

The Power of the Weisfeiler-Leman Algorithm to Decompose Graphs

Sandra Kiefer 

RWTH Aachen University, Aachen, Germany
kiefer@cs.rwth-aachen.de

Daniel Neuen 

RWTH Aachen University, Aachen, Germany
neuen@cs.rwth-aachen.de

Abstract

The Weisfeiler-Leman procedure is a widely-used approach for graph isomorphism testing that works by iteratively computing an isomorphism-invariant coloring of vertex tuples. Meanwhile, a fundamental tool in structural graph theory, which is often exploited in approaches to tackle the graph isomorphism problem, is the decomposition into bi- and triconnected components.

We prove that the 2-dimensional Weisfeiler-Leman algorithm implicitly computes the decomposition of a graph into its triconnected components. Thus, the dimension of the algorithm needed to distinguish two given graphs is at most the dimension required to distinguish the corresponding decompositions into 3-connected components (assuming dimension at least 2).

Our result implies that for $k \geq 2$, the k -dimensional algorithm distinguishes k -separators, i.e., k -tuples of vertices that separate the graph, from other vertex k -tuples. As a byproduct, we also obtain insights about the connectivity of constituent graphs of association schemes.

In an application of the results, we show the new upper bound of k on the Weisfeiler-Leman dimension of graphs of treewidth at most k . Using a construction by Cai, Fürer, and Immerman, we also provide a new lower bound that is asymptotically tight up to a factor of 2.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Theory of computation → Finite Model Theory; Theory of computation → Graph algorithms analysis

Keywords and phrases Weisfeiler-Leman, separators, first-order logic, counting quantifiers

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.45

Related Version <http://arxiv.org/abs/1908.05268>

1 Introduction

Originally introduced in [37], the Weisfeiler-Leman (WL) algorithm has become a – if not the – fundamental subroutine in the context of isomorphism testing for graphs. It is used in theoretical as well as in practical approaches to tackle the graph isomorphism problem (see e.g. [5, 19, 31, 32, 35]), among them also Babai’s recent quasipolynomial-time isomorphism test [4]. For every $k \geq 1$, there is a k -dimensional version of the algorithm which colors the vertex k -tuples of the input graph and iteratively refines the coloring in an isomorphism-invariant manner.

There are various characterizations of the algorithm, which link it to other areas in theoretical computer science (see also *Further Related Work*). For example, very recent results in the context of machine learning show that the 1-dimensional version of the algorithm is as expressive as graph neural networks with respect to distinguishing graphs [34]. Following Grohe [16], an indicator to investigate the expressive power of the algorithm is the so-called *WL dimension* of a graph, defined as the minimal dimension of the WL algorithm required in order to distinguish the graph from every other non-isomorphic graph.

There is no fixed dimension of the algorithm that decides graph isomorphism in general, as was proved by Cai, Fürer, and Immerman [9]. Still, when focusing on particular graph classes, often a bounded dimension of the algorithm suffices to identify every graph in the



© Sandra Kiefer and Daniel Neuen;
licensed under Creative Commons License CC-BY

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).

Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 45; pp. 45:1–45:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

class. This proves that for the considered class, graph isomorphism is solvable in polynomial time, since the k -dimensional algorithm can be implemented in time $\mathcal{O}(n^{k+1} \log n)$ [27]. For example, it suffices to apply the 3-dimensional WL algorithm to identify every planar graph [29]. Also, the WL dimension of graphs of treewidth at most k is bounded by $k + 2$ [18]. More generally, by a celebrated result by Grohe, for all graph classes with an excluded minor, the WL dimension is bounded [15]. Very recent work provides explicit upper bounds on the WL dimension, which are linear in the rank width [19] and in the Euler genus [17], respectively, of the graph.

Regarding combinatorial techniques, to handle graphs with complex structures, the decomposition into connected, biconnected, and triconnected components provides a fundamental tool from structural graph theory. The decomposition can be computed in linear time (see e.g. [25, 36]). Hopcroft and Tarjan used the decomposition of a graph into its triconnected components to obtain an algorithm that decides isomorphism for planar graphs in quasi-linear time [22, 23, 24], which was improved to linear time by Hopcroft and Wong [26].

Also, in [29], to prove the bound on the WL dimension for the class of planar graphs, the challenge of distinguishing two arbitrary planar graphs is reduced to the case of two arc-colored triconnected planar graphs, by exploiting the fact that the 3-dimensional WL algorithm is able to implicitly compute the decomposition of a graph into its triconnected components. Similarly, the bound on the WL dimension for graphs parameterized by their Euler genus from [17] relies on an isomorphism-invariant decomposition of the graphs into their triconnected components.

Our Contribution. We show that for $k \geq 2$, the k -dimensional WL algorithm implicitly computes the decomposition into the triconnected components of a given graph. More specifically, we prove that already the 2-dimensional WL algorithm distinguishes separating pairs, i.e., pairs of vertices that separate the given graph, from other vertex pairs. This improves on a result from [29], where an analogous statement was proved for the 3-dimensional WL algorithm. Using the decomposition techniques discussed there, we conclude that for the k -dimensional WL algorithm with $k \geq 2$, to identify a graph, it suffices to determine vertex orbits on all arc-colored 3-connected components of it. Since it is easy to see that $k = 1$ does not suffice to distinguish vertices contained in 2-separators from others, our upper bound of 2 is tight.

The expressive power of the k -dimensional algorithm corresponds to definability in the logic \mathcal{C}^{k+1} , the extension of the $(k + 1)$ -variable fragment of first-order logic by counting quantifiers [9, 27]. Exploiting this correspondence, our results imply that for every $n \in \mathbb{N}$, there is a formula $\varphi_n(x_1, x_2) \in \mathcal{C}^3$ (first-order logic with counting quantifiers over three variables) such that for an n -vertex graph G , it holds that $G \models \varphi_n(v, w)$ if and only if $\{v, w\}$ is a 2-separator in G . With only three variables at our disposal, it is not possible to take the route of [29] by comparing certain numbers of walks between different pairs of vertices. Instead, the formulas obtained from our proof are essentially a disjunction over all n -vertex graphs and subformulas for two distinct graphs may look completely different, exploiting specific structural properties of the graphs. While this makes the proof rather involved, it also stresses the power of the 2-dimensional WL algorithm and equivalently, the expressive power of the logic \mathcal{C}^3 . We show that for all $n, s \in \mathbb{N}$, there is a formula $\varphi_{n,s}(x_1, x_2, x_3) \in \mathcal{C}^3$ such that for an n -vertex graph G , it holds that $G \models \varphi_{n,s}(u, v, w)$ if and only if $s = |C|$, where C is the vertex set of the connected component containing u after removing v and w from the graph G .

Our result can also be viewed in a combinatorial setting. In 1985, Brouwer and Mesner [8] proved that the vertex connectivity of a strongly regular graph equals its degree and that in fact, the only minimal disconnecting vertex sets are neighborhoods. Later, Brouwer conjectured this to be true for any constituent graph of an association scheme (i.e., any graph consisting in a single color class of the association scheme) [6]. While some progress has been made on certain special cases [13], most prominently distance-regular graphs [7], the general question is still open. Our results imply that any connected constituent graph of an association scheme is either a cycle or 3-connected. Such a statement was previously only known for symmetric association schemes [30], which are far more restricted than the general ones.

A natural use case of these results is to determine or to improve upper bounds on the WL dimension of certain graph classes. As a first application in this direction, we obtain a new upper bound of k on the WL dimension for graphs of treewidth at most k . Based on [10], we also provide a new lower bound for this graph class, thus delimiting the value of the WL dimension of graphs of treewidth bounded by k to the interval $[\lceil \frac{k}{2} \rceil - 3, k]$.

Due to space restrictions, some of the proofs and proof details have been omitted. For a full version of this paper, we refer the reader to [28].

Further Related Work. Apart from its correspondence to counting logics, the WL algorithm has further surprising links to other areas. For example, the algorithm has a close connection to Sherali-Adams relaxations of particular linear programs [3, 20] and captures the same information as certain homomorphism counts [11]. It can also be characterized via winning strategies in so-called pebble games [21], which are a particular family of Ehrenfeucht-Fraïssé games.

As mentioned above, the 1-dimensional WL algorithm essentially corresponds to graph neural networks. In order to make them more powerful, the authors of [34] propose an extension of graph neural networks based on the k -dimensional WL algorithm (see also [33]).

Towards understanding the expressive power of the algorithm, in a related direction of research, it has been studied which graph properties the WL algorithm can detect, which may become particularly relevant in the graph-learning framework. In this context, Fürer [14] as well as Arvind et al. [2] obtained results concerning the ability of the algorithm to detect and count certain subgraphs.

2 Preliminaries

2.1 Graphs

A *graph* is a pair $G = (V(G), E(G))$ of a *vertex set* $V(G)$ and an *edge set* $E(G) \subseteq \{\{u, v\} \mid u, v \in V(G)\}$. To give explicit reference to G , we also write $V(G)$ for V and $E(G)$ for E . All graphs considered in this paper are finite, simple (i.e., they contain no loops or multiple edges), and undirected. For $v, w \in V$, we also write vw as a shorthand for $\{v, w\}$. The *neighborhood* of v is denoted by $N(v)$, and the *closed neighborhood* of v is $N[v] := N(v) \cup \{v\}$. The *degree* of v , denoted by $\deg(v)$, is the number of edges incident with v . For $X \subseteq V(G)$ we define $N(X) := (\bigcup_{v \in X} N(v)) \setminus X$.

A *walk of length k* from v to w is a sequence of vertices $v = u_0, u_1, \dots, u_k = w$ such that $u_{i-1}u_i \in E$ for all $i \in \{1, \dots, k\}$. A *path of length k* from v to w is a walk of length k from v to w for which all occurring vertices are pairwise distinct. We refer to the *distance* between two vertices $v, w \in V(G)$ by $\text{dist}(v, w)$. For a set $A \subseteq V(G)$, we denote by $G[A]$ the *induced subgraph* of G on vertex set A . Also, we denote by $G - A$ the subgraph induced by the

complement of A , that is, the graph $G - A := G[V(G) \setminus A]$. A set $S \subseteq V(G)$ is a *separator* of G if $G - S$ has more connected components than G . A *k-separator* of G is a separator of G of size k . A vertex $v \in V(G)$ is a *cut vertex* if $\{v\}$ is a separator of G . The graph G is *k-connected* if it is connected and has no $(k - 1)$ -separator.

An *isomorphism* from G to another graph H is a bijection $\varphi: V(G) \rightarrow V(H)$ that respects the edge relation, that is, for all $v, w \in V(G)$, it holds that $vw \in E(G)$ if and only if $\varphi(v)\varphi(w) \in E(H)$. Two graphs G and H are *isomorphic* ($G \cong H$) if there is an isomorphism from G to H . We write $\varphi: G \cong H$ to denote that φ is an isomorphism from G to H .

A *vertex-colored graph* is a tuple (G, χ) , where G is a graph and $\chi: V(G) \rightarrow \mathcal{C}$ is a mapping into some set \mathcal{C} of colors. Similarly, an *arc-colored graph* is a tuple (G, χ) , where G is a graph and $\chi: \{(v, v) \mid v \in V(G)\} \cup \{(u, v) \mid \{u, v\} \in E(G)\} \rightarrow \mathcal{C}$ is a mapping into some color set \mathcal{C} . Typically, \mathcal{C} is chosen to be an initial segment $[n] := \{1, \dots, n\}$ of the natural numbers. Isomorphisms between vertex- and arc-colored graphs have to respect the colors of the vertices and arcs.

For details on the *treewidth* of a graph, we refer the reader to [12].

2.2 The Weisfeiler-Leman Algorithm

Let $\chi_1, \chi_2: V^k \rightarrow \mathcal{C}$ be colorings of the k -tuples of vertices of G , where \mathcal{C} is some finite set of colors. We say χ_2 *refines* χ_1 if for all $\bar{v}, \bar{w} \in V^k$ we have $(\chi_2(\bar{v}) = \chi_2(\bar{w}) \Rightarrow \chi_1(\bar{v}) = \chi_1(\bar{w}))$. The k -dimensional WL algorithm is a procedure that, given a graph G and a coloring χ of its k -tuples of vertices, computes an isomorphism-invariant refinement of the coloring.

We describe the mechanisms of the algorithm in the following. For an integer $k > 1$ and a vertex-colored graph (G, χ) , we let $\chi_{G,k}^0: V^k \rightarrow \mathcal{C}$ be the coloring where each k -tuple is colored with the isomorphism type of its underlying ordered colored subgraph. More formally, $\chi_{G,k}^0(v_1, \dots, v_k) = \chi_{G,k}^0(w_1, \dots, w_k)$ if and only if for all $i \in [k]$ it holds that $\chi(v_i) = \chi(w_i)$ and for all $i, j \in [k]$, it holds that $v_i = v_j \Leftrightarrow w_i = w_j$ and $v_i v_j \in E(G) \Leftrightarrow w_i w_j \in E(G)$. If G is arc-colored, the arc colors must be respected accordingly.

We then recursively define the coloring $\chi_{G,k}^i$ obtained after i rounds of the algorithm. Let $\chi_{G,k}^{i+1}(v_1, \dots, v_k) := (\chi_{G,k}^i(v_1, \dots, v_k); \mathcal{M})$, where \mathcal{M} is a multiset defined as

$$\{\{\chi_{G,k}^i(\bar{v}[w/1]), \chi_{G,k}^i(\bar{v}[w/2]), \dots, \chi_{G,k}^i(\bar{v}[w/k])\} \mid w \in V\}$$

where $\bar{v}[w/i] := (v_1, \dots, v_{i-1}, w, v_{i+1}, \dots, v_k)$.

For the 1-dimensional algorithm (i.e. $k = 1$), the definition is similar, but we iterate only over the neighbors of v_1 , that is, the multiset \mathcal{M} equals $\{\{\chi_{G,1}^i(w) \mid w \in N(v_1)\}\}$.

By definition, every coloring $\chi_{G,k}^{i+1}$ induces a refinement of the partition of the k -tuples of vertices of the graph G with coloring $\chi_{G,k}^i$. Thus, there is a minimal i such that the partition of the vertex k -tuples induced by $\chi_{G,k}^{i+1}$ is not strictly finer than the one induced by $\chi_{G,k}^i$. For this value of i , we call the coloring $\chi_{G,k}^i$ the *stable* coloring of G and denote it by $\chi_{G,k}$.

The original WL algorithm is its 2-dimensional variant [37]. Since that version is the central algorithm of this paper, we omit the index 2 and write χ_G instead of $\chi_{G,2}$.

For $k \in \mathbb{N}$, the *k-dimensional WL algorithm* takes as input a (vertex- or arc-)colored graph (G, χ) and returns the coloring $\chi_{G,k}$. The procedure can be implemented in time $O(n^{k+1} \log n)$ [27]. For two graphs G and H , we say that the k -dimensional WL algorithm *distinguishes* G and H if there is a color c such that the sets $\{\bar{v} \mid \bar{v} \in (V(G))^k, \chi_{G,k}(\bar{v}) = c\}$ and $\{\bar{w} \mid \bar{w} \in (V(H))^k, \chi_{H,k}(\bar{w}) = c\}$ have different cardinalities. We write $G \simeq_k H$ if the k -dimensional WL algorithm does not distinguish G and H . The algorithm *identifies* G if it distinguishes G from every non-isomorphic graph H .

Pebble Games. For further analysis, it is often cumbersome to work with the WL algorithm directly and more convenient to use the following characterization via pebble games, which is known to capture the same information. Let $k \in \mathbb{N}$. For graphs G and H on the same number of vertices and with vertex colorings χ and χ' , respectively, we define the *bijective k -pebble game* $\text{BP}_k(G, H)$ as follows:

- The game has two players called *Spoiler* and *Duplicator*.
- The game proceeds in rounds, each of which is associated with a pair of positions (\bar{v}, \bar{w}) with $\bar{v} \in (V(G))^\ell$ and $\bar{w} \in (V(H))^\ell$, where $0 \leq \ell \leq k$.
- The initial position of the game is a pair of vertex tuples of equal length ℓ with $0 \leq \ell \leq k$. If not specified otherwise, the initial position is the pair $((), ())$ of empty tuples.
- Each round consists of the following steps. Suppose the current position of the game is $(\bar{v}, \bar{w}) = ((v_1, \dots, v_\ell), (w_1, \dots, w_\ell))$. First, Spoiler chooses whether to remove a pair of pebbles or to play a new pair of pebbles. The first option is only possible if $\ell > 0$, and the latter option is only possible if $\ell < k$.

If Spoiler wishes to remove a pair of pebbles, he picks some $i \in [\ell]$ and the game moves to position $(\bar{v} \setminus i, \bar{w} \setminus i)$ where $\bar{v} \setminus i := (v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_\ell)$, and the tuple $(\bar{w} \setminus i)$ is defined in the analogous way. Otherwise, the following steps are performed.

(D) Duplicator picks a bijection $f: V(G) \rightarrow V(H)$.

(S) Spoiler chooses $v \in V(G)$. The new position is then $((v_1, \dots, v_\ell, v), (w_1, \dots, w_\ell, f(v)))$.

If for the current position $((v_1, \dots, v_\ell), (w_1, \dots, w_\ell))$, the induced ordered subgraphs of G and H are not isomorphic, Spoiler wins the play. More precisely, Spoiler wins if there is an $i \in [\ell]$ such that $\chi(v_i) \neq \chi'(w_i)$, or there are $i, j \in [\ell]$ such that $v_i = v_j \not\leftrightarrow w_i = w_j$ or $v_i v_j \in E(G) \not\leftrightarrow w_i w_j \in E(H)$. If there is no position of the play such that Spoiler wins, then Duplicator wins.

We say that Spoiler (and Duplicator, respectively) *wins the bijective k -pebble game* $\text{BP}_k(G, H)$ if Spoiler (and Duplicator, respectively) has a winning strategy for the game.

The following theorem describes the correspondence between the Weisfeiler-Leman algorithm and the introduced pebble games.¹

► **Theorem 1** (see e.g. [9]). *Let G and H be two graphs. Then $G \simeq_k H$ if and only if Duplicator wins the game $\text{BP}_{k+1}(G, H)$.*

Association Schemes. Let V be a set. An *association scheme* on V is an ordered partition (R_0, \dots, R_d) of V^2 such that

1. $R_0 = \{(v, v) \mid v \in V\}$, and
2. for every $i \in [d]$, there is a $j \in [d]$ such that the set $R_i^\top := \{(w, v) \mid (v, w) \in R_i\}$ equals R_j , and
3. for all i, j, k , there are numbers $p_{i,j}^k$ such that for all $(v, w) \in R_k$,

$$p_{i,j}^k = |\{x \in V \mid (v, x) \in R_i \text{ and } (w, x) \in R_j\}|.$$

An association scheme is *symmetric* if $R_i^\top = R_i$ for all $i \in [d]$. With each R_i , we associate a directed *constituent graph* $G(R_i)$ of the association scheme, defined as $G(R_i) := (V, R_i \cup R_i^\top)$.

¹ The pebble games in [9] are defined slightly differently. Still, a player has a winning strategy in the game described there if and only if they have one in our game and thus, Theorem 1 holds for both versions of the game.

Every association scheme induces a coloring on V^2 , in which every (v, w) is colored with the relation it is contained in. This coloring is stable in the sense that it is not refined by the 2-dimensional WL algorithm (when V is interpreted as the vertex set of a complete directed graph). That is, for all $j \in [d]$ and all $(v_1, v_2), (w_1, w_2) \in R_j$, it holds that $\chi_{G(R_i)}(v_1, v_2) = \chi_{G(R_i)}(w_1, w_2)$. Conversely, every colored graph G with $\chi_G(v, v) = \chi_G(w, w)$ for all $v, w \in V(G)$ induces an association scheme in which the relations R_i are the color classes of the coloring $\chi_G = \chi_{G,2}$.

3 One Color

Our first goal is to prove that the 2-dimensional WL algorithm distinguishes vertex pairs that are separators in a graph from other pairs of vertices. We start with an analysis of the graphs in which all vertices are assigned the same color by the WL algorithm. In particular, this includes all constituent graphs of association schemes.

A main tool for the analysis are distance patterns of vertices. For a graph G and a vertex $v \in V(G)$, let $D(v) := \{\{\text{dist}(v, w) \mid w \in V(G)\}\}$. Note that for vertices $u, v \in V(G)$ it holds that $\chi_G(u, u) \neq \chi_G(v, v)$ whenever $D(u) \neq D(v)$ since the 2-dimensional WL algorithm detects distances between vertex pairs.“

► **Lemma 2.** *Let G be a graph and $uv \in E(G)$. Suppose that $D(u) = D(v)$. Then*

$$\begin{aligned} & \{\{\text{dist}(u, w) \mid w \in V(G) : \text{dist}(u, w) < \text{dist}(v, w)\}\} \\ &= \{\{\text{dist}(v, w) \mid w \in V(G) : \text{dist}(v, w) < \text{dist}(u, w)\}\}. \end{aligned}$$

Throughout the remainder of this section, if not explicitly stated otherwise, we make the following assumption.

► **Assumption 3.** G is a connected graph on n vertices with the following properties:

1. $\chi_G(u, u) = \chi_G(v, v)$ for all $u, v \in V(G)$, and
2. G has a 2-separator $\{w_1, w_2\}$.

In the rest of this section, we analyze the structure of G and ultimately prove that G must be a cycle. In particular, this completely characterizes constituent graphs of association schemes that are connected, but not 3-connected.

Note that Assumption 3 implies that G is regular, i.e., $\deg(u) = \deg(v)$ for all $u, v \in V(G)$.

► **Lemma 4.** G is 2-connected, i.e., G does not contain any cut vertex.

This is a consequence of Condition 1 in Assumption 3, since the 2-dimensional WL algorithm distinguishes cut vertices from other vertices (see [29, Corollary 7]). Note that the lemma implies that each of w_1 and w_2 has at least one neighbor in each of the connected components of $G - w_1w_2$.

► **Lemma 5.** *Let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| < \frac{n}{2}$ and let $v \in C$. Then there is no vertex $u \in N(v)$ such that $\text{dist}(u, w_1) < \text{dist}(v, w_1)$ and $\text{dist}(u, w_2) < \text{dist}(v, w_2)$.*

Proof. Suppose towards a contradiction that such a vertex $u \in N(v)$ exists. For all $w \in V(G)$, we have $|\text{dist}(v, w) - \text{dist}(u, w)| \leq 1$, since $uv \in E(G)$. Furthermore, it holds that $\sum_{w \in V(G)} (\text{dist}(v, w) - \text{dist}(u, w)) = 0$ because $D(u) = D(v)$ due to Condition 1 in Assumption 3. But $\text{dist}(v, w) > \text{dist}(u, w)$ for all $w \in V(G) \setminus C$, and $|V(G) \setminus C| > \frac{n}{2}$. This is a contradiction. ◀

► **Lemma 6.** *Let $d := \text{dist}(w_1, w_2)$ and let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| \leq \frac{n-2}{2}$. Then for all $v \in C \cup \{w_1, w_2\}$ and all $i \in \{1, 2\}$, it holds that $\text{dist}(v, w_i) \leq d$.*

Proof. By symmetry, it suffices to prove $\text{dist}(v, w_2) \leq d$. The statement is proved by induction on $\ell := \text{dist}(v, w_1)$. For $\ell = 0$, it holds that $v = w_1$ and $\text{dist}(w_1, w_2) = d$. So suppose the statement holds for all $u \in C \cup \{w_1, w_2\}$ with $\text{dist}(u, w_1) \leq \ell$. Obviously, the statement is true if $v = w_1$ or $v = w_2$. So pick $v \in C$ with $\text{dist}(v, w_1) = \ell + 1$. Let $u \in N(v)$ such that $\text{dist}(u, w_1) \leq \ell$. Then $\text{dist}(v, w_2) \leq \text{dist}(u, w_2) \leq d$ by Lemma 5 and the induction hypothesis. ◀

► **Lemma 7.** $w_1w_2 \notin E(G)$.

Proof. Suppose towards a contradiction that $w_1w_2 \in E(G)$. Let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| \leq \frac{n-2}{2}$. By Lemma 6, we conclude that $C \subseteq N(w_1) \cap N(w_2)$. Let $v \in C$. Since G is 2-connected, the vertex w_1 must have at least one neighbor in $V(G) \setminus C$, in addition to being adjacent to C and to w_2 . Thus, $\deg(w_1) \geq |C| + 2 > |C| - 1 + |\{w_1, w_2\}| \geq \deg(v)$, which contradicts G being a regular graph. ◀

► **Lemma 8.** *Suppose that $N(w_1) \cap N(w_2) \neq \emptyset$. Then G is a cycle.*

Proof. By Lemma 7, it holds that $w_1w_2 \notin E(G)$. Furthermore, by the assumption of the lemma, we have $\text{dist}(w_1, w_2) = 2$. Let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| \leq \frac{n-2}{2}$. Also let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$. For $i, j \geq 1$ let

$$C_{i,j} := \{v \in C \mid \text{dist}(v, w_1) = i \text{ and } \text{dist}(v, w_2) = j\}.$$

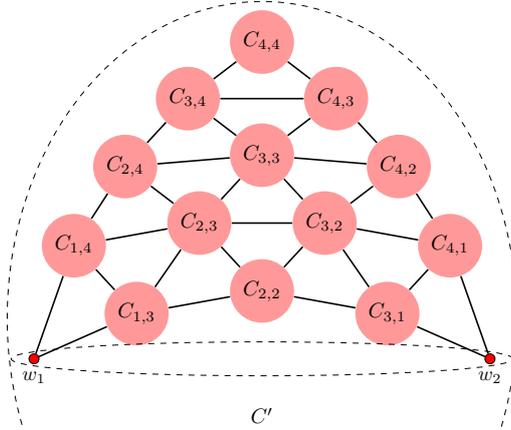
By Lemma 6, we conclude that $C_{i,j} = \emptyset$ unless $(i, j) \in \{(1, 1), (1, 2), (2, 1), (2, 2)\}$.

Suppose there exists $v \in C_{1,2}$. We have $D(v) = D(w_1)$ and, by Lemma 4, also $N(w_1) \cap C' \neq \emptyset$. Thus, there is some vertex $u' \neq w_1$ such that $\text{dist}(w_1, u') < \text{dist}(v, u')$ and therefore, by Lemma 2, there is also a vertex $u \neq v$ such that $\text{dist}(v, u) < \text{dist}(w_1, u)$. For every such vertex u , it holds that $\text{dist}(w_1, u) \leq 2$ and thus, $u \in N(v)$. Therefore, by Lemma 2, for every vertex $v' \neq w_1$ with $\text{dist}(w_1, v') < \text{dist}(v, v')$, it holds that $v' \in N(w_1)$. This implies that there is no $v' \in C'$ such that $\text{dist}(w_1, v') = 2$ since such a vertex would satisfy $3 = \text{dist}(v, v') > \text{dist}(w_1, v')$.

Because w_2 is not a cut vertex (cf. Lemma 4), from every $v' \in C'$, there is a path to w_1 that does not contain w_2 . However, this is only possible if there is no vertex $v' \in C'$ such that $\text{dist}(w_1, v') > 1$, in other words: $C' \subseteq N(w_1)$. Since G is regular and $|N(w_1) \setminus C'| \geq 1$, it follows that $\deg(v) \geq |C'| + 1$. But $N(v) \subseteq (C \cup \{w_1\}) \setminus \{v\}$, which implies $\deg(v) \leq |C|$. The combination of both inequalities yields $|C'| + 1 \leq |C|$, which implies $n = 2 + |C| + |C'| \leq 1 + 2|C| \leq n - 1$, a contradiction. So $C_{1,2} = \emptyset$ and by symmetry, it also holds that $C_{2,1} = \emptyset$. But then $C_{2,2} = \emptyset$ by Lemma 5.

So $C = C_{1,1}$, which means that $C \subseteq N(w_1) \cap N(w_2)$. In particular, $\deg(w_1) \geq |C| + 1$ since $|N(w_1) \cap C'| \geq 1$. Since G is regular, this implies that $\deg(v) \geq |C| + 1$ for every $v \in C$, which is only possible if $N[v] = C \cup \{w_1, w_2\}$. Because $C \neq \emptyset$, this means that there is a vertex $v \in V(G)$ such that $G[N[v]]$ contains only one non-edge. Now by Condition 1 in Assumption 3, this also has to hold for w_1 , and hence, since no vertex in $C \cap N(w_1)$ is adjacent to any vertex in $C' \cap N(w_1)$, it must hold that $\deg(w_1) = 2$. Therefore, by regularity, all vertices in G have degree 2 and thus, being connected, G is a cycle. ◀

► **Lemma 9.** G is a cycle.



■ **Figure 1** Visualization of the sets $C_{i,j}$ for $d = 4$ in the proof of Lemma 9. An arc between two sets indicates that there may be edges connecting vertices from the two sets.

Proof Sketch. We only present a sketch here. It suffices to prove the lemma for the case that G is a graph with a maximum edge set that satisfies Assumption 3. Indeed, if such a graph G is a cycle, then G has n edges, and the lemma trivially holds for every graph with less edges since every connected regular graph has at least n edges.

Let $d := \text{dist}(w_1, w_2)$. By Lemmas 7 and 8, we can assume that $d \geq 3$. Let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| \leq \frac{n-2}{2}$. Also let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$. For $i, j \geq 1$ let

$$C_{i,j} := \{v \in C \mid \text{dist}(v, w_1) = i \text{ and } \text{dist}(v, w_2) = j\}.$$

By Lemma 6, we conclude that $C_{i,j} = \emptyset$ unless $i, j \leq d$. Furthermore, by the definition of d , we have that $C_{i,j} = \emptyset$ unless $i + j \geq d$. This situation is also visualized in Figure 1.

Using similar ideas as in the proof of Lemma 8, the first step is to show that $C_{d,d} = C_{d-1,d} = C_{d,d-1} = \emptyset$. The main part of the proof is based on the following claim, which exploits the edge maximality of the graph G .

▷ **Claim 10.** Let $u, v \in V(G)$ such that $\text{dist}(u, v) < d$. Then there is a unique shortest path from u to v .

Indeed, if there were a pair of vertices $u, v \in V(G)$ of distance $\ell < d$ with two shortest paths, then u and v would be “on the same side” of the separator w_1w_2 . Hence, since the 2-dimensional WL algorithm is capable of detecting such pairs, adding all of them to the edge set of G would result in a graph with a larger number of edges that still fulfills Assumption 3. Using similar arguments, we also obtain the following related insight.

▷ **Claim 11.** Let $u, v \in V(G)$ such that $\ell := \text{dist}(u, v) < d$. Furthermore, suppose there is a walk $u = u_0, \dots, u_{\ell+1} = v$ of length $\ell + 1$ from u to v . Then there is some $i \in [\ell]$ such that $u_{i-1}u_{i+1} \in E(G)$.

These two claims drastically restrict the structure of the graph G and allow us to prove that G is a cycle. Intuitively speaking, the claims imply that, when looking towards the connected component $G[C]$ from any of the w_i , the graph has a tree-like shape, i.e., the initial segments of paths up to length $d - 1$ starting in w_i form a tree rooted in w_i . However, this

only works out if G is a cycle. More formally, for $k' \in \{d, \dots, 2d\}$, let $\mathcal{C}_{k'} := \bigcup_{i,j:i+j=k'} C_{i,j}$ and let $k \in \{d, \dots, 2d\}$ be the maximal number such that $\mathcal{C}_k \neq \emptyset$. Then $k \leq 2d - 2$ since we already know that $C_{d,d} = C_{d-1,d} = C_{d,d-1} = \emptyset$.

Consider a set $C_{i,k-i}$ for some i with $k - d + 1 \leq i \leq d - 1$ (i.e., none of the extremal sets). Then, for all $v \in C_{i,k-i}$, it holds that $\text{dist}(v, w_1) < d$ and $\text{dist}(v, w_2) < d$, which allows us to apply Claims 10 and 11 for shortest paths from v to w_1 and w_2 . With this, we show that if $C_{i,k-i} \neq \emptyset$, then there is a vertex of degree 2, which implies that G is a cycle, since it is regular. In the other case, $C_{j,k-j} = \emptyset$ for all j with $k - d + 1 \leq j \leq d - 1$. We then consider the extremal sets and derive a contradiction from there. ◀

Reformulating the previous lemma, we obtain the following theorem.

► **Theorem 12.** *Let G be a graph such that $\chi_G(u, u) = \chi_G(v, v)$ for all $u, v \in V(G)$. Then (exactly) one of the following holds:*

1. G is not connected, or
2. G is 3-connected, or
3. G is a cycle of length $\ell \geq 4$.

Note that the complete graphs on two and three vertices are 3-connected. The theorem also implies that a connected constituent graph of an association scheme is either 3-connected or a cycle (for other work on the connectivity of relations in association schemes, see e.g. [6, 7, 8, 13]). It thus provides a generalization of Kodalen's and Martin's result in [30], where they proved the theorem in case the graph stems from a symmetric association scheme.

4 Two Colors

Recall that our overall goal is to prove that the 2-dimensional WL algorithm assigns special colors to 2-separators in a graph. We will use Lemma 9 to prove this in case the tuples (u, u) and (v, v) of a 2-separator $\{u, v\}$ obtain the same color under the 2-dimensional WL algorithm. To treat the much more difficult case that u and v obtain distinct colors, we intend to generalize the results of the previous section to two vertex colors. Maybe somewhat surprisingly, we obtain a similar statement to Lemma 9. However, now we require the input graphs to be 2-connected (instead of only being connected). This is a necessary condition, since for example the star graphs $K_{1,n}$ for $n \geq 2$ are neither 3-connected nor cycles but still have only two vertex colors under the 2-dimensional WL algorithm.

The route to proving the statement is similar to the one described in Section 3. Still, two colors allowing for more complexity in the graph structure, the statements and lemmas become more involved and additional cases need to be considered.

► **Lemma 13.** *Let G be a graph and suppose there are $c_1 \neq c_2$ such that $\{\chi_G(v, v) \mid v \in V(G)\} = \{c_1, c_2\}$ for some $c_1 \neq c_2 \in \mathbb{N}$. Let $U := \{u \in V(G) \mid \chi_G(u, u) = c_1\}$ and $V := \{v \in V(G) \mid \chi_G(v, v) = c_2\}$. Also let U_1, \dots, U_k be the vertex sets of the connected components of $G[U]$ and let V_1, \dots, V_ℓ be the vertex sets of the connected components of $G[V]$. Let G' be the graph with $V(G') = \{U_1, \dots, U_k, V_1, \dots, V_\ell\}$ and $U_i V_j \in E(G')$ if and only if there are $u \in U_i, v \in V_j$ such that $uv \in E(G)$.*

Then $\chi_{G'}(U_i, U_i) = \chi_{G'}(U_j, U_j)$ for all $i, j \in [k]$ and $\chi_{G'}(V_i, V_i) = \chi_{G'}(V_j, V_j)$ for all $i, j \in [\ell]$.

► **Theorem 14.** *Let G be a 2-connected graph with the following properties:*

1. G has a 2-separator $w_1 w_2$, and
2. for every $v \in V(G)$, there is an $i \in \{1, 2\}$ such that $\chi_G(v, v) = \chi_G(w_i, w_i)$.

Then G is a cycle.

Proof Idea. By Lemma 9, we can assume without loss of generality that $\chi_G(w_1, w_1) \neq \chi_G(w_2, w_2)$. The statement is proved by induction on the graph size n . For $n \leq 4$, a simple case analysis among the possible graphs G yields the statement.

So let $n \geq 5$. Again, it suffices to prove the statement for the case that G is an n -vertex graph with a maximum edge set that satisfies the requirements of the theorem. Let $U := \{u \in V(G) \mid \chi_G(u, u) = \chi_G(w_1, w_1)\}$ and $V := \{v \in V(G) \mid \chi_G(v, v) = \chi_G(w_2, w_2)\}$. Let U_1, \dots, U_k be the vertex sets of the connected components of $G[U]$ and let V_1, \dots, V_ℓ be the vertex sets of the connected components of $G[V]$. Without loss of generality, assume that $w_1 \in U_1$ and $w_2 \in V_1$. Let C be the vertex set of a connected component of $G - w_1w_2$ such that $|C| \leq \frac{n-2}{2}$. Also let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$.

Consider the graph G' defined in the same way as in Lemma 13. First suppose that G' has fewer vertices than the original graph G . If the induction hypothesis is applicable to the graph G' , it follows that G' is a cycle and from there, it is not difficult to prove that G also must be a cycle. However, it may happen that G' is 3-connected in which case the induction hypothesis cannot be applied. But this can only happen in very specific cases, which we then treat separately.

If $|V(G')| = |V(G)|$, then there are no edges connecting vertices of the same color with respect to the 2-dimensional WL algorithm and thus, G is bipartite. In this case, we can proceed very similarly to the proof of Lemma 9. \blacktriangleleft

5 Detecting Decompositions with Weisfeiler and Leman

Let S be a set of colors. We say a path u_0, \dots, u_ℓ *avoids* S if $\chi_G(u_i, u_i) \notin S$ for every $i \in [\ell - 1]$. Note that we impose no restriction on the colors of the endpoints of the path. It is easy to see that, given two vertices $u, v \in V(G)$, the 2-dimensional WL algorithm is aware of whether there is a path from u to v that avoids S .

► **Lemma 15.** *Let G be a graph and let $X := \{\chi_G(v, v) \mid v \in V(G)\}$. Furthermore, let $S \subseteq X$ and define $G[[S]] = (V, E)$, where $V = \{v \in V(G) \mid \chi_G(v, v) \in S\}$ and*

$$E = \{uv \mid \text{there is a path from } u \text{ to } v \text{ in } G \text{ that avoids } S\}.$$

Then $\chi_{G[[S]]}(u, u) = \chi_{G[[S]]}(v, v)$ for all $u, v \in V$ with $\chi_G(u, u) = \chi_G(v, v)$.

► **Theorem 16.** *Let G and H be 2-connected graphs and let $w_1, w_2 \in V(G)$ such that w_1w_2 forms a 2-separator in G . Also let $v_1, v_2 \in V(H)$ and suppose $\chi_G(w_1, w_2) = \chi_H(v_1, v_2)$. Then v_1v_2 forms a 2-separator in H .*

Proof. Let $S := \{\chi_G(w_1, w_1), \chi_G(w_2, w_2)\}$ and let $G' := G[[S]]$. Clearly, the graph G' is connected. We argue that G' is 2-connected. Suppose towards a contradiction that there is a separating vertex w in G' . Let C and C' be the vertex sets of two connected components of $G' - w$ and let $v \in C$ and $v' \in C'$. We show that w separates v from v' in G . Towards a contradiction, suppose there is a path P from v to v' in G that does not pass w . Then there is a corresponding path P' in G' , which simply skips all inner vertices of P not contained in S . In particular, P' connects v and v' , but avoids w . This contradicts w being a cut vertex in G' . Hence, G' is 2-connected.

First suppose that $|V(G')| = 2$. Let $A := \{v \in V(H) \mid \chi_H(v, v) \in S\}$. Then $|A| = 2$ and thus $A = \{v_1, v_2\}$. Moreover, $H - A$ is disconnected, since the 2-dimensional WL algorithm detects that $G - w_1w_2$ is disconnected. Hence, v_1v_2 forms a 2-separator in H .

Now assume $|V(G')| \geq 3$ and suppose there is a vertex set C of a connected component of $G - w_1w_2$ such that $V(G') \subseteq C \cup \{w_1, w_2\}$. Let C' be the vertex set of a second connected component of $G - w_1w_2$ and let $v \in C'$. Then w_1 and w_2 are the only vertices with color in S that can be reached from v via a path that avoids S . Hence, using the expressive power of the 2-dimensional WL algorithm, it is not hard to see that there must also be some $u \in V(H)$ such that v_1 and v_2 are the only vertices with color in S that can be reached from u via a path that avoids S . Since $|V(G')| \geq 3$, there is some $u' \in V(H)$ such that $v_1 \neq u' \neq v_2$ and $\chi_H(u', u') \in S$ because in order not to be distinguished, unions of color classes with color in S must have the same cardinality in both graphs. But then v_1v_2 separates u from u' in H and thus, v_1v_2 forms a 2-separator in H .

In the other case, w_1w_2 forms a 2-separator in G' and hence, G' is a cycle by Lemma 15 and Theorem 14. Note that $|V(G')| \geq 4$ and $w_1w_2 \notin E(G')$. It follows that $H[[S]]$ is also a cycle, since otherwise, the 2-dimensional WL algorithm would distinguish the graphs. Also, $|V(H[[S]])| \geq 4$ and $v_1v_2 \notin E(H[[S]])$. So v_1v_2 forms a 2-separator in $H[[S]]$ and thus, it also forms a 2-separator in H . ◀

► **Corollary 17.** *Suppose $k \geq 2$. Let G and H be connected graphs. Suppose $\{w_1, \dots, w_k\} \subseteq V(G)$ is a k -separator in G . Let $\{v_1, \dots, v_k\} \subseteq V(H)$ and suppose $\chi_{G,k}(w_1, \dots, w_k) = \chi_{H,k}(v_1, \dots, v_k)$. Then $\{v_1, \dots, v_k\}$ forms a k -separator in H .*

Using the corollary, we can prove a strengthened version of Theorem 13 in [29]. Following [29], we say that the k -dimensional WL algorithm *determines orbits* in a graph class \mathcal{G} if for all arc-colored graphs (G, λ) , (G', λ') with $G, G' \in \mathcal{G}$, arc colorings λ, λ' and for all vertices $v \in V(G)$ and $v' \in V(G')$ there exists an isomorphism from (G, λ) to (G', λ') mapping v to v' if and only if $\chi_{G,k}(v, \dots, v) = \chi_{G',k}(v', \dots, v')$.

► **Theorem 18.** *Let \mathcal{G} be a minor-closed graph class and assume $k \geq 2$. Suppose the k -dimensional WL algorithm determines orbits on all arc-colored 3-connected graphs in \mathcal{G} . Then the k -dimensional WL algorithm distinguishes all non-isomorphic graphs in \mathcal{G} .*

Thus, since by [29], the WL dimension of the class of planar graphs is 2 or 3, the concrete value only depends on the dimension needed to determine orbits on arc-colored triconnected planar graphs.

For a graph G and $v_1, v_2, v_3 \in V(G)$ we define $s_G(v_1, v_2, v_3) := |C|$, where C is the vertex set of the connected component of $G - v_1v_2$ that contains v_3 (if $v_3 \in \{v_1, v_2\}$, then $s_G(v_1, v_2, v_3) := 0$).

► **Theorem 19.** *Let G and H be two 2-connected graphs. Also let $v_1, v_2, v_3 \in V(G)$ and $w_1, w_2, w_3 \in V(H)$ such that $\chi_G(v_i, v_j) = \chi_H(w_i, w_j)$ for all $i, j \in \{1, 2, 3\}$. Then $s_G(v_1, v_2, v_3) = s_H(w_1, w_2, w_3)$.*

The last theorem can also be formulated in terms of the expressive power of the 3-variable fragment \mathbf{C}^3 of first-order logic with counting quantifiers of the form $\exists^{\geq k} x \varphi(x)$. Indeed, it implies that for all $n, s \in \mathbb{N}$, there is a formula $\varphi_{n,s}(x_1, x_2, x_3) \in \mathbf{C}^3$ such that, for every 2-connected n -vertex graph G and $v_1, v_2, v_3 \in V(G)$, it holds that $G \models \varphi_{n,s}(v_1, v_2, v_3)$ if and only if $s_G(v_1, v_2, v_3) = s$ (for details about the connection between the WL algorithm and counting logics, see e.g. [9, 27]).

6 New Bounds for Graphs of Treewidth k

As an application of the results presented so far, we investigate the WL dimension of graphs of treewidth at most k . Up to this point, the best known upper bound on the WL dimension

of such graphs has been $k + 2$, i.e., the $(k + 2)$ -dimensional WL algorithm identifies every graph of treewidth at most k [18]. In this section, we present new upper and lower bounds.

6.1 Upper Bound

The basic idea for proving a new upper bound is to provide a winning strategy for Spoiler in the corresponding bijective pebble game and it works similarly to the proof that the $(k + 2)$ -dimensional WL algorithm identifies every graph of treewidth at most k [18]. The main difference is a much more careful implementation of the general strategy in order to get by with the desired number of pebbles. As a major ingredient, we exploit that separators can be detected using fewer pebbles.

For a $(k + 1)$ -tuple $(v_1, \dots, v_k, v_{k+1})$ of vertices of a graph G , we define $s_G(v_1, \dots, v_k, v_{k+1}) := |C|$ where C is the unique connected component of $G - \{v_1, \dots, v_k\}$ with $v_{k+1} \in C$.

► **Corollary 20.** *Suppose $k \geq 2$. Let G, H be two graphs and let $v_1, \dots, v_{k+1} \in V(G)$ and $w_1, \dots, w_{k+1} \in V(H)$ such that $s_G(v_1, \dots, v_k, v_{k+1}) \neq s_H(w_1, \dots, w_k, w_{k+1})$. Then Spoiler wins the game $\text{BP}_{k+1}(G, H)$ from the initial position $((v_1, \dots, v_{k+1}), (w_1, \dots, w_{k+1}))$.*

To build Spoiler's strategy along a given tree decomposition, we use the following characterization of treewidth. Let G be a graph of treewidth k . For a k -separator $S \subseteq V(G)$ and the vertex set C of a connected component of $G - S$, we define $G(S, C)$ to be the graph on vertex set $S \cup C$ obtained by inserting a clique between the vertices in S into $G[S \cup C]$.

► **Lemma 21** (Arnborg et al. [1]). *Suppose $G(S, C)$ has at least $k + 2$ vertices. Then $G(S, C)$ has treewidth at most k if and only if there exists $v \in C$ such that for every connected component A of $G[C \setminus \{v\}]$, there is a k -element separator $S_A \subseteq S \cup \{v\}$ such that*

1. *no vertex in A is adjacent to the unique element from $S \setminus S_A$, and*
2. *$G(S_A, V(A))$ has treewidth at most k .*

Suppose $G(S, C)$ has treewidth at most k . Let $D_G(S, C)$ denote the set of possible vertices $v \in C$ that satisfy Lemma 21.

► **Theorem 22.** *Suppose $k \geq 2$. Let G be a graph of treewidth at most k . Then the k -dimensional WL algorithm identifies G .*

Proof Idea. Given a graph G of treewidth at most k and a second non-isomorphic graph H , we show that Spoiler wins the game $\text{BP}_{k+1}(G, H)$. For simplicity of notation, we view tuples $\bar{a} = (a_1, \dots, a_k)$ also as sets $\{a_1, \dots, a_k\}$. Suppose the game is at a position $(\bar{a}, \bar{b}) \in (V(G))^k \times (V(H))^k$ such that $G(\bar{a}, C)$ has treewidth k for every connected component C of $G - \bar{a}$. Let $m := m(\bar{a}, \bar{b})$ be the smallest number such that $G - \bar{a}$ and $H - \bar{b}$ are not isomorphic when only considering connected components of size m . We prove by induction on m that Spoiler wins the game. The case $m = 1$ is easy, since there is still one pebble left. For the case $m > 1$, Spoiler finds a connected component C_G of $G - \bar{a}$ of size m that differs from the corresponding connected component C_H (specified by Duplicator's bijection) in the graph $H - \bar{b}$. By Corollary 20, we can assume that $|C_G| = |C_H|$. Then Spoiler places a pebble on a vertex in the set $D_G(\bar{a}, C_G)$. This splits C_G into smaller connected components. Then, it can be proved that one pebble can be removed so that afterwards, we find connected components of size at most $m - 1$ in which G and H differ (again using Corollary 20). ◀

6.2 Lower Bound

For the lower bound we use a construction introduced by Cai, Fürer, and Immerman [9]. For a graph G , we let $\text{CFI}(G)$ be the graph obtained by applying the standard Cai, Fürer, Immerman (CFI-) construction to G , and $\text{CFI}^\times(G)$ be the graph obtained by applying the CFI-construction with one pair of edges twisted (see [9]). We shall actually not require any further details on how the graphs $\text{CFI}(G)$ and $\text{CFI}^\times(G)$ look, since we exploit the following theorem.

► **Theorem 23** (Dawar and Richerby [10]). *Let G be a connected graph such that $\text{tw}(G) \geq k+1$ and $\deg(v) \geq 2$ for all $v \in V(G)$. Then $\text{CFI}(G) \simeq_k \text{CFI}^\times(G)$.*

The strategy to obtain a good lower bound is to find graphs G where we can show a sufficiently good upper bound on the treewidth of $\text{CFI}(G)$ and $\text{CFI}^\times(G)$. For $n \geq 2$, let $G_{n,n}$ be the $n \times n$ grid.

► **Lemma 24.** *For $n \geq 2$, it holds that $\text{tw}(\text{CFI}(G_{n,n})) \leq 2n+5$ and $\text{tw}(\text{CFI}^\times(G_{n,n})) \leq 2n+5$.*

► **Theorem 25.** *For every $k \geq 2$, there are non-isomorphic graphs G_k and H_k of treewidth at most $2k+7$ such that $G_k \simeq_k H_k$.*

Proof. Let $G_k := \text{CFI}(G_{k+1,k+1})$ and $H_k := \text{CFI}^\times(G_{k+1,k+1})$. Then the statement follows from Theorem 23 and Lemma 24. ◀

For a graph class \mathcal{C} , denote by $\text{dim}_{\text{WL}}(\mathcal{C})$ the *WL dimension* of \mathcal{C} , i.e., the minimum $k \in \mathbb{N} \cup \{\infty\}$ such that the k -dimensional WL algorithm identifies every graph $G \in \mathcal{C}$. As a corollary from Theorems 22 and 25, we obtain the following result.

► **Corollary 26.** *Let $k \geq 2$. Then $\lceil \frac{k}{2} \rceil - 3 \leq \text{dim}_{\text{WL}}(\mathcal{T}_k) \leq k$, where \mathcal{T}_k denotes the class of graphs of treewidth at most k .*

7 Conclusion

We have proved that for $k \geq 2$, the k -dimensional WL algorithm implicitly computes the decomposition of its input graph into its triconnected components. As a by-product, we found that every connected constituent graph of an association scheme is either a cycle or 3-connected.

We have applied this insight to improve on the upper bound on the WL dimension of graphs of bounded treewidth and have also provided a lower bound that is asymptotically only a factor of 2 away from the upper bound.

A natural use case of our results may be determining the WL dimension of certain graph classes that satisfy the requirements of Theorem 18. We conjecture that the 2-dimensional WL algorithm identifies every planar graph. Indeed, using the results of this paper, it essentially suffices to show this for triconnected planar graphs.

References

- 1 Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of Finding Embeddings in a K-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, April 1987. doi:10.1137/0608024.
- 2 Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. On Weisfeiler-Leman Invariance: Subgraph Counts and Related Graph Properties. *CoRR*, abs/1811.04801, 2018. arXiv:1811.04801.

- 3 Albert Atserias and Elitza N. Maneva. Sherali-Adams Relaxations and Indistinguishability in Counting Logics. *SIAM J. Comput.*, 42(1):112–137, 2013. doi:10.1137/120867834.
- 4 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016. doi:10.1145/2897518.2897542.
- 5 László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, and John Wilmes. Faster Canonical Forms for Strongly Regular Graphs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 157–166. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.25.
- 6 Andries E. Brouwer. Spectrum and connectivity of graphs. *CWI Quarterly*, 9(1-2):37–40, 1996.
- 7 Andries E. Brouwer and Jack H. Koolen. The vertex-connectivity of a distance-regular graph. *Eur. J. Comb.*, 30(3):668–673, 2009. doi:10.1016/j.ejc.2008.07.006.
- 8 Andries E. Brouwer and Dale M. Mesner. The Connectivity of Strongly Regular Graphs. *Eur. J. Comb.*, 6(3):215–216, 1985. doi:10.1016/S0195-6698(85)80030-5.
- 9 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. doi:10.1007/BF01305232.
- 10 Anuj Dawar and David Richerby. The Power of Counting Logics on Restricted Classes of Finite Structures. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 84–98. Springer, 2007. doi:10.1007/978-3-540-74915-8_10.
- 11 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Leman. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 40:1–40:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.40.
- 12 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 13 Sergei Evdokimov and Ilia Ponomarenko. On the vertex connectivity of a relation in an association scheme. *Journal of Mathematical Sciences*, 134(5):2354–2357, May 2006. doi:10.1007/s10958-006-0112-z.
- 14 Martin Fürer. On the Combinatorial Power of the Weisfeiler-Lehman Algorithm. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, volume 10236 of *Lecture Notes in Computer Science*, pages 260–271, 2017. doi:10.1007/978-3-319-57586-5_22.
- 15 Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *J. ACM*, 59(5):27:1–27:64, 2012. doi:10.1145/2371656.2371662.
- 16 Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*. Lecture Notes in Logic. Cambridge University Press, 2017.
- 17 Martin Grohe and Sandra Kiefer. A Linear Upper Bound on the Weisfeiler-Leman Dimension of Graphs of Bounded Genus. *CoRR*, abs/1904.07216, 2019. arXiv:1904.07216.
- 18 Martin Grohe and Julian Mariño. Definability and Descriptive Complexity on Databases of Bounded Tree-Width. In Catriel Beeri and Peter Buneman, editors, *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings.*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer, 1999. doi:10.1007/3-540-49257-7_6.
- 19 Martin Grohe and Daniel Neuen. Canonisation and Definability for Graphs of Bounded Rank Width. *CoRR*, abs/1901.10330, 2019. arXiv:1901.10330.
- 20 Martin Grohe and Martin Otto. Pebble Games and linear equations. *J. Symb. Log.*, 80(3):797–844, 2015. doi:10.1017/jsl.2015.28.

- 21 Lauri Hella. Logical Hierarchies in PTIME. *Inf. Comput.*, 129(1):1–19, 1996. doi:10.1006/inco.1996.0070.
- 22 John E. Hopcroft and Robert Endre Tarjan. A V^2 Algorithm for Determining Isomorphism of Planar Graphs. *Inf. Process. Lett.*, 1(1):32–34, 1971. doi:10.1016/0020-0190(71)90019-6.
- 23 John E. Hopcroft and Robert Endre Tarjan. Isomorphism of Planar Graphs. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 131–152. Plenum Press, New York, 1972.
- 24 John E. Hopcroft and Robert Endre Tarjan. A $V \log V$ Algorithm for Isomorphism of Triconnected Planar Graphs. *J. Comput. Syst. Sci.*, 7(3):323–331, 1973. doi:10.1016/S0022-0000(73)80013-3.
- 25 John E. Hopcroft and Robert Endre Tarjan. Dividing a Graph into Triconnected Components. *SIAM J. Comput.*, 2(3):135–158, 1973. doi:10.1137/0202012.
- 26 John E. Hopcroft and J. K. Wong. Linear Time Algorithm for Isomorphism of Planar Graphs (Preliminary Report). In Robert L. Constable, Robert W. Ritchie, Jack W. Carlyle, and Michael A. Harrison, editors, *Proceedings of the 6th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1974, Seattle, Washington, USA*, pages 172–184. ACM, 1974. doi:10.1145/800119.803896.
- 27 Neil Immerman and Eric Lander. Describing Graphs: A First-Order Approach to Graph Canonization. In *Complexity Theory Retrospective*, pages 59–81. Springer New York, 1990. doi:10.1007/978-1-4612-4478-3_5.
- 28 Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. *CoRR*, abs/1908.05268, 2019. arXiv:1908.05268.
- 29 Sandra Kiefer, Iliia Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005107.
- 30 Brian G. Kodalen and William J. Martin. On the Connectivity of Graphs in Association Schemes. *Electr. J. Comb.*, 24(4):P4.39, 2017. URL: <http://www.combinatorics.org/ojs/index.php/eljc/article/view/v24i4p39>.
- 31 Brendan D. McKay. Practical graph isomorphism. *Congr. Numer.*, 30:45–87, 1981.
- 32 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.
- 33 Christopher Morris and Petra Mutzel. Towards a practical k-dimensional Weisfeiler-Leman algorithm. *CoRR*, abs/1904.01543, 2019. arXiv:1904.01543.
- 34 Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks. *CoRR*, abs/1810.02244, 2018. arXiv:1810.02244.
- 35 Daniel Neuen. Graph Isomorphism for Unit Square Graphs. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, volume 57 of *LIPICs*, pages 70:1–70:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.ESA.2016.70.
- 36 William T. Tutte. *Graph theory*. Encyclopedia of mathematics and its applications. Addison-Wesley Pub. Co., Advanced Book Program, 1984.
- 37 Boris 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.