Computational complexity of the Weisfeiler-Leman dimension

Moritz Lichter **□ 0**

RWTH Aachen University

Simon Raßmann

□

TU Darmstadt

Pascal Schweitzer

□

TU Darmstadt

Abstract

The Weisfeiler-Leman dimension of a graph G is the least number k such that the k-dimensional Weisfeiler-Leman algorithm distinguishes G from every other non-isomorphic graph, or equivalently, the least k such that G is definable in (k+1)-variable logic with counting. The dimension is a standard measure of the descriptive or structural complexity of a graph and recently finds various applications in particular in the context of machine learning. This paper studies the complexity of computing the Weisfeiler-Leman dimension. We observe that deciding whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, even if G is restricted to have 4-bounded color classes. Therefore, we study parameterized versions of the problem. For each fixed $k \geq 2$, we give a polynomial-time algorithm that decides whether the Weisfeiler-Leman dimension of a given graph with 5-bounded color classes is at most k. Moreover, we show that for these bounds on the color classes, this is optimal because the problem is P-hard under logspace-uniform AC_0 -reductions. Furthermore, for each larger bound c on the color classes and each fixed $k \geq 2$, we provide a polynomial-time decision algorithm for the abelian case, that is, for structures of which each color class has an abelian automorphism group.

While the graph classes we consider may seem quite restrictive, graphs with 4-bounded abelian colors include CFI-graphs and multipedes, which form the basis of almost all known hard instances and lower bounds related to the Weisfeiler-Leman algorithm.

2012 ACM Subject Classification Mathematics of computing \rightarrow Graph algorithms; Theory of computation \rightarrow Problems, reductions and completeness; Theory of computation \rightarrow Complexity theory and logic

Keywords and phrases Weisfeiler-Leman algorithm, dimension, complexity, coherent configurations

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (EngageS: grant agreement No. 820148).

Moritz Lichter: The research of this author has received further funding by the European Union (ERC, SymSim, 101054974).

1 Introduction

The Weisfeiler-Leman algorithm is a simple combinatorial procedure studied in the context of the graph isomorphism problem. For every $k \geq 1$, the algorithm has a k-dimensional variant, k-WL for short, that colors k-tuples of vertices according to how they structurally sit inside the whole graph: if two tuples get different colors, they cannot be mapped onto each other by an automorphism of the graph (while the converse is not always true). The 1-dimensional algorithm, which is also known as color refinement, starts by coloring each vertex according to its degree, and then repeatedly refines this coloring by including into each vertex color the multisets of colors of its neighbors. The k-dimensional variant generalizes this idea and colors k-tuples of vertices instead of single vertices [42, 11].

2 Computational complexity of the Weisfeiler-Leman dimension

The Weisfeiler-Leman algorithm plays an important role in both theoretic and practical approaches to the graph isomorphism problem, but is also related to a plethora of seemingly unrelated areas: to finite model theory and descriptive complexity via the correspondence of k-WL to (k+1)-variable first-order logic with counting [11, 26], to machine learning via a correspondence to the expressive power of (higher-dimensional) graph neural networks [35], to the Sherali-Adams hierarchy in combinatorial optimization [3, 24], and to homomorphism counts from treewidth-k graphs [14].

On the side of practical graph isomorphism, the color refinement procedure is a basic building block of the so-called *individualization-refinement framework*, which is the basis of almost every modern practical solver for the graph isomorphism problem [34, 27, 28, 1]. On the side of theoretical graph isomorphism, Babai's quasipolynomial-time algorithm for the graph isomorphism problem [4] uses a combination of group-theoretic techniques and a logarithmic-dimensional Weisfeiler-Leman algorithm.

The Weisfeiler-Leman algorithm is a powerful algorithm for distinguishing non-isomorphic graphs on its own. For every k, k-WL can be used as an incomplete polynomial-time isomorphism test: if the multiset of colors of k-tuples of two graphs G and H differ, then G and H cannot be isomorphic. In this case, k-WL distinguishes G and H, otherwise G and H are k-WL-equivalent. For a given graph G, we say that the k-dimensional Weisfeiler-Leman algorithm k-WL identifies G if it distinguishes G from every non-isomorphic graph. The smallest such k is known as the Weisfeiler-Leman dimension of G [20].

It is known that almost all graphs have Weisfeiler-Leman dimension 1 [5]. However, color refinement fails spectacularly on regular graphs, where it always returns the monochromatic coloring. For these, it is known that 2-WL identifies almost all regular graphs [10, 31]. In contrast to these positive results, for every k there is some graph G (even of order linear in k, of maximum degree 3, and with 4-bounded abelian color classes, i.e., such that no more than 4 vertices can share the same vertex-color and every color class induces a graph with abelian automorphism group) that is not identified by k-WL [11]. These so-called CFI-graphs have high Weisfeiler-Leman dimension and are thus hard instances for combinatorial approaches to the graph isomorphism problem.

The situation changes for restricted classes of graphs. If the Weisfeiler-Leman dimension over some class of graphs is bounded by k, then the k-dimensional Weisfeiler-Leman algorithm correctly decides isomorphism over this class. And since k-WL can be implemented in polynomial time $O(n^{k+1}\log n)$ [26], this puts graph isomorphism over such classes into polynomial time. Examples of graph classes with bounded Weisfeiler-Leman dimension include graphs of bounded tree-width [22], graphs of bounded rank-width [23], graphs with 3-bounded color classes. [26], planar graphs [29], and more generally every non-trivial minor-closed graph class [20].

In this paper, we study the computational complexity of computing the Weisfeiler-Leman dimension. We call the problem of deciding whether the Weisfeiler-Leman dimension of a given graph is at most k the k-WL-identification problem. For upper complexity bounds, non-identification of a graph G can be witnessed by providing a graph H that is not distinguished from G by k-WL but is also not isomorphic to G. As the latter can be checked in co-NP, this places the identification problem into the class Π_2^P of the polynomial hierarchy. If the graph isomorphism problem is solvable in polynomial time, this complexity bound collapses to co-NP. However, there is no apparent reason why the identification problem should not be polynomial-time decidable.

On the side of lower complexity bounds, the 1-WL-identification problem is complete for polynomial time under uniform reductions in the circuit complexity class AC_0 [30, 2].

Hardness of the 1-WL-identification problem does, however, not easily imply any hardness results for the k-WL-identification problem for higher values of k. Indeed, no hardness results are known for $k \geq 2$. The 2-WL-identification problem in particular includes the problem of deciding whether a given strongly regular graph is determined up to isomorphism by its parameters, which is a baffling problem from classic combinatorics far beyond our current knowledge. To understand the difficulties of the k-WL-identification problem better, we can again consider classes of graphs. On every class of graphs with bounded color classes, graph isomorphism is solvable in polynomial time [6, 16], which puts the identification problem over this class into co-NP for every $k \geq 2$. Graphs with 3-bounded color classes are identified by 2-WL [26], which makes their identification problem trivial. As shown by the CFI-graphs [11], this is no longer true for graphs with 4-bounded color classes. Nevertheless, as shown by Fuhlbrück, Köbler, and Verbitsky, identification of graphs with 5-bounded color classes by 2-WL is efficiently decidable [15]. For higher dimensions or bounds on the color classes essentially nothing is known.

Contribution. We extend the results of [15] from 2-WL to k-WL and give a polynomial-time algorithm deciding whether a graph with 5-bounded color classes is identified by k-WL:

▶ **Theorem 1.** For every k, there is an algorithm that decides the k-WL-identification problem for vertex- and edge-colored, directed graphs with 5-bounded color classes in time $O_k(n^{O(k)})$. If such a graph G is not identified by k-WL, the algorithm provides a witness for this, i.e., a graph H that is not isomorphic to G and not distinguished from G by k-WL.

Via the correspondence of k-WL to (k+1)-variable counting logic, Theorem 1 implies that definability of graphs with 5-bounded color classes in this logic is decidable in polynomial time. While the restriction to 5-bounded color classes may seem stark, almost all known hardness results and lower bounds for the Weisfeiler-Leman algorithm remain true for graphs with bounded color classes and in most cases even 4-bounded color classes suffice [19, 13, 38, 37, 36].

Towards generalizing Theorem 1 to arbitrary relational structures and larger color classes, we consider structures with abelian color classes, i.e., structures of which each color class induces a structure with an abelian automorphism group. Such structures were previously considered in the context of descriptive complexity theory [43], and include both CFI-graphs [11] and multipedes [37, 36] over ordered base graphs, which form the basis of all known constructions of graphs with high Weisfeiler-Leman dimension. For many case in descriptive complexity theory, restricting to 4-bounded abelian color classes is sufficient, but in some cases larger (but still abelian) color classes are required [25, 18, 32, 33]. For such structures, we obtain a polynomial-time algorithm as before:

▶ Theorem 2. For every $k \in \mathbb{N}$ and $c, r \leq k$, there is an algorithm that decides the k-WL-identification problem for r-ary relational structures with c-bounded abelian color classes in time $O_k(n^{O(k)})$. If such a structure $\mathfrak A$ is not identified by k-WL, the algorithm provides a witness for this, i.e., a second structure $\mathfrak B$ that is not isomorphic to $\mathfrak A$ and not distinguished from $\mathfrak A$ by k-WL.

On the side of hardness results, we first prove that when the dimension k is part of the input, the identification problem is NP-hard. Note that a similar result was recently independently observed by Seppelt [40].

▶ **Theorem 3.** The problem of deciding, given a graph G and a natural number k, whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, both over uncolored simple graphs, and over simple graphs with 4-bounded abelian color classes.

4 Computational complexity of the Weisfeiler-Leman dimension

Furthermore, we extend the P-hardness results for 1-WL [2] to arbitrary k and prove that, when k is fixed, the k-WL-identification problem is hard for polynomial time:

▶ Theorem 4. For every $k \ge 1$, the k-WL-identification problem is P-hard under uniform AC_0 -reductions over both uncolored simple graphs, and simple graphs with 4-bounded abelian color classes.

Techniques. To prove Theorem 1, we exploit the close connection between the coloring computed by k-WL and k-ary coherent configurations. These structures come with two notions of isomorphisms, algebraic ones and combinatorial ones. Similarly to [15], we reduce the k-WL-identification problem to the separability problem for k-ary coherent configurations, that is, to decide whether algebraic and combinatorial isomorphisms for a given k-ary coherent configuration coincide. We make two crucial observations: First, we show that the k-ary coherent configurations obtained from graphs are fully determined by their underlying 2-ary configurations. We call such configurations 2-induced. Second, we reduce the separability problem for arbitrary k-ary coherent configurations to that of k-ary coherent configurations where no interspace contains a disjoint union of stars. Combining both observations, we show that two 2-induced, star-free k-ary coherent configurations obtained from k-WL-equivalent graphs must be isomorphic. Given such a k-ary coherent configuration obtained from a graph, it thus suffices to decide whether there is another non-isomorphic graph yielding the same configuration. Finally, we solve this problem by encoding it into the graph isomorphism problem for structures with bounded color classes, which is polynomial-time solvable [6, 16].

The main obstacle to generalize Theorem 1 to larger color classes or relational structures of higher arity is the existence of k-WL-equivalent structures that yield non-isomorphic star-free k-ary coherent configurations, which greatly increases the space of possibly equivalent bot non-isomorphic structures.

To make up for this, we consider structures with abelian color classes. Using both the bijective pebble game [25] and ideas from the theory of coherent configurations, we provide structural insights for the class of k-ary coherent configurations with abelian fibers which allows us to finally prove that in this case, it does suffice to consider other relational structures yielding the same k-ary coherent configuration.

NP-hardness in Theorem 3 is proved by combining the known relationship between the Weisfeiler-Leman dimension of CFI-graphs [11] and the tree-width of the underlying base graphs with the recent result that computing the tree-width of cubic graphs is NP-hard [9]. With the same techniques, we can also prove that deciding k-WL-equivalence of graphs is co-NP-hard when the dimension k is considered part of the input.

For the P-hardness result of the k-WL-identification problem in Theorem 4, we adapt a construction by Grohe [19] to encode monotone boolean circuits into graphs using different types of gadgets. This simultaneously reduces the monotone circuit value problem, which is known to be hard for polynomial time, to the k-WL-equivalence and k-WL-identification problem. The main difficulty was showing identification of Grohe's gadgets, specifically his so-called *one-way switches*. We give an alternative construction of these one-way switches based on the CFI-construction [11]. This construction simplifies proofs and more importantly yields graphs with 4-bounded abelian color classes for every k. This shows hardness for the k-WL-equivalence and k-WL-identification problems even for graphs with 4-bounded abelian color classes.

2 The Weifeiler-Leman algorithm and coherent configurations

Preliminaries.

For $n \in \mathbb{N}$, we set $[n] := \{1, \ldots, n\}$. For a set A, the set of all k-element subsets of A is denoted by $\binom{A}{k}$. For two runtime-bounding functions f and g with parameters including κ , we write $f \in O_{\kappa}(g)$ if f/g is bounded by a function of κ . A simple graph is a pair G = (V(G), E(G)) of a set V(G) of vertices and a set $E(G) \subseteq \binom{V(G)}{2}$ of undirected edges. For a directed graph, we allow $E(G) \subseteq V(G)^2 \setminus \{(v,v) : v \in V(G)\}$. For either graph type, we write uv for the edge $\{u,v\}$ or (u,v) respectively. For a simple or directed graph G, a vertex-coloring of G is a map $\chi \colon V(G) \to C$ for some finite, ordered set C of colors. Similarly, an edge-coloring is a map $\eta \colon E(G) \to C$. A (vertex-)color class is a set $\chi^{-1}(c)$ for some vertex color $c \in C$. If all color classes have order at most q, we say that the colored graph (G,χ) has q-bounded color classes.

Relational structures are a higher-arity analogue of graphs. Formally, a k-ary relational structure \mathfrak{A} is a tuple $(V(\mathfrak{A}), R_1, \ldots, R_\ell)$ of vertices $V(\mathfrak{A})$ and relations $R_i \subseteq V(\mathfrak{A})^{r_i}$ with $r_i \leq k$. The number r_i is the arity of the relation R_i . We again allow relational structures to come with a vertex-coloring and define q-bounded color classes as before.

An isomorphism between graphs G and H is a bijection $\varphi \colon V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $\varphi(u)\varphi(v) \in E(H)$. In this case G and H are isomorphic and we write $G \cong H$. An isomorphism between edge- or vertex-colored graphs must also preserve the vertex- and edge-colors. Similarly, an isomorphism between (vertex-colored) relational structures is a (color-preserving) bijection between the vertex sets that preserves all relations and their complements. An automorphism is an isomorphism from a structure to itself. The set of automorphisms of a structure forms a group, and we say that a graph or relational structure $\mathfrak A$ has abelian color classes if for every color class C, the induced substructure $\mathfrak A[C]$ has an abelian automorphism group.

The Weisfeiler-Leman algorithm.

For every $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm (k-WL) computes an isomorphism-invariant coloring of k-tuples of vertices of a given graph G via an iterative refinement process. Initially, the algorithm colors each k-tuple according to its isomorphism type, i.e., $\mathbf{x} = (x_1, \ldots, x_k), \mathbf{y} = (y_1, \ldots, y_k) \in V(G)^k$ get the same color if and only if mapping $x_i \mapsto y_i$ for every $i \in [k]$ is an isomorphism of the induced subgraphs $G[\{x_1, \ldots, x_k\}]$ and $G[\{y_1, \ldots, y_k\}]$. In each iteration, this coloring is refined as follows: if $\chi_r^G : V(G)^k \to C_i$ is the coloring obtained after i refinement rounds, the coloring $\chi_{r+1}^G : V(G)^k \to C_{i+1}$ is defined as $\chi_{r+1}^G(\mathbf{x}) := (\chi_r^G(\mathbf{x}), M_{\mathbf{x}}^i)$, where

$$M_{\mathbf{x}}^{r} = \left\{ \left(\chi_{r}^{G} \left(\mathbf{x} \frac{y}{1} \right), \dots, \chi_{r}^{G} \left(\mathbf{x} \frac{y}{k} \right) \right) \colon y \in V(G) \right\}$$

and $\mathbf{x}_{i}^{\underline{y}}$ denotes the tuple obtained from \mathbf{x} by replacing the *i*-th entry by y. If χ_{r+1}^{G} does not induce a finer color partition on $V(G)^{k}$ than χ_{r}^{G} , the algorithm terminates and returns the stable coloring $\chi_{\infty}^{G} \coloneqq \chi_{r}^{G}$. This must happen before the n^{k} -th refinement round.

We say that k-WL distinguishes two k-tuples $\mathbf{x}, \mathbf{y} \in V(G)^k$ if $\chi_{\infty}^G(\mathbf{x}) \neq \chi_{\infty}^G(\mathbf{y})$ and that k-WL distinguishes two ℓ -tuples $\mathbf{x}, \mathbf{y} \in V(G)^{\ell}$ for $\ell < k$ if k-WL distinguishes the two k-tuples we get by repeating the last entries of \mathbf{x} respectively \mathbf{y} . Finally, k-WL distinguishes two graphs G and H if there is a color c such that

$$\left|\left\{\mathbf{x} \in V(G): \chi_{\infty}^{G}(\mathbf{x}) = c\right\}\right| \neq \left|\left\{\mathbf{x} \in V(H): \chi_{\infty}^{H}(\mathbf{x}) = c\right\}\right|.$$

Otherwise, G and H are k-WL-equivalent and we write $G \equiv_{k\text{-WL}} H$. A graph G is identified by k-WL if k-WL distinguishes G from every other non-isomorphic graph. Every n-vertex graph is identified by n-WL, and the least number k such that k-WL identifies G is called the Weisfeiler-Leman dimension of G, denoted by WL-dim(G).

If c_1, \ldots, c_k are colors assigned by the stable coloring χ_r^G , then the multiplicity of the tuple (c_1, \ldots, c_k) in the multiset $M_{\mathbf{x}}^r$ is given by

$$p^{\mathbf{x}}_{c_1,...,c_k} \coloneqq \left| \left\{ y \in V(G) \colon \chi^G_r\left(\mathbf{x}\frac{y}{i}\right) = c_i \text{ for all } i \in [k] \right\} \right|.$$

These numbers, which are called *intersection numbers*, fully determine the multiset $M_{\mathbf{x}}^{r}$. In the theoretical study of the Weisfeiler-Leman algorithm, these numbers are often quite handy to work with compared to the multiset view from the definition. For example, the statement that a coloring χ is stable under k-WL-refinement can be expressed as the intersection numbers $p_{c_1,\ldots,c_k}^{\mathbf{x}}$ being determined by $\chi(\mathbf{x})$ and not depending on \mathbf{x} itself. This leads to the notion of coherent configurations.

As every coloring of k-tuples also induces a coloring of ℓ -tuples for $\ell \leq k$ by repeating the last entry, k-WL is at least as powerful in distinguishing graphs as ℓ -WL and this hierarchy is actually strict [11]. Completely analogously, k-WL can be applied to relational structures.

Coherent configurations.

For an introduction to (2-ary) coherent configurations we refer to [12] and for their connection to the Weisfeiler-Leman algorithm we refer to [15]. For $k \geq 2$, a k-ary rainbow is a pair (V, \mathcal{R}) of a finite set of vertices V and a partition \mathcal{R} of V^k , whose elements are called basis relations, that satisfies the following two conditions:

- (R1) For every basis relations $R \in \mathcal{R}$, all tuples $\mathbf{x}, \mathbf{y} \in R$ have the same equality type, i.e., $x_i = x_j$ if and only if $y_i = y_j$. We also call this the equality type of the relation R.
- (R2) \mathcal{R} is closed under permuting indices: For all basis relations $R \in \mathcal{R}$ and permutations σ of [k], the set $R^{\sigma} := \{(x_{\sigma(1)}, \dots, x_{\sigma(k)}) : (x_1, \dots, x_k) \in R\}$ is a basis relation.

Because the vertex set V is determined by the partition \mathcal{R} , we also write \mathcal{R} to denote the rainbow (V, \mathcal{R}) and in this case write $V(\mathcal{R})$ for its vertex set V.

A k-ary coherent configuration is a k-ary rainbow \mathcal{C} that is stable under k-WL-refinement. More formally, this means that

(C) for all basis relations $R, R_1, \ldots, R_k \in \mathcal{C}$, the intersection number

$$p(R; R_1, \dots, R_k) := \left| \left\{ y \in V(\mathcal{C}) \colon \mathbf{x} \frac{y}{i} \in R_i \text{ for all } i \in [k] \right\} \right|$$

is the same for all choices of $\mathbf{x} \in R$ and is thus well-defined.

For $\ell \leq k$, the partition of k-vertex tuples of an ℓ -ary relational structure according to their isomorphism type always yields a k-ary rainbow. The connection of k-WL and k-ary coherent configurations is that the partition of k-vertex tuples of a graph according to their k-WL-colors always forms a k-ary coherent configuration.

Induced configurations. If \mathcal{R} is an ℓ -ary rainbow for $\ell \leq k$, we can interpret \mathcal{R} as the k-ary rainbow $\mathcal{R}^{|k|}$ by partitioning k-tuples according to the basis relations of the ℓ -subtuples they contain. Formally, let $\sim_{\mathcal{R}}$ be the equivalence relation on $V(\mathcal{R})^{\ell}$ whose equivalence classes are the basis relations of \mathcal{R} . We define the equivalence relation $\sim_{\mathcal{R}}^k$ on $V(\mathcal{R})^k$ by writing $\mathbf{x} \sim_k \mathbf{y}$ if and only if for all $I \in {[k] \choose \ell}$ we have $\mathbf{x}|_I \sim_{\mathcal{R}} \mathbf{y}|_I$, where $\mathbf{x}|_I$ is the subtuple of \mathbf{x} for which all indices not in I are deleted. The basis relations of $\mathcal{R}|^k$ are the equivalence classes of $\sim_{\mathcal{R}}^k$. For every k-ary rainbow \mathcal{R} , there is a unique coarsest k-ary coherent configuration $\operatorname{WL}_k(\mathcal{R})$ that is at least as fine as \mathcal{R} and is called the k-ary coherent closure of \mathcal{R} . For an $\ell \leq k$ and an ℓ -ary rainbow \mathcal{R} , we also write $\operatorname{WL}_k(\mathcal{R})$ for $\operatorname{WL}_k(\mathcal{R}|^k)$. Similarly, for an ℓ -ary relational structure \mathfrak{A} , we write $\operatorname{WL}_k(\mathfrak{A})$ for the partition of $V(\mathfrak{A})^k$ into k-WL-color classes.

Every k-ary coherent configuration \mathcal{C} induces the ℓ -ary coherent configuration $\mathcal{C}|_{\ell}$ for every $\ell \leq k$ by considering the partition of tuples of the form $(x_1, \ldots, x_\ell, \ldots, x_\ell) \in V(\mathcal{C})^k$. This ℓ -ary coherent configuration is called the ℓ -skeleton of \mathcal{C} . For every basis relation $R \in \mathcal{C}$ and every subset $I \in {[k] \choose \ell}$ of the indices, the set $R_I := \{\mathbf{x}|_I \colon \mathbf{x} \in R\}$ is a basis relation of $\mathcal{C}|_{\ell}$ and called the I-face of R. For basis relations $R \in \mathcal{C}|_{\ell}$ and $T \in \mathcal{C}$, we get well-defined extension numbers $p(R;T) := |\{\mathbf{y} \in V(\mathcal{C})^{k-\ell} \colon \mathbf{x}\mathbf{y} \in T\}|$ for some (and every) $\mathbf{x} \in R$.

For $\ell = 1$, the 1-skeleton yields a partition of $V(\mathcal{C})$, whose partition classes are called *fibers*. We denote the set of fibers by $F(\mathcal{C})$. \mathcal{C} has *c-bounded fibers* if all fibers of \mathcal{C} have order at most c. Between two fibers X and Y, the induced configuration $\mathcal{C}|_2$ further induces a partition $\mathcal{C}|_2[X,Y]$ of $X \times Y$, called an *interspace*.

We call a k-ary coherent configuration \mathcal{C} ℓ -induced if it is the coherent closure of its ℓ -skeleton, i.e., if $\mathcal{C} = \operatorname{WL}_k(\mathcal{C}|_{\ell})$. This is equivalent to \mathcal{C} being the coherent closure of some ℓ -ary rainbow. In particular, the k-ary coherent closure of a (directed, colored) graph is 2-induced and, more generally, the k-ary coherent closure of an ℓ -ary relational structure is ℓ -induced for every $k \geq \ell$.

For a k-ary rainbow $\mathcal{R} = (V, \{R_1, \dots, R_\ell\})$, the vertex-colored k-ary relational structure $(V, R_1, \dots, R_\ell, \chi)$ where χ maps every vertex to its fiber is a *colored variant* of \mathcal{R} . Note that this requires choosing an ordering of the basis relations; colored variants are thus not unique.

Algebraic and combinatorial isomorphisms. There are two notions of isomorphism for two k-ary coherent configurations \mathcal{C} and \mathcal{D} . First, a combinatorial isomorphism is a bijection $\varphi \colon V(\mathcal{C}) \to V(\mathcal{D})$ that preserves the partition into basis relations, i.e., for every basis relation $R \in \mathcal{C}$, the mapped set $R^{\varphi} := \{(\varphi(x_1), \dots, \varphi(x_k)) \colon (x_1, \dots, x_k) \in R\}$ is a basis relation of \mathcal{D} . Combinatorial isomorphisms are thus isomorphisms between certain colored variants of \mathcal{C} and \mathcal{D} and the notion also applies to rainbows.

Second, an algebraic isomorphism is a map $f: \mathcal{C} \to \mathcal{D}$ between the two partitions that preserves the intersection numbers. More formally, we require that

- (A1) for all $R \in \mathcal{C}$, the relations R and f(R) have the same equality type,
- (A2) for all $R \in \mathcal{C}$ and permutations σ of [k], we have $f(R^{\sigma}) = f(R)^{\sigma}$, and
- **(A3)** for all $R, T_1, ..., T_k \in \mathcal{C}$, we have $p(R; T_1, ..., T_k) = p(f(R); f(T_1), ..., f(T_k))$,

but Property (A3) already implies the former two. Algebraic isomorphisms can be thought of as maps preserving the Weisfeiler-Leman colors and thus as a functional perspective on Weisfeiler-Leman equivalence. More formally, if for k-ary relational structures $\mathfrak A$ and $\mathfrak B$, $f \colon \operatorname{WL}_k(\mathfrak A) \to \operatorname{WL}_k(\mathfrak B)$ is an algebraic isomorphism that preserves the relations of $\mathfrak A$ and $\mathfrak B$, then f is the unique map that maps every color class of the stable coloring computed by k-WL on $\mathfrak A$ to the corresponding color class of the stable coloring computed by k-WL on $\mathfrak B$. In particular, we get $\mathfrak A \equiv_{k\text{-WL}} \mathfrak B$ in this case.

If $f: \mathcal{C} \to \mathcal{D}$ is an algebraic isomorphism, then f induces an algebraic isomorphism $f|_{\ell}: \mathcal{C}|_{\ell} \to \mathcal{D}|_{\ell}$ for every $\ell \leq k$. A combinatorial (respectively algebraic) automorphism of \mathcal{C} is a combinatorial (respectively algebraic) isomorphism from \mathcal{C} to itself. Every combinatorial isomorphism induces an algebraic isomorphism, but the converse is not true. Algebraic isomorphisms behave nicely with coherent closures as seen in the next lemma (the proof is analogue to the k=2 case [15, Lemma 2.4]):

- ▶ Lemma 5. Let \mathcal{R} be a k-ary rainbow, $\mathcal{C} = \mathrm{WL}_k(\mathcal{R})$, and $f: \mathcal{C} \to \mathcal{D}$ an algebraic isomorphism. Then
- 1. $\mathcal{D} = WL_k(\mathcal{R}^f)$, in particular, if \mathcal{C} is ℓ -induced, then so is \mathcal{D} ,
- **2.** f is fully determined by its action on basis relations in \mathcal{R} , and
- **3.** if $f|_{\mathcal{R}}$ is induced by a combinatorial isomorphism φ , then φ induces f.

A k-ary coherent configuration C is called *separable* if every algebraic isomorphism $f: \mathcal{C} \to \mathcal{D}$ from \mathcal{C} is induced by a combinatorial one. There is a close relation to the power of the Weisfeiler-Leman algorithm (the proof is analogue to the k=2 case [15, Theorem 2.5]):

▶ Lemma 6. Let $\ell < k$ and $\mathfrak A$ be an ℓ -ary relational structure. Then $\mathfrak A$ is identified by the k-dimensional Weisfeiler-Leman algorithm if and only if $\mathrm{WL}_k(\mathfrak{A})$ is separable.

Bounded variable counting logics

The k-dimensional Weisfeiler-Leman algorithm has an alternative characterization in terms of the distinguishing power of some logic, namely (k+1)-variable counting logic. First-order counting logic C is the extension of first-order logic by the counting quantifiers $\exists^{\geq k}$ for all natural numbers k, which state that there exist at least k distinct elements satisfying the formula that follows. But because first-order logic has the ability to simulate the counting quantifier $\exists^{\geq k}$ by a sequence of k usual existential quantifiers, adding counting quantifiers does not actually increase the expressive power of first-order logic. This situation changes when we restrict the number of variables. For a natural number $k \geq 2$, we define k-variable counting logic C^k to be the fragment of C which only uses the variables x_1, \ldots, x_k . In order to not restrict the expressive power of these logics too much, we do, however, allow requantifications, that is, quantifications over a variable within the scope of another quantification over the same variable. As an example, the following is a C^2 -formula stating that

$$\forall x_1 \exists x_2 \left(Ex_1 x_2 \wedge \left(\exists^{\geq 5} x_1 Ex_2 x_1 \right) \wedge \neg \exists^{\geq 6} x_1 Ex_2 x_1 \right),$$

which states that every vertex is adjacent to a vertex of degree 5.

The bijective pebble game.

The question whether k-WL can distinguish two graphs G and H has another characterization in terms of the so-called bijective (k+1)-pebble game. In this game, there are two players: Spoiler and Duplicator. Game positions are partial maps $\mathbf{g} \mapsto \mathbf{h}$ between G and H, where both tuples contain at most k+1 elements. We also sometimes identify such partial maps with the set $P = \{g_i \mapsto h_i : i \leq |\mathbf{g}|\}$. We think of these maps as k+1 pairs of corresponding pebbles placed in the two graphs.

If such a partial map is not a partial isomorphism, i.e., not an isomorphisms on the induced subgraphs, Spoiler wins immediately. Otherwise, at the beginning of each turn, Spoiler picks up one pebble pair, either from the board if all k+1 pairs are placed, or from the side if there are pebble pairs left. Duplicator responds by giving a bijection $\varphi \colon V(G) \to V(H)$ between the two graphs. Spoiler then places the pebble pair they picked up on a pair $(q, \varphi(q))$ of vertices of their choice. The game then continues in the resulting new position.

We say that Spoiler wins if the graphs have differing cardinality or they can reach a position that is no longer a partial isomorphism (and thus win immediately). Duplicator wins the game if they can find responses to Spoiler's moves indefinitely.

▶ Lemma 7 ([11], [25]). Let $\mathfrak A$ and $\mathfrak B$ be two relational structures of arity at most k, and $\mathbf{a} \in V(\mathfrak{A})^k$ and $\mathbf{b} \in V(\mathfrak{B})^k$ two tuples of vertices. Then the following are equivalent:

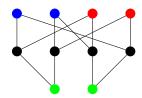


Figure 1 A CFI-gadget for a vertex of degree 3, consisting of four inner vertices and three outer pairs

- (i) Duplicator has a winning strategy in position $\mathbf{a} \mapsto \mathbf{b}$ of the bijective (k+1)-pebble game between \mathfrak{A} and \mathfrak{B} ,
- (ii) for every C^{k+1} -formula $\varphi(x_1,\ldots,x_k)$, we have $(\mathfrak{A},\mathbf{a}) \models \varphi$ if and only if $(\mathfrak{B},\mathbf{b}) \models \varphi$,
- (iii) the stable colors computed by k-WL for the tuples \mathbf{a} and \mathbf{b} agree. Further, every stable color class is definable by a single C^{k+1} -formula.

In particular, the Weisfeiler-Leman dimension of a structure is precisely one less than the number of variables needed to define the structure in first-order counting logic.

The CFI-construction.

CFI-graphs are certain graphs with high Weisfeiler-Leman dimension [11]. To construct them, we start with a base graph G, which is a connected simple graph, and a function $f: E(G) \to \mathbb{F}_2$. For a vertex $v \in V(G)$, we denote the set of edges incident to v by $E[v] := \{uv \colon v \in N_G(v)\} \subseteq E(G)$. Now, to construct the CFI-graph CFI(G, f), we replace each vertex $v \in V(G)$ by a gadget X_v which consists of inner vertices $I_v := \{v\} \times \{\mathbf{x} \in \mathbb{F}_2^{E[v]} \colon \sum \mathbf{x} = 0\}$ and outer vertices $\{v\} \times \{(e,i) \colon e \in E[v], i \in \mathbb{F}_2\}$. Inside each gadget, the inner and outer vertices each form an independent set, and an inner vertex (v, \mathbf{x}) and outer vertex (v, e, i) are connected by an edge if and only if $\mathbf{x}_e = i$. The resulting gadget for a vertex of degree 3 is depicted in Figure 1.

Next, we define the edge set between different gadgets. For every edge $e = uv \in E(G)$, we connect the outer vertices (u, e, i) and (v, e, j) if and only if i + j = f(e) and add no further edges. Thus, corresponding outer vertex pairs (u, e, \cdot) and (v, e, \cdot) are always connected by a matching, which is either *untwisted* if f(e) = 0, or *twisted* if f(e) = 1.

Finally, we define a vertex coloring on this graph. For every vertex v, we turn the set I_v of inner vertices into a color class of size $2^{d(v)-1}$. Moreover, we turn each outer pair $\{(v,e,0),(v,e,1)\}$ into a color class of size 2. This finishes the construction of CFI-graphs.

It turns out that for two functions $f, g: E(G) \to \mathbb{F}_2$, we have $CFI(G, f) \cong CFI(G, g)$ if and only if $\sum f = \sum g$, meaning that every even number of twists cancels out. Thus, we also write CFI(G, 0) and CFI(G, 1) for the *untwisted* and *twisted* CFI-graphs over the base graph G.

To understand the power of the Weisfeiler-Leman algorithm on CFI-graphs, it is convenient to study *tree-width*, which is a graph parameter that intuitively measures how far a graph is from being a tree. In this work, we do not need the formal definition of tree-width, and refer to [8]. The power of the Weisfeiler-Leman algorithm to distinguish CFI-graphs can now conveniently be expressed in terms of the tree-width of the base graphs, see [21].

▶ **Lemma 8.** For every base graph G of tree-width $tw(G) \ge 2$, we have

$$WL\text{-}dim(CFI(G, 0)) = WL\text{-}dim(CFI(G, 1)) = tw(G).$$

Proof. We first show that CFI-graphs can be easily distinguished from other graphs.

 \triangleright Claim 9. For every base graph G, 2-WL distinguishes CFI-graphs over G from all other graphs.

Proof. Let G be a base graph, $i \in \mathbb{F}_2$, and H be some graph not isomorphic to $\mathrm{CFI}(G,0)$ or $\mathrm{CFI}(G,1)$. If H has a color class of different size than the one of the same color in $\mathrm{CFI}(G,i)$, then 2-WL certainly distinguishes H and $\mathrm{CFI}(G,i)$. Let $u \in V(G)$ be of degree d. Let c be the color of inner vertices of the gadget of u and c_1,\ldots,c_d be the colors of outer vertices of this gadget. Then every vertex of color c has exactly one neighbor of each c_i and no others. For each $i \in [d]$, the two vertices of color c_i have the same number of neighbors in c, no common neighbor in c, and no neighbor in all c_j . Finally, for every two c-vertices u and v, the number of $i \in [d]$, for which u and v have a different c_i -neighbor, is even. All these conditions can easily be recognized by 2-WL and hence 2-WL identifies all CFI-gadgets. If $e = uv \in E(G)$, let c and d be the color classes for the outer vertices of the gadgets for u respectively v for the edge e. Then c and d are of size two and connected by a matching, which is also easily identified by 2-WL. Hence, 2-WL distinguishes $\mathrm{CFI}(G,i)$ from H.

We present another modified CFI-construction. For a base graph G and a function $f: E(G) \to \mathbb{F}_2$, let $\mathrm{CFI}'(G,f)$ be the graph obtained from $\mathrm{CFI}(G,f)$ in the following way: The vertex set of $\mathrm{CFI}'(G,f)$ is the set of all inner vertices of all CFI-gadgets in $\mathrm{CFI}(G,f)$. Two inner vertices (u,\mathbf{x}) and (v,\mathbf{y}) are adjacent in $\mathrm{CFI}'(G,f)$ whenever $uv \in E(G)$ and $\mathbf{x}_{uv} + \mathbf{y}_{uv} = f(uv)$. This is equivalent to that there is a path of length 3 between (u,\mathbf{x}) and (v,\mathbf{y}) (namely the path $(u,\mathbf{x}),(u,uv,\mathbf{x}_{uv}),(v,uv,\mathbf{y}_{uv}),(v,\mathbf{y})$). These modified CFI-construction shares all important properties with the one presented here, in particular, $\mathrm{CFI}'(G,f) \cong \mathrm{CFI}'(G,g)$ if and only if $\sum f = \sum g$. The following claim is well-known (see [11, 13, 21]):

 \triangleright Claim 10. For every $k \ge 3$ and every base graph G, Spoiler wins the bijective k-pebble game on $\mathrm{CFI}'(G,0)$ and $\mathrm{CFI}'(G,1)$ if and only if Spoiler wins the bijective k-pebble game on $\mathrm{CFI}(G,0)$ and $\mathrm{CFI}(G,1)$.

The main insight to prove this lemma is that it is never beneficial for Spoiler to place a pebble on an outer vertex apart from the very end of the game because an outer vertex (u, e, i) is uniquely identified by an inner vertex (u, \mathbf{x}) such that $\mathbf{x}_e = i$.

 \triangleright Claim 11 ([21]). For every $k \ge 3$ and every base graph G, Spoiler wins the bijective (k+1)-pebble game on $\mathrm{CFI}'(G,0)$ and $\mathrm{CFI}'(G,1)$ if and only if G has tree-width $\mathrm{tw}(G) \le k$.

Now finally let G be a base graph of tree-width $\operatorname{tw}(G) \geq 2$. By Claims 10 and 11 and Lemma 7, k-WL distinguishes $\operatorname{CFI}(G,0)$ and $\operatorname{CFI}(G,1)$ if and only if $k \geq \operatorname{tw}(G)$. This implies WL-dim $(\operatorname{CFI}(G,i)) \geq \operatorname{tw}(G)$ for both $i \in \mathbb{F}_2$. By Claim 9, 2-WL distinguishes CFI-graphs over G from all other graphs. Since there are only two non-isomorphic CFI-graphs over G and $\operatorname{tw}(G)$ -WL distinguishes them, $\operatorname{tw}(G)$ -WL identifies CFI-graphs over G and thus WL-dim $(\operatorname{CFI}(G,i)) \leq \operatorname{tw}(G)$ for both $i \in \mathbb{F}_2$. This implies WL-dim $(\operatorname{CFI}(G,0)) = \operatorname{WL-dim}(\operatorname{CFI}(G,1)) = \operatorname{tw}(G)$.

3 Deciding identification for graphs with 5-bounded color classes

As recently shown [15], identification of a given graph with 5-bounded color classes by 2-WL is polynomial-time decidable. We extend this result to arbitrary dimensions of the Weisfeiler-Leman algorithm. We adapt the approach of [15] and solve the separability for 2-induced k-ary coherent configurations with 5-bounded fibers instead. We generalize the

elimination of interspaces containing a matching and interspaces of type $2K_{1,2}$ in order to reduce to star-free 2-induced k-ary coherent configurations. By characterizing separability using certain automorphism groups, we provide a new reduction of the separability problem for such configurations to the isomorphism problem for graphs with bounded color classes, which can be solved in polynomial time.

3.1 Functional basis relations and disjoint unions of stars

Let \mathcal{C} be a k-ary coherent configuration and $X,Y \in F(\mathcal{C})$ two distinct fibers. A disjoint union of stars between X and Y is a basis relation $S \in \mathcal{C}|_2[Y,X]$ such that every vertex in Y is incident to exactly one outgoing edge in S. If no interspace of \mathcal{C} contains a disjoint union of stars, \mathcal{C} is called star-free.

In this section, we want to show that whenever $\mathcal{C}|_2[X,Y]$ contains a disjoint union of stars, then \mathcal{C} is separable if and only if $\mathcal{C} \setminus X := \mathcal{C}[V(\mathcal{C}) \setminus X]$ is separable. This allows us to reduce the separability problem to the class of star-free k-ary coherent configurations. As we will see, this is possible because the configuration \mathcal{C} is fully determined by the configuration $\mathcal{C} \setminus X$, together with the information how the disjoint union of stars in $\mathcal{C}[X,Y]$ attaches to Y.

Slightly more generally, we call a binary basis relation $S \in \mathcal{C}|_2$ functional if every vertex of \mathcal{C} is incident to at most one outgoing edge in S. Besides disjoint unions of stars, examples of such basis relations are matchings or directed cycles within a fiber.

For a functional basis relation $S \in \mathcal{C}|_2$, we define a map

$$\nu_S \colon V(\mathcal{C}) \to V(\mathcal{C}),$$

$$y \mapsto \begin{cases} x & \text{if } yx \in S, \\ y & \text{if no such } x \text{ exists.} \end{cases}$$

Moreover, we denote for every $I \subseteq [k]$ by $\nu_S^I : V(\mathcal{C})^k \to V(\mathcal{C})^k$ the map defined via $\nu_S^I(y_1, \ldots, y_k) := (z_1, \ldots, z_k)$, where $z_i := \nu_S(y_i)$ if $i \in I$ and $z_i := y_i$ for every $i \in [k] \setminus I$.

▶ Lemma 12. Let C be a k-ary coherent configuration, and $S \in C|_2$ a functional basis relation. For every subset $I \subseteq [k]$ and every basis relation $R \in C$, the set $\nu_S^I(R)$ is also a basis relation of C.

Moreover, for every algebraic isomorphism $f: \mathcal{C} \to \mathcal{D}$, the basis relation f(S) is also functional in \mathcal{D} and $f \circ \nu_S^I = \nu_{f(S)}^I \circ f$ for every $I \subseteq [k]$.

Proof. If $I = \{i_1, \dots, i_\ell\}$, then we can write $\nu_S^I = \nu_S^{\{i_1\}} \circ \dots \circ \nu_S^{\{i_\ell\}}$. As both claims of the lemma are compatible with compositions, it thus suffices to consider the case that $I = \{i\}$ is a singleton.

Now, consider a colored variant \mathfrak{C} of \mathcal{C} . Because the partition \mathcal{C} is stable under k-WL-refinement, it is also equal to the partition into C^{k+1} -types by Lemma 7. In particular, for every formula $\varphi(x_1,\ldots,x_\ell)\in C^{k+1}$ with $\ell\leq k$, the set $[\![\varphi]\!]_{\mathfrak{C}}:=\{\mathbf{x}\in V(\mathcal{C})^\ell\colon \mathfrak{C},\mathbf{x}\models\varphi\}$ is a union of basis relations and conversely, every union of basis relations is of the form $[\![\varphi]\!]_{\mathfrak{C}}$ for an appropriate formula φ .

Thus, we find a formula $\nu_S(x,y)$ which encodes the graph of the function ν_S , and further, we find for every basis relation $R \in \mathcal{C}$, a formula φ_R encoding R. But then, the formula $\psi_{\nu_S^i(R)} := \exists x_{k+1} \nu_S(x_{k+1}, x_i) \wedge \varphi_R \frac{x_{k+1}}{x_i}$ is a C^{k+1} -formula defining the set $\nu_S^i(R)$. Thus, $\nu_S^i(R)$ is again a union of basis relations. If it was a proper union of basis relations, then there would exist some basis relation $T \subsetneq \nu_S^i(R)$. But then, the formula $\psi_R \wedge \exists x_{k+1} \nu_S(x_i, x_{k+1}) \wedge \varphi_T \frac{x_{k+1}}{x_i}$ would define a proper subset of R, which is impossible. Thus, $\nu_S^i(R)$ is a basis relation.

For the second claim, consider an algebraic isomorphism $f: \mathcal{C} \to \mathcal{D}$ and construct a colored variant of \mathcal{D} in such a way that f becomes a color-preserving map from \mathfrak{C} to \mathfrak{D} . Then f also preserves Weisfeiler-Leman colors and thus C^{k+1} -types. In particular, this means that $f(\llbracket \varphi \rrbracket_{\mathfrak{C}}) = \llbracket \varphi \rrbracket_{\mathfrak{D}}$ for every formula φ . In particular, for every basis relation $R \in \mathcal{C}$, we get

$$f(\nu_S^i(R)) = f(\llbracket \psi_{\nu_S^i(R)} \rrbracket_{\mathfrak{C}}) = \llbracket \psi_{\nu_S^i(R)} \rrbracket_{\mathfrak{D}} = \nu_{f(S)}^i(f(R)),$$

which is what we wanted to show.

From now on, we fix S to be a disjoint union of stars in C[X,Y]. In this case, ν_S maps the fiber Y onto the fiber X while fixing every other fiber. Then, Lemma 12 in particular implies that the basis relations of C are fully determined by those of $C \setminus X$. Indeed, if R is a basis relation of C and $\mathbf{x} \in R$, let $I \subseteq [k]$ be the set of X-components of \mathbf{x} . Then, for every $i \in I$, we pick a ν_S -preimage y_i of x_i . If we now define $\mathbf{y} := \mathbf{x} \left(\frac{y_i}{i} \right)_{i \in I}$, we get $\mathbf{x} = \nu_S^I(\mathbf{y})$ and thus $R = T^{\nu_S^I}$, where $T \in C \setminus X$ is the basis relation containing \mathbf{y} .

Together with Lemma 12, this implies that the basis relations of \mathcal{C} are precisely those of the form $T^{\nu_S^I}$ for $T \in \mathcal{C} \setminus X$ and $I \subseteq [k]$. Moreover, we can assume that I is a subset of the set of Y-components of T. These representations of basis relations of \mathcal{C} are, however, not necessarily unique.

Before we are ready to eliminate all interspaces with a disjoint union of stars, we need one more technical observation

▶ **Lemma 13.** If $S \in \mathcal{C}|_2[X,Y]$ is a disjoint union of stars, then the relation

$$\mathsf{Eq}_S \coloneqq \{ vw \in Y^2 \colon \nu_S(v) = \nu_S(w) \}$$

is a union of 2-ary basis relations. Moreover, for every algebraic isomorphism $f: \mathcal{C} \to \mathcal{D}$, we have $f(\mathsf{Eq}_S) = \mathsf{Eq}_{f(S)}$.

Proof. It suffices to assume that \mathcal{C} is a 2-ary coherent configuration. Then, $vw \in \mathsf{Eq}_S$ if and only if for some z we have $zv, zw \in S$. If E is the basis relation of \mathcal{C} containing vw, this is equivalent to $p\left(E;S,S^{-1}\right) \geq 1$, where $S^{-1} = \{yx\colon xy \in S\}$. Because this property depends only on the basis relation containing vw and not on vw itself, the first claim follows.

For the second claim, note that $R \subseteq \mathsf{Eq}_S$ if and only if $\nu_S^{\{1\}}(R) = S$. Lemma 12 now implies

$$f(S) = (f \circ \nu_S^{\{1\}})(R) = (\nu_{f(S)}^{\{1\}} \circ f)(R) = \nu_{f(S)}^{\{1\}}(f(R))$$

and thus $f(R) \subseteq \mathsf{Eq}_{f(S)}$.

We are now ready to prove one of the implications of the claimed equivalence:

▶ **Lemma 14.** Let C be a k-ary coherent configuration, $X, Y \in F(C)$ two distinct fibers and $S \in C|_2[X,Y]$ a disjoint union of stars between X and Y. If $C \setminus X$ is separable, then so is C.

Proof. Assume that $\mathcal{C} \setminus X$ is separable and let $f: \mathcal{C} \to \mathcal{D}$ be an algebraic isomorphism. Then f restricts to an algebraic isomorphism $f_0: \mathcal{C} \setminus X \to \mathcal{D} \setminus f(X)$, which is induced by a combinatorial isomorphism $\varphi_0: \mathcal{C} \setminus X \to \mathcal{D} \setminus f(X)$ because $\mathcal{C} \setminus X$ is separable. We extend this map to all of \mathcal{C} by defining

$$\varphi \colon V(\mathcal{C}) \to V(\mathcal{D}),$$

$$x \mapsto \begin{cases} \varphi_0(x) & \text{if } x \notin X, \\ (\nu_{f(S)} \circ \varphi_0)(y) & \text{if } x \in X \text{ and } x = \nu_S(y) \text{ for some } y \in Y. \end{cases}$$

We claim that φ is a well-defined combinatorial isomorphism inducing f.

To see well-definedness, we need to argue that the definition of $\varphi(x)$ for vertices $x \in X$ does not depend on the choice of its ν_S -preimage $y \in Y$. But indeed, if $\nu_S(y) = \nu_S(y')$, then $yy' \in \mathsf{Eq}_S$. By Lemma 13, it holds that $f(\mathsf{Eq}_S) = \mathsf{Eq}_{f(S)}$. As φ_0 induces f_0 , we get $\varphi_0(y)\varphi_0(y') \in \mathsf{Eq}_{f(S)}$. But this means that $\nu_{f(S)}(\varphi_0(y)) = \nu_{f(S)}(\varphi_0(y'))$ which implies that φ is well-defined. Moreover, it is a bijection as φ_0 is a bijection and $\varphi|_X \colon X \to f(X)$ is a bijection. To show that φ induces f, consider a basis relation $\nu_S^I(R) \in \mathcal{C}$ where $R \in \mathcal{C} \setminus X$ and I is a subset of the Y-components of X. Applying φ to this basis relation yields

$$\varphi(\nu_S^I(R)) = (\varphi \circ \nu_S^I)(R).$$

Using the definition of φ , this is equal to

$$= \left(\nu^I_{f(S)} \circ \varphi_0\right)(R) = \left(\nu^I_{f(S)} \circ f_0\right)(R) = \left(\nu^I_{f(S)} \circ f\right)(R),$$

which, by Lemma 13, is equal to

$$= (f \circ \nu_S^I)(R) = f(\nu_S^I(R)).$$

This proves that φ induces f and thus in particular that φ is a combinatorial isomorphism. Thus, \mathcal{C} is separable.

Next, we want to prove the converse of Lemma 14, which will allow us to eliminate all disjoint unions of stars without affecting separability of the configuration. For this, we need one auxiliary lemma:

▶ Lemma 15. For every k-ary coherent configuration C, fiber $Y \in F(C)$ and every union of basis relations $E \in (C|_2[Y])^{\cup}$ which forms an equivalence relation on Y, there exists an extension C_E^{\star} of C by a single fiber X such that the interspace $C_E^{\star}[X,Y]$ contains a disjoint union of stars S satisfying $E = \mathsf{Eq}_S$. Moreover, this extension is unique up to combinatorial isomorphism.

Furthermore, the construction satisfies the following properties:

- 1. For every C and E as above, we have $(C|_2)_E^{\star} = (C_E^{\star})|_2$.
- **2.** For every configuration C and every interspace C[X,Y] that contains a disjoint union of stars S, we have $C \cong (C \setminus X)^*_{\mathsf{Eq}_S}$.

Proof. Define X := Y/E as the quotient of Y by the equivalence relation E, and $\nu_S \colon Y \to X$ as the natural projection. Now, we define $V(\mathcal{C}_E^{\star}) := V(\mathcal{C}) \cup X$, and extend ν_S to all of $V(\mathcal{C}_E^{\star})$ by letting it act as the identity on every fiber besides Y. Once again, we write ν_S^I for the map $V(\mathcal{C}_E^{\star})^k \to V(\mathcal{C}_E^{\star})^k$ defined by $\nu_S^I(y_1, \ldots, y_k) := (z_1, \ldots, z_k)$, where $z_i := \nu_S^I(y_i)$ for all $i \in I$ and $z_i := y_i$ for all $i \notin I$. To simplify notation, we also abbreviate this map as (I).

By Lemma 12, all of the sets $R^{(I)}$ with $R \in \mathcal{C}$ and I a subset of the Y-components of R must be basis relations of \mathcal{C}_E^{\star} if we want to make \mathcal{C}_E^{\star} coherent. Hence, we can only define the basis relations of \mathcal{C}_E^{\star} in a unique way:

$$\mathcal{C}_E^{\star} \coloneqq \left\{ R^{(I)} \colon R \in \mathcal{C}, I \subseteq [k], R_I \subseteq Y^{|I|} \right\}.$$

 \triangleright Claim 16. \mathcal{C}_E^{\star} is a k-ary rainbow with $\mathcal{C}_E^{\star} \setminus X = \mathcal{C}$.

Proof. The fact that every k-tuple in \mathcal{C}_E^* is contained in at least one of these basis relations follows from surjectivity of ν_S . To see that every two distinct basis relations are disjoint, we note that for every basis relation $R^{(I)} \in \mathcal{C}_E^*$, the set I can be reconstructed as the set

of X-components of $R^{(I)}$. This implies that if $R^{(I)} \cap R'^{(I')} \neq \emptyset$, then I = I'. For every k-tuple \mathbf{x} in this intersection, we can write $\mathbf{x} = \mathbf{y}^{(I)} = (\mathbf{y}')^{(I)}$ for some $\mathbf{y} \in R$ and $\mathbf{y} \in R'$. Then for every $i \in I$, we have $\nu_S(y_i) = x_i = \nu_S(y_i')$ and thus $y_i y_i' \in E$.

Now, fix some $i \in I$, and let R_i be the basis relation containing $\mathbf{y} \frac{y_i'}{i}$. For all $j \neq i$, we set

$$R_j := \{ \mathbf{z} \in V(\mathcal{C}) \colon z_i z_j \in E \}.$$

The sets R_i are unions of basis relations, and we have

$$p\left(R;R_1,\ldots,R_k\right)\geq 1,$$

because $\mathbf{y} \frac{y_i'}{j} \in R_j$ for all $j \in [k]$. But as the relations R_j for $j \neq i$ also ensure that any witness z to this inequality must satisfy $y_i z \in E$, we get that for every $\mathbf{y} \in R$, there is some $z \in V(\mathcal{C})$ with $y_i z \in E$ such that $\mathbf{y} \frac{z}{i} \in R_i$. We can iterate this construction with R_i instead of R and a different $i \in I$. If we do this for all $i \in I$, we finally get that for every $\mathbf{y} \in R$ and every $i \in I$, there is some $z_i \in V(\mathcal{C})$ such that $y_i z_i \in E$ and furthermore, these z_i satisfy $\mathbf{y} \left(\frac{z_i}{i}\right)_{i \in I} \in R'$. But this means $R^{(I)} \subseteq R'^{(I)}$ and by symmetry of the argument, $R^{(I)} = R'^{(I)}$.

This concludes the proof that the basis relations $R^{(I)}$ form a partition of $V(\mathcal{C}_E^*)^k$. The fact that this partition is compatible with equality types follows from the fact that for $\mathbf{x} = \mathbf{y}^{(I)} \in R^{(I)}$ we have $x_i = x_j$ if and only if either $i, j \in I$ and $y_i y_j \in E$ or $i, j \notin I$ and $y_i = y_j$. The fact that the partition is compatible with permutations follows from the observation that $(R^{(I)})^{\sigma} = (R^{\sigma})^{(I^{\sigma})}$ for every permutation σ on [k]. This concludes the proof that \mathcal{C}_E^* is a k-ary rainbow. As furthermore $R^{(\emptyset)} = R$ for all $R \in \mathcal{C}$, this rainbow extends \mathcal{C} .

In order to finish the proof that \mathcal{C}_E^{\star} is a k-ary coherent configuration, it remains to show that the intersection numbers of \mathcal{C}_E^{\star} are well-defined, that is, independent on the specific choice of vertex tuple in the basis relation. For this, we write $R \sim_I R'$ for basis relations $R, R \in \mathcal{C}$ if $R^{(I)} = R'^{(I)}$. Moreover, we write

$$R/I \coloneqq \bigcup_{\substack{R' \in \mathcal{C} \\ R' \sim_I R}} R'.$$

Now, to prove coherence of \mathcal{C}_E^{\star} , let $R^{(I)}, T_1^{(I_1)}, \dots, T_k^{(I_k)} \in \mathcal{C}_E^{\star}$ be arbitrary basis relations and pick some representative $\mathbf{x} = \mathbf{y}^{(I)} \in R^{(I)}$ with $\mathbf{y} \in R$. For every $z \in V(\mathcal{C})$, we then have

$$\mathbf{x} \frac{z}{i} = \mathbf{y}^{(I)} \frac{z}{i} = \left(\mathbf{y} \frac{z}{i}\right)^{(I \setminus \{i\})}.$$

This means that $\mathbf{x}_{i}^{\underline{z}} \in T_{i}^{(I_{i})}$ if and only if $I_{i} = I \setminus \{i\}$ and $\mathbf{y}_{i}^{\underline{z}} \in T_{i}[\sim_{I_{i}}]$. For all $x \in X$ and $y \in Y$ such that $x = \nu_{S}(y)$, the analogous calculation instead yields

$$\mathbf{x}\frac{x}{i} = \mathbf{y}^{(I)} \frac{\nu_S(y)}{i} = \left(\mathbf{y}\frac{y}{i}\right)^{(I \cup \{i\})}.$$

This means that $\mathbf{x}_{i}^{\underline{x}} \in T_{i}^{(I_{i})}$ if and only if $I_{i} = I \cup \{i\}$ and $\mathbf{y}_{i}^{\underline{y}} \in T_{i}[\sim_{I_{i}}]$. As one of the S-neighbors of x satisfies this last condition if and only if every S-neighbor does, we overcount the number of such $x \in X$ by a factor of p(X; S) if we count the number of such $y \in Y$ instead.

In total, this means that if for some $\mathbf{x} \in R^{(I)}$ we define

$$p\left(\mathbf{x}; T_1^{(I_1)}, \dots, T_k^{(I_k)}\right) \coloneqq \left|\left\{y \in V(\mathcal{C}) \colon \mathbf{x} : \frac{y}{i} \in R_i \text{ for all } i \in [k]\right\}\right|,$$

then this intersection number can be expressed as

$$p\left(\mathbf{x}; T_{1}^{(I_{1})}, \dots, T_{k}^{(I_{k})}\right) = \begin{cases} p\left(R; T_{1}[\sim_{I_{1}}], \dots, T_{k}[\sim_{I_{k}}]\right) & \text{if } I_{i} = I \setminus \{i\} \text{ for all } i \in [k], \\ \frac{p\left(R; T_{1}[\sim_{I_{1}}], \dots, T_{k}[\sim_{I_{k}}]\right)}{p\left(X; S\right)} & \text{if } I_{i} = I \cup \{i\} \text{ for all } i \in [k], \end{cases}$$

$$0 \qquad \text{otherwise.}$$

As the right-hand side of Equation 1 does not depend on \mathbf{x} but only on the basis relation R containing \mathbf{y} which can be chosen to be the same for all $\mathbf{x} = \mathbf{y}^{(I)} \in R^{(I)}$, the intersection number $p\left(R^{(I)}; T_1^{(I_1)}, \dots, T_k^{(I_k)}\right)$ is well-defined. This proves coherence of \mathcal{C}_E^{\star} .

Property 1 follows from the observation that for all $I \subseteq [k]$, the [2]-face of $R^{(I)}$ is given by $(R_{[2]})^{(I \cap [2])}$. Property 2 follows from uniqueness of the extension.

Now, we are ready to prove the converse of Lemma 14.

- ▶ **Lemma 17.** Let C be a k-ary coherent configuration, $X, Y \in F(C)$ two distinct fibers, and $S \in C|_2[X,Y]$ a disjoint union of stars between X and Y.
- **1.** If C is separable, so is $C \setminus X$.
- **2.** If C is 2-induced, so is $C \setminus X$.

Proof. We start with the first claim. Assume that \mathcal{C} is separable and let $f_0 \colon \mathcal{C} \setminus X \to \mathcal{D}$ be an arbitrary algebraic isomorphism. Then $E \coloneqq f_0(\mathsf{Eq}_S)$ is a union of 2-ary basis relations in $\mathcal{D}[f_0(Y)]$ which forms an equivalence relation. We want to construct an extension $f \colon \mathcal{C} \to \mathcal{D}_E^{\star}$ of f. Then, by separability of \mathcal{C} , we would find a combinatorial isomorphism φ that induces f and whose restriction to $\mathcal{C} \setminus X$ induces f_0 .

For this, recall that the basis relations of \mathcal{C} are all of the form $R^{(I)} = \nu_S^I(R)$ with $R \in \mathcal{C} \setminus X$ and I a subset of the Y-components of R; and the basis relations of \mathcal{D}_E^{\star} are of the form $T^{(I)} = \nu_{f(S)}^I(T)$ for $T \in \mathcal{D}$ and I a subset of the f(Y)-components of T. We define the extension f of f_0 by

$$f\left(R^{(I)}\right) := \left(f_0(R)\right)^{(I)}$$
.

 \triangleright Claim 18. The map f is a well-defined extension of f_0 .

Proof. To see that this map is well-defined, note that the set I is precisely the set of X-components of $R^{(I)}$ and is thus determined by the basis relation. Thus, we only need to show that whenever $R^{(I)} = R'^{(I)}$, then also $(f_0(R))^{(I)} = (f_0(R'))^{(I)}$. The former is the case if and only if R and R' can be transformed into one another by replacing for all $\mathbf{x} \in R$ the X-components of \mathbf{x} by one or multiple Eq_S -equivalent components. As the analogous claim is true in \mathcal{D}_E^{\star} and as f_0 sends Eq_S to $\mathsf{Eq}_{f(S)}$ by construction, it follows that f is well-defined. Moreover, f extends f_0 , as for $I = \emptyset$, we get $f(R) = f_0(R)$ for all $R \in \mathcal{C} \setminus X$.

 \triangleright Claim 19. The map f is an algebraic isomorphism.

Proof. The intersection numbers of both $\mathcal{C} \cong (\mathcal{C} \setminus X)_{\mathsf{Eq}_S}^{\star}$ and \mathcal{D}_E^{\star} are determined in terms of those of $\mathcal{C} \setminus X$ and \mathcal{D} by Equation (1). To see that f preserves the right-hand side of this formula, recall that we defined

$$R/I = \bigcup_{\substack{R' \in \mathcal{C} \backslash X \\ R^{(I)} = (R')^{(I)}}} R'.$$

It now remains to show that $f_0(R/I) = f_0(R)/I$ for all $R \in \mathcal{C} \setminus X$ and every subset I of the Y-components of R.

We again use the observation that $R^{(I)} = (R')^{(I)}$ if and only if R can be obtained from R' by replacing some of its Y-components by Eq_S -neighbors, and that the analogous statement is true in \mathcal{D}_E^\star with respect to $\mathsf{Eq}_{f(S)}$. Therefore, we have $R^{(I)} = (R')^{(I)}$ if and only if $(f_0(R))^{(I)} = (f_0(R'))^{(I)}$. This implies f(R/I) = f(R)/I and thus that f preserves intersection numbers.

We have now successfully extended the algebraic isomorphism $f_0: \mathcal{C} \setminus X \to \mathcal{D}$ to an algebraic isomorphism $f: \mathcal{C} \to \mathcal{D}_E^{\star}$. Because \mathcal{C} is separable, f is induced by a combinatorial isomorphism φ which restricts to an isomorphism φ_0 inducing f_0 .

For the second claim, assume that \mathcal{C} is 2-induced. To show that $\mathcal{C} \setminus X = \operatorname{WL}_k((\mathcal{C} \setminus X)|_2)$, we first note that the left-hand side is a k-ary coherent configuration on $V(\mathcal{C}) \setminus X$ that is at least as fine as $(\mathcal{C} \setminus X)|_2$ and thus also at least as fine as $\operatorname{WL}_k((\mathcal{C} \setminus X)|_2)$. For the converse, consider the k-ary coherent configuration $\mathcal{C}' := \operatorname{WL}_k((\mathcal{C} \setminus X)|_2)^*_{\mathsf{Eq}_S}$. By choosing the obvious bijection between the fiber $X \in F(\mathcal{C})$ and the additional fiber of \mathcal{C}' , we can assume that \mathcal{C}' is a k-ary coherent configuration on $V(\mathcal{C})$, which is furthermore at most as fine as \mathcal{C} . Now, note that

$$\mathcal{C}'|_{2} = \left(\mathrm{WL}_{k} \big((\mathcal{C} \setminus X)|_{2} \big) \big|_{2} \right)_{\mathsf{Eq}_{S}}^{\star} = \big((\mathcal{C} \setminus X)|_{2} \big)_{\mathsf{Eq}_{S}}^{\star} = (\mathcal{C} \setminus X)_{\mathsf{Eq}_{S}}^{\star} \big|_{2} = \mathcal{C}|_{2},$$

where the first and third equality follow from Property 1 of Lemma 15, the second equality follows from the fact that $\mathcal{C} \setminus X$ is a k-ary coherent configuration, and the last equality follows from Property 2 of Lemma 15. As \mathcal{C} is 2-induced, this yields $\mathcal{C}' \preceq \operatorname{WL}_k(\mathcal{C}|_2) = \mathcal{C}$ and thus $\mathcal{C} = \mathcal{C}'$. Finally,

$$\mathcal{C} \setminus X = \mathcal{C}' \setminus X = (\operatorname{WL}_k((\mathcal{C} \setminus X)|_2))_E^* \setminus X = \operatorname{WL}_k((\mathcal{C} \setminus X)|_2)$$

shows that $\mathcal{C} \setminus X$ is 2-induced.

Lemma 14 together with Lemma 17 allows us to remove fibers from coherent configurations containing a disjoint union of stars without affecting the separability of the configuration. Thus, we have reduced the separability problem for arbitrary k-ary coherent configurations to the same problem for star-free configurations. Furthermore by Lemma 17, this process preserves 2-inducedness of a configuration. Thus, in order to solve the separability problem for 2-induced k-ary coherent configurations, it only remains to consider 2-induced star-free configurations. This simultaneously generalizes the elimination of interspaces containing a matching and the elimination of fibers of size 2 from [15].

3.2 Structure of k-ary coherent configurations with 5-bounded fibers

A list of all isomorphism types of 2-ary coherent configurations on a single fiber of order at most 5 is known [15, 39], see Figure 2. Furthermore, the authors of [15] also give a complete list of possible interspaces between fibers of order up to 4. Here, it is always possible that the interspace C[X,Y] is uniform, meaning that it consists of only a single basis relation $X \times Y$. Uniform interspaces are usually easy to handle and the other interspace types are the more interesting ones. As we already eliminated interspaces containing a disjoint union of stars, Figure 3 contains only the two remaining non-uniform and star-free interspaces.

In [15, Lemma 5.2], it was shown that both of these interspaces enforce the existence of a matching in both incident fibers. These matchings are denoted by dotted lines in Figure 3.

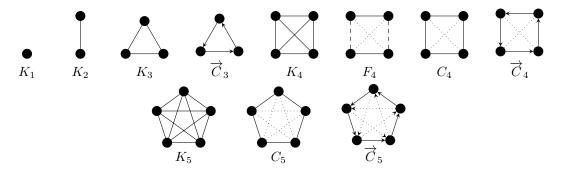


Figure 2 The complete list of 2-ary coherent configurations on a single fiber of order up to 4 from [15], and the three 2-ary coherent configurations on a single fiber of order 5 from [39].

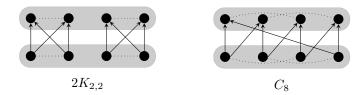


Figure 3 All non-uniform and star-free interspace types between two fibers of order up to 5. In each case, one of the basis relations between the two fibers is missing and can be reconstructed as the complement of the one drawn.

As there can be no non-uniform interspaces between fibers of coprime orders [15, Lemma 3.1], we are thus only missing an enumeration of the non-uniform star-free interspaces between two fibers of order 5 each. However, as every such interspace must contain a relation which has degree 2 in both fibers, the only candidate for such an interspace is the C_{10} , the interspace containing a basis relation forming a 10-cycle between the two fibers. But this 10-cycle induces a matching between opposite vertices. Thus, in a star-free 2-ary coherent configuration with 5-bounded fibers, there can be no non-uniform interspaces incident to any fibers of order 5.

In the k-dimensional case, interspaces are generally not between two, but between k fibers and their enumeration and analysis is more complicated. But as we are only dealing with 2-induced k-ary coherent configuration, the two-dimensional interspaces suffice for our purposes.

We need the following fact about star-free (and indeed even matching-free) configurations:

▶ Lemma 20 ([15, Lemma 6.2]). In a star-free 2-ary coherent configuration, no two distinct interspaces of type C_8 can be incident to a common fiber.

This allows us to prove the following generalization of [15, Lemma 7.1]:

▶ **Lemma 21.** Let C be a star-free, 2-ary coherent configuration with 5-bounded fibers. If $f: C \to D$ is an algebraic isomorphism, then there exists a combinatorial isomorphism $\varphi: V(C) \to V(D)$ which satisfies $\varphi(X) = f(X)$ for every fiber $X \in F(C)$.

Proof. As every 2-ary coherent configuration with at most 8 vertices is separable [15], we can pick for every C_8 -interspace $\mathcal{C}[X,Y]$ a combinatorial isomorphism $\varphi_{X,Y}\colon X\cup Y\to f(X)\cup f(Y)$ which induces $f|_{\mathcal{C}[X\cup Y]}$ and for every fiber Z not incident to a C_8 -interspace an isomorphism $\varphi_Z\colon Z\to f(Z)$ inducing $f|_{\mathcal{C}[Z]}$.

By Lemma 20, the domains of these partial combinatorial isomorphisms form a partition of $V(\mathcal{C})$. Thus, we can combine these bijections to get a total bijection $\varphi \colon V(\mathcal{C}) \to V(\mathcal{D})$. It remains to show that φ is a combinatorial isomorphism on every interspace.

On uniform interspaces, the claim is immediate and further it is true by construction on interspaces of type C_8 . The only remaining interspace type is thus $2K_{2,2}$. But here, the interspace is uniquely determined by its isomorphism-type and the matchings it induces in the incident fibers. On such interspaces φ thus either induces f or switches the two binary basis relations. In both cases, φ is a combinatorial isomorphism on C.

We call an algebraic automorphism f of a k-ary coherent configuration \mathcal{C} strict if it fixes every fiber, i.e. satisfies f(X) = X for every fiber $X \in F(\mathcal{C})$. The strict algebraic automorphisms of \mathcal{C} form a group, which we denote by $\mathbb{A}(\mathcal{C})$. Now, we get the following reformulation of separability:

▶ Lemma 22. A star-free and 2-induced k-ary coherent configuration C with 5-bounded fibers is separable if and only if every strict algebraic automorphism is induced by a combinatorial automorphism.

Proof. The forward implication is immediate. For the converse implication, let $f: \mathcal{C} \to \mathcal{D}$ be an algebraic isomorphism. By Lemma 21, we find a combinatorial isomorphism $\varphi \colon V(\mathcal{C}) \to V(\mathcal{D})$ which satisfies $\varphi(X) = f(X)$ for every fiber $X \in F(\mathcal{C})$. Because \mathcal{C} is 2-induced, this map is also a combinatorial isomorphism between \mathcal{C} and \mathcal{D} and thus induces an algebraic isomorphism $g_{\varphi} \colon \mathcal{C} \to \mathcal{D}$ such that $f(X) = g_{\varphi}(X)$ for every fiber $X \in \mathcal{C}$. Now, $g_{\varphi}^{-1} \circ f$ is a strict algebraic automorphism of \mathcal{C} and is thus induced by some combinatorial automorphism ψ . But then f is induced by $\varphi \circ \psi$, which proves separability of \mathcal{C} .

3.3 Strict algebraic automorphisms

In this section, we give a polynomial time algorithm to decide whether every strict algebraic automorphism of a k-ary coherent configuration \mathcal{C} is induced by a combinatorial automorphism.

We will heavily make use of the fact that the graph isomorphism problem is polynomialtime solvable for graphs with bounded color classes:

▶ Lemma 23 ([6, 16]). Isomorphism of k-ary relational structures of order n and c-bounded color classes can be decided in time $O_{k,c}(n^{O(k)})$. Moreover, a generating set of $Aut(\mathfrak{A})$ for of a k-ary relational structure \mathfrak{A} of order n and c-bounded color classes can also be computed in the same time bounds.

By encoding the algebraic structure of a k-ary coherent configuration into a graph $G_{\mathcal{C}}$ such that the strict algebraic automorphisms of \mathcal{C} become automorphisms of $G_{\mathcal{C}}$, we obtain the following:

▶ Lemma 24. There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k-ary coherent configuration C of order n and c-bounded fibers, computes a generating set of A(C).

Proof. We construct a graph $G_{\mathcal{C}}$ with bounded color classes and order $O(n^k)$ whose automorphism group is isomorphic to $\mathbb{A}(\mathcal{C})$. Lemma 23 then yields the claim.

For this, we call a k-tuple $\mathbf{R} = (R_1, \dots, R_k)$ of basis relations of \mathcal{C} compatible if for some basis relation $R \in \mathcal{C}$ we have $p(R; R_1, \dots, R_k) > 0$. Let $\mathcal{C}^{(k)}$ be the collection of all such tuples. Because for every compatible tuple \mathbf{R} of basis relations there must exist some $\mathbf{x} \in V(\mathcal{C})^k$ and $x' \in V(\mathcal{C})$ such that $\mathbf{x} \frac{x'}{i} \in R_i$ for all $i \in [k]$, there are at most n^{k+1}

compatible tuples, and these can be found in time $n^{O(k)}$ by enumerating all such pairs (\mathbf{x}, x') noting which compatible tuple of basis relations they correspond to.

Now, we define the graph $G_{\mathcal{C}}$ on the vertex set $V(G_{\mathcal{C}}) := \mathcal{C} \cup \mathcal{C}^{(k)}$ by putting in the following labeled edges:

- \blacksquare connect R and **R** with an edge labeled $p(R; \mathbf{R})$,
- \blacksquare connect R and **R** with an edge labeled $\{i \in [k]: R_i = R\},\$

Finally, we define a vertex coloring on $G_{\mathcal{C}}$:

- color each basis relation R using the tuple $(R_{\{i\}})_{i\in[k]}$ of fibers of its components,
- \blacksquare color each compatible tuple of basis relations ${f R}$ using the tuple of colors we assigned to its components.

This graph can clearly be constructed in time $n^{O(k)}$ and has order $n^{O(k)}$.

Because the fibers of \mathcal{C} have order at most c, there can be at most c^k basis relations sharing the same color. Thus, the number of compatible tuples of basis relations sharing a color is bounded by $(c^k)^k = c^{k^2} \in O_{k,c}(1)$. Moreover, its automorphism group is as required: each automorphism is determined by how it permutes the basis relations, the edge-colors ensure that this automorphism is an algebraic automorphism, and the vertex-colors ensure strictness.

As the collection of those strict algebraic automorphisms which are induced by combinatorial ones forms a subgroup of $\mathbb{A}(\mathcal{C})$, it now suffices to check whether each of the polynomially many elements of the generating set of $\mathbb{A}(\mathcal{C})$ is induced by a combinatorial automorphism. This can also be checked in polynomial time:

- ▶ **Lemma 25.** There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k-ary coherent configuration \mathcal{C} with c-bounded fibers, and a strict algebraic automorphism $f \in \mathbb{A}(\mathcal{C})$, outputs whether f is induced by a combinatorial automorphism.
- **Proof.** Let $\mathfrak C$ be an arbitrary colored variant of $\mathcal C$ and $\mathfrak C^f$ another colored variant such that f becomes a color-preserving map between the k-ary color classes of these two structures. Combinatorial automorphisms inducing f now naturally correspond to isomorphisms between $\mathfrak C$ and $\mathfrak C^f$, meaning that f is induced by a combinatorial automorphism if and only if $\mathfrak C \cong \mathfrak C^f$. As $\mathcal C$ has bounded fibers, so does $\mathfrak C$. Thus, we can decide the latter in the required time by Lemma 23.
- ▶ Corollary 26. There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k-ary coherent configuration C with c-bounded fibers, either
- 1. outputs a $\varphi \in \mathbb{A}(\mathcal{C})$ which not induced by a combinatorial automorphism, or
- 2. correctly returns that no such automorphisms exist.

This allows us to finally prove our first main theorem:

- ▶ **Theorem 1.** For every k, there is an algorithm that decides the k-WL-identification problem for vertex- and edge-colored, directed graphs with 5-bounded color classes in time $O_k(n^{O(k)})$. If such a graph G is not identified by k-WL, the algorithm provides a witness for this, i.e., a graph H that is not isomorphic to G and not distinguished from G by k-WL.
- **Proof.** In a first step, we run k-WL on G to get the 2-induced configuration $\mathcal{C} := \operatorname{WL}_k(G)$. By Lemma 6, it remains to decide whether \mathcal{C} is separable. Now, we eliminate disjoint unions of stars using Lemma 14, and Lemma 17, while maintaining 2-inducedness of \mathcal{C} . By Lemma 22, it remains to decide whether every strict algebraic automorphism is induced by a combinatorial one. This can be achieved using Corollary 26.

If this is the case, the input structure is identified by k-WL. Otherwise, we obtain a strict algebraic automorphism f which is not induced by a combinatorial automorphism. By adding back all interspaces containing a disjoint union of stars, we can extend f to an algebraic isomorphism $f \colon \mathrm{WL}_k(G) \to \mathcal{D}$ which is not induced by a combinatorial isomorphism. But then, we can obtain a witnessing graph H from G by replacing its edge set by its $f|_{2}$ -image and similarly translating vertex- and edge-colors along f.

4 Identification for structures with bounded abelian color classes

The approach we used in Section 3 to decide the k-WL-identification problem for graphs with 5-bounded color classes does not easily generalize to graphs with larger color classes or to relational structures of higher arity. In particular, Lemma 22 was crucial in the reduction of k-WL-identification to a statement on strict algebraic automorphisms which could be handled using group-theoretic techniques. The proof of the lemma was based on an explicit case distinction on the possible isomorphism types of interspaces, and fails for graphs with larger color classes. In this section, we show that Lemma 22 remains true in the special case of relational structures with bounded abelian color classes, i.e., structures for which the automorphism group of the structure induced on each color class is abelian. Such structures were already considered in the context of descriptive complexity theory [43] and include both CFI-graphs [11] and multipedes [37, 36] on ordered base graphs.

4.1 Coherent configurations with abelian fibers

To start, we translate the concept of abelian color classes to the corresponding concept of abelian fibers for k-ary coherent configurations.

A combinatorial automorphism φ of a k-ary coherent configuration \mathcal{D} is color-preserving if φ fixes every basis relation of \mathcal{D} . This is equivalent to φ being an automorphism of every colored variant of \mathcal{D} or to the algebraic automorphism induced by φ being the identity (recall that combinatorial automorphisms are not required to fix every basis relation, but only the partition of $V(\mathcal{D})^k$ into basis relations). We say that a coherent configuration \mathcal{C} has abelian fibers if, for each fiber $X \in F(\mathcal{C})$, the group of color-preserving combinatorial automorphisms of $\mathcal{C}[X]$ is abelian.

Lemma 27. Let $\mathfrak A$ be a relational structure of arity at most k. If $\mathfrak A$ has abelian color classes, then $WL_k(\mathfrak{A})$ has abelian fibers.

Recall that an algebraic automorphism of a k-ary coherent configuration \mathcal{C} is strict if it fixes every basis relation inside a fiber. Similarly, we call a combinatorial automorphism φ strict if it fixes every basis relation inside a fiber, i.e., if it fixes every fiber and its restriction to every fiber is color-preserving. This is equivalent to φ inducing a strict algebraic automorphism.

Towards understanding the structure of abelian fibers, we start with one simple grouptheoretic observation.

▶ Lemma 28. Let $\Gamma \subseteq \operatorname{Sym}(\Omega)$ be an abelian group acting transitively on a set Ω . Then the group action is regular.

Proof. Because Γ acts transitively, the stabilizer subgroup of all elements in Ω are pairwise conjugated. But because Γ is abelian, this implies that they are equal. Thus, every permutation that stabilizes some element already stabilizes all, which is only true for the identity.

In order to apply this observation to abelian fibers, we need one more definition. For a fiber $X \in F(\mathcal{C})$, a binary basis relation $S \in \mathcal{C}|_2[X]$ is called *thin* if every vertex in X is incident to exactly one ingoing and exactly one outgoing S-edge, that is, if S is either a matching or a union of directed cycles. The fiber X is called *thin* if all basis relations $R \in \mathcal{C}|_2[X]$ are thin and if this is true for all fibers of \mathcal{C} , we say that \mathcal{C} has *thin fibers*.

▶ Corollary 29. Let C be a k-ary coherent configuration. Then every abelian fiber of order at most k is thin.

Proof. Let $X \in F(\mathcal{C})$ be an abelian fiber of order at most k. Then $\mathcal{C}|_2[X]$ is the partition of X^2 into orbits under the natural action of the group of color-preserving automorphisms.

Pick some $x \in X$ and assume that some binary basis relation $S \in \mathcal{C}|_2[X]$ contains two pairs xy and xy' for $y, y' \in X$. But this implies that there is a color-preserving automorphism φ of $\mathcal{C}[X]$ that maps xy to xy'. But as the group of color-preserving automorphism of $\mathcal{C}[X]$ is abelian and acts transitvely, $\varphi(x) = x$ implies $y' = \varphi(y) = y$. Thus, the basis relation S is thin.

Finally, we need one well-known lemma on the structure of thin fibers, which essentially states that thin fibers correspond to Cayley graphs of their automorphism groups.

▶ **Lemma 30** ([12, Section 2.1.4]). Let C be a 2-ary coherent configuration on a single thin fiber. Then the basis relations of C are precisely those of the form $S_{\varphi} := \{x\varphi(x) : x \in V(C)\}$ for color-preserving combinatorial automorphisms φ of C.

4.2 Separability of configurations with bounded thin fibers

Let C be a k-ary coherent configuration. Recall that in Section 3.1, we defined for every functional basis relation, that is, a basis relation with out-degree at most 1 at every vertex of C, a map

$$\nu_S \colon V(\mathcal{C}) \to V(\mathcal{C}),$$

$$v \mapsto \begin{cases} w & \text{if } vw \notin S, \\ v & \text{if no such } w \text{ exists.} \end{cases}$$

and, for $I \subseteq [k]$, further the maps $\nu_S^I \colon V(\mathcal{C})^k \to V(\mathcal{C})^k$ which act as ν_S on all components in I and as the identity on all components not in I.

Because every thin basis relation S lies within a single fiber, every vertex of that fiber also has in-degree exactly 1 with respect to S. This means that not only is every thin basis relation functional, but the maps ν_S are bijective in this case. Thus, the following lemma is an immediate consequence of Lemma 12.

▶ Lemma 31. For every k-ary coherent configuration C, every thin basis relation $S \in C|_2[X]$, and every $I \subseteq [k]$, the map ν_S^I induces an algebraic automorphism of C. Furthermore, for every algebraic isomorphism $f: C \to D$, we get $f \circ \nu_S^I = \nu_{f(S)}^I \circ f$.

Next, we show that k-ary coherent configurations with few, thin fibers are separable:

▶ **Lemma 32.** Let C be a k-ary coherent configuration with at most k fibers. If C has thin fibers, then C is separable.

Proof. Let \mathfrak{C} be a colored variant of \mathcal{C} , where we additionally also add all thin basis relations within each fiber as binary basis relations inside the color classes. Then $\mathcal{C} = \mathrm{WL}_k(\mathfrak{C})$, which, using Lemma 6, implies that \mathcal{C} is separable if and only if \mathfrak{C} is identified by k-WL.

Thus, assume $\mathfrak{C} \equiv \mathfrak{D}$. We argue using the bijective (k+1)-pebble game that $\mathfrak{C} \cong \mathfrak{D}$. We start by considering a winning position $x \mapsto y$ for Duplicator in the bijective (k+1)-pebble game between $\mathfrak C$ and $\mathfrak D$ with only a single placed pebble pair. Assume Spoiler picks up a second pebble pair and let $f: V(\mathfrak{C}) \to V(\mathfrak{D})$ be the bijection that Duplicator provides according to their winning strategy. Note that for every thin basis relation S in the color class of x, the map f must map the unique S-neighbor of x to the unique S-neighbor of y, which completely determines the map f on the color class of x. If this is not the case, Spoiler can place pebbles on this vertex pair and wins in the next round. Further note that for every different vertex x' in the color class of x', this same bijection on the color class must also be winning in position $xx' \mapsto yf(x')$ and thus also in position $x' \mapsto f(x')$. Thus, as long as Spoiler never picks up all pebbles from this color class, the bijections provided by Duplicator stay fixed on this color class.

Now, assume Spoiler places a pebble in every color class of $\mathfrak C$ and thus reaches a position $\mathbf{x} \mapsto \mathbf{y}$, which is still winning for Duplicator. Again, let f be the bijection that Duplicator provides when Spoiler picks the last remaining pebble pair.

We argue that f is an isomorphism, by showing that for every k-tuple of vertices $\mathbf{z} \in V(\mathfrak{C}^k)$, Spoiler can force the game to reach the position $\mathbf{z} \mapsto f(\mathbf{z})$. Because Duplicator has a winning strategy, these positions must be partial isomorphisms, which then implies that f is indeed an isomorphism.

By our previous remarks, it suffices to observe that Spoiler can clearly pebble all vertices in z without ever removing all pebbles from some color class that contains some vertex from z. This way, the bijections provided by Duplicator must always agree with f on all color classes containing a vertex from \mathbf{z} , which proves the claim.

Finally, we are ready to once again reduce the question of separability to only strict algebraic automorphisms, which we can again deal with using Corollary 26.

▶ Lemma 33. Let C be a k-ary coherent configurations with thin fibers. Then C is separable if and only if every strict algebraic automorphism of C is induced by a combinatorial automorphism.

Proof. The forward implication is immediate, so it suffices to show the backward implication. So assume that every strict algebraic automorphism of \mathcal{C} is induced by a combinatorial automorphism and let $f: \mathcal{C} \to \mathcal{D}$ be an algebraic isomorphism. We need to show that f is induced by a combinatorial isomorphism.

By Lemma 32, f is induced by a combinatorial isomorphism $\psi_{\mathbf{X}}: \mathcal{C}[\mathbf{X}] \to \mathcal{D}[f(\mathbf{X})]$ on every union $\mathbf{X} = X_1 \cup \cdots \cup X_k$ of k fibers and thus in particular by a combinatorial isomorphism $\varphi_X \colon \mathcal{C}[X] \to \mathcal{D}[f(X)]$ on every single fiber. We define a bijection $\varphi \colon V(\mathcal{C}) \to V(\mathcal{D})$ by setting $\varphi|_X := \varphi_X$ for every fiber $X \in F(\mathcal{C})$ and claim that φ is a combinatorial isomorphism that induces f on every fiber.

Clearly, $\varphi|_X$ is a combinatorial isomorphism inducing f for every fiber $X \in F(\mathcal{C})$, hence it remains to show that φ is also a combinatorial isomorphism on the whole configuration \mathcal{C} . Because every basis relation $R \in \mathcal{C}$ is contained in a subconfiguration $\mathcal{C}[\mathbf{X}]$ induced on the union of k fibers $\mathbf{X} = X_1 \cup \cdots \cup X_k$, it suffices to show that $\varphi|_{\mathbf{X}} \colon \mathbf{X} \to f(\mathbf{X})$ is a combinatorial isomorphism from $\mathcal{C}[\mathbf{X}]$ to $\mathcal{D}[f(\mathbf{X})]$ for every such \mathbf{X} .

For this, we first note that the map $\psi_{\mathbf{X}}$ is such a combinatorial isomorphism, which implies that for every combinatorial automorphism θ of $\mathcal{C}[\mathbf{X}]$, the map $\psi_{\mathbf{X}} \circ \theta$ is such a combinatorial isomorphism as well. Thus, it would suffice to prove that the permutation $\psi_{\mathbf{X}}^{-1} \circ \varphi|_{\mathbf{X}}$ is a combinatorial automorphism of $\mathcal{C}[X]$. For this, note since both $\psi_{\mathbf{X}}$ and $\varphi|_{\mathbf{X}}$ induce f on every fiber, the composition $\psi_{\mathbf{X}}^{-1} \circ \varphi|_{\mathbf{X}}$ is a color-preserving automorphism restricted to every fiber $X_i \subseteq \mathbf{X}$.

By Lemma 30, this implies that for every fiber $X_i \subseteq \mathbf{X}$, there is a thin basis relation $S_i \in \mathcal{C}|_2[X_i]$ such that $\psi_{\mathbf{X}}^{-1} \circ \varphi|_{X_i} = \nu_{S_i}|_{X_i}$. Thus, we can write

$$\psi_{\mathbf{X}}^{-1} \circ \varphi|_{\mathbf{X}} = \prod_{i=1}^k \nu_{S_i},$$

which is a composition of combinatorial automorphisms and thus also a combinatorial automorphism. Thus, we get that $\varphi|_{\mathbf{X}} = \psi_{\mathbf{X}} \circ (\psi_{\mathbf{X}}^{-1} \circ \varphi|_{\mathbf{X}})$ is indeed a combinatorial isomorphism for every union of k fibers \mathbf{X} , which implies that φ is a combinatorial isomorphism which induces f on every fiber.

Finally, it follows that $\varphi^{-1} \circ f$ is a strict algebraic automorphism which, by assumption, is induced by a combinatorial automorphism θ . But then, $\varphi \circ \theta$ induces f.

▶ **Theorem 2.** For every $k \in \mathbb{N}$ and $c, r \leq k$, there is an algorithm that decides the k-WL-identification problem for r-ary relational structures with c-bounded abelian color classes in time $O_k(n^{O(k)})$. If such a structure \mathfrak{A} is not identified by k-WL, the algorithm provides a witness for this, i.e., a second structure \mathfrak{B} that is not isomorphic to \mathfrak{A} and not distinguished from \mathfrak{A} by k-WL.

Proof. Let \mathfrak{A} be a relational structure of arity r. Then \mathfrak{A} is identified by k-WL if and only if $\operatorname{WL}_k(\mathfrak{A})$ is separable. Because $\operatorname{WL}_k(\mathfrak{A})$ has c-bounded thin fibers by Lemma 27 and Lemma 29, Lemma 33 implies that separability of $\operatorname{WL}_k(\mathfrak{A})$ is equivalent to every strict algebraic automorphism of $\operatorname{WL}_k(\mathfrak{A})$ being induced by a combinatorial automorphism. This can be checked in the given time using Corollary 26, and in case of a negative answer, we can construct a non-isomorphic but non-distinguished structure from the strict algebraic automorphism not induced by a combinatorial one as in Theorem 1.

Note that the restriction to relational structures of arity at most k is insubstantial, because the standard variant of the Weisfeiler-Leman algorithm given in Section 2 does not identify any relational structure of arity larger than k, simply because it does not consider tuples of length larger than k and thus cannot even detect whether a relation of arity larger than k is empty. While there are variants of k-WL which identify some (k+1)-ary relational structures, these variants can be treated similarly to decide identification by those algorithms.

5 Hardness

In this section, we prove hardness results that complement the positive results in the previous two sections. We start by showing that, when the dimension k is considered part of the input, the k-WL-equivalence problem and the k-WL-identification problem are co-NP-hard and NP-hard, respectively. This is achieved via reductions from TREE-WIDTH, which is NP-hard even over cubic graphs [9]. The reduction is based on the CFI-construction, see Section 2.

▶ **Theorem 3.** The problem of deciding, given a graph G and a natural number k, whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, both over uncolored simple graphs, and over simple graphs with 4-bounded abelian color classes.

Proof. TREE-WIDTH is the problem to decide whether the tree-width of a given graph G is at most a given number k. This problem is NP-hard even over cubic graphs [9]. By Lemma 8, we have $\operatorname{tw}(G) = \operatorname{WL-dim}(\operatorname{CFI}(G,0))$. Thus, computing the tree-width of a cubic graph G reduces to computing the Weisfeiler-Leman dimension of its CFI-graph. Because the CFI-graphs of cubic graphs have 4-bounded color classes, and CFI-graphs of cubic graphs can be efficiently computed, the hardness result for graphs with 4-bounded color classes follows.

The claim for the class of simple graphs follows from the observation that we can encode colors into gadgets we attach to every vertex, and that these gadgets can be chosen such that they do not affect the Weisfeiler-Leman dimension.

Because in the above proof, we could explicitly construct a non-isomorphic but equivalent graph, this also yields co-NP-hardness of the k-WL-equivalence problem, which was also independently observed in [40].

▶ **Theorem 34.** The problem of deciding, for a given pair of graph G and H and a natural number $k \ge 1$, whether $G \equiv_{k\text{-WL}} H$, is co-NP-hard.

Now, we once again turn to the k-WL-identification problem for a fixed dimension $k \geq 2$, and show that both over uncolored simple graphs, and over simple graphs with 4-bounded color classes, the problem is P-hard under logspace-uniform AC_0 -reductions.

We reduce from the P-hard monotone circuit value problem MCVP [17]. Our construction of a graph from a monotone circuit closely resembles the reductions of Grohe [19] to show P-hardness of the k-WL-equivalence problem. A similar reduction was also used to prove P-hardness of the identification problem for the color refinement algorithm (1-WL) [2].

The reduction based one so-called *one-way switches*, which were introduced by Grohe [19]. These graph gadgets allow color information computed by the Weisfeiler-Leman algorithm to pass in one direction, but block it from passing in the other. And while Grohe provides one-way switches for every dimension of the Weisfeiler-Leman algorithm, his gadgets have large color classes and are difficult to analyze. Instead, we give a new construction of such gadgets with 4-bounded color classes based on the CFI-construction. We then use these one-way switches to construct a graph from an instance of the monotone circuit value problem from the identification of which we can read off the answer to the initial MCVP-query.

5.1 One-way switches

In the following sections, we fix a dimension $k \geq 2$ of the Weisfeiler-Leman algorithm. A k-one-way switch is a graph gadget with a pair of input vertices $\{y_1, y_2\}$, and a pair of output vertices $\{x_1, x_2\}$, which each form a color class of size 2. We say that a pair of vertices is split if the two vertices are colored differently. We say that k-WL splits a pair if the coloring computed by k-WL splits the pair.

The fundamental property of the k-one-way switch is the following: whenever the input pair $\{y_1, y_2\}$ of the one-way switch is split, k-WL also splits the output pair, but not the other way around. One-way switches thus only allow one-way flow of k-WL-color information. In contrast to Grohe's gadgets, our one-way switches are based on the CFI-construction, see Section 2.

We start by defining our base graph. Consider a wall graph consisting of k-1 rows of k bricks each. Then, we attach a new vertex v to the two upper corner vertices of the first row. The resulting graph B_k is depicted in Figure 4.

▶ **Lemma 35.** The graph B_k has tree-width k+1, while B_k-v has tree-width k.

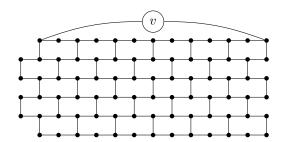


Figure 4 The base graph B_6 of the CFI-graphs underlying our one-way switches.

Proof. Because the tree-width of a graph is invariant under subdividing edges, we can delete v and directly connect the two upper corner vertices of B_k . Let B'_k be the resulting graph.

For every positive integer ℓ , let $G'_{k,\ell}$ be the graph obtained from a $k \times \ell$ -grid graph by directly connecting two corners that lie on a common side of length ℓ . Then $G'_{k,k+1}$ is a minor of B'_k and thus $\operatorname{tw}(B'_k) \geq \operatorname{tw}(G'_{k,k+1})$. Similarly, up to a subdivision of some edges, which does not affect the tree-width, B'_k is a subgraph of $G'_{k,2k+1}$. Thus,

$$\operatorname{tw}(G'_{k,k+1}) \le \operatorname{tw}(B'_k) \le \operatorname{tw}(G'_{k,2k+1}).$$

It thus suffices to argue that $\operatorname{tw}(G'_{k,\ell}) = k+1$ for all $\ell > k$. Because adding a single vertex or edge to a graph increases the tree-width by at most 1, and the $k \times \ell$ -grid graph has tree-width $\min(k,\ell)$, we have $\operatorname{tw}(B_k - v) = k$ and $\operatorname{tw}(G'_{k,\ell}) \le k+1$.

For the matching lower bound, we use the theory of brambles [41]. Recall that a bramble \mathcal{B} of a finite graph H is a collection of connected subgraphs of H such that every two subgraphs in \mathcal{B} either overlap, or are connected by an edge. The order of a bramble is the size of a minimal hitting set, i.e., the smallest $k \in \mathbb{N}$ such that there exist k vertices v_1, \ldots, v_k such that every set $B \in \mathcal{B}$ contains one of the vertices v_i . It turns out that the maximal order of a bramble in H is precisely one more than the tree-width of H [41].

For the matching lower bound, it thus suffices to construct a bramble in $G'_{k,\ell}$ of order at least k+2. Let the vertex set of $G'_{k,\ell}$ be $[k] \times [\ell]$, with the extra edge connecting (1,1) to $(1,\ell)$. For each $i \in [k] \setminus \{k\}$ and $j \in [\ell] \setminus \{\ell\}$, we define a set

$$U_{i,j} := \{(a,b) \in ([k] \setminus \{k\}) \times ([\ell] \setminus \{\ell\}) : a = i \text{ or } b = j\}$$

to be the union of the i-th row and j-th column, while excluding the vertices from the last row or column. Further, we take the two sets

$$R_k := \{k\} \times [\ell] \quad \text{and} \quad C_\ell := ([k] \setminus \{1, k\}) \times \{\ell\}.$$

Finally, we define our bramble to be

$$\begin{aligned}
& \left\{ R_k, C_\ell \right\} \\
& \cup \left\{ U_{i,j} \colon i \in [k] \setminus \{1, k\}, j \in [\ell] \setminus \{\ell\} \right\} \\
& \cup \left\{ \left(U_{1,j} \cup \{(1, \ell)\} \right) \setminus \{(1, i)\} \colon j \in [\ell] \setminus \{\ell\}, i \in [\ell] \setminus \{j\} \right\}.
\end{aligned}$$

To argue that this bramble has order at least k + 2, we need to show that it does not admit a hitting set of order at most k + 1, that is, we must argue that for every choice of k + 1

vertices there exists some set in the bramble which contains none of the vertices. Assume for contradiction that $H \subseteq V(G'_{k,\ell})$ was such a hitting set. Because the sets R_k and C_ℓ are disjoint from all other sets, H must contain at least one vertex in each of these sets. Hence, only k-1 vertices remain to hit all other sets in the bramble. These k-1 vertices must miss one of the first $\ell-1$ columns, say column j. If they also miss one of the first k-1 rows, say row i, then they miss the set $U_{i,j}$. Otherwise, there is precisely one vertex per row. Let (1,i) be the vertex in the first row. But then the set misses $(U_{1,i} \cup \{(1,\ell)\}) \setminus \{(1,i)\}$.

Now, we are ready to construct our one-way switches.

- ▶ **Lemma 36** (compare [19, Lemma 14]). For every $k \ge 2$, there is a colored graph O^k with 4-bounded color classes, called k-one-way switch, with an input pair $\{y_1, y_2\}$ and an output pair $\{x_1, x_2\}$ satisfying the following properties:
- 1. The graph O_{split}^k obtained by splitting the input pair $\{y_1,y_2\}$ is identified by k-WL.
- **2.** k-WL splits the output pair $\{x_1, x_2\}$ of O_{split}^k .
- **3.** There is no automorphism of O^k exchanging the output vertices x_1 and x_2 . Furthermore, there are sets of positions in the bijective (k+1)-pebble game between O^k and itself, called trapped and twisted such that
- **4.** every trapped or twisted position is a partial isomorphism,
- 5. Duplicator can avoid non-trapped positions from trapped ones and non-twisted positions from twisted ones,
- **6.** for every trapped position $\mathbf{a} \mapsto \mathbf{b}$, the position $\mathbf{a}x_1 \mapsto \mathbf{b}x_1$ is also trapped, ¹
- **7.** for every twisted position $\mathbf{a} \mapsto \mathbf{b}$, the position $\mathbf{a} x_1 \mapsto \mathbf{b} x_2$ is also twisted,
- **8.** the positions $y_1y_2 \mapsto y_1y_2$ and $y_1y_2 \mapsto y_2y_1$ are both trapped and twisted,
- 9. every subposition of a trapped position is trapped, and every subposition of a twisted position is twisted

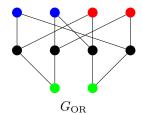
Proof. Let O^k be the (untwisted) CFI-graph of B_k , but with a CFI-gadget of degree 3 added for the vertex v instead of a gadget of degree 2. This leaves one outer pair of this gadget free which we use as our output pair $\{x_1, x_2\}$. Furthermore, we use one of the other two outer pairs of this same CFI-gadget as the input pair $\{y_1, y_2\}$.

Now, if we fix the output pair $\{x_1, x_2\}$ by individualizing one of the two vertices, the resulting graph corresponds to the usual CFI-graph of H, while switching the pair $\{x_1, x_2\}$ corresponds to the twisted CFI-graph of H. In particular, as these graphs are not isomorphic, there is no automorphism of O^k switching the pair $\{x_1, x_2\}$, which proves Property 3.

Moreover, splitting the input pair $\{y_1, y_2\}$ has the same effect to the power of k-WL as removing one of the two edges incident to v in the base graph B_k has. When removing this edge in the base graph, the resulting graph is essentially equivalent to the CFI-graph of the $k \times (k+1)$ -wall graph with one corner vertex replaced by a CFI-gadget of degree 3 instead of 2. Because exchanging the two vertices of the free outer pair of this degree-3 gadget interchanges the twisted and untwisted CFI-graphs over the base graph, and k-WL can distinguish CFI-graphs from all other graphs, the resulting graph is identified by k-WL. This proves Property 1.

To show Property 2, we start the bijective (k+1)-pebble game in position $x_1 \mapsto x_2$. Then, Spoiler uses the usual strategy of pebbling a wall which they then move from one side

If the position $\mathbf{a}x_1 \mapsto \mathbf{b}x_1$ contains more than k+1 pebbles, this means that every subposition on at most k+1 pebbles is trapped



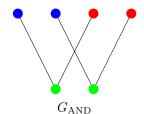


Figure 5 The gadgets G_{OR} and G_{AND} encoding OR- and AND-gates. We call the two pairs at the top their *input pairs* and their bottom pair their *output pair*.

of the wall graph to the other. But because the game started in position $x \mapsto x'$, the two graphs the game is played on differ in a twist which will finally force Duplicator to lose.

Now, consider again the original graph O^k without splitting the input pair. On this graph, we can extend every winning position for Duplicator in the bijective k-pebble game between the untwisted CFI-graph CFI(B_k , 0) and the twisted CFI-graph CFI(B_k , 1) to a position in the bijective k-pebble game between O^k and itself which is compatible with $x_1 \mapsto x_2$. Similarly, we can extend every winning position for Duplicator in the bijective k-pebble game between the untwisted CFI-graph CFI(B_k , 0) and itself to a position between O^k and itself which is compatible with $x_1 \mapsto x_1$.

We call the former positions *twisted* and the latter positions *trapped*. Properties 4, 6, 7 and 9 are then immediate, and Property 5 follows from Lemma 8 together with Lemma 35.

Because v lies on a cycle in B_k , there exists an automorphism of $CFI(B_k)$ which twists both outer pairs of the gadget corresponding to v. Lifting this automorphism to O^k yields an automorphism switching y_1 and y_2 whilst fixing x_1 and x_2 . This proves Property 8.

In the bijective (k+1)-pebble game on O^k , we say that Duplicator follows a trapped or twisted strategy if Spoiler can never reach a non-trapped or non-twisted position respectively. Note that Properties 4 and 5 together imply that trapped and twisted strategies are winning strategies.

5.2 From monotone circuits to graphs

We now reduce the monotone circuit value problem MCVP to the k-WL-identification problem. A monotone circuit M is a circuit consisting of input nodes, each of which has value either True or False, a distinguished output node, and inner nodes, which are either AND- or Ornodes with two inputs each. We write V(M) for the set of nodes of M. With the monotone circuit M, we can associate the evaluation function $\operatorname{val}_M \colon V(M) \to \{\text{True}, \text{False}\}$, which is defined in the obvious way. The monotone circuit value problem MCVP is the following problem: given a monotone circuit M with output node c, decide whether $\operatorname{val}_M(c) = \operatorname{True}$. This problem is known to be hard for polynomial time [17].

Now, let M be such a monotone circuit. We construct a colored graph G_M such that for every node $a \in V(M)$, there is a vertex pair $\{a_1, a_2\}$ in G_M which will be split by k-WL if and only if $\operatorname{val}_M(a) = \operatorname{False}$. We use the two graphs G_{OR} and G_{AND} in Figure 5 as gadgets to replace the logic gates in our construction of G_M . These gadgets both have two input pairs and one output pairs such that exchanging the two output vertices by an automorphism requires the two vertices of one (for G_{OR}) or both (for G_{AND}) input pairs to also be exchanged.

We now start with the formal construction of G_M , which is depicted in Figure 6. For every node a of M, we add a pair of vertices $\{a_1, a_2\}$ forming a color class to G_M . To encode

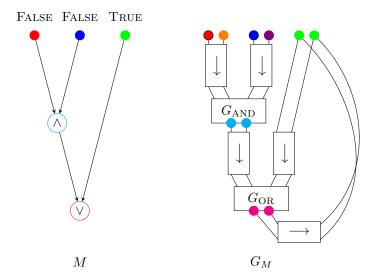


Figure 6 A simple monotone circuit M and the graph G_M obtained from it. The colors in M are just for illustration purposes and not part of the actual circuit.

the input valuation of the circuit, we split every pair $\{a_1, a_2\}$ corresponding to an input node a of value FALSE.

For every AND-node $a \in V(M)$ with input nodes b and b', we add a freshly colored copy of the gadget G_{AND} from Figure 5 and identify its output pair with the pair $\{a_1, a_2\}$. Next, we connect its two input pairs via freshly colored one-way switches O_{ba}^k and $O_{b'a}^k$ to the pairs $\{b_1, b_2\}$ and $\{b'_1, b'_2\}$ respectively. More precisely, we identify the input pairs of these one-way switches with $\{b_1, b_2\}$ or $\{b'_1, b'_2\}$ respectively and identify their output pair with the respective input pair of the copy of G_{AND} . Analogously, we add a copy of G_{OR} for every OR-node $a \in V(M)$ and connect it to its input via one-way switches as before.

This concludes the translation of the circuit itself, but for our reduction to the identification problem we need one more step: we connect all input pairs $\{a_1, a_2\}$ with $\operatorname{val}_M(a) = \operatorname{True}$ to the output pair $\{c_1, c_2\}$ via additional one-way switches O_{ca}^k , whose input pair we identify with $\{c_1, c_2\}$ and whose output pair we identify with $\{a_1, a_2\}$.

Let G_M be the resulting graph. Because we color different gadgets using distinct colors, and every gadget has 4-bounded color classes, the resulting graph G_M also has 4-bounded color classes, which could be made abelian by introducing colored edges within the gadgets. Indeed, note that every pair of vertices in a color class of order 4 has precisely one common neighbor. If we connect each such pair by an edge whose color encodes the color of the common neighbor, the resulting color class is abelian. Note further that this coloring is computed by 2-WL, which means that adding it does not affect the power of the Weisfeiler-Leman algorithm on these graphs.

Moreover, the graph G_M can be constructed in AC_0 because we only need to replace every node and edge of M by a gadget of bounded size that only depends on whether the node is an input node, an AND-node or an OR-node.

Recall that G_M contains, for every node a of M, a vertex pair $\{a_1, a_2\}$. We first prove that the vertex pairs of all nodes evaluating to FALSE are indeed split by k-WL.

▶ Lemma 37 (compare [19, Section 5.4], [2, Proof of Theorem 7.11]). For every monotone circuit M with output node c, and every node a of M, if either $val_M(a) = FALSE$ or $val_M(c) = FALSE$, then $(G_M, a_1) \not\equiv_{k\text{-WL}} (G_M, a_2)$

Proof. We split the proof into two claims.

```
\triangleright Claim 38. If \operatorname{val}_M(a) = \operatorname{FALSE}, then (G_M, a_1) \not\equiv_{k\text{-WL}} (G_M, a_2).
```

Proof. We argue by induction on the depth of a in the circuit M. If a is an input node, then a_1 and a_2 have different colors and are thus distinguished by construction. Thus, assume that a is an inner node with parents b and b' and assume by induction that the claim is true for both b and b'.

If a is an AND-node, assume w.l.o.g. that $\operatorname{val}_M(b) = \operatorname{False}$. By the induction hypothesis, $(G_M, b_1) \not\equiv_{k\text{-WL}} (G_M, b_2)$. But this means that the input pair of the one-way switch O_{ba}^k is split, which implies by Property 2 of the one-way switch that also the output pair is split. But this output pair is one of the input pairs of the AND-gadget G_{AND} , whose output pair is $\{a_1, a_2\}$. Thus, k-WL also splits the pair $\{a_1, a_2\}$.

If a is an OR-node, then $\operatorname{val}_M(b) = \operatorname{val}_M(b') = \operatorname{False}$. Thus, by the induction hypothesis, both of the pairs $\{b_1, b_2\}$ and $\{b'_1, b'_2\}$ are split by k-WL. Using Property 2 of the one-way switches O_{ba}^k and $O_{b'a}^k$, this means that k-WL also splits both input pairs of the OR-gadget G_{OR} whose output pair is $\{a_1, a_2\}$. However, all inner vertices of the gadget G_{OR} are uniquely determined by their neighborhood among the input pairs, and the two vertices of the output pair have distinct neighborhoods among these inner vertices. Thus, k-WL also splits the pair $\{a_1, a_2\}$.

 \triangleright Claim 39. If $\operatorname{val}_M(c) = \operatorname{False}$, then for all nodes a of M it holds that $(G_M, a_1) \not\equiv_{k\text{-WL}} (G_M, a_2)$.

Proof. Using the previous claim, we get that k-WL splits the pair $\{c_1, c_2\}$. Using the one-way switches O_{ca}^k connecting the pair $\{c_1, c_2\}$ to all pairs $\{a_1, a_2\}$ corresponding to input nodes of value True, this means that k-WL now splits all pairs corresponding to input nodes by Property 2 of these one-way switches. Thus, the graph now simulates the circuit with all input nodes set to False. But as the circuit is monotone, this means that also all inner nodes now evaluate to False. By the previous claim, k-WL thus also splits all pairs $\{a_1, a_2\}$ corresponding to inner nodes.

Combining the two claims finishes the proof.

Next, we prove the converse: Nodes of value True get interpreted as vertex pairs which are not split by k-WL.

▶ Lemma 40 (compare [19, Section 5.4], [2, Proof of Theorem 7.11]). For every monotone circuit M with output node c, and every node $a \in V(M)$, if $\operatorname{val}_M(a) = \operatorname{True}$ and $\operatorname{val}_M(c) = \operatorname{True}$, then $(G_M, a_1) \equiv_{k\text{-WL}} (G_M, a_2)$.

Proof. To prove the desired equivalences, we provide winning strategies for Duplicator in the bijective (k+1)-pebble game on G_M . We call a vertex pair $\{v_1, v_2\}$ in G_M true if it either corresponds to a node v of M with $\operatorname{val}_M(v) = \operatorname{True}$, or is the output pair of a one-way switch whose input pair corresponds to a node of value True . Otherwise, we call $\{v_1, v_2\}$ false. We call a one-way switch in G_M true if its input pair is a true pair, and false if its input pair is a false pair.

Now, we call a position P in the bijective k+1-pebble game on G_M safe, if there is an automorphism φ of the subgraph induced by all vertices of G_M besides the inner vertices of one-way switches satisfying the following conditions:

1. The position $P \cup \varphi$, that is, the position obtained from P by adding all pebble pairs $x \mapsto \varphi(x)$ for $x \in \text{dom}(\varphi)$, is a partial isomorphism.

- 2. For every false one-way switch O, φ fixes both input and both output vertices of O, and the restriction $P|_O$ of the position P to vertices of O is the identity map.
- **3.** One of the following is true:
 - a. There is a single true one-way switch O with output pair $\{x_1, x_2\}$ that contains all k+1 pebble pairs, and either P is trapped and φ fixes both x_1 and x_2 , or P is twisted and φ exchanges x_1 and x_2 .
 - b. Every true one-way switch contains at most k pebble pairs, and for every true one-way switch O with input pair $\{y_1, y_2\}$ and output pair $\{x_1, x_2\}$, either $P|_O \cup \{y_1 \mapsto \varphi(y_1)\}$ is trapped and φ fixes both x_1 and x_2 , or $P|_O \cup \{y_1 \mapsto \varphi(y_1)\}$ is twisted and φ exchanges x_1 and x_2 .

We call a position *dangerous* if it is not safe.

ightharpoonup Claim 41. For every node $a \in V(M)$ with $\mathrm{val}_M(a) = \mathrm{True}$, the position $a_1 \mapsto a_2$ is safe. Moreover, the witnessing automorphism φ can be chosen such that it is the identity on every gadget but the gadget corresponding to the node a.

Proof. Let a be an arbitrary node of C. If a is an input node, let φ be the map that exchanges a_1 and a_2 and fixes every other vertex. This clearly satisfies Conditions 1 and 2. Condition 3 is trivially true, because every color-preserving permutation of the input- and output vertices of a one-way switch is either twisted or trapped by Properties 6, 7 and 8 of the one-way switches.

If a is an AND-node, pick φ to be the unique non-trivial automorphism on the corresponding AND-gadget G_{AND} , and the identity everywhere else. Again, all three conditions are clearly satisfied in this case because both parents of a also have value True. If a is an OR-node, let b be a parent node of a with $\text{val}_M(b) = \text{True}$, and let φ be the unique automorphism that exchanges a_1 and a_2 , as well as b_1 and b_2 on the corresponding OR-gadget G_{OR} , and that is the identity everywhere else. Again, all three conditions are clear by construction.

 \triangleright Claim 42. Every subposition of a safe position P is safe.

Proof. Assume P is a safe position and let φ be the automorphism from the definition of safeness for P.

If P satisfies Condition 3b, then φ also witnesses that every subposition of P is safe. Indeed, because subpositions of partial isomorphisms are again partial isomorphisms, Condition 1 is clear. Similarly, Condition 2 is true because restrictions of the identity are again the identity on the restricted domain. Condition 3b follows from the observation that every subposition of a trapped position is trapped, and every subposition of a twisted position is twisted by Property 9 of the one-way switches.

If P satisfies Condition 3a instead, then no proper subposition of P does also satisfy Condition 3a. Let O be the unique true one-way switch that contains all k+1 pebble pairs, and let $P_0 \subsetneq P$ be a proper subposition. By Condition 3a of P, the position P and thus also P_0 is either trapped or twisted. In order to prove Condition 3b for the position P_0 , we need to show that we can extend this trapped or twisted position by the vertex y_1 . For this, consider the bijective (k+1)-pebble game in position P_0 on O. Assume Spoiler picks up an unused pebble pair, and let $\psi \colon V(O) \to V(O)$ be the bijection that Duplicator responds with when following a trapped or twisted strategy. Then the position $P_0 \cup \{y_1 \mapsto \psi(y_1)\}$ is again trapped or twisted. Thus, in order to ensure Condition 3b for P_0 , it remains to construct the automorphism φ_0 for P_0 such that $\varphi_0(y_1) = \psi(y_1)$.

To construct this automorphism, we start with the automorphism φ that witnessed safeness of P. The position P_0 together with φ satisfies Condition 1. If $\varphi(y_1) = \psi(y_1)$, we

are thus done. Hence, assume from now on that $\varphi(y_1) \neq \psi(y_1)$. Because the input pair $\{y_1, y_2\}$ corresponds to a node of value True, Claim 41 gives us a second automorphism θ_y which exchanges y_1 and y_2 , and is the identity on every gadget besides the input-, AND- or OR-gadget containing y_1 and y_2 . We set $\varphi_0 := \theta_y \circ \varphi$. Then φ_0 is an automorphism satisfying $\varphi_0(y_1) = \psi(y_1)$, which furthermore agrees with φ everywhere but on the gadget containing y_1 and y_2 . Moreover, by Condition 2 applied to the position $y_1 \mapsto y_2$ and θ , the automorphism φ_0 disagrees with φ only on y_1 and y_2 , where $P_0 \cup \varphi_0$ is a partial isomorphism by construction, and possibly on output pairs of one-way switches whose input pairs correspond to a node of value True. But for these one-way switches, we can independently exchange or fix the two vertices of the input and the output pair while keeping the resulting position either trapped or twisted using Properties 6, 7 and 8 of the one-way switches. This proves Conditions 1 and 2.

 \triangleright Claim 43. Duplicator can avoid dangerous positions from safes ones in the bijective (k+1)-pebble game on G_M .

Proof. Let P be a safe position and φ the witnessing automorphism. As subpositions of safe positions are again safe, it suffices to assume that P contains at most k of the possible k+1 pebble pairs. Now, if a single true one-way switch O contains all k pebbles, then the position $P = P|_O$ is trapped or twisted, and Duplicator picks a bijection $\psi|_O$ according to a trapped or twisted strategy.

If $\psi|_O$ and φ agree on the input pair $\{y_1, y_2\}$ of O, then $\psi|_O$ is compatible with φ . Otherwise, we again pick the automorphism θ_y that Claim 41 yields for the position $y_1 \mapsto y_2$ and replace φ by $\theta_y \circ \varphi$, which is now compatible with $\psi|_O$.

As all other one-way switches O' do not contain any pebble pairs on inner vertices, and the position P is safe, the restrictions $\varphi|_{O'}$ are the identity on all false one-way switches, and either trapped or twisted on all true one-way switches by Properties 6, 7 and 8. We choose $\psi|_{O'}$ according to a trapped, twisted or identity strategy on these one-way switches. Combining the maps $\psi|_{O'}$ for all one-way switches with φ yields a total bijection ψ , which Duplicator can choose.

Now, if Spoiler places the free pebbles on vertices x and $\psi(x)$, we need to argue that the resulting position is still safe. However, Condition 1 is true because ψ was chosen according to a winning strategy on each gadget, and these strategies are compatible at the intersections of the gadgets. Conditions 2 and 3 are true by construction. Thus, the position $P \cup \{x \mapsto \psi(x)\}$ is again safe.

Assume now that every true one-way switch of G_M contains at most k-1 pebble pairs of P, and let φ be the map which witnesses that P is safe. Then, Condition 3b guarantees that for all one-way switches O with input pair $\{y_1, y_2\}$ and output pair $\{x_1, x_2\}$, we can pick the bijection $\psi|_O$ according to a trapped, twisted or identity strategy on $P|_O \cup \{x_1 \mapsto \varphi(x_1), y_1 \mapsto \varphi(y_1)\}$. This ensures that $\psi|_O$ is compatible with φ , which means that we get a total bijection ψ , which Duplicator chooses. Again, each of the positions $P \cup \{x \mapsto \psi(x)\}$ is safe by construction.

Thus, for every node a with $\operatorname{val}_M(a) = \operatorname{True}$, Duplicator has a winning strategy in position $a_1 \mapsto a_2$ by always staying in safe positions. This proves $(G_M, a_1) \equiv_{k\text{-WL}} (G_M, a_2)$.

Similar to [19], combining Lemma 37 with Lemma 40 yields:

▶ Corollary 44. The k-WL-equivalence problem for vertices is P-hard under uniform AC_0 reductions, both over simple graphs with 4-bounded abelian color classes, and over uncolored simple graphs.

Proof. The claim for colored simple graphs with 4-bounded abelian color classes follows from P-hardness of the monotone circuit value problem, the observation that the graph G_M can be constructed in AC₀ together with Lemma 37 and Lemma 37.

To show P-hardness of the k-WL-equivalence problem over uncolored simple graphs, we need to eliminate the vertex-colors without affecting the k-WL-color partition on the original graph. To do this let $\chi\colon G_M\to C$ be the vertex-coloring of G_M , where C is a set of colors of polynomial size. Assume w.l.o.g. that χ is surjective, i.e., all colors in C are used.

We construct an uncolored simple graph G_M^C from G_M as follows: we start by adding to G_M a fresh universal vertex u. Then, we add a path P to G_M with vertex set C, and further add a fresh vertex v_0 to one end of the path. Finally, we connect each vertex $x \in V(G_M)$ to the vertex $\chi(x) \in C$. This finishes the construction of G_M^C . This construction can be carried out in AC_0 : the path P is independent of the input and can hence just be hardcoded into the circuit, and the additional edges only depend on the vertex-color.

Next, we argue that replacing the colored graph G_M by the uncolored graph G_M^C does not affect the power of the Weisfeiler-Leman algorithm. For this, note that the universal vertex u is the unique vertex in G_M^C of maximal degree, and hence distinguished even by color refinement. Thus, its neighborhood, which is the vertex set of $G_M \subseteq G_M^C$ is also distinguished from its non-neighborhood, which is the path P. The path P contains a single vertex of degree 1, namely the vertex v_0 we added to one end, which means that color refinement completely discretizes P. But because the color classes of $G_M \subseteq G_M^C$ are precisely the neighborhoods of the vertices on P in G_M , color refinement recomputes the color partition induced on G_M by χ , while discretizing every vertex we added. The claim now follows from the observation that adding discrete vertices which are homogeneously connected to every color class does not affect the power of the Weisfeiler-Leman algorithm.

Thus, Lemma 37 and Lemma 40 are also true for the graph G_M^C , which proves P-hardness of the k-WL-identification problem for uncolored simple graphs.

In order to further reduce to the identification problem, we need to consider identification of the graph G_M .

▶ Lemma 45. Let M be a monotone circuit with output node c which evaluates to FALSE. Then G_M is identified by k-WL.

Proof. By Lemma 37, all pairs $\{a_1, a_2\}$ corresponding to a node $a \in V(M)$ are split by k-WL. Translating these splits through the one-way switches, this implies that everything in G_M besides the one-way switches is discretized by k-WL. But as this includes all input and output pairs of the one-way switches, the fact that the graphs O_{split}^k are identified by Property 1, implies that all one-way switches are identified as subgraphs of G_M .

All individual parts of the graph G_M are thus identified as subgraphs of G_M . As moreover all edges are contained in one fo the individual parts, and different parts only overlap in input and output pairs, all of which are discretized by k-WL, it follows that the whole graph G_M is identified by k-WL.

Consider now the modified graph G_M^* we get by adding another freshly colored one-way switch O_*^k to G_M whose input pair is $\{c_1, c_2\}$, i.e., the vertex pair corresponding to the output node of the circuit M. Additionally, we color one of the output vertices of O_*^k in another color.

- ▶ **Lemma 46.** The following are equivalent:
- (i) $val_M(c) = FALSE$,
- (ii) $(G_M, c_1) \not\equiv_{k\text{-WL}} (G_M, c_2)$ and
- (iii) G_M^* is identified by k-WL.

Proof. The equivalence of Conditions 1 and 2 follows from Lemma 37 and Lemma 40. We show that Condition 2 implies Condition 3. Assume that $(G_M, c_1) \not\equiv_{k\text{-WL}} (G_M, c_2)$. This splits the input pair of O_*^k . By Property 1 of the one-way switches, O_*^k is now identified as a subgraph of G_M^* , and because G_M is also identified by Lemma 45 and both parts interact only at the split pair $\{c_1, c_2\}$, the whole graph G_M^* is identified.

We finally show that Condition 3 implies Condition 2 by contraposition. Assume $(G_M, c_1) \equiv_{k\text{-WL}} (G_M, c_2)$, and consider the graph $(G_M^*)'$ we obtain by following the same construction as for G_M^* , but then coloring the other output vertex of O_*^k . These two graphs are non-isomorphic, as the two output vertices of our one-way switches do not lie in the same orbit. They are, however, equivalent under k-WL: Duplicator can follow a twisted strategy to win between the two one-way switches and use an arbitrary winning strategy on G_M . If none of the two parts contains all pebble pairs, these two strategies can be made compatible at the pair $\{c_1, c_2\}$ by enforcing which of the two maps $c_1 \mapsto c_1$ or $c_1 \mapsto c_2$ both strategies should be compatible with. If all pebble pairs lie in one of the two parts, Duplicator can choose their strategy on this part first and then extend it to a winning strategy on the whole graph by using that $(G_M, c_1) \equiv_{k\text{-WL}} (G_M, c_2)$. This proves that $G_M^* \equiv_{k\text{-WL}} (G_M^*)'$ and thus that G_M^* is not identified by k-WL.

This finally yields our hardness result.

▶ **Theorem 4.** For every $k \ge 1$, the k-WL-identification problem is P-hard under uniform AC_0 -reductions over both uncolored simple graphs, and simple graphs with 4-bounded abelian color classes.

Proof. P-hardness of the identification problem for graphs with 4-bounded abelian color classes follows from Lemma 46 and the hardness of the monotone circuit value problem. Hardness for the class of simple graphs follows from the procedure for the elimination of vertex colors which we already used in the proof of Corollary 44.

6 Conclusion

We have shown on the one hand that when the dimension k is part of the input, the k-WL-equivalence problem and the k-WL-identification problem are co-NP-hard and NP-hard, respectively.

On the other hand, when the dimension k is fixed, the equivalence problem is trivially solvable in polynomial time, and we have shown that the identification problem is solvable in polynomial time over graphs with 5-bounded color classes and on relational structures with k-bounded abelian color classes. Still, the identification problem is P-hard in both cases. As an immediate corollary, we obtain the same polynomial-time solvability and hardness results for definability and equivalence in the bounded-variable logic with counting C^k .

It would be interesting to know whether the k-WL-identification problem can be solved in polynomial time for larger color classes or indeed on general graphs when k is fixed. Indeed, our NP-hardness reduction was based on whether the tree-width of a given graph is at most k, which can be solved in linear time for every fixed k [7], and thus does not even yield a super-linear lower bound when k is fixed. Still, we would expect that neither the

identification nor the equivalence problem can be solved in time $n^{o(k)}$. It might be fruitful to study these problems from the lens of parameterized complexity or provide lower complexity bounds based on the (strong) exponential time hypothesis.

References

- 1 Markus Anders and Pascal Schweitzer. Parallel computation of combinatorial symmetries. In 29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference), volume 204 of LIPIcs, pages 6:1–6:18. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ESA.2021.6.
- Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. Graph isomorphism, color refinement, and compactness. Comput. Complex., 26(3):627–685, 2017. doi:10.1007/S00037-016-0147-6.
- 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 *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 and Ludek Kucera. Canonical labelling of graphs in linear average time. In 20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979, pages 39-46. IEEE Computer Society, 1979. doi:10.1109/SFCS.1979.8.
- 6 László Babai. Monte-Carlo algorithms in graph isomorphism testing. Technical Report 79-10, Université de Montréal, 1979.
- 7 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput., 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.
- 8 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci., 209(1-2):1-45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 9 Hans L. Bodlaender, Édouard Bonnet, Lars Jaffke, Dušan Knop, Paloma T. Lima, Martin Milanič, Sebastian Ordyniak, Sukanya Pandey, and Ondřej Suchý. Treewidth is NP-complete on cubic graphs. In 18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands, volume 285 of LIPIcs, pages 7:1–7:13. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS. IPEC.2023.7.
- Béla Bollobás. Distinguishing vertices of random graphs. *North-holland Mathematics Studies*, 62:33–49, 1982. doi:10.1016/S0304-0208(08)73545-X.
- Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Comb.*, 12(4):389–410, 1992. doi:10.1007/BF01305232.
- G. Chen and I. Ponomarenko. Lectures on Coherent Configurations. Central China Normal University Press, 2019. A draft is available at https://www.pdmi.ras.ru/~inp/.
- Anuj Dawar and David Richerby. The power of counting logics on restricted classes of finite structures. In 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.
- Zdenek Dvorák. On recognizing graphs by numbers of homomorphisms. J. Graph Theory, 64(4):330-342, 2010. doi:10.1002/JGT.20461.
- Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. Identifiability of graphs with small color classes by the Weisfeiler-Leman algorithm. SIAM J. Discret. Math., 35(3):1792–1853, 2021. doi:10.1137/20M1327550.
- Merrick Furst, John Hopcroft, and Eugene M. Luks. A subexponential algorithm for trivalent graph isomorphism. Technical report, Cornell University, USA, 1980.

- 17 Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for P. SIGACT News, 9(2):25-29, 1977. doi:10.1145/1008354.1008356.
- 18 Erich Grädel and Wied Pakusa. Rank logic is dead, long live rank logic! *J. Symb. Log.*, 84(1):54–87, 2019. doi:10.1017/jsl.2018.33.
- 19 Martin Grohe. Equivalence in finite-variable logics is complete for polynomial time. *Comb.*, 19(4):507–532, 1999. doi:10.1007/S004939970004.
- 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.
- 21 Martin Grohe, Moritz Lichter, Daniel Neuen, and Pascal Schweitzer. Compressing CFI graphs and lower bounds for the Weisfeiler-Leman refinements. In 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023, pages 798–809. IEEE, 2023. doi:10.1109/F0CS57990.2023.00052.
- 22 Martin Grohe and Julian Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *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.
- 23 Martin Grohe and Daniel Neuen. Canonisation and definability for graphs of bounded rank width. ACM Trans. Comput. Log., 24(1):6:1–6:31, 2023. doi:10.1145/3568025.
- 24 Martin Grohe and Martin Otto. Pebble games and linear equations. J. Symb. Log., 80(3):797–844, 2015. doi:10.1017/JSL.2015.28.
- 25 Lauri Hella. Logical hierarchies in PTIME. Inf. Comput., 129(1):1-19, 1996. doi:10.1006/INCO.1996.0070.
- Neil Immerman and Eric S. Lander. Describing Graphs: A First-Order Approach to Graph Canonization, pages 59–81. Springer New York, New York, NY, 1990. doi: 10.1007/978-1-4612-4478-3_5.
- 27 Tommi A. Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In Proceedings of the Nine Workshop on Algorithm Engineering and Experiments, ALENEX 2007, New Orleans, Louisiana, USA, January 6, 2007. SIAM, 2007. doi:10.1137/1.9781611972870.13.
- 28 Tommi A. Junttila and Petteri Kaski. Conflict propagation and component recursion for canonical labeling. In Theory and Practice of Algorithms in (Computer) Systems First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings, volume 6595 of Lecture Notes in Computer Science, pages 151-162. Springer, 2011. doi: 10.1007/978-3-642-19754-3_16.
- Sandra Kiefer, Ilia Ponomarenko, and Pascal 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.
- 30 Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. *ACM Trans. Comput. Log.*, 23(1):1:1–1:31, 2022. doi:10.1145/3417515.
- 31 Ludek Kucera. Canonical labeling of regular graphs in linear average time. In 28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987, pages 271–279. IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.11.
- 32 Moritz Lichter. Separating rank logic from polynomial time. J. ACM, 70(2), 03 2023. doi:10.1145/3572918.
- Moritz Lichter. Witnessed symmetric choice and interpretations in fixed-point logic with counting. In 50th International Colloquium on Automata, Languages, and Programming (ICALP 2023), volume 261 of LIPIcs, pages 133:1–133:20. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.133.
- 34 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.
- 35 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. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI

- 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 4602-4609. AAAI Press, 2019. doi:10.1609/AAAI.V33I01.33014602.
- Daniel Neuen and Pascal Schweitzer. Benchmark graphs for practical graph isomorphism. In 25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria, volume 87 of LIPIcs, pages 60:1-60:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.ESA.2017.60.
- 37 Daniel Neuen and Pascal Schweitzer. An exponential lower bound for individualizationrefinement algorithms for graph isomorphism. In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018, pages 138-150. ACM, 2018. doi:10.1145/3188745.3188900.
- 38 Thomas Schneider and Pascal Schweitzer. An upper bound on the Weisfeiler-Leman dimension, 2024. arXiv:2403.12581.
- 39 Kyoungah See and Sung Y. Song. Association schemes of small order. Journal of Statistical Planning and Inference, 73(1):225-271, 1998. doi:10.1016/S0378-3758(98)00064-0.
- Tim Seppelt. An Algorithmic Meta Theorem for Homomorphism Indistinguishability. In 49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024), volume 306 of Leibniz International Proceedings in Informatics (LIPIcs), pages 82:1-82:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.MFCS.2024.82.
- Paul D. Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. $J.\ Comb.\ Theory,\ Ser.\ B,\ 58(1):22-33,\ 1993.\ doi:10.1006/JCTB.1993.1027.$
- 42 B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. Nauchno-Technicheskaya Informatsia, Seriya 2, 9:12–16, 1968. An english translation due to Grigory Ryabov is available at https://www.iti.zcu.cz/wl2018/ pdf/wl_paper_translation.pdf.
- 43 Faried Abu Zaid, Erich Grädel, Martin Grohe, and Wied Pakusa. Choiceless polynomial time on structures with small abelian colour classes. In Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I, volume 8634 of Lecture Notes in Computer Science, pages 50-62. Springer, 2014. doi:10.1007/978-3-662-44522-8_5.