Logarithmic Weisfeiler-Leman Identifies All Planar Graphs

Martin Grohe
RWTH Aachen University, Germany

Sandra Kiefer
University of Warsaw, Poland
RWTH Aachen University, Germany

Abstract

The Weisfeiler-Leman (WL) algorithm is a well-known combinatorial procedure for detecting symmetries in graphs and it is widely used in graph-isomorphism tests. It proceeds by iteratively refining a colouring of vertex tuples. The number of iterations needed to obtain the final output is crucial for the parallelisability of the algorithm.

We show that there is a constant k such that every planar graph can be identified (that is, distinguished from every non-isomorphic graph) by the k-dimensional WL algorithm within a logarithmic number of iterations. This generalises a result due to Verbitsky (STACS 2007), who proved the same for 3-connected planar graphs.

The number of iterations needed by the k-dimensional WL algorithm to identify a graph corresponds to the quantifier depth of a sentence that defines the graph in the $(k+1)$-variable fragment C^{k+1} of first-order logic with counting quantifiers. Thus, our result implies that every planar graph is definable with a C^{k+1}-sentence of logarithmic quantifier depth.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Mathematics of computing → Graph theory

Keywords and phrases Weisfeiler-Leman algorithm, finite-variable logic, isomorphism testing, planar graphs, quantifier depth, iteration number

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.134

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version `https://arxiv.org/abs/2106.16218`

Funding *Sandra Kiefer*: supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant LIPA, agreement no. 683080).

1 Introduction

The Weisfeiler-Leman (WL) algorithm is a combinatorial procedure for detecting symmetries in graphs. It is widely used in approaches to tackle the graph-isomorphism problem, both from a theoretical ([4, 5, 24]) and from a practical perspective ([7, 23, 31, 32]). The algorithm is derived from a technique called *naïve vertex classification* (or *Colour Refinement*), which may be viewed as the 1-dimensional version WL^1 of the WL algorithm. For every $k \geq 1$, the k-dimensional WL algorithm (WL^k) iteratively colours k-tuples of vertices of a graph by propagating local information until it reaches a *stable colouring*. Weisfeiler and Leman introduced the 2-dimensional version WL^2, today known as the *classical WL algorithm*, in [37]. The algorithm WL^k can be implemented to run in time $O(n^{k+1} \log n)$ on graphs of order n [22].

The algorithm has striking connections to numerous areas of mathematics and computer science, which surely is a reason why research on it has been active since its introduction over half a century ago. For example, there are tight connections to linear and semidefinite programming [2, 3, 20], homomorphism counting [8, 10], and the algebra of coherent configurations [6]. Most recently, the WL algorithm has been applied in several interesting machine-learning contexts [1, 16, 33, 34, 39].

A very strong and highly exploited link between the algorithm and logic was established by Immerman and Lander [22] and Cai, Fürer, and Immerman [5]: WL^k assigns the same colour to two k-tuples of vertices if and only if these tuples satisfy the same formulas of the $(k+1)$-variable fragment C^{k+1} of first-order logic with counting quantifiers. Cai, Fürer, and Immerman [5] used this correspondence and an Ehrenfeucht-Fraïssé game that characterises equivalence for the logic C^{k+1} to prove that, for every k, there are non-isomorphic graphs of order $O(k)$ that are not distinguished by WL^k. Here we say that WL^k *distinguishes* two graphs if WL^k computes different stable colourings on them, that is, there is some colour such that the numbers of k-tuples of that colour differ in the two graphs.

We say that WL^k *identifies* a graph G if it distinguishes G from all graphs G' that are not isomorphic to G. It has been shown that for suitable constants k, the algorithm WL^k identifies all planar graphs [13], all graphs of bounded tree width [18], and all graphs in many other natural graph classes [12, 14, 15, 17, 19]. For some of these classes, fairly tight bounds for the optimal value of k, called the *Weisfeiler-Leman (WL) dimension*, are known. Notably, interval graphs have WL dimension 2 [12], graphs of tree width k have WL dimension in the range $\lceil k/2 \rceil - 3$ to k [26], and, most relevant for us, planar graphs have WL dimension 2 or 3 [27].

Another parameter of the WL algorithm that has received recent attention is the number of iterations it needs to reach its final, stable colouring. Since a set of size n^k can only be partitioned $n^k - 1$ times, a natural upper bound on the number of iterations to reach the final output is $n^k - 1$ (n always denotes the number of vertices of the input graph). This bound cannot be improved for WL^1, since there are infinitely many graphs on which the algorithm takes $n-1$ iterations to compute its final output [25]. However, for WL^2, it was shown that the bound $\Theta(n^2)$ is asymptotically not tight [28]. Currently, the best upper bound on the iteration number for WL^2 is $O(n \log n)$ [30].

The number of iterations of WL^k is crucial for the parallelisability of the algorithm: for $\ell \geq \log n$, it holds that ℓ iterations of WL^k can be simulated in $O(\ell)$ steps on a PRAM with $O(n^k)$ processors [21, 29]. In particular, if for a class $\mathcal{C}$ of graphs, all $G, G' \in \mathcal{C}$ (of order n) can be distinguished by WL^k in $O(\log n)$ iterations, then the isomorphism problem for graphs in $\mathcal{C}$ is in the complexity class AC^1. Grohe and Verbitsky [21] proved that this is the case for all classes of graphs of bounded tree width and all maps (graphs embedded into a surface together with a rotation system specifying the embedding), and Verbitsky [36] proved it for the class of 3-connected planar graphs.

Our results

We say that WL^k distinguishes two graphs *in ℓ iterations* if the colouring obtained by WL^k in the ℓ-th iteration differs among the two graphs, and we say WL^k identifies a graph *in ℓ iterations* if it distinguishes the graph from every non-isomorphic graph in ℓ iterations.

▶ **Theorem 1.** *There is a constant k such that WL^k identifies every n-vertex planar graph in $O(\log n)$ iterations.*

The correspondence between WL^k and the logic C^{k+1} can be refined to a correspondence between the number of iterations and the quantifier depth: WL^k assigns the same colour to two k-tuples of vertices in the ℓ-th iteration if and only if these two k-tuples satisfy the same C^{k+1}-formulas of quantifier depth ℓ. Thus, the following theorem is equivalent to Theorem 1.

▶ **Theorem 2.** *There is a constant k such that for every n-vertex planar graph G, there is a C^k-sentence of quantifier depth $O(\log n)$ that identifies G (that is, characterises G up to isomorphism).*

We exploit the logical characterisation of the WL algorithm in our proof, so it is actually Theorem 2 that we prove. We first show that every planar graph G has a tree decomposition of logarithmic height where each bag consists of at most four 3-connected components of G and the adhesion is at most 6. Then we inductively construct a formula to identify G by ascending through the tree, encoding all information about isomorphism types of the parsed subgraphs in subformulas. At each node of the tree, we use Verbitsky's result to deal with the 3-connected components.

2 Preliminaries

All graphs in this paper are finite, simple, and undirected. For a graph G, we denote by $V(G)$ and $E(G)$ its set of vertices and edges, respectively. The *order* of G is $|G| := |V(G)|$. We write edges without parenthesis, as in vw. For $v \in V(G)$, we let $N_G(v) := \{w \mid vw \in E(G)\}$.

A *subgraph* of G is a graph H with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We set $N_G(H) := \bigcup_{v \in V(H)} N_G(v) \setminus V(H)$. We call a graph H a *topological subgraph* of G if a subdivision of H (i.e., a graph obtained from H by replacing some edges with paths) is a subgraph of G. For $W \subseteq V(G)$, we let $G[W] := (W, E(G) \cap \{uv \mid u, v \in W\})$ and, for arbitrary sets W, we let $G \setminus W := G[V(G) \setminus W]$.

A graph G is *k-connected* if $|G| > k$ and there is no set $S \subseteq V(G)$ with $|S| \leq k-1$ such that $G \setminus S$ is disconnected.

2.1 Logic

We denote by C the extension of first-order logic FO by *counting quantifiers* $\exists^{\geq m} x$ with the obvious meaning. C is only a syntactical extension of FO, because $\exists^{\geq m} x \varphi(x)$ is equivalent to $\exists x_1 \dots \exists x_m \Big(\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_i \varphi(x_i) \Big)$. However, we are mainly interested in the fragments C^k of C consisting of all formulae with at most k variables (which can, however, be reused within the formula). If $m > k$, then $\exists^{\geq m} x$ cannot be expressed in the k-variable fragment of FO, this is why we add the counting quantifiers.

We write $\varphi(x_1, \dots, x_\ell)$ to indicate that the free variables of φ are among $x_1, \dots, x_\ell$. Then for a graph G and vertices $u_1, \dots, u_\ell \in V(G)$, we write $G \models \varphi(u_1, \dots, u_\ell)$ to denote that G satisfies φ if, for all i, the variable x_i is interpreted by u_i. Moreover, we write $\varphi[G, u_1, \dots, u_i, x_{i+1}, \dots, x_\ell]$ to denote the set of all $(\ell - i)$-tuples $(u_{i+1}, \dots, u_\ell)$ such that $G \models \varphi(u_1, \dots, u_\ell)$.

The *quantifier depth* $\mathrm{qd}(\varphi)$ of a formula $\varphi \in \mathsf{C}$ is its depth of quantifier nesting. More formally,

- if φ is atomic, then $\mathrm{qd}(\varphi) = 0$.
- $\mathrm{qd}(\neg\varphi) = \mathrm{qd}(\varphi)$.
- $\mathrm{qd}(\varphi_1 \vee \varphi_2) = \mathrm{qd}(\varphi_1 \wedge \varphi_2) = \max\{\mathrm{qd}(\varphi_1), \mathrm{qd}(\varphi_2)\}$.
- $\mathrm{qd}(\exists^{\geq p} x \varphi) = \mathrm{qd}(\varphi) + 1$.

We denote the set of all C^k-formulas of quantifier depth at most ℓ by C^k_ℓ.

It will often be convenient to use asymptotic notation, such as $\mathsf{C}^{O(1)}_{O(\log n)}$. The parameter n always refers to the order of the input graph, and we will typically make assertions such as: *For every n, there exists a $\mathsf{C}^{O(1)}_{O(\log n)}$-formula $\varphi^{(n)}(x)$ such that for all graphs G of order $|G| = n$ and all $v \in V(G)$, [something holds].* What this means is that *there is a constant k and a function $\ell(n) \in O(\log n)$ such that for every n, there exists a $\mathsf{C}^k_{\ell(n)}$-formula $\varphi^{(n)}(x)$ such that for all graphs G of order $|G| = n$ and all $v \in V(G)$, [something holds].*

Throughout this paper, we will have to express properties of graphs and their vertices using $\mathsf{C}^{O(1)}_{O(\log n)}$-formulas. The basic building blocks that we use are connectivity statements with formulas of logarithmic quantifier depth, as illustrated in the following example.

▶ **Example 3.** For every $k \geq 0$, we define a $\mathsf{C}^3_{\lceil \log n \rceil}$-formula $\mathsf{dist}_{\leq k}$ such that for every graph G of order at most n and all vertices $u, u' \in V(G)$, it holds that $G \models \mathsf{dist}_{\leq k}(u, u')$ if and only if u and u' have distance at most k in G. We let

$$\mathsf{dist'}_{\leq k}(x, x') := \begin{cases} x = x' & \text{if } k = 0 \\ E(x, x') \vee x = x' & \text{if } k = 1 \\ \exists y_k \big(\mathsf{dist'}_{\leq \lfloor \frac{k}{2} \rfloor}(x, y_k) \wedge \mathsf{dist'}_{\leq \lceil \frac{k}{2} \rceil}(y_k, x')\big) & \text{otherwise.} \end{cases}$$

Thus, for $k \leq n$, the quantifier depth of $\mathsf{dist'}_{\leq k}$ is bounded by $\lceil \log n \rceil$. Now, it suffices to note that we can actually get by with the three variables x, x', y_k by reusing them in the subformulas that are defined inductively. We hence obtain the desired $\mathsf{C}^3_{\lceil \log n \rceil}$-formula $\mathsf{dist}_{\leq k}$. Note that, for $k \geq 1$, the $\mathsf{C}^3_{\lceil \log n \rceil}$-formula $\mathsf{dist}_{=k}(x, x') := \mathsf{dist}_{\leq k}(x, x') \wedge \neg \mathsf{dist}_{\leq k-1}(x, x')$ states that x and x' have distance exactly k. Moreover, in every graph of order at most n, the $\mathsf{C}^3_{\lceil \log n \rceil}$-formula $\mathsf{comp}(x, x') := \mathsf{dist}_{\leq n-1}(x, x')$ states that x and x' lie in the same connected component and the $\mathsf{C}^3_{\lceil \log n \rceil}$-sentence $\mathsf{conn}_n := \forall x \forall x' \mathsf{dist}_{\leq n-1}(x, x')$ states that the graph is connected. ⌟

2.2 The WL Algorithm

We briefly review the WL algorithm. For details, we refer to the recent survey [24].

Let $k \geq 1$. The *atomic type* $\mathrm{atp}(G, \bar{u})$ of a k-tuple $\bar{u} = (u_1, \ldots, u_k)$ of vertices of a graph G is the set of all atomic facts satisfied by these vertices, that is, all adjacencies and equalities between the vertices. Hence, tuples $\bar{u} = (u_1, \ldots, u_k)$ and $\bar{v} = (v_1, \ldots, v_k)$ of vertices of graphs G, H, respectively, have the same atomic type if and only if the mapping $u_i \mapsto v_i$ is an isomorphism from the graph $G[\{u_1, \ldots, u_k\}]$ to $H[\{v_1, \ldots, v_k\}]$.

The algorithm WL^k (the *k-dimensional Weisfeiler-Leman algorithm*) takes a graph G as input and computes the following sequence of *colourings* wl^k_i of $V(G)^k$ for $i \geq 0$, until it returns $\mathsf{wl}^k_\infty := \mathsf{wl}^k_i$ for the smallest i such that, for all $\bar{u}, \bar{v}$, it holds that $\mathsf{wl}^k_i(\bar{u}) = \mathsf{wl}^k_i(\bar{v}) \iff \mathsf{wl}^k_{i+1}(\bar{u}) = \mathsf{wl}^k_{i+1}(\bar{v})$. Set $\mathsf{wl}^k_0(\bar{u}) := \mathrm{atp}(G, \bar{u})$. In the $(i+1)$-st *iteration*, the colouring wl^k_{i+1} is defined by $\mathsf{wl}^k_{i+1}(\bar{u}) := \big(\mathsf{wl}^k_i(\bar{u}), M_i(\bar{u})\big)$, where, for $\bar{u} = (u_1, \ldots, u_k)$, we let $M_i(\bar{u})$ be the multiset

$$\Big\{\!\!\Big\{ \big(\mathrm{atp}(G, (u_1, \ldots, u_k, v)), \mathsf{wl}^k_i(u_1, \ldots, u_{k-1}, v), \\ \mathsf{wl}^k_i(u_1, \ldots, u_{k-2}, v, u_k), \ldots, \mathsf{wl}^k_i(v, u_2, \ldots, u_k)\big) \mid v \in V \Big\}\!\!\Big\}$$

The algorithm WL^k *distinguishes* two graphs G, H *in ℓ iterations* if there is a colour c in the range of wl^k_ℓ such that the number of tuples $\bar{u} \in V(G)^k$ with $\mathsf{wl}^k_\ell(\bar{u}) = c$ is different from the number of tuples $\bar{v} \in V(H)^k$ with $\mathsf{wl}^k_\ell(\bar{v}) = c$. In this case, we say WL^k_ℓ *distinguishes G and H*. Moreover, WL^k_ℓ *identifies* G if it distinguishes G from all graphs H that are not isomorphic to G.

▶ **Theorem 4** ([5, 22])**.** *Let $k \in \mathbb{N}$. Let G and H be graphs with $|G| = |H|$ and let $\bar{u} := (u_1, \ldots, u_k) \in V(G)^k$ and $\bar{v} := (v_1, \ldots, v_k) \in V(H)^k$. Then, for all $i \in \mathbb{N}$, the following are equivalent.*

1. $\mathsf{wl}^k_i(\bar{u}) = \mathsf{wl}^k_i(\bar{v})$.
2. $G \models \varphi(u_1, \ldots, u_k) \iff H \models \varphi(v_1, \ldots, v_k)$ *holds for every C^{k+1}_i-formula $\varphi(x_1, \ldots, x_k)$.*

3 3-Connected Planar Graphs

Verbitsky [36] proved that $\mathsf{WL}^{O(1)}_{O(\log n)}$ distinguishes any two 3-connected planar graphs. Before we discuss the specific version of this result that we need here, let us briefly review some background on planar graphs. Intuitively, a *plane graph* is a graph drawn into the plane with no edges crossing. A *planar graph* is an abstract graph G isomorphic to a plane graph; an isomorphism from G to a plane graph is a *planar embedding* of G. Now suppose G is a plane graph. If we cut the plane along all edges of the graph, the pieces that remain are the *faces* of G (note that one of these faces is unbounded). The closed walk along the vertices and edges in the boundary of a face is the *facial walk* associated with this face. If G is 2-connected, then every facial walk is a cycle. If G is 3-connected, we can describe the facial cycles combinatorially: a cycle C is a facial cycle of G if and only if C is an induced subgraph of G and $G \setminus V(C)$ is connected. (This is the statement of *Whitney's Theorem* [38].) This implies that all planar embeddings of a 3-connected planar graph have the same facial cycles, which can be interpreted as saying that, combinatorially, all planar embeddings of the graph are the same. Another way of describing a planar embedding combinatorially is by specifying, for each vertex, the cyclic order in which the edges incident to this vertex appear. This is what is known as a *rotation system*. It is easy to see that a rotation system determines all facial walks, and, conversely, the facial walks determine the rotation system. One last fact that we need to know about plane graphs is *Euler's formula*: if G is a connected plane graph with n vertices, m edges, and f faces, then $n - m + f = 2$. (For details and more background, we refer the reader to [9].)

Let us now turn to the version of Verbitsky's theorem about 3-connected planar graphs that we need here. It says that, in a 3-connected planar graph, we can find three vertices such that once these vertices are fixed, we can identify every other vertex by a $\mathsf{C}^{O(1)}_{O(\log n)}$-formula.

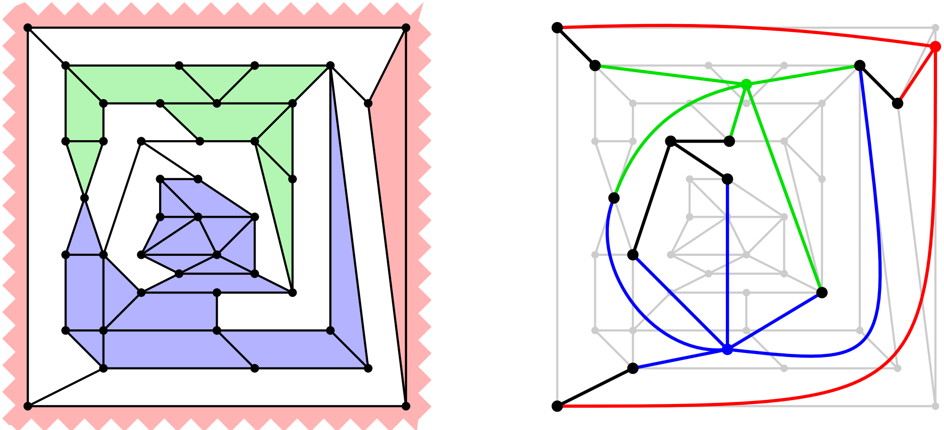
▶ **Theorem 5** ([36]). *Let $n \in \mathbb{N}$ and let G be a 3-connected planar graph of order $|G| \le n$ and $v_1v_2 \in E(G)$. Then there is a $v_3 \in N_G(v_2)$ and for every $w \in V(G)$ a $\mathsf{C}^{O(1)}_{O(\log n)}$-formula $\mathsf{id}_w(x_1, x_2, x_3, y)$ such that $G \models \mathsf{id}_w(v_1, v_2, v_3, w)$ and $G \not\models \mathsf{id}_w(v_1, v_2, v_3, w')$ for all $w' \in V(G) \setminus \{w\}$.*

The key step in Verbitsky's proof is to define the rotation system underlying the unique planar embedding of a 3-connected planar graph. To state this formally, we use the terminology of [13, 15]. An *angle* of a plane graph G *at* a vertex v is a triple (w, v, w') of vertices such that vw and vw' are successive edges in a facial walk of G. Two angles (v_1, v_2, v_3) and (w_1, w_2, w_3) are *aligned* if $w_1 = v_2$ and $w_2 = v_3$ and both angles appear in the same facial walk. Observe that, if we know all angles at a vertex v, we can define the cyclic permutation of the edges incident with v induced by the embedding. If we know all angles of G and the alignment relation between them, we can define the rotation system. By Whitney's Theorem, all planar embeddings of a 3-connected planar graph G have the same angles; we call them *the angles of G*. Similarly, we can define abstractly if two angles of a 3-connected planar graph are aligned.

▶ **Lemma 6** ([36]). *There are $\mathsf{C}^{O(1)}_{O(\log n)}$-formulas $\mathsf{ang}^{(n)}(x_1, x_2, x_3)$ and $\mathsf{aln}^{(n)}(x_1, x_2, x_3, x_4)$ such that for all 3-connected planar graphs G of order $|G| = n$ and all $v_1, v_2, v_3, v_4 \in V(G)$, we have*

$$G \models \mathsf{ang}^{(n)}(v_1, v_2, v_3) \iff (v_1, v_2, v_3) \textit{ is an angle of } G,$$

$$G \models \mathsf{aln}^{(n)}(v_1, \dots, v_4) \iff (v_1, v_2, v_3), (v_2, v_3, v_4) \textit{ are aligned angles of } G.$$



(a) (b)

Figure 1 Defining the faces of a 3-connected planar graph: (a) shows a 3-connected planar graph G with 3 regions formed by faces with at most 6 edges in their boundary; (b) shows the derived graph $G^{(1)}$; the faces of $G^{(1)}$ are in one-to-one correspondence to the white faces of G.

This lemma is an easy consequence of the results in [36, Section 4]. The terminology there is different, the notion corresponding to (aligned) angles is that of a *layout system*. Verbitsky's proof is based on a careful (and tedious) analysis of how two paths between the neighbours of a vertex may intersect.

To give the reader some intuition about the lemma, we sketch an alternative proof, which is based on ideas from [14] (also see [15, Section 10.4]). Let G be a 3-connected planar graph, and let us think of G as being embedded in the plane. It follows from Euler's formula that in every plane graph of minimum degree 3, a constant fraction of the edges is contained in facial walks of length at most 6. Using Whitney's Theorem, we can define the set of all 6-tuples that determine a facial cycle of length at most 6 using a C^9-formula of logarithmic quantifier depth. This gives us all the angles associated with these cycles and the alignment relation on these angles. The faces corresponding to these facial cycles of size at most 6 can be partitioned into *regions*, where two faces belong to the same region if their boundaries share an edge (see Figure 1(a)).

We define a new graph $G^{(1)}$ as follows: for every region R of G, we delete all vertices contained in the interior of R, all vertices on the boundary of R that have no neighbours outside the region, and all edges that are either in the interior or on the boundary of the region. Then we add a fresh vertex v_R and edges from v_R to all vertices that remain in the boundary of the region R (see Figure 1(b)). Each face of $G^{(1)}$ corresponds to a face of G that we have not found yet. Applying Euler's formula again, we can prove that a constant fraction of the edges of G that remain edges of $G^{(1)}$ are contained in facial walks of $G^{(1)}$ that contain at most six vertices of degree ≥ 3. We can define the facial walks of the corresponding edges in G, again using Whitney's Theorem to test if a cycle is facial. Note that, for this, we do not need $G^{(1)}$ to be 3-connected (in general, it is not); we always define facial cycles in the original graph G. The new facial cycles together with those found in the first step give us new regions (covering more faces of G), and from these, we construct a graph $G^{(2)}$. Iterating the construction, we obtain a sequence of graphs $G^{(i)}$. The construction stops once we have found all facial walks of G. Since we always use a constant fraction of the edges, this happens after at most logarithmically many iterations. This completes our proof sketch of Lemma 6.

Proof of Theorem 5. Let G be a 3-connected planar graph of order $|G| = n$. For angles $\overline{v} = (v_1, v_2, v_3)$, $\overline{w} = (w_1, w_2, w_3)$, we write $\overline{v} \curvearrowright \overline{w}$ if $\overline{v}, \overline{w}$ are aligned, and we write $\overline{v} \curlywedge \overline{w}$ if $w_1 = v_3$ and $w_2 = v_2$ and $w_3 \neq v_1$. Note that, for every angle $\overline{v}$, there is a unique $\overline{w}$ such that $\overline{v} \curvearrowright \overline{w}$, because, by the 3-connectedness of G, every angle is in the boundary of a unique face, and the aligned angle belongs to the same face. There is also a unique $\overline{w}'$ such that $\overline{v} \curlywedge \overline{w}'$, determined by the cyclic order of the edges and faces around a vertex. An *angle walk* is a sequence $\overline{v}_0, \ldots, \overline{v}_\ell$ of angles such that for all $i \in [\ell]$, we have $\overline{v}_{i-1} \curvearrowright \overline{v}_i$ or $\overline{v}_{i-1} \curlywedge \overline{v}_i$. The *direction* of the angle walk $\overline{v}_0, \ldots, \overline{v}_\ell$ is the tuple $\overline{\delta} = (\delta_1, \ldots, \delta_\ell) \in \{\curvearrowright, \curlywedge\}^\ell$ such that for every $i \in [\ell]$, we have $\overline{v}_{i-1}\delta_i\overline{v}_i$. Using Lemma 6, it is straightforward to prove that for every $\overline{\delta} \in \{\curvearrowright, \curlywedge\}^{\leq n}$, there is a $\mathsf{C}^{O(1)}_{O(\log n)}$-formula $\mathsf{awalk}^{(n)}_{\overline{\delta}}(\overline{x}, \overline{y})$ such that for all $\overline{v}, \overline{w} \in V(G)^3$, we have $G \models \mathsf{awalk}^{(n)}_{\overline{\delta}}(\overline{v}, \overline{w})$ if and only if there is an angle walk of direction $\overline{\delta}$ from $\overline{v}$ to $\overline{w}$. Now let $v_1v_2 \in E(G)$. Then there is a v_3 such that (v_1, v_2, v_3) is an angle. Let $\overline{v} := (v_1, v_2, v_3)$. Note that, for every $w \in V(G) \setminus \{v_1, v_2, v_3\}$, there is an angle walk of length at most n from $\overline{v}$ to some $\overline{w} = (w_1, w_2, w_3)$ with $w_3 = w$, simply because every path in G can be extended to an angle walk. Let $\Delta(w)$ be the set of all directions $\overline{\delta}$ of length at most n such that there is an angle walk of direction $\overline{\delta}$ from $\overline{v}$ to some $\overline{w} = (w_1, w_2, w_3)$ with $w_3 = w$. Note that the sets $\Delta(w)$ for $w \in V(G) \setminus \{v_1, v_2, v_3\}$ are mutually disjoint. Let $\mathsf{id}_{\overline{\delta}}(x_1, x_2, x_3, y) := \exists y_1 \exists y_2 \mathsf{awalk}^{(n)}_{\overline{\delta}}(x_1, x_2, x_3, y_1, y_2, y)$. Then for $\overline{\delta} \in \Delta(w)$, we have $G \models \mathsf{id}_{\overline{\delta}}(v_1, v_2, v_3, w)$ and $G \not\models \mathsf{id}_{\overline{\delta}}(v_1, v_2, v_3, w')$ for all $w' \neq w$. ◀

4 Decomposition into Blocks

Let G be a graph. A *tree decomposition* of G is a pair (T, β) where T is a tree and $\beta \colon V(T) \to 2^{V(G)}$ is a function such that for every $v \in V(G)$, the set $\{t \in V(T) \mid v \in \beta(t)\}$ is non-empty and induces a connected subgraph in T, and for every $e \in E(G)$, there is a $t \in V(T)$ such that $e \subseteq \beta(t)$. For $t \in V(T)$, we call $\beta(t)$ a *bag* of (T, β). The *adhesion* of (T, β) is $\mathrm{ad}(T, \beta) := \max\left\{|\beta(t) \cap \beta(u)| \;\middle|\; tu \in E(T)\right\}$ (or 0 if $E(T) = \emptyset$). The *width* of (T, β) is $\mathrm{wd}(T, \beta) := \max_{t \in V(T)} |\beta(t)| - 1$.

We denote the root of a rooted tree T by r^T. For better readability, if the rooted tree is referred to as T^*, we set $r^* := r^{T^*}$. The *height* of T is the maximum length of a path from r^T to a leaf of T. We denote the descendant order of T by $\trianglelefteq^T$. That is, $t \trianglelefteq^T u$ if t occurs on the path from r^T to u. A *rooted tree decomposition* is a tree decomposition where the tree is rooted.

▶ **Lemma 7** (Folklore). *Let T be a tree and $\chi \colon V(T) \to \mathbb{R}_{\geq 0}$. Then there is a node $t \in V(T)$ such that for every connected component C of $T \setminus \{t\}$, it holds that*

$$\sum_{t \in V(C)} \chi(t) \leq \frac{1}{2} \sum_{t \in V(T)} \chi(t).$$

Proof. Orient all edges towards the larger sum of χ-weights in the connected components that the removal of the edge would induce, breaking ties arbitrarily. There will be a node such that all incident edges are oriented towards it. This node has the desired property. ◀

The following lemma is known in its essence (for example, [11]), though we are not aware of a reference where it is stated in this precise form, which we will need later.

▶ **Lemma 8.** *Let T be a tree, and let $B \subseteq V(T)$ be a set of size $|B| \leq 3$. Then there is a rooted tree decomposition (T^*, β^*) of T with $B \subseteq \beta^*(r^*)$ and the following additional properties:*

(i) *The height of T^* is at most* $2\log|T|$.
(ii) *The width of* (T^*, β^*) *is at most* 3.
(iii) *The adhesion of* (T^*, β^*) *is at most* 3.
(iv) *For every* $t^* \in V(T^*)$ *and every child* u^* *of* t^*, *the graph* $T\left[\left(\bigcup_{v^* \trianglerighteq_{T^*} u^*} \beta^*(v^*)\right) \setminus \beta^*(t^*)\right]$ *is connected.*

Proof. Condition (iv) is something that we can easily achieve for every rooted tree decomposition: if, for the rooted subtree at some node, the subgraph induced by the bags in this subtree is not connected, we simply create one copy of the subtree for each connected component and only keep the vertices of that connected component in the copy. Moreover, the adhesion of a tree decomposition of width 3 can only be larger than 3 if there are adjacent nodes with the same bag. If this is the case, we can simply contract the edge between the nodes. Repeating this, we can turn the decomposition into a decomposition of adhesion at most 3. So we only need to take care of Conditions (i) and (ii).

The proof is by induction on $n := |T|$. We prove a slightly stronger statement; in addition to $B \subseteq \beta^*(r^*)$, we require $|\beta^*(r^*) \setminus B| \leq 1$.

The base case $n \leq 4$ is easy: for $n = 1$, the 1-node tree decomposition of height 0 has all the desired properties, and for $2 \leq n \leq 4$, we can take a 2-node tree decomposition of height 1 where the root bag is B and the leaf bag is $V(T)$.

For the inductive step, suppose $n > 4$.

Case 1: $|B| < 3$.

By Lemma 7, there is a node $b \in V(T)$ such that for every connected component C of $T \setminus \{b\}$, it holds that $|V(C)| \leq \frac{n}{2}$.

Let $C_1, \ldots, C_m$ be the vertex sets of the connected components of $T \setminus \{b\}$. For every $i \in [m]$, let c_i be the unique neighbour of b in C_i, and let $B_i := (B \cap V(C_i)) \cup \{c_i\}$. Note that $|B_i| \leq 3$.

By the induction hypotheses, for every i, there is a rooted tree decomposition (T_i, β_i) of C_i with the desired properties. In particular, the height of T_i is at most $2\log(n/2) = 2\log n - 2$.

For every i, let r_i be the root of T_i. We form a new tree T^* by taking the disjoint union of all the T_i, adding fresh nodes r^* and r_i^* for $i \leq m$, and adding edges $r^*r_i^*$, $r_i^*r_i$ for all $i \in [m]$. We define $\beta^* \colon V(T^*) \to 2^{V(T)}$ by

$$\beta^*(t) := \begin{cases} B \cup \{b\} & \text{if } t = r^*, \\ B_i \cup \{b\} & \text{if } t = r_i^*, \\ \beta_i(t) & \text{if } t \in V(T_i). \end{cases}$$

Then (T^*, β^*) is a tree decomposition of T of width at most 3 and height at most $2\log n$.

Case 2: $|B| = 3$.

By Lemma 7 applied to the characteristic function of B, there is a node $b \in V(T)$ such that for every connected component C of $T \setminus \{b\}$, it holds that $|V(C) \cap B| \leq 1$.

Let $C_1, \ldots, C_\ell$ be the connected components of $T \setminus \{b\}$, and for every i, let $B_i := B \cap V(C_i)$. Then $|B_i| \leq 1$.

$\triangleright$ Claim 9. For every $i \in [\ell]$, there is a tree decomposition (T_i, β_i) of width at most 3 such that the height of T_i is at most $2\log n - 1$ and for the root r_i of T_i it holds that $B_i \subseteq \beta_i(r_i)$ and $|\beta(r_i)| \leq 2$.

Proof. Let $\in [\ell]$ and $n_i \coloneqq |C_i|$. By Lemma 7, there is a $c \in V(C_i)$ such that for every connected component D of $C_i \setminus \{c\}$, it holds that $|D| \leq n_i/2$. Choose such a c and let $D_1, \dots, D_m$ be the connected components of $C_i \setminus \{c\}$. For every $j \in [m]$, let d_j be the unique neighbour of c in D_j. Let $B_{ij} \coloneqq (B_i \cap D_j) \cup \{d_j\}$. Then $|B_{ij}| \leq 2$.
By the induction hypotheses, for every j, there is a rooted tree decomposition (T_{ij}, β_{ij}) of D_j of width 3 such that the height of T_{ij} is at most $2\log|D_i| \leq 2\log(n_i/2) \leq 2\log n - 2$. Furthermore, for the root r_{ij} of T_{ij}, it holds that $B_{ij} \subseteq \beta_{ij}(r_{ij})$ and $|\beta_{ij}(r_{ij}) \setminus B_{ij}| \leq 1$. This implies $|\beta_{ij}(r_{ij})| \leq 3$.
We form a new tree T_i by taking the disjoint union of all the T_{ij} for $j \in [m]$, adding a fresh node r_i, and adding edges r_{ij} for all $j \in [m]$. We define $\beta_i \colon V(T_i) \to 2^{V(C_i)}$ by

$$\beta_i(t) \coloneqq \begin{cases} B_i \cup \{c\} & \text{if } t = r_i, \\ \beta_{ij}(r_{ij}) \cup \{c\} & \text{if } t = r_{ij}, \\ \beta_{ij}(t) & \text{if } t \in V(T_{ij}) \setminus \{r_{ij}\}. \end{cases}$$

Then (T_i, β_i) is a tree decomposition of C_i with the desired properties. ◁

To complete the proof of the lemma, we form a new tree T^* by taking the disjoint union of the T_i of Claim 9 for $i \in [\ell]$, adding a fresh node r^*, and adding edges r^*r_i for all $i \in [\ell]$. We define $\beta^* \colon V(T^*) \to 2^{V(T)}$ by

$$\beta^*(t) \coloneqq \begin{cases} B \cup \{b\} & \text{if } t = r^*, \\ \beta(r_i) \cup \{b\} & \text{if } t = r_i, \\ \beta_i(t) & \text{if } t \in V(T_i) \setminus \{r_i\}. \end{cases}$$

Then (T^*, β^*) is a tree decomposition of T of width at most 3 and height at most $2\log n$. ◀

Let us now turn to decompositions of a graph into its 3-connected components. We need a few more definitions. In the following, let G be a connected graph and $X \subseteq V(G)$. The *torso* of X is the graph $G[\![X]\!]$ with vertex set X and edge set

$$\left\{ vw \in \binom{X}{2} \;\middle|\; vw \in E(G) \text{ or } v, w \in N_G(C) \text{ for some connected component } C \text{ of } G \setminus X \right\}.$$

The *adhesion* of X is the maximum of $|N_G(C)|$ for all connected components C of $G \setminus X$. It is easy to see that if the adhesion of X is at most 2, then the torso $G[\![X]\!]$ is a topological subgraph of G and if the adhesion of X is at most 1, then the torso $G[\![X]\!]$ is just the induced subgraph $G[X]$.

A *block*[1] of G is a set $B \subseteq V(G)$ such that

- either $G[\![B]\!]$ is 3-connected and the adhesion of B is at most 2,
- or $G[\![B]\!]$ is a complete graph of order 3 and the adhesion of B is at most 2,
- or $G[\![B]\!]$ is a complete graph of order 2 and the adhesion of B is at most 1.

We call blocks with 3-connected torsos *proper blocks* and blocks of cardinality at most 3 *degenerate blocks* of *order* 3 and 2, respectively. It is easy to see that for distinct blocks B, B', neither $B \subseteq B'$ nor $B' \subseteq B$ holds and, furthermore, $|B \cap B'| \leq 2$. A *block separator*

[1] Our usage of the term "block" is non-standard. If anything, what we call a "block" might better be called "2-block". But just using "block" is more convenient.

is a set $S \subseteq V(G)$ such that there are distinct blocks B, B' with $S = B \cap B'$, and the two sets $B \setminus S$ and $B' \setminus S$ belong to different connected components of $G \setminus S$. Note that by the definition of blocks, block separators have cardinality at most 2.

Observe that the torsos of all blocks of a graph are topological subgraphs. As all topological subgraphs of a planar graph are planar, the torsos of the blocks of a planar graph are planar. In particular, the torsos of proper blocks are 3-connected planar graphs. This will be important later.

Call a tree decomposition (T, β) *small* if for all distinct nodes $t, u \in V(T)$, it holds that $\beta(t) \not\subseteq \beta(u)$.

▶ **Lemma 10** ([35]). *Every connected graph G has a small tree decomposition (T, β) of adhesion at most 2 such that for all $t \in V(T)$, the bag $\beta(t)$ is a block of G.*

The decomposition in this lemma is essentially Tutte's well-known decomposition of a graph into its 3-connected components described in a slightly non-standard way. The two main differences are that, normally, the decomposition is only described for 2-connected graphs, whereas arbitrary connected graphs are first decomposed into their 2-connected components. We merge these decompositions into one. The second difference is that Tutte decomposes a 2-connected graph into 3-connected pieces (our proper blocks) and cycles. Instead of cycles, we only allow triangles, i.e., degenerate blocks of order 3. This is possible because every cycle can be decomposed into triangles. What we lose with our form of decomposition is the canonicity: a graph may have several structurally different decompositions of the form described in the lemma.

In the following, we apply Lemma 8 to the tree of the decomposition of Lemma 10 and obtain a decomposition of logarithmic height that is still essentially a decomposition into 3-connected components.

▶ **Lemma 11.** *Every connected graph G has a rooted tree decomposition (T^*, β^*) with the following properties.*

(i) *The height of T^* is at most $2 \log |G|$.*

(ii) *For every $t^* \in V(T^*)$, there are sets $B_1, \dots, B_4$ (not necessarily distinct or disjoint) such that $\beta^*(t^*) = \bigcup_{i=1}^{4} B_i$ and each B_i is either a block or a block separator.*

(iii) *The adhesion of (T^*, β^*) is at most 6.*

(iv) *For every $t^* \in V(T^*)$ and every child u^* of t^*, the induced subgraph*

$$G\left[\Big(\bigcup_{v^* \trianglerighteq^{T^*} u^*} \beta^*(v^*)\Big) \setminus \beta^*(t^*)\right]$$

is connected.

Proof. Let (T, β) be the decomposition of G into its blocks obtained from Lemma 10. Let (T^*, β^*_T) be the rooted tree decomposition of T obtained from Lemma 8. Let r^* be the root of T^*, and let $\trianglelefteq^* := \trianglelefteq^{T^*}$ be the partial descendant order associated with T^*. For every $t^* \in V(T^*)$, let

$$\gamma^*_T(t^*) := \bigcup_{u^* \trianglerighteq^* t^*} \beta^*_T(u^*)$$

and

$$\sigma^*_T(t^*) := \begin{cases} \emptyset & \text{if } t^* = r^*, \\ \beta^*_T(s^*) \cap \beta^*_T(t^*) & \text{for the parent } s^* \text{ of } t^* \text{ in } T^*, \text{ otherwise}. \end{cases}$$

For every $t \in V(T)$, we let $\min^*(t)$ be the unique $\trianglelefteq^*$-minimal node $t^* \in V(T^*)$ such that $t \in \beta^*_T(t^*)$. The uniqueness follows from the fact that the set of all $t^* \in V(T^*)$ with $t \in \beta^*_T(t^*)$ is connected in T^*.

Let us call $t \in V(T)$ *active* in $t^* \in V(T^*)$ if $t \in \beta^*_T(t^*)$ and $t^* \neq \min^*(t)$ and there is a $u \in N_T(t)$ such that $t^* \trianglelefteq \min^*(u)$. We call u an *activator* of t in t^*.

▷ Claim 12. Suppose that $t \in V(T)$ is active in $t^* \in V(T^*)$. Then there is a unique activator of t in t^*.

Proof. Since $t \in \beta^*_T(t^*)$ and $t^* \neq \min^*(t)$, we have $\min^*(t) \triangleleft t^*$ and $t \in \beta^*_T(\min^*(t)) \cap \beta^*_T(t^*) \subseteq \sigma^*_T(t^*)$. Moreover, for every activator u of t, it holds that $t^* \trianglelefteq \min^*(u)$, which implies $u \in \gamma^*_T(t^*) \setminus \sigma^*_T(t^*)$.

Suppose towards a contradiction that t has two activators u_1, u_2 in t^*. Then $u_1, u_2 \in N_T(t) \cap \big(\gamma^*_T(t^*) \setminus \sigma^*_T(t^*)\big)$. By Lemma 8(iv), the induced subgraph $T[\gamma^*_T(t^*) \setminus \sigma^*_T(t^*)]$ is connected. Thus, there is a path from u_1 to u_2 in $T[\gamma^*_T(t^*) \setminus \sigma^*_T(t^*)]$. As $u_1, u_2 \in N_T(t)$ and $t \in \sigma^*_T(t^*)$, there is a cyle in T, which is a contradiction. ◁

Hence, in the following we can speak of *the* activator of a node. Observe that if t is active in t^*, then t is also active in all u^* with $\min^*(t) \triangleleft u^* \triangleleft t^*$, with the same activator.

Now we are ready to define our tree decomposition (T^*, β^*) of G. The tree T^* is the same as in the decomposition (T^*, β^*_T) of T. We define $\beta^* \colon V(T^*) \to 2^{V(G)}$ by letting $\beta^*(t^*)$ for $t^* \in V(T^*)$ be the union of the following sets:

- for all $t \in \beta^*_T(t^*)$ such that $t^* = \min^*(t)$: the block $\beta(t)$, and
- for all $t \in \beta^*_T(t^*)$ such that t is active in t^* with activator u: the block separator $\beta(t) \cap \beta(u)$.

▷ Claim 13. (T^*, β^*) is a tree decomposition of G.

Proof. Every edge $e \in E(G)$ is contained in some bag $\beta(t)$, and $\beta(t) \subseteq \beta^*(\min^*(t))$.

Now consider a vertex $v \in V(G)$. Let

$$S_v := \{t \in V(T) \mid v \in \beta(t)\},$$
$$S^*_v := \{t^* \in V(T^*) \mid S_v \cap \beta^*_T(t^*) \neq \emptyset\}.$$

Since (T, β) is a tree decomposition, S_v is connected in T, and as (T^*, β^*_T) is a tree decomposition, S^*_v is connected in T^*. Thus, there is a unique $\trianglelefteq^*$-minimal node s^* in S^*_v. Let $s \in S_v \cap \beta^*_T(s^*)$. Then $s^* = \min^*(s)$ and therefore $v \in \beta^*(s^*)$.

Let $t^* \in V(T^*)$ such that $v \in \beta^*(t^*)$. We shall prove that $v \in \beta^*(v^*)$ for all v^* on the path from t^* to s^*. This will prove that the set of all t^* for which $v \in \beta^*(t^*)$ holds is connected in T^*.

By the definition of β^*, since $v \in \beta^*(t^*)$, there is a $t \in \beta^*_T(t)$ such that $v \in \beta(t)$ and either $t^* = \min^*(t)$ or t is active in t^*. We choose such a t. Then $t \in S_v$ and therefore $t^* \in S^*_v$. By the minimality of s^*, this implies $s^* \trianglelefteq^* t^*$.

The proof that $v \in \beta^*(v^*)$ holds for all v^* on the path from t^* to s^* is by induction on the distance d between t^* and s^*. The base case $d = 0$ is trivial. So let us assume that $d \geq 1$. It follows from the definition of β^* that $v \in \beta^*(v^*)$ holds for all v^* on the path from t^* to $\min^*(t)$. Thus, without loss of generality, we may assume that $t^* = \min^*(t)$.

Let $t = t_1, \dots, t_m = s$ be the path from t to s in T. Note that $v \in \beta(t_i)$ holds for all $i \in [m]$. The edge $tt_2 = t_1t_2$ must be covered by some bag $\beta^*_T(u^*)$ that contains both t and t_2. Since $t^* = \min^*(t)$, we have $t^* \trianglelefteq^* u^*$. As the pre-image of the path $t_1, \dots, t_m$ in T^* is connected and $s^* \trianglelefteq^* t^* \trianglelefteq^* u^*$, there is an $i > 1$ such that $t_i \in \beta^*(t^*)$. If $\min^*(t_i) = t^*$, we find a $j > i$ such that $t_j \in \beta^*(t)$, and, repeating this, we eventually arrive at a $t_k \in \beta^*(t)$ such that $\min^*(t_k) \lhd t^*$. Arguing as above, we find that $v \in \beta^*(v^*)$ holds for all v^* on the path from t^* to $\min^*(t_k)$. Since $\min^*(t_k)$ is closer to s^* than t^*, we can now apply the induction hypothesis to conclude that $v \in \beta^*(v^*)$ holds for all v^* on the path from $\min^*(t_k)$ to s^*. $\lhd$

Let us turn to proving that the tree decomposition (T^*, β^*) has the desired properties.

Since (T, β) is a small decomposition, we have $|T| \leq |G|$. Thus, Condition (i) follows from Lemma 8(i). Also, Condition (ii) follows from Lemma 8(ii) and the definition of $\beta^*(t)$.

To prove Condition (iii), let u^* be a child of t^*. Let us assume that $\beta^*_T(t^*) = \{t_1, \ldots, t_4\}$ and $\beta^*_T(u^*) = \{u_1, \ldots, u_4\}$ with $t_1 = u_1, t_2 = u_2$, and $t_3 = u_3$ and $t_4 \neq u_i, u_4 \neq t_i$ for $i \in [4]$. The cases of smaller bags $\beta^*_T(t^*)$, $\beta^*_T(u^*)$ or a smaller intersection between them can be dealt with similarly.

Let us first deal with the common elements $t_i = u_i$ for $i \in [3]$. Note that $\min^*(t_i) \unlhd t^* \lhd u^*$. If t_i is not active in u^*, then it does not contribute to the $\beta^*(u^*)$ and hence not to the intersection of the two bags. If t_i is active in u^*, say, with activator v_i, then the block separator $S_i := \beta(t_i) \cap \beta(v_i)$ is contained in $\beta^*(u^*)$. To simplify the notation, in the following, we let $S_i := 0$ if t_i is not active in u^*.

Either t_i is active in t^* as well with the same activator and we have $S_i \subseteq \beta^*(t^*)$, or $t^* = \min^*(t_i)$ and $S_i \subseteq \beta(t_i) \subseteq \beta^*(t^*)$. In both cases,

$$S_i \subseteq \beta^*(t^*) \cap \beta(u^*). \tag{1}$$

Next, let us look at the contribution of t_4 and u_4. The contribution of t_4 to $\beta^*(t^*)$ is contained in $\beta(t_4)$, and the contribution of u_4 to $\beta^*(u^*)$ is contained in $\beta(u_4)$. Since the only neighbour of t_i in $\gamma^*_T(u^*) \setminus \sigma^*_T(u^*) = \gamma^*_T(u^*) \setminus \{t_1, t_2, t_3\}$ is v_i (if t_i is active in u^*, otherwise there is no neighbour), all paths from t_i to u_4 go through v_i. This implies that

$$\beta(t_i) \cap \beta(u_4) \subseteq \beta(t_i) \cap \beta(v_i) = S_i. \tag{2}$$

All paths from t_4 to u_4 go through t_1, t_2, t_3, and therefore

$$\beta(t_4) \cap \beta(u_4) \subseteq \bigcup_{i=1}^{3} \beta(t_i) \cap \beta(u_4) \subseteq S_1 \cup S_2 \cup S_3. \tag{3}$$

Thus, overall, we have $\beta^*(t^*) \cap \beta^*(u^*) \subseteq S_1 \cup S_2 \cup S_3$.

To prove that Condition (iv) holds, let $t^* \in V(T^*)$ and and let u^* be a child of t^*. To simplify the notation, let $\sigma^*(u^*) := \beta(u^*) \cap \beta^*(t^*)$ and

$$\gamma^*(u^*) := \bigcup_{v^* \unrhd u^*} \beta^*(v^*). \tag{4}$$

We need to prove that $G[\gamma^*(u^*) \setminus \sigma^*(u^*)]$ is connected. The key observation is that

$$\gamma^*(u^*) \setminus \sigma^*(u^*) = \bigcup_{t \in \gamma^*_T(u^*) \setminus \sigma^*_T(u^*)} \beta(t). \tag{5}$$

The reason for this is that, for all $t \in \gamma^*_T(u^*) \setminus \sigma^*_T(u^*)$, it holds that $u^* \unlhd \min^*(t)$, which implies that $\beta(t) \subseteq \beta^*(\min^*(t))$ appears on the right-hand side of (4). It follows from Lemma 8(iv) that the set $\gamma^*_T(u^*) \setminus \sigma^*_T(u^*)$ is connected in T, and this implies that the union on the right-hand side of (5) is connected. ◀

Our next goal will be to define the decomposition in the logic $\mathsf{C}^{O(1)}_{O(\log n)}$. The following lemma yields a way to define blocks via triplets of vertices.

▶ **Lemma 14.** *Let G be a graph, and let B be a proper block of G. Let $b_1, b_2, b_3 \in B$ be pairwise distinct vertices. Then B is the set of all $v \in V(G)$ such that there is no set $S \subseteq V(G) \setminus \{v\}$ of cardinality at most 2 separating v from $\{b_1, b_2, b_3\}$.*

Proof. Let $v \in B$. Since $G[\![B]\!]$ is 3-connected, there are paths $P_i \subseteq G[\![B]\!]$ from v to b_i that are internally disjoint, that is, $V(P_i) \cap V(P_j) = \{v\}$ for $i \neq j$. As $G[\![B]\!]$ is a topological subgraph of G, these paths can be expanded to paths P_i' from v to b_i in G, and the P_i' are still internally disjoint. Since every $S \subseteq V(G) \setminus \{v\}$ of cardinality at most 2 has an empty intersection with at least one of the paths P_i', it does not separate v from $\{b_1, b_2, b_3\}$.

Conversely, let $v \in V(G) \setminus B$, and let C be the connected component of $G \setminus B$ with $v \in V(C)$, and let $S := N_G(C)$. Then $|S| \leq 2$. Then S separates v from $\{b_1, b_2, b_2\}$. ◀

Let G be a graph and $S, X \subseteq V(G)$. We say that S *separates* X if there are two distinct connected components C_1, C_2 of $G \setminus S$ such that $X \cap V(C_i) \neq \emptyset$ for both $i = 1, 2$.

▶ **Lemma 15.** *Let G be a graph, and let $b_1, b_2, b_3 \in V(G)$ be mutually distinct. Then there is a proper block B with $b_1, b_2, b_3 \in B$ if and only if there is a vertex $b_4 \in V(G) \setminus \{b_1, b_2, b_3\}$ such that no set $S \subseteq V(G)$ of cardinality at most 2 separates $\{b_1, b_2, b_3, b_4\}$.*

Proof. For the forward direction, suppose that B is a proper block with $b_1, b_2, b_3 \in B$. Let $b_4 \in B \setminus \{b_1, b_2, b_3\}$. Then it follows from Lemma 14 that there is no S of cardinality at most 2 that separates $\{b_1, b_2, b_3, b_4\}$.

For the backward direction, let B be the set of all $v \in V(G)$ such that no set $S \subseteq V(G) \setminus \{v\}$ of cardinality at most 2 separates v from $\{b_1, b_2, b_3\}$. Then $b_1, b_2, b_3 \in B$ and $|B| \geq 4$. It is easy to prove that B is a block. ◀

▶ **Lemma 16.** *For all $n \in \mathbb{N}$, there exist $\mathsf{C}^{O(1)}_{O(\log n)}$-formulas $\mathsf{block}^{(n)}(x_1, x_2, x_3, y)$ and $\mathsf{torso}^{(n)}(x_1, x_2, x_3, y, z)$ such that for all graphs G of order at most n and all $b_1, b_2, b_3, v \in V(G)$, we have*

$$G \models \mathsf{block}^{(n)}(b_1, b_2, b_3, v)$$

if and only if one of the following holds:
- *either $\{b_1, b_2, b_3\}$ is a degenerate block and $v \in \{b_1, b_2, b_3\}$,*
- *or b_1, b_2, b_3 are mutually distinct and there is a proper block B such that $b_1, b_2, b_3, v \in B$.*

Moreover, for all $b_1, b_2, b_3, v, w \in V(G)$, we have

$$G \models \mathsf{torso}^{(n)}(b_1, b_2, b_3, v, w)$$

if and only if $G \models \mathsf{block}^{(n)}(b_1, b_2, b_3, v)$ and $G \models \mathsf{block}^{(n)}(b_1, b_2, b_3, w)$ and vw is an edge of the torso of the block determined by b_1, b_2, b_3.

Proof. It is easy to express in $\mathsf{C}^{O(1)}_{O(\log n)}$ that $\{b_1, b_2, b_3\}$ is a degenerate block. For proper blocks, we use Lemmas 14 and 15. ◀

As an immediate consequence, we obtain a formula to define a block separator.

▶ **Corollary 17.** *For all $n \in \mathbb{N}$, there exists a $\mathsf{C}^{O(1)}_{O(\log n)}$-formula $\mathsf{blocksep}^{(n)}(x_1, x_2)$ such that for all graphs G of order at most n and all $s_1, s_2 \in V(G)$, we have*

$$G \models \mathsf{blocksep}^{(n)}(s_1, s_2)$$

if and only if $\{s_1, s_2\}$ is a block separator of G.

We are ready to define the formula that yields the decomposition from Lemma 11.

▶ **Lemma 18.** *For all $h \geq 0$, $n \geq 1$, there is a $\mathsf{C}^{O(1)}_{O(h+\log n)}$-formula $\mathsf{dec}_h^{(n)}(x_i^j, y_k \mid i \in [4], j \in [3], k \in [6])$ such that the following holds. Let G be a graph of order $|G| \leq n$ and $b_i^j, s_k \in V(G)$ for $i \in [4], j \in [3], k \in [6]$ (not necessarily distinct). Then*

$$G \models \mathsf{dec}_h^{(n)}(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6])$$

if and only if the following conditions are satisfied.

(i) *For all $i \in [4]$, either $B_i := \{b_i^1, b_i^2, b_i^3\}$ is a block separator or $B_i := \{b_i^1, b_i^2, b_i^3\}$ is a degenerate block or b_i^1, b_i^2, b_i^3 are mutually distinct and there is a (unique) block B_i that contains b_i^1, b_i^2, b_i^3.*
Let $B := B_1 \cup \ldots \cup B_4$.

(ii) *$S := \{s_1, \ldots, s_6\} \subset B$.*

(iii) *There is a (unique) connected component C of $G \setminus S$ such that $B \subseteq S \cup V(C)$.*

(iv) *The induced subgraph $G[S \cup V(C)]$ has a rooted tree decomposition (T^*, β^*) of height at most h with $B = \beta^*(r^*)$ for the root r^* of T^*.*

(v) *The tree decomposition (T^*, β^*) satisfies Conditions* (ii)–(iv) *of Lemma 11, where all blocks are blocks of the graph G (rather than of the subgraph $G[S \cup C]$).*

Proof. The proof is by induction on $h \geq 0$.

However, before we begin the induction, we observe that using Lemma 16 and Corollary 17, we can write a formula in the variables x_i^j that expresses Condition (i). It is straightforward to express Condition (ii), and, again using Lemma 16, to express Condition (iii). So in the induction, we will focus on Conditions (iv) and (v).

For the case $h = 0$, observe that a decomposition of height 0 consists of a single node that covers the whole graph. So we need to express that for the component C we obtain in (iii), we have $V(C) \cup S = B$. Then the 1-node tree decomposition of $G[B]$ satisfies (iv) and (v).

For a 1-node decomposition, Conditions (iii) and (iv) of Lemma 11 are void, and Condition (ii) of Lemma 11 follows from Condition (i) of this lemma.

For the inductive step $h \to h+1$, suppose we have a graph G and elements b_i^j, s_k satisfying Conditions (i)–(iii) for suitable sets B, S, C. It suffices to express that for each connected component C' of $G[S \cup V(C)] \setminus B$, we can find a decomposition of height h that covers C' and attaches to B in a way that satisfies the conditions of Lemma 11.

So let $G' := G[S \cup V(C)]$, and let C' be a connected component of $G' \setminus B$. Let $S' := N_G(C')$. If $|S'| > 6$, then there definitely is no decomposition with the desired properties. Suppose that $S' = \{s'_1, \ldots, s'_6\}$. Then, if there are $b_i'^j \in S' \cup V(C')$ such that $G \models \mathsf{dec}_h^{(n)}(b_i'^j, s'_k \mid i \in [4], j \in [3], k \in [6])$, the desired decomposition that covers C' exists by the induction hypothesis. If this is the case for all C', we can combine the decompositions to form the desired decomposition of G'. Conversely, if there is a decomposition of $G[S' \cup V(C')]$ of height h in the sense of Lemma 11 such that for the root u^*, the bag $\beta^*(u^*)$ contains S', then there are blocks or block separators $B'_1, \ldots, B'_4$ such that $\beta^*(u^*) = B'_1 \cup \ldots \cup B'_4$. From the B'_i, we obtain $b_i'^j$ such that $G \models \mathsf{dec}_h^{(n)}(b_i'^j, s'_k \mid i \in [4], j \in [3], k \in [6])$, again by the induction hypothesis.

To conclude, in addition to the subformulas taking care of Conditions (i)–(iii), the formula $\mathsf{dec}_{h+1}^{(n)}$ must have a subformula stating that, for all connected components C' of $G' \setminus B$, there exist $s'_k \in B$ for $k \in [6]$ and $b_i'^j \in S' \cup V(C')$ for $i \in [4]$, $j \in [3]$ such that $\{s'_1, \ldots, s'_6\} = N_G(C')$ and $\mathsf{dec}_h^{(n)}(b_i'^j, s'_k \mid i \in [4], j \in [3], k \in [6])$

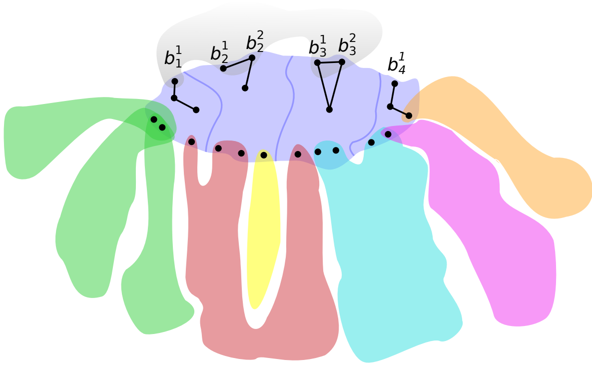


Figure 2 A (simplified) schematic visualisation of the rooted tree decomposition (T^*, β^*) from Lemma 11. For simplicity, all B_i in the bag of the purple node are depicted as distinct proper blocks.

Note that, in each step $h \to h+1$, we need to use formulas of quantifier depth $O(\log n)$ to express the desired connectivity conditions, for example to speak about components C', and to express that the b_i^j define blocks. However, the formula $\mathsf{dec}_h^{(n)}$ occurs only in the scope of constantly many (19, to be precise) quantifiers ranging over an element of the component(s) C' and the $b_i'^j, s_k'$. Overall, the quantifier depth will be $O(h) + O(\log n)$. ◀

5 Canonisation

In this section, we finally prove Theorems 1 and 2. By the logical characterisation of the WL algorithm given in Theorem 4, we obtain Theorem 1 as a corollary from Theorem 2, which we prove below.

In the following, for a graph G and a list of vertices $v_1, \dots, v_\ell \in V(G)$, we denote by $(G, v_1, \dots, v_\ell)$ the graph G with *individualised* vertices $v_1, \dots, v_\ell$. That is, $(G, v_1, \dots, v_\ell)$ and $(G', v_1', \dots, v_{\ell'}')$ have the same isomorphism type if and only if $\ell = \ell'$ and there is an isomorphism from G to G' that maps v_i to v_i' for every $i \in [\ell]$.

▶ **Lemma 19.** *For all $h \geq 0$, $n \geq 1$ and all connected planar graphs G of order $|G| \leq n$, and all $b_i^j, s_k \in V(G)$ for $i \in [4], j \in [3], k \in [6]$ (not necessarily distinct) such that*

$$G \models \mathsf{dec}_h^{(n)}(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6]),$$

there is a $\mathsf{C}_{O(h+\log n)}^{O(1)}$-formula $\mathsf{iso}_h^{(n)}(x_i^j, y_k \mid i \in [4], j \in [3], k \in [6])$ (which depends on the b_i^j and the s_k) such that the following holds. Let H be a connected graph of order $|H| \leq n$ and $b_i'^j, s_k' \in V(H)$ for $i \in [4], j \in [3], k \in [6]$ (not necessarily distinct) and assume $H \models \mathsf{dec}_h^{(n)}(b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6])$. Then

$$H \models \mathsf{iso}_h^{(n)}(b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6])$$

if and only if for the connected components C_G, C_H that Lemma 18 yields for G and H, it holds that $\big(H[\{s_1', \dots, s_6'\} \cup V(C_H)], (b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6])\big) \cong \big(G[\{s_1, \dots, s_6\} \cup V(C_G)], (b_i^j, s_k \mid i \in [4], j \in [3], k \in [6])\big)$.

Proof. For the following arguments, see also Figure 2 for a better intuition.

Let $n \in \mathbb{N}$ and let G be a connected planar graph with $|G| \leq n$. The proof is by induction on $h \geq 0$.

First, given a second connected graph H of order at most $|G|$ that satisfies the $\mathsf{dec}_h^{(n)}$-formula, we can assume that the first four triplets of vertices form the same types of blocks and block separators (of corresponding sizes), respectively, in H as in G, since otherwise we can distinguish the graphs using the formulas from Lemma 16 and Corollary 17.

Note that there is a formula $\mathsf{bag}^{(n)}(x_1^1, \ldots, x_4^3, y) \in \mathsf{C}_{O(\log n)}^{O(1)}$ such that for all graphs H of order at most n and all $b_1'^1, b_1'^2, b_1'^3, \ldots, b_4'^1, b_4'^2, b_4'^3, v \in V(H)$, it holds that $H \models \mathsf{bag}^{(n)}(b_1'^1, \ldots, b_4'^3, v)$ if and only if each set $\{b_i'^j \mid j \in [3]\}$ for $i \in [4]$ is a block separator B_i or a degenerate block B_i or contained in a proper block B_i of H and v is in $B \coloneqq \bigcup_{i=1}^4 B_i$.

The case that $h = 0$ follows analogously as the formula for the isomorphism type of the root bag in the inductive step. We therefore focus on the inductive step. Assume that for every list of vertices $(b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6]) \in V(G)^{18}$, where

$$G \models \mathsf{dec}_h^{(n)}(b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6]),$$

there is a $\mathsf{C}_{O(h+\log n)}^{O(1)}$-formula

$$\mathsf{iso}_{G,(b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6])}(x_1'^1, \ldots, x_4'^3, y_1', \ldots, y_6')$$

that defines the isomorphism type of $(G[\{s_1', \ldots, s_6'\} \cup V(C')], (b_i'^j, s_k' \mid i \in [4], j \in [3], k \in [6]))$, where C' is the connected component from Parts (iii)–(v) in Lemma 18.

Let $(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6]) \in V(G)^{18}$ be a list of vertices such that

$$G \models \mathsf{dec}_{h+1}^{(n)}(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6]).$$

For $B_1, B_2, B_3, B_4, B, C, S$ as described in Lemma 18, let (T^*, β^*) be the rooted tree decomposition from Condition (iv) in Lemma 18. Let r^* be the root of T^*. By Condition (iv) in Lemma 18, it holds that $\beta^*(r^*) = B = \bigcup_{i=1}^4 B_i$. Consider a B_i with $|B_i| \geq 4$. Then B_i is a proper block, in which, by Theorem 5, we can find vertices v_i^1, v_i^2, v_i^3 such that for all $w \in B_i$, there is a $\mathsf{C}_{O(\log n)}^{O(1)}$-formula $\mathsf{id}'_w(x_i^1, x_i^2, x_i^3, y)$ such that $G[[B_i]] \models \mathsf{id}'_w(v_i^1, v_i^2, v_i^3, w)$ and $G[[B_i]] \not\models \mathsf{id}'_w(v_i^1, v_i^2, v_i^3, w')$ for every $w' \in B_i \setminus \{v\}$. (In every B_i with $|B_i| \leq 3$, such vertex-identifying formulas with four free variables exist trivially and they also identify the vertex the entire graph G.)

For simplicity, first assume that for all i with $|B_i| \geq 4$, the vertex v_i^j equals b_i^j for $j \in [3]$. Then by replacing in $\mathsf{id}'_v(x_i^1, x_i^2, x_i^3, y)$ every subformula of the form $\exists^{\geq k} x \psi$ with $\exists^{\geq k} x(\psi \wedge \mathsf{block}^{(n)}(x_i^1, x_i^2, x_i^3, x))$ and every $E(x, y)$ with $\mathsf{torso}^{(n)}(x_i^1, x_i^2, x_i^3, x, y)$, we easily obtain for each $v \in B$ a $\mathsf{C}_{O(\log n)}^{O(1)}$-formula $\mathsf{id}_v(x_1^1, \ldots, x_4^3, y)$ with $\mathsf{id}_v[G, b_1^1, \ldots, b_4^3, y] = \{v\}$.

Now we can use these formulas to address each vertex individually. More formally, we can define the edge relation of $G[B]$ by setting, for $v, w \in B$ with $v \neq w$,

$$\varphi_{v,w}(x, y) \coloneqq \begin{cases} E(x, y) & \text{if } vw \in E(G), \\ \neg E(x, y) & \text{otherwise.} \end{cases}$$

Then the $\mathsf{C}_{O(\log n)}^{O(1)}$-formula

$$\begin{aligned} \mathsf{iso}_B(x_1^1, \ldots, x_4^3) \coloneqq & \bigwedge_{v,w \in B} \exists^{=1} x \Big(\mathsf{id}_v(x_1^1, \ldots, x_4^3, x) \wedge \exists^{=1} x' \big(\mathsf{id}_w(x_1^1, \ldots, x_4^3, x') \wedge \varphi_{v,w}(x, x') \big) \Big) \\ & \wedge \neg \exists x \Big(\mathsf{bag}^{(n)}(x_1^1, \ldots, x_4^3, x) \to \bigwedge_{w \in B} \neg \mathsf{id}_w(x_1^1, \ldots, x_4^3, x) \Big) \wedge \\ & \bigwedge_{v \neq w \in B} \neg \exists x \Big(\mathsf{id}_v(x_1^1, \ldots, x_4^3, x) \wedge \mathsf{id}_w(x_1^1, \ldots, x_4^3, x) \Big) \end{aligned}$$

defines the isomorphism type of $(G[B], b_1^1, \ldots, b_4^3)$ (see the purple bag in Figure 2).

We now construct a formula that describes how the connected components of $G[S \cup V(C)] \setminus B$ are attached to $G[B]$. Let $G' := G[S \cup V(C)]$. By Condition (iii) in Lemma 11, for every connected component C' of $G' \setminus B$, it holds that $|N_G(C')| \leq 6$ (see the coloured shapes attached to the purple one in Figure 2). Hence, we iterate over all tuples $(s'_1, \dots, s'_6) \in B^6$: let $\mathcal{M}^{s'_1,\dots,s'_6}$ be the multiset of isomorphism types of the graphs $(G[S' \cup C'], s'_1, \dots, s'_6)$, where $S' := \{s'_i \mid i \in [6]\}$ and C' is a connected component of $G' \setminus B$ with $N_G(C') = S'$.

Since $G \models \mathsf{dec}^{(n)}_{h+1}(b^j_i, s_k \mid i \in [4], j \in [3], k \in [6])$, for every $(s'_1, \dots, s'_6) \in B^6$ and every connected component C' of $G' \setminus B$ with $N_G(C') = \{s'_1, \dots, s'_6\}$, there exist vertices $(b'^j_i \mid i \in [4], j \in [3]) \in (S' \cup V(C'))^{12}$ such that

$$G[S' \cup V(C')] \models \mathsf{dec}^{(n)}_h(b'^j_i, s'_k \mid i \in [4], j \in [3], k \in [6]).$$

So, by the induction hypothesis, there is a formula $\mathsf{iso}_M(x'^1_1, \dots, x'^3_4, y'_1, \dots, y'_6) \in \mathsf{C}^{O(1)}_{O(h+\log n)}$ for the isomorphism type M of $(G[\{s'_1, \dots, s'_6\} \cup V(C')], (b'^j_i, s'_k \mid i \in [4], j \in [3], k \in [6]))$. Note that by Condition (ii) in Lemma 18, at least one of the vertices b'^j_i will lie outside B. Using the counting quantifiers, we can use the iso_M to make sure that every isomorphism type appears with the correct multiplicity. More precisely, we first group all components with equal isomorphism types. The fact that they are of the same size enables us to define their number (e.g. the three green shapes in Figure 2). This then allows us to build a formula $\mathsf{iso}'_{\mathcal{M}}(y'_1, \dots, y'_6)$ which identifies the graph $(G[S' \cup \bigcup_{C' : N_G(C') = S'} V(C')], s'_1, \dots, s'_6)$ (where $\mathcal{M} := \mathcal{M}^{s'_1,\dots,s'_6}$ and the C' are connected components of $G' \setminus B$). Using the $\mathsf{dec}^{(n)}_h$-formula, we can turn $\mathsf{iso}'_{\mathcal{M}}(y'_1, \dots, y'_6)$ into a formula $\mathsf{iso}_{\mathcal{M}}(x^1_1, \dots, x^3_4, y_1, \dots, y_6, y'_1, \dots, y'_6)$ that ensures that $\mathsf{iso}_{\mathcal{M}}(b^1_1, \dots, b^3_4, s_1, \dots, s_6, s'_1, \dots, s'_6)$ describes for $S' := \{s'_1, \dots, s'_6\}$ the subgraph $(G[S' \cup \bigcup_{C' : N_G(C') = S'} V(C')], s'_1, \dots, s'_6)$, where the C' are connected components of $G' \setminus B$, up to isomorphism.

Hence, it suffices to conjugate $\mathsf{iso}_B(x^1_1, \dots, x^3_4)$ with a conjunction over all $(s'_1, \dots, s'_6) \in B^6$ of the following formula

$$\exists y'_1 \dots \exists y'_6 \left(\bigwedge_{i=1}^{6} \mathsf{id}_{s'_i}(x^1_1, \dots, x^3_4, y'_i) \wedge \mathsf{iso}_{\mathcal{M}}(x^1_1, \dots, x^3_4, y_1, \dots, y_6, y'_1, \dots, y'_6) \right),$$

where $\mathcal{M} := \mathcal{M}^{s_1,\dots,s_6}$, to obtain the desired $\mathsf{iso}_{G,(b^j_i, s_k \mid i \in [4], j \in [3], k \in [6])}(x^1_1, \dots, x^3_4, y_1, \dots, y_6)$.

We now consider the general case where it does not necessarily hold for all i, j that $v^j_i = b^j_i$. We assume for notational simplicity that for all i, the b^1_i, b^2_i, b^3_i define a block. It is easy to adapt the following construction to the situation that block separators are present.

We introduce one nested existential quantifier $\exists \tilde{x}^j_i$ for each of the v^j_i so that our resulting formula $\mathsf{iso}_{G,(b^j_i, s_k \mid i \in [4], j \in [3], k \in [6])}(x^1_1, \dots, x^3_4, y_1, \dots, y_6)$ looks as follows:

$$\exists \tilde{x}^1_1 \dots \exists \tilde{x}^3_4 \Bigg(\bigwedge_{j=1}^{3} \bigwedge_{i=1}^{4} \mathsf{block}^{(n)}(x^1_i, x^2_i, x^3_i, \tilde{x}^j_i) \wedge \mathsf{iso}_B(\tilde{x}^1_1, \dots, \tilde{x}^3_4) \wedge \bigwedge_{(s'_1,\dots,s'_6) \in B^6} \exists y'_1 \dots \exists y'_6 \Bigg(\bigwedge_{i=1}^{6} \mathsf{id}_{s'_i}(\tilde{x}^1_1, \dots, \tilde{x}^3_4, y'_i) \wedge \mathsf{iso}_{\mathcal{M}}(x^1_1, \dots, x^3_4, y_1, \dots, y_6, y'_1, \dots, y'_6) \Bigg) \Bigg).$$

The bounds on the quantifier depth and the number of variables follow similarly as in the proof of Lemma 18. ◀

Applying Lemma 8, we can deduce Theorem 2.

Proof of Theorem 2. Let $n \in \mathbb{N}$ and let G be a planar graph with order $|G| = n$. If G is not connected, we construct one formula for each connected component of G (as described in the following) and join them to obtain the identifying sentence.

So suppose G is connected. Then by Lemma 11, G has a rooted tree decomposition (T^*, β^*) of logarithmic height and adhesion at most 6 for which every bag is a union of four (not necessarily distinct) blocks or block separators and also Condition (iv) of the lemma holds. Let $b_1^1, \dots, b_4^3$ be vertices that determine the blocks and block separators in the root bag B of (T^*, β^*).

If there is a vertex $s \in B$ such that there is a unique connected component C of $G \setminus \{s\}$ with $B \subseteq \{s\} \cup V(C)$, then there are vertices b_i^j, s_k for $i \in [4], j \in [3], k \in [6]$ (e.g. $s_k = s$ for all k) such that G satisfies $\mathsf{dec}^{(n)}_{2\log|G|}(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6])$. Then the sentence

$$\exists x_1^1 \dots \exists x_4^3 \exists y_1 \dots \exists y_6 \mathsf{iso}_{G,(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6])}(x_1^1, \dots, x_4^3, y_1, \dots, y_6)$$

identifies G, where $\mathsf{iso}_{G,(b_i^j, s_k \mid i \in [4], j \in [3], k \in [6])}$ is the formula from Lemma 19.

Otherwise, let $s \in B$ be a vertex such that $G \setminus \{s\}$ has multiple connected components C_i and let $G_i \coloneqq G[V(C_i) \cup \{s\}]$. Then the restriction of (T^*, β^*) to each G_i still satisfies the conditions of Lemma 11, because the block structure of G_i is just the block structure induced by G on $V(G_i)$ (that is, the blocks of G_i are precisely those blocks of G contained in $V(G_i)$, and similarly for the block separators). This yields by Lemma 19 an identifying formula $\varphi_i(y)$ for each (G_i, s), which we can join by isomorphism type of (G_i, s) to obtain an identifying sentence. ◀

We can directly deduce Theorem 1.

Proof of Theorem 1. The theorem follows from Theorems 2 and 4. ◀

6 Conclusion

We prove that planar graphs are identified by the WL algorithm with constant dimension in a logarithmic number of iterations, thereby completing a project started by Verbitsky fourteen years ago with his proof of the same result in the special case of 3-connected planar graphs. Our proof is based on the careful analysis of a novel logarithmic-depth decomposition of graphs into their 3-connected components.

It is unclear which dimension of the WL algorithm is necessary to identify planar graphs in logarithmically many iterations and if there is a (provable) trade-off between dimension and iteration number. This is not only interesting for planar graphs, and many questions remain open.

We leave it as another interesting open project whether our result can be extended to graph classes of bounded genus. As it stands, our proof heavily relies on properties of 3-connected planar graphs that are not shared by 3-connected graphs of higher genus. Similarly, we pose as a challenge to find good bounds on the iteration number of the WL algorithm on other parameterised graph classes, such as graphs with a certain excluded minor or graphs of bounded rank width.

References

1 B. Ahmadi, K. Kersting, M. Mladenov, and S. Natarajan. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning Journal*, 92(1):91–132, 2013. doi:10.1007/s10994-013-5385-0.

2 A. Atserias and E. N. Maneva. Sherali-Adams relaxations and indistinguishability in counting logics. *SIAM Journal on Computing*, 42(1):112–137, 2013. doi:10.1137/120867834.

3 A. Atserias and J. Ochremiak. Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '18)*, pages 66–75, 2018. doi:10.1145/3209108.3209186.

4 L. Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, pages 684–697, 2016. doi:10.1145/2897518.2897542.

5 J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992. doi:10.1007/BF01305232.

6 G. Chen and I. Ponomarenko. Lectures on coherent configurations. Lecture notes available at http://www.pdmi.ras.ru/~inp/ccNOTES.pdf, 2019.

7 P. T. Darga, M. H. Liffiton, K. A. Sakallah, and I. L. Markov. Exploiting structure in symmetry detection for CNF. In *Proceedings of the 41st Design Automation Conference (DAC '04)*, pages 530–534. ACM, 2004. doi:10.1145/996566.996712.

8 H. Dell, M. Grohe, and G. Rattan. Lovász meets Weisfeiler and Leman. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP '18)*, pages 40:1–40:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.40.

9 R. Diestel. *Graph Theory*. Springer Verlag, 5th edition, 2016.

10 Z. Dvorák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010. doi:10.1002/jgt.20461.

11 M. Elberfeld, A. Jakoby, and T. Tantau. Logspace versions of the theorems of Bodlaender and Courcelle. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '10)*, pages 143–152, 2010. doi:10.1109/FOCS.2010.21.

12 S. Evdokimov, I. N. Ponomarenko, and G. Tinhofer. Forestal algebras and algebraic forests (on a new class of weakly compact graphs). *Discrete Mathematics*, 225(1-3):149–172, 2000. doi:10.1016/S0012-365X(00)00152-7.

13 M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science (LICS '98)*, pages 6–15, 1998. doi:10.1109/LICS.1998.705639.

14 M. Grohe. Isomorphism testing for embeddable graphs through definability. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC '00)*, pages 63–72, 2000. doi:10.1145/335305.335313.

15 M. Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781139028868.

16 M. Grohe. The logic of graph neural networks. In *Proceedings of the 36th ACM-IEEE Symposium on Logic in Computer Science (LICS '21))*, 2021. arXiv version at arXiv:2104.14624.

17 M. Grohe and S. Kiefer. A linear upper bound on the Weisfeiler-Leman dimension of graphs of bounded genus. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP '19)*, pages 117:1–117:15, 2019. doi:10.4230/LIPIcs.ICALP.2019.117.

18 M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *Proceedings of the 7th International Conference on Database Theory (ICDT '99)*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer, 1999. doi:10.1007/3-540-49257-7_6.

19 M. Grohe and D. Neuen. Canonisation and definability for graphs of bounded rank width. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '19)*, pages 1–13, 2019. doi:10.1109/LICS.2019.8785682.

20 M. Grohe and M. Otto. Pebble games and linear equations. *Journal of Symbolic Logic*, 80(3):797–844, 2015. doi:10.1017/jsl.2015.28.

21 M. Grohe and O. Verbitsky. Testing graph isomorphism in parallel by playing a game. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP '06)*, pages 3–14, 2006. doi:10.1007/11786986_2.

22 N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity theory retrospective*, pages 59–81. Springer-Verlag, 1990.

23 T. A. Junttila and P. Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments (ALENEX '07)*. SIAM, 2007. doi:10.1137/1.9781611972870.13.

24 S. Kiefer. The Weisfeiler-Leman algorithm: An exploration of its power. *ACM SIGLOG News*, 7(3):5–27, 2020. doi:10.1145/3436980.3436982.

25 S. Kiefer and B. D. McKay. The iteration number of Colour Refinement. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP '20)*, volume 168 of *LIPIcs*, pages 73:1–73:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.73.

26 S. Kiefer and D. Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. In *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS '19)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.45.

27 S. Kiefer, I. Ponomarenko, and P. Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. *J. ACM*, 66(6):44:1–44:31, 2019. doi:10.1145/3333003.

28 S. Kiefer and P. Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first-order logic. *Log. Methods Comput. Sci.*, 15(2), 2019. doi:10.23638/LMCS-15(2:19)2019.

29 J. Köbler and O. Verbitsky. From invariants to canonization in parallel. In *Proceedings of the 3rd International Computer Science Symposium in Russia (CSR '08)*, volume 5010 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2008. doi:10.1007/978-3-540-79709-8_23.

30 M. Lichter, I. Ponomarenko, and P. Schweitzer. Walk refinement, walk logic, and the iteration number of the Weisfeiler-Leman algorithm. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '19)*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785694.

31 B. D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

32 B. D. McKay and A. Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.

33 C. Morris, M. Ritzert, M. Fey, W. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019. doi:10.1609/aaai.v33i01.33014602.

34 N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

35 W. T. Tutte. *Graph Theory*. Addison-Wesley, 1984.

36 O. Verbitsky. Planar graphs: Logical complexity and parallel isomorphism tests. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS '07)*, pages 682–693, 2007. doi:10.1007/978-3-540-70918-3_58.

37 B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series 2*, 1968. English translation by G. Ryabov available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.

38 H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54:150–168, 1932.

39 K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations (ICLR '19)*, 2019.