $r \in \{0, 1, 2\}$. By Lemma 29, the desired number of equitable list 3-colorings is exactly the corresponding sum of coefficients of p. Since every monomial of p has total degree n, the polynomial p contains at most $\binom{n+2}{2} = O(n^2)$ monomials. Moreover, each coefficient is at most 3^n and therefore has $O(n)$ bits. Hence, once p is computed, extracting the required coefficients and summing them takes only $\mathrm{poly}(n)$ time. Correctness follows from Lemma 29. ◀

References

1 Hans L. Bodlaender and Fedor V. Fomin. Equitable colorings of bounded treewidth graphs. *Theor. Comput. Sci.*, 349(1):22–30, 2005. doi:10.1016/J.TCS.2005.09.027.

2 Flavia Bonomo, Maria Chudnovsky, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38(4):779–801, 2018. doi:10.1007/S00493-017-3553-8.

3 Bor-Liang Chen and Ko-Wei Lih. Equitable coloring of trees. *Journal of Combinatorial Theory, Series B*, 61(1):83–87, 1994. doi:10.1006/JCTB.1994.1032.

4 Bor-Liang Chen, Ko-Wei Lih, and Pou-Lin Wu. Equitable coloring and the maximum degree. *European journal of combinatorics*, 15(5):443–447, 1994. doi:10.1006/EUJC.1994.1047.

5 Bor-Liang Chen, Ko-Wei Lih, and Jing-Ho Yan. Equitable coloring of interval graphs and products of graphs. *arXiv preprint*, 2009. arXiv:0903.1396.

6 Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-coloring p 6-free graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1239–1256. SIAM, 2019. doi:10.1137/1.9781611975482.76.

7 Gennaro Cordasco, Luisa Gargano, and Adele A Rescigno. Iterated type partitions. In *Combinatorial Algorithms: 31st International Workshop, IWOCA 2020, Bordeaux, France, June 8–10, 2020, Proceedings*, pages 195–210. Springer, 2020. doi:10.1007/978-3-030-48966-3_15.

8 Derek G Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM Journal on Computing*, 34(4):825–847, 2005. doi:10.1137/S0097539701385351.

9 Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *Journal of computer and system sciences*, 46(2):218–270, 1993. doi:10.1016/0022-0000(93)90004-G.

10 Bruno Courcelle, Johann A Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/S002249910009.

11 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015. doi:10.1007/978-3-319-21275-3.

12 Konrad K Dabrowski and Daniël Paulusma. Clique-width of graph classes defined by two forbidden induced subgraphs. *The Computer Journal*, 59(5):650–666, 2016. doi:10.1093/COMJNL/BXV096.

13 Wolfgang Espelage, Frank Gurski, and Egon Wanke. How to solve np-hard graph problems on clique-width bounded graphs in polynomial time. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 117–128. Springer, 2001. doi:10.1007/3-540-45477-2_12.

14 Michael R Fellows, Fedor V Fomin, Daniel Lokshtanov, Frances Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143–153, 2011. doi:10.1016/J.IC.2010.11.026.

15 Guilherme CM Gomes, Matheus R Guedes, and Vinicius F dos Santos. Structural parameterizations for equitable coloring. In *Latin American Symposium on Theoretical Informatics*, pages 129–140. Springer, 2020.

16 Guilherme de Gomes, Carlos VGC Lima, and Vinícius F dos Santos. Parameterized complexity of equitable coloring. *arXiv preprint*, 2018. arXiv:1810.13036.

17 Carla Groenland, Karolina Okrasa, Paweł Rzążewski, Alex Scott, Paul Seymour, and Sophie Spirkl. H-colouring pt-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019. doi:10.1016/J.DAM.2019.04.010.

18 András Hajnal and Endre Szemerédi. Proof of a conjecture of P. Erdös. *Combin. Theory Appl.*, 2:601, 1970.

19 Chính T Hoàng, Marcin Kamiński, Vadim Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of p 5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010. doi:10.1007/S00453-008-9197-8.

20 Shenwei Huang. Improved complexity results on k-coloring pt-free graphs. *European Journal of Combinatorics*, 51:336–346, 2016. doi:10.1016/J.EJC.2015.06.005.

21 Daniel Kobler and Udi Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003. doi:10.1016/S0166-218X(02)00198-1.

22 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM Journal on Discrete Mathematics*, 34(3):1538–1558, 2020. doi:10.1137/19M1280326.

23 Michaël Rao. Msol partitioning problems on graphs of bounded treewidth and clique-width. *Theoretical Computer Science*, 377(1-3):260–267, 2007. doi:10.1016/J.TCS.2007.03.043.

24 Yi Zhang and H-P Yap. Equitable colorings of planar graphs. *Journal of combinatorial mathematics and combinatorial computing*, 27:97–106, 1998.