An Algorithmic Meta Theorem for Homomorphism Indistinguishability

Tim Seppelt
RWTH Aachen University, Germany

Abstract

Two graphs G and H are *homomorphism indistinguishable* over a family of graphs $\mathcal{F}$ if for all graphs $F \in \mathcal{F}$ the number of homomorphisms from F to G is equal to the number of homomorphism from F to H. Many natural equivalence relations comparing graphs such as (quantum) isomorphism, cospectrality, and logical equivalences can be characterised as homomorphism indistinguishability relations over various graph classes.

The wealth of such results motivates a more fundamental study of homomorphism indistinguishability. From a computational perspective, the central object of interest is the decision problem HomInd($\mathcal{F}$) which asks to determine whether two input graphs G and H are homomorphism indistinguishable over a fixed graph class $\mathcal{F}$. The problem HomInd($\mathcal{F}$) is known to be decidable only for few graph classes $\mathcal{F}$. Due to a conjecture by Roberson (2022) and results by Seppelt (MFCS 2023), homomorphism indistinguishability relations over minor-closed graph classes are of special interest. We show that HomInd($\mathcal{F}$) admits a randomised polynomial-time algorithm for every minor-closed graph class $\mathcal{F}$ of bounded treewidth.

This result extends to a version of HomInd where the graph class $\mathcal{F}$ is specified by a sentence in counting monadic second-order logic and a bound k on the treewidth, which are given as input. For fixed k, this problem is randomised fixed-parameter tractable. If k is part of the input, then it is coNP- and coW[1]-hard. Addressing a problem posed by Berkholz (2012), we show coNP-hardness by establishing that deciding indistinguishability under the k-dimensional Weisfeiler–Leman algorithm is coNP-hard when k is part of the input.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases homomorphism indistinguishability, graph homomorphism, graph minor, recognisability, randomised algorithm, Courcelle's Theorem

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.82

Related Version *Full Version*: https://arxiv.org/abs/2402.08989

Funding German Research Foundation (DFG) within Research Training Group 2236/2 (UnRAVeL), European Union (ERC, SymSim, 101054974). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements I would like to thank Martin Grohe and Louis Härtel for fruitful discussions and their support. In particular, I thank Martin Grohe for making me aware of Chinese remaindering.

1 Introduction

In 1967, Lovász [32] proved that two graphs G and H are isomorphic if and only if they are *homomorphism indistinguishable* over all graphs, i.e. they admit the same number of homomorphisms from every graph F. Subsequently, many graph isomorphism relaxations have been characterised as homomorphism indistinguishability relations. For example, two graphs are quantum isomorphic if and only if they are homomorphism indistinguishable over

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 82; pp. 82:1–82:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

all planar graphs [34]. Moreover, two graphs satisfy the same sentences in k-variable first-order logic with counting quantifiers if and only if they are homomorphism indistinguishable over all graphs of treewidth less than k [20, 19]. A substantial list of similar results characterises notions from quantum information [34, 5], finite model theory [20, 24, 22], convex optimisation [19, 28, 38], algebraic graph theory [19, 28], machine learning [35, 41, 25], and category theory [18, 1] as homomorphism indistinguishability relations.

The wealth of such examples motivates a more principled study of homomorphism indistinguishability [3, 37, 40]. Notably, all graph classes featured in the results listed above are minor-closed and this is not a mere coincidence [40, Theorem 1]. Therefore, homomorphism indistinguishability relations of minor-closed graph classes are of central interest in light of the emerging theory of homomorphism indistinguishability. In [37], Roberson conjectured that *any two distinct graph classes which are closed under disjoint unions and taking minors have distinct homomorphism indistinguishability relations.* From a computational perspective, the central question on homomorphism indistinguishability concerns the complexity and computability of the following decision problem for a fixed graph class $\mathcal{F}$ [37, Question 9]:

HomInd($\mathcal{F}$)
Input Graphs G and H.
Question Are G and H homomorphism indistinguishable over $\mathcal{F}$?

The graphs G and H may be arbitrary graphs and do not necessarily have to be in $\mathcal{F}$. Typically, the graph class $\mathcal{F}$ is infinite. Thus, the trivial approach to HomInd($\mathcal{F}$) of checking whether G and H have the same number of homomorphisms from every $F \in \mathcal{F}$ does not even render HomInd($\mathcal{F}$) decidable. Beyond this observation, the understanding of the problems HomInd($\mathcal{F}$) is limited to a short list of examples of graph classes $\mathcal{F}$: For the class $\mathcal{G}$ of all graphs, HomInd($\mathcal{G}$) is graph isomorphism [32], a problem representing a long standing complexity-theoretic challenge and currently only known to be in quasi-polynomial time [6]. For the class $\mathcal{P}$ of all planar graphs, HomInd($\mathcal{P}$) is quantum isomorphism and undecidable [34]. Finally, for the class $\mathcal{TW}_k$ of all graphs of treewidth at most k, HomInd($\mathcal{TW}_k$) can be decided in polynomial time with the well-known k-dimensional Weisfeiler–Leman algorithm [20, 19].

These results illustrate that the complexity of HomInd($\mathcal{F}$) is not monotone in the sense that if $\mathcal{F}_1 \subseteq \mathcal{F}_2$, then HomInd($\mathcal{F}_1$) is at most as hard as HomInd($\mathcal{F}_2$). For example, despite that $\mathcal{TW}_2 \subseteq \mathcal{P} \subseteq \mathcal{G}$, deciding homomorphism indistinguishability over $\mathcal{TW}_2$, $\mathcal{P}$, and $\mathcal{G}$ is polynomial-time, undecidable, and quasi-polynomial-time, respectively. Furthermore, although HomInd($\mathcal{TW}_k$) is in polynomial-time for every k, there are infinitely many minor-closed graph classes $\mathcal{F}$ of bounded treewidth, e.g. the classes of k-outerplanar graphs, for which HomInd($\mathcal{F}$) could yet be undecidable. Our main result shows that this is not the case: HomInd($\mathcal{F}$) is in randomised polynomial time for every minor-closed graph class $\mathcal{F}$ of bounded treewidth.

▶ **Theorem 1.** *Let $k \geq 1$. If $\mathcal{F}$ is a k-recognisable class of graphs of treewidth at most $k-1$, then* HomInd($\mathcal{F}$) *is in* coRP*.*

Spelled out, Theorem 1 asserts that there exists a randomised algorithm for HomInd($\mathcal{F}$) which always runs in polynomial time, accepts all YES-instances and incorrectly accepts NO-instances with probability less than one half. Recognisability is a fairly general property that arises in the context of Courcelle's theorem [14], cf. Definition 10. Courcelle showed that every graph class definable in counting monadic second-order logic CMSO_2 is recognisable. This subsumes graph classes defined by finitely many forbidden (induced) subgraphs and minors, and by the Robertson–Seymour Theorem, all minor-closed graph classes.

Thereby, Theorem 1 applies to e.g. the class of graphs of bounded branchwidth, k-outerplanar graphs, and the class of trees of bounded degree. As a concrete application, we resolve an open question from [38] by showing in Theorem 22 that the exact feasibility of the Lasserre semidefinite programming hierarchy for graph isomorphism can be decided in randomised polynomial-time.

The proof of Theorem 1 combines Courcelle's graph algebras [15] with the homomorphism tensors from [34, 28]. Graph algebras comprise labelled graphs and operations on them such as series and parallel composition. Homomorphism tensors keep track of homomorphism counts of labelled graphs. We show that recognisability and bounded treewidth guarantee that homomorphism tensors yield finite-dimensional representations of suitable graph algebras which certify homomorphism indistinguishability and are efficiently computable. The algorithm in Theorem 1 is randomised as it employs arithmetic modulo random primes to deal with integers which would otherwise grow to exponential size. For graph classes of bounded pathwidth, this issue can be avoided:

▶ **Theorem 2.** *Let $k \geq 1$. If $\mathcal{F}$ is a k-recognisable class of graphs of pathwidth at most $k-1$, then* HomInd$(\mathcal{F})$ *is in polynomial time.*

The connection to Courcelle's theorem motivates considering the parametrised problem HomInd. Here, the CMSO_2-sentence φ allows the graph class to be specified as part of the input. Using results by Courcelle [14], we generalise Theorem 1 in Theorem 3.

HomInd
Input Graphs G and H, a CMSO_2-sentence φ, an integer k.
Parameter $|\varphi| + k$.
Question Are G and H homomorphism indistinguishable over the graph class $\mathcal{F}_{\varphi,k}$ of graphs of treewidth at most $k-1$ satisfying φ?

▶ **Theorem 3.** *There exists a computable function $f \colon \mathbb{N} \to \mathbb{N}$ and a randomised algorithm for* HomInd *of runtime $f(|\varphi| + k)n^{O(k)}$ for $n := \max\{|V(G)|, |V(H)|\}$ which accepts all* YES*-instances and accepts* NO*-instances with probability less than one half.*

Equipped with the parametrised perspective offered by HomInd, we finally consider lower bounds on the complexity of this problem. Firstly, we show that it is coW[1]-hard and that the runtime in Theorem 3 is optimal under the Exponential Time Hypothesis (ETH).

▶ **Theorem 4.** HomInd *is* coW[1]*-hard under fpt-reductions. Unless* ETH *fails, there is no algorithm for* HomInd *that runs in time $f(|\varphi| + k)n^{o(|\varphi|+k)}$ for any computable function $f \colon \mathbb{N} \to \mathbb{N}$.*

Secondly, we show that, when disregarding the parametrisation, HomInd is coNP-hard. We do so by showing coNP-hardness of deciding indistinguishability under the k-dimensional Weisfeiler–Leman (WL) algorithm (recently, the same result was independently obtained by [31]). WL is an important heuristic in graph isomorphism and tightly related to notions in finite model theory [13] and graph neural networks [41, 35, 25]. The fastest known algorithm for WL runs in time $O(k^2 n^{k+1} \log n)$ [29], which is exponential when regarding k as part of the input. It was shown that when k is fixed, then the problem is PTIME-complete [23]. Establishing lower bounds on the complexity of WL is a challenging problem [7, 8, 26]. Theorem 5 is a first step towards resolving a question posed by Berkholz [7]: Is the decision problem in Theorem 5 EXPTIME-complete?

▶ **Theorem 5.** *The problem of deciding given graphs G and H and an integer $k \in \mathbb{N}$ whether G and H are k-*WL *indistinguishable is* coNP*-hard under polynomial-time many-one reductions.*

2 Preliminaries

All graphs in this work are finite, undirected, and without multiple edges and loops. For a graph G, we write $V(G)$ for its vertex set and $E(G)$ for its edge set. A *homomorphism* $h\colon F \to G$ from a graph F to a graph G is a map $V(F) \to V(G)$ such that $h(u)h(v) \in E(G)$ for all $uv \in E(F)$. Write $\hom(F, G)$ from the number of homomorphisms from F to G.

Two graphs G and H are *homomorphism indistinguishable* over a class of graph $\mathcal{F}$, in symbols $G \equiv_{\mathcal{F}} H$, if $\hom(F, G) = \hom(F, H)$ for all $F \in \mathcal{F}$. For an integer $n \geq 2$, G and H are *homomorphism indistinguishable over* $\mathcal{F}$ *modulo* n, in symbols $G \equiv^n_{\mathcal{F}} H$, if $\hom(F, G) \equiv \hom(F, H) \mod n$ for all $F \in \mathcal{F}$, cf. [21, 30].

2.1 (Bi)labelled Graphs and Homomorphism Tensors

Let $k, \ell \geq 1$. A *distinctly k-labelled graph* is a tuple $\boldsymbol{F} = (F, \boldsymbol{u})$ where F is a graph and $\boldsymbol{u} \in V(F)^k$ is such that $u_i \neq u_j$ for all $1 \leq i < j \leq k$. We say $u_i \in V(F)$, the i-th entry of $\boldsymbol{u}$, *carries the i-th label.* Write $\mathcal{D}(k)$ for the class of distinctly k-labelled graphs.

A *distinctly (k, ℓ)-bilabelled graph* is a tuple $\boldsymbol{F} = (F, \boldsymbol{u}, \boldsymbol{v})$ where F is a graph and $\boldsymbol{u} \in V(F)^k$ and $\boldsymbol{v} \in V(F)^\ell$ are such that $u_i \neq u_j$ for all $1 \leq i < j \leq k$ and $v_i \neq v_j$ for all $1 \leq i < j \leq \ell$. Note that $\boldsymbol{u}$ and $\boldsymbol{v}$ might share entries. We say $u_i \in V(F)$ and $v_i \in V(F)$ *carry the i-th in-label* and *out-label*, respectively. Weite $\mathcal{D}(k, \ell)$ for the class of distinctly (k, ℓ)-bilabelled graphs.

For a graph G, and $\boldsymbol{F} = (F, \boldsymbol{u}) \in \mathcal{D}(k)$ define the *homomorphism tensor* $\boldsymbol{F}_G \in \mathbb{N}^{V(G)^k}$ of $\boldsymbol{F}$ w.r.t. G whose $\boldsymbol{v}$-th entry is equal to the number of homomorphisms $h\colon F \to G$ such that $h(u_i) = v_i$ for all $i \in [k]$. Analogously, for $\boldsymbol{F} \in \mathcal{D}(k, \ell)$, define $\boldsymbol{F}_G \in \mathbb{N}^{V(G)^k \times V(G)^\ell}$. As the entries of homomorphism tensors are integral, we can view them as vectors in vector spaces over $\mathbb{R}$ as in Section 3 or over finite fields $\mathbb{F}_p$ as in Section 4.

As observed in [34, 28], (bi)labelled graphs and their homomorphism tensors are intriguing due to the following correspondences between combinatorial operations on the former and algebraic operations on the latter:

Dropping labels corresponds to sum-of-entries (soe). For $\boldsymbol{F} = (F, \boldsymbol{u}) \in \mathcal{D}(k)$, define the underlying unlabelled graph $\operatorname{soe}(\boldsymbol{F}) \coloneqq F$ of $\boldsymbol{F}$. Then for all graphs G, $\hom(\operatorname{soe} \boldsymbol{F}, G) = \sum_{\boldsymbol{v} \in V(G)^k} \boldsymbol{F}_G(\boldsymbol{v}) \eqqcolon \operatorname{soe}(\boldsymbol{F}_G)$.

Gluing corresponds to Schur products. For $\boldsymbol{F} = (F, \boldsymbol{u})$ and $\boldsymbol{F}' = (F', \boldsymbol{u}')$ in $\mathcal{D}(k)$, define $\boldsymbol{F} \odot \boldsymbol{F}' \in \mathcal{D}(k)$ as the k-labelled graph obtained by taking the disjoint union of F and F' and placing the i-th label at the vertex obtained by merging u_i with u'_i for all $i \in [k]$. Then for every graph G and $\boldsymbol{v} \in V(G)^k$, $(\boldsymbol{F} \odot \boldsymbol{F}')_G(\boldsymbol{v}) = \boldsymbol{F}_G(\boldsymbol{v})\boldsymbol{F}'_G(\boldsymbol{v}) \eqqcolon (\boldsymbol{F}_G \odot \boldsymbol{F}'_G)(\boldsymbol{v})$. One may similarly define the gluing product of two (k, ℓ)-bilabelled graphs.

Series composition corresponds to matrix products. For bilabelled graphs $\boldsymbol{K} = (K, \boldsymbol{u}, \boldsymbol{v})$ and $\boldsymbol{K}' = (K', \boldsymbol{u}', \boldsymbol{v}')$ in $\mathcal{D}(k, k)$, define $\boldsymbol{K} \cdot \boldsymbol{K}' \in \mathcal{D}(k, k)$ as the bilabelled graph obtained by taking the disjoint union of K and K', merging the vertices v_i and u'_i for $i \in [k]$, and placing the i-th in-label (out-label) on u_i (on v'_i) for $i \in [k]$. Then for all graphs G and $\boldsymbol{x}, \boldsymbol{z} \in V(G)^k$, $(\boldsymbol{K} \cdot \boldsymbol{K}')_G(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{y} \in V(G)^k} \boldsymbol{K}_G(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{K}'_G(\boldsymbol{y}, \boldsymbol{z}) \eqqcolon (\boldsymbol{K}_G \cdot \boldsymbol{K}'_G)(\boldsymbol{x}, \boldsymbol{z})$. One may similarly compose a graph in $\mathcal{D}(k, k)$ with a graph in $\mathcal{D}(k)$ obtaining one in $\mathcal{D}(k)$. This operation corresponds to the matrix-vector product.

2.2 Labelled Graphs of Bounded Treewidth

Labelled graphs of bounded treewidth represent the technical foundation of most proofs in this article. Several versions of such have been used in previous works [28, 15, 22].

Let F be a graph. A *tree decomposition* of F is a pair (T, β) where T is a tree and $\beta \colon V(T) \to 2^{V(F)}$ is a map such that

1. the union of the $\beta(t)$ for $t \in V(T)$ is equal to $V(F)$,
2. for every edge $uv \in E(F)$ there exists $t \in V(T)$ such that $\{u, v\} \subseteq \beta(t)$,
3. for every vertex $u \in V(F)$ the set of vertices $t \in V(T)$ such that $u \in \beta(t)$ is connected in T.

The *width* of (T, β) is the maximum over all $|\beta(t)| - 1$ for $t \in V(T)$. The *treewidth* $\operatorname{tw} F$ of F is the minimum width of a tree decomposition of F. A *path decomposition* is a tree decomposition (T, β) where T is a path. The *pathwidth* $\operatorname{pw} F$ of F is the minimum width of a path decomposition of F.

Building on [9, Lemma 8], we show in the following Lemma 6 that every tree decomposition can be rearranged such that the depth of the decomposition tree gives a bound on the number of vertices in the decomposed graph.

▶ **Lemma 6.** *Let $k \geq 1$ and F be a graph such that $\operatorname{tw} F \leq k - 1$ and $|V(F)| \geq k$. Then F admits tree decomposition (T, β) such that*

1. *$|\beta(t)| = k$ for all $t \in V(T)$,*
2. *$|\beta(s) \cap \beta(t)| = k - 1$ for all $st \in E(T)$,*
3. *there exists a vertex $r \in V(T)$ such that the out-degree of every vertex in the rooted tree (T, r) is at most k.*

Proof. By [9, Lemma 8], there exists a tree decomposition (T, β) of F satisfying the first two assertions. Pick a root $r \in V(T)$ arbitrarily. To ensure that the last property holds, the tree decomposition is modified recursively as follows:

By merging vertices, it can be ensured that no two children of r carry the same bag, i.e. that there exist no two children $s_1 \neq s_2$ of r such that $\beta(s_1) = \beta(s_2)$.

For every $v \in \beta(r)$, let $C(v)$ denote the set of all children t of r such that $\beta(r) \setminus \beta(t) = \{v\}$, i.e. $C(v)$ is the set of all children of r whose bags do not contain the vertex v. The collection $C(v)$ for $v \in \beta(r)$ is a partition of the children of r in at most k parts. Note that for two distinct children $t_1 \neq t_2$ in the same part $C(v)$ it holds that $|\beta(t_1) \cap \beta(t_2)| = k - 1$. Rewire the children of r as follows: For every $v \in \beta(r)$ with $C(v) \neq \emptyset$, pick $t \in C(v)$, make t a child of r and all other elements of $C(v)$ children of t. The vertex r now has at most k children and the new tree decomposition still satisfies the first two assertions. Proceed by processing the children of r. ◀

Inspired by Lemma 6, we consider the following family of distinctly labelled graphs. The *depth* of a rooted tree (T, r) is the maximal number of vertices on any path from r to a leaf.

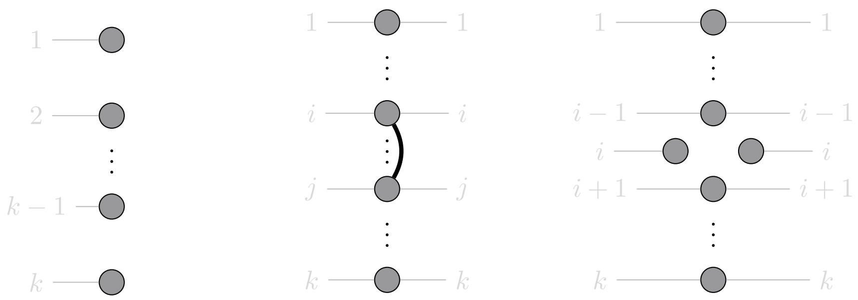
▶ **Definition 7.** *Let $k, d \geq 1$. Define $\mathcal{TW}_d(k)$ as the set of all $\boldsymbol{F} = (F, \boldsymbol{u}) \in \mathcal{D}(k)$ such that F admits a tree decomposition (T, β) of width $\leq k - 1$ satisfying the assertions of Lemma 6 with some $r \in V(T)$ such that $\beta(r) = \{u_1, \dots, u_k\}$ and (T, r) is of depth at most d. Let $\mathcal{TW}(k) \coloneqq \bigcup_{d \geq 1} \mathcal{TW}_d(k)$.*

Every graph in $\mathcal{TW}_d(k)$ has at least k vertices. Conversely, Definition 7 permits the following upper bound on the size of the graphs in $\mathcal{TW}_d(k)$ in terms of d and k.

▶ **Lemma 8.** *Let $k, d \geq 1$. Every $\boldsymbol{F} \in \mathcal{TW}_d(k)$ has at most $\max\{k^d, d\}$ vertices.*

Proof. Let (T, β) and $r \in V(T)$ be as in Definition 7. If $k = 1$, then every vertex in (T, r) has out-degree 1 and F at most d vertices.

Suppose that $k \geq 2$. The proof is by induction on the depth d of the rooted tree (T, r). If $d = 1$, then T contains only a single vertex and $\boldsymbol{F}$ has at most k vertices.



(a) $\mathbf{1} \in \mathcal{TW}(k)$. **(b)** $\boldsymbol{A}^{ij} \in \mathcal{D}(k,k)$. **(c)** $\boldsymbol{J}^{i} \in \mathcal{D}(k,k)$.

Figure 1 The (bi)labelled generators of $\mathcal{TW}(k)$ in wire notion of [34]. A vertex carries in-label (out-label) i if it is connected to the index i on the left (right) by a wire. Actual edges and vertices of the graph are depicted in black.

For the inductive step, let $\boldsymbol{F}$ be of depth d. If r has only a single neighbour s, then $S := T - r$ is such that (S, s) is of depth $d-1$. By the inductive hypothesis, $\left|\bigcup_{s \in V(S)} \beta(s)\right| \leq k^d$. Furthermore, $\left|\bigcup_{t \in V(T)} \beta(t) \setminus \bigcup_{s \in V(S)} \beta(s)\right| = 1$. Hence, $\boldsymbol{F}$ has at most $k^{d-1} + 1 \leq k^d$ many vertices.

If r has multiple neighbours, observe that due to Lemma 6 every vertex in $\beta(r)$ is also in $\beta(s)$ for some neighbour s of r. Hence, the number of vertices in $\boldsymbol{F}$ is bounded by the number of vertices covered by the subtrees of $T - r$ rooted in s. Thus, $\boldsymbol{F}$ has at most $k^{d-1} \cdot k \leq k^d$ many vertices. ◀

Clearly, if $\boldsymbol{F} \in \mathcal{TW}(k)$, then $\operatorname{tw}(\operatorname{soe} \boldsymbol{F}) \leq k-1$. Conversely, by Lemma 6, for every F with $\operatorname{tw} F \leq k-1$ and $|V(F)| \geq k$, there exists $\boldsymbol{u} \in V(F)^k$ such that $(F, \boldsymbol{u}) \in \mathcal{TW}(k)$. Thus the underlying unlabelled graphs of the labelled graphs in $\mathcal{TW}(k)$ are exactly the graphs of treewidth $\leq k-1$ on $\geq k$ vertices.

The family $\mathcal{TW}(k)$ is generated by certain small building blocks under series composition and gluing as follows: Let $\mathbf{1} \in \mathcal{TW}_1(k)$ be the distinctly k-labelled graph on k vertices without any edges. For $i \in [k]$, let $\boldsymbol{J}^i = (J^i, (1, \dots, k), (1, \dots, i-1, \widehat{i}, i+1, \dots, k))$ the distinctly (k,k)-bilabelled graph with $V(J^i) := [k] \cup \{\widehat{i}\}$ and $E(J^i) := \emptyset$. Writing $\binom{[k]}{2}$ for the set of pairs of distinct elements in $[k]$, let $\boldsymbol{A}^{ij} = (A^{ij}, (1, \dots, k), (1, \dots, k))$ for $ij \in \binom{[k]}{2}$ be the distinctly (k,k)-bilabelled graph with $V(A^{ij}) := [k]$ and $E(A^{ij}) := \{ij\}$. These graphs are depicted in Figure 1. Let $\mathcal{B}(k) := \{\boldsymbol{J}^i \mid i \in [k]\} \cup \{\boldsymbol{A}^{ij} \mid ij \in \binom{[k]}{2}\}$.

It can be readily verified that if $\boldsymbol{F} \in \mathcal{TW}(k)$ and $\boldsymbol{B} \in \mathcal{B}(k)$, then $\boldsymbol{B} \cdot \boldsymbol{F} \in \mathcal{TW}(k)$. Furthermore, if $\boldsymbol{F}, \boldsymbol{F}' \in \mathcal{TW}(k)$, then $\boldsymbol{F} \odot \boldsymbol{F}' \in \mathcal{TW}(k)$. Conversely, the elements of $\mathcal{B}(k)$ generate $\mathcal{TW}(k)$ in the following sense:

▶ **Lemma 9.** *Let $k \geq 1$. For every $\boldsymbol{F} \in \mathcal{TW}(k)$, one of the following holds:*
1. $\boldsymbol{F} = \mathbf{1}$,
2. $\boldsymbol{F} = \prod_{ij \in A} \boldsymbol{A}^{ij} \cdot \boldsymbol{F}'$ *for some* $A \subseteq \binom{[k]}{2}$ *and* $\boldsymbol{F}' \in \mathcal{TW}(k)$ *with less edges than* $\boldsymbol{F}$,
3. $\boldsymbol{F} = \boldsymbol{J}^i \cdot \boldsymbol{F}'$ *for some* $i \in [k]$ *and* $\boldsymbol{F}' \in \mathcal{TW}(k)$ *with less vertices than* $\boldsymbol{F}$,
4. $\boldsymbol{F} = \boldsymbol{F}_1 \odot \boldsymbol{F}_2$ *for* $\boldsymbol{F}_1, \boldsymbol{F}_2 \in \mathcal{TW}(k)$ *on less vertices than* $\boldsymbol{F}$.

2.3 Recognisability and $CMSO_2$

Recognisability is a property of a class of unlabelled graphs which is shared by most named graph classes. It is the assumption of Theorem 1 which rules out esoteric graph classes as constructed in [12, Theorem 1]. We consider the following definition of recognisability.

(a) $\mathbf{1}$. **(b)** $\boldsymbol{P}$. **(c)** $\boldsymbol{Q}$. **(d)** $\boldsymbol{C}$.

Figure 2 Representatives for $\sim^1_{\mathcal{W}}$ and $\mathcal{W}$ the class of paths from Example 11.

▶ **Definition 10** ([11]). *Let $k \geq 1$. For class of unlabelled graphs $\mathcal{F}$, define the equivalence relation $\sim^k_{\mathcal{F}}$ on the class of distinctly k-labelled graphs $\mathcal{D}(k)$ by letting $\boldsymbol{F}_1 \sim^k_{\mathcal{F}} \boldsymbol{F}_2$ if and only if for all $\boldsymbol{K} \in \mathcal{D}(k)$ it holds that*

$$\text{soe}(\boldsymbol{K} \odot \boldsymbol{F}_1) \in \mathcal{F} \iff \text{soe}(\boldsymbol{K} \odot \boldsymbol{F}_2) \in \mathcal{F}.$$

*The class $\mathcal{F}$ is k-*recognisable *if $\sim^k_{\mathcal{F}}$ has finitely many equivalence classes. The number of classes of $\sim^k_{\mathcal{F}}$ is the k-*recognisability index *of $\mathcal{F}$.*

To parse Definition 10, first recall that $\boldsymbol{K} \odot \boldsymbol{F}_1$ is the k-labelled graph obtained by gluing $\boldsymbol{K}$ and $\boldsymbol{F}_1$ together at their labelled vertices. The soe-operator drops the labels yielding unlabelled graphs. Intuitively, $\boldsymbol{F}_1 \sim^k_{\mathcal{F}} \boldsymbol{F}_2$ iff both or neither of their underlying unlabelled graphs are in $\mathcal{F}$ and the positions of the labels in $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ is equivalent with respect to membership in $\mathcal{F}$. This intuition is made more concrete in the following example:

▶ **Example 11.** The class $\mathcal{W}$ of all paths is 1-recognisable. Its 1-recognisability index is 4. The equivalence classes are described by the representatives in Figure 2.

Proof. To show that the labelled graphs in Figure 2 cover all equivalence classes, let $\boldsymbol{F} = (F, u)$ be arbitrary. If F is not a path, then $\boldsymbol{F} \sim^1_{\mathcal{W}} \boldsymbol{C}$. Indeed, for every $\boldsymbol{K} \in \mathcal{D}(1)$, F is a subgraph of soe$(\boldsymbol{K} \odot \boldsymbol{F})$. Hence, regardless of $\boldsymbol{K}$, both soe$(\boldsymbol{K} \odot \boldsymbol{F})$ and soe$(\boldsymbol{K} \odot \boldsymbol{C})$ are not paths. If F is a path,, then $\boldsymbol{F}$ and $\mathbf{1}$, $\boldsymbol{P}$, or $\boldsymbol{Q}$ are equivalent depending on whether the degree of u is 0, 1, or 2.

To show that the representatives in Figure 2 are in distinct classes, observe for example that soe$(\boldsymbol{P} \odot \boldsymbol{P}) \in \mathcal{W}$ while soe$(\boldsymbol{P} \odot \boldsymbol{Q}) \notin \mathcal{W}$, thus $\boldsymbol{P} \not\sim^1_{\mathcal{W}} \boldsymbol{Q}$. Similarly, soe$(\mathbf{1} \odot \boldsymbol{Q}) \in \mathcal{W}$ whereas soe$(\boldsymbol{P} \odot \boldsymbol{Q}) \notin \mathcal{W}$, thus $\mathbf{1} \not\sim^1_{\mathcal{W}} \boldsymbol{P}$. ◀

A more involved example is the following. Analogously, one may argue that every class defined by forbidden minors is recognisable.

▶ **Example 12.** Let $\mathcal{F}$ be the family of H-subgraph-free graphs for some graph H. Then $\mathcal{F}$ is k-recognisable for every $k \geq 1$.

Proof. Suppose H has m vertices. For a distinctly k-labelled graph $\boldsymbol{F} = (F, \boldsymbol{u})$, consider the set $\mathcal{H}(\boldsymbol{F})$ of (isomorphism types of) distinctly k-labelled graphs $\boldsymbol{F}' = (F', \boldsymbol{u})$ where F' is a subgraph of F such that $V(F') \supseteq \{u_1, \dots, u_k\}$ has at most $k + m$ vertices. Clearly, there are only finitely many possible sets $\mathcal{H}(\boldsymbol{F})$. Furthermore, if $\mathcal{H}(\boldsymbol{F}_1) = \mathcal{H}(\boldsymbol{F}_2)$, then $\boldsymbol{F}_1 \sim^k_{\mathcal{F}} \boldsymbol{F}_2$. Indeed, if $\boldsymbol{K} \in \mathcal{D}(k)$ is such that soe$(\boldsymbol{K} \odot \boldsymbol{F}_1)$ contains H as a subgraph, then so does soe$(\boldsymbol{K} \odot \boldsymbol{F}_2)$ since $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ contain the same subgraphs on $k + m$ vertices containing their labelled vertices. ◀

Courcelle [14] proved that every CMSO_2-definable graph class is *recognisable*, i.e. it is k-recognisable for every $k \in \mathbb{N}$. Conversely, Bojańczyk and Pilipczuk [11] proved that if a recognisable class $\mathcal{F}$ has bounded treewidth, then it is CMSO_2-definable. Furthermore, they conjecture that k-recognisability is a sufficient condition for a graph class of treewidth at most $k - 1$ to be CMSO_2-definable.

Here, *counting monadic second-order logic* CMSO_2 is the extension of first-order logic by (1) variables that range over sets of vertices or edges, or over edges, (2) atomic formulas $\mathrm{inc}(x, y)$ which evaluate to true if x is assigned a vertex v and y is assigned an edge e such that v is incident with e, and (3) atomic formulas $\mathrm{card}_{p,q}(X)$ for integers $p, q \in \mathbb{N}$ and set variables X expressing that $|X| \equiv p \mod q$. See [15] for further details.

3 Decidability

As a first step, we show that the problem $\textsc{HomInd}(\mathcal{F})$ is decidable for every k-recognisable graph class $\mathcal{F}$ of treewidth at most $k-1$. We do so by establishing a bound on the maximum size of a graph $F \in \mathcal{F}$ for which $\hom(F, G) = \hom(F, H)$ needs to be checked in order to conclude whether $G \equiv_{\mathcal{F}} H$. For a graph class $\mathcal{F}$ and $\ell \in \mathbb{N}$, define the class $\mathcal{F}_{\leq \ell} := \{F \in \mathcal{F} \mid |V(F)| \leq \ell\}$.

▶ **Theorem 13.** *Let $k \geq 1$. Let $\mathcal{F}$ be a graph class of treewidth $\leq k-1$ with k-recognisability index C. For graphs G and H on at most n vertices, with $f_{k,C}(n) := \max\{k^{2Cn^k}, 2Cn^k\}$,*

$$G \equiv_{\mathcal{F}} H \iff G \equiv_{\mathcal{F}_{\leq f_{k,C}(n)}} H.$$

Fix throughout a graph class $\mathcal{F}$ as in Theorem 13. In reminiscence of Courcelle's theorem, we let Q denote the set of equivalence classes of $\sim^k_{\mathcal{F}}$, as defined in Definition 10, and call them *states*. A state $q \in Q$ is *accepting* if for an (and equivalently, every) $\boldsymbol{F}$ in q it holds that $\mathrm{soe}(\boldsymbol{F}) \in \mathcal{F}$. Write $A \subseteq Q$ for the set of all accepting states.

To every state $q \in Q$, we associate a finite-dimensional vector space spanned by the homomorphism tensors of the k-labelled graphs $\boldsymbol{F}$ that belong to state q. We show that these vector spaces certify homomorphism indistinguishability. Using a dimensionality argument, we show that these vector spaces are spanned by homomorphism tensors of graphs whose size is bounded by the function f from Theorem 13. To that end, we decompose the labelled graphs $\boldsymbol{F} \in \mathcal{TW}(k)$ using the operations considered in Lemma 9.

Formally, we associate to a state $q \in Q$ and an integer $d \geq 1$ the vector space[1]

$$S_d(q) := \langle \{\boldsymbol{F}_G \oplus \boldsymbol{F}_H \mid \boldsymbol{F} \in \mathcal{TW}_d(k) \text{ in state } q\} \rangle \subseteq \mathbb{R}^{V(G)^k \cup V(H)^k}.$$

Here, $\boldsymbol{F}$ is a k-labelled graph of bounded treewidth in the state q. The vector $\boldsymbol{F}_G \oplus \boldsymbol{F}_H := \binom{\boldsymbol{F}_G}{\boldsymbol{F}_H} \in \mathbb{R}^{V(G)^k \cup V(H)^k}$ is obtained by stacking the homomorphisms vectors of $\boldsymbol{F}$ w.r.t. G and H. Since $\mathcal{TW}_d(k) \subseteq \mathcal{TW}_{d+1}(q)$, the space $S_d(q)$ is a subspace of $S_{d+1}(q)$ for every $d \geq 1$. Ultimately, we are interested in $S(q) := \bigcup_{d \geq 1} S_d(q)$, i.e. the vector space spanned by the homomorphism vectors of all labelled graphs of treewidth $\leq k-1$ in state q.

By the following Lemma 14, the vectors in $S(q)$ for $q \in A$ can be used to infer whether G and H are homomorphism indistinguishable over $\mathcal{F}$. For a labelled graph $\boldsymbol{F} \in \mathcal{TW}(k)$, the number $\mathbf{1}_G^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H)$ is equal to the number of homomorphisms from the underlying unlabelled graph of $\boldsymbol{F}$ to G. This observation readily yields the backward implication in Lemma 14. For the forward implication, observe that the space $S(q)$ is spanned by homomorphism tensors $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$, which satisfy the assertion by assumption.

▶ **Lemma 14.** *Two graph G and H are homomorphism indistinguishable over $\mathcal{F}_{\geq k} := \{F \in \mathcal{F} \mid |V(F)| \geq k\}$ if and only if $\mathbf{1}_G^T v = \mathbf{1}_H^T v$ for every $q \in A$ and every $v \in S(q)$.*

---

[1] Wlog we may suppose that $V(G)$ and $V(H)$ are disjoint.

Proof. For the forward direction, note that $S(q)$ is spanned by the $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$ where $\boldsymbol{F} \in \mathcal{TW}(k)$ is in state q. Let $\boldsymbol{F}$ in $q \in A$ be arbitrary. Then $\operatorname{soe}(\boldsymbol{F}) =: F \in \mathcal{F}_{\geq k}$ and it holds that $\boldsymbol{1}_G^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H) = \operatorname{soe}(\boldsymbol{F}_G) = \hom(F, G) = \hom(F, H) = \boldsymbol{1}_H^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H)$. Conversely, let $F \in \mathcal{F}_{\geq k}$ be arbitrary. Since $\operatorname{tw} F \leq k-1$, by Lemma 6, there exists $\boldsymbol{u} \in V(F)^k$ such that $\boldsymbol{F} := (F, \boldsymbol{u}) \in \mathcal{TW}(k)$. Furthermore, $\boldsymbol{F}$ belongs to some accepting $q \in Q$. Thus, $\boldsymbol{F}_G \oplus \boldsymbol{F}_H \in S(q)$, and hence $\hom(F, G) = \boldsymbol{1}_G^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H) = \boldsymbol{1}_H^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H) = \hom(F, H)$. ◀

By Lemma 8, the space $S_d(q)$ for $d \geq 1$ is spanned by homomorphism tensors of graphs of size $\max\{k^d, d\}$. Thus, Theorem 13 follows once we establish that $S_{d'}(q) = S(q)$ for all $q \in Q$ and $d' := 2Cn^k$. This d' arises as an upper bound on the dimension of the space $\bigoplus_{q \in Q} S(q)$. The spaces $\bigoplus_{q \in Q} S_d(q)$ for $d \geq 1$ form a chain of nested subspaces in $\bigoplus_{q \in Q} S(q)$. The following Lemma 15 shows that once this chain becomes stationary, then the maximal subspace is reached.

▶ **Lemma 15.** *If $S_d(q) = S_{d+1}(q)$ for $d \geq 1$ and all $q \in Q$, then $S_d(q) = S(q)$ for all $q \in Q$. In particular, $S_{2Cn^k}(q) = S(q)$ for all $q \in Q$.*

The proof of Lemma 15 relies on the properties of the relation $\sim_{\mathcal{F}}^k$. In particular, it uses the fact that series composition and gluing, the operations under which $\mathcal{TW}(k)$ is generated by Lemma 9, preserve the relation $\sim_{\mathcal{F}}^k$.

▶ **Lemma 16.** *For $\boldsymbol{F}, \boldsymbol{F}', \boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_1', \boldsymbol{F}_2' \in \mathcal{D}(k)$, $\boldsymbol{L} \in \mathcal{D}(k, k)$,*
1. *if $\boldsymbol{F}_1 \sim_{\mathcal{F}}^k \boldsymbol{F}_1'$ and $\boldsymbol{F}_2 \sim_{\mathcal{F}}^k \boldsymbol{F}_2'$, then $\boldsymbol{F}_1 \odot \boldsymbol{F}_2 \sim_{\mathcal{F}}^k \boldsymbol{F}_1' \odot \boldsymbol{F}_2'$,*
2. *if $\boldsymbol{F} \sim_{\mathcal{F}}^k \boldsymbol{F}'$, then $\boldsymbol{L} \cdot \boldsymbol{F} \sim_{\mathcal{F}}^k \boldsymbol{L} \cdot \boldsymbol{F}'$.*

Proof. Let $\boldsymbol{K} = (K, \boldsymbol{u}) \in \mathcal{D}(k)$ be arbitrary. Then $\operatorname{soe}((\boldsymbol{K} \odot \boldsymbol{F}_1) \odot \boldsymbol{F}_2) \in \mathcal{F} \Leftrightarrow \operatorname{soe}((\boldsymbol{K} \odot \boldsymbol{F}_1) \odot \boldsymbol{F}_2') \in \mathcal{F} \Leftrightarrow \operatorname{soe}(\boldsymbol{K} \odot \boldsymbol{F}_1' \odot \boldsymbol{F}_2') \in \mathcal{F}$.

For a (k, k)-bilabelled graph $\boldsymbol{L} = (L, \boldsymbol{u}, \boldsymbol{v})$, write $\boldsymbol{L}^* := (L, \boldsymbol{v}, \boldsymbol{u})$ for the (k, k)-bilabelled graph obtained by swapping the in-labels and out-labels. Then $\operatorname{soe}(\boldsymbol{K} \odot (\boldsymbol{L} \cdot \boldsymbol{F})) = \operatorname{soe}((\boldsymbol{L}^* \cdot \boldsymbol{K}) \odot \boldsymbol{F})$. Thus the second claim follows from the first. ◀

The algebraic operations on homomorphism tensors corresponding to series composition and gluing are the matrix-vector product and Schur product. Crucially, these operations are linear and bilinear, respectively. This allows Lemma 15 to be proven by structural induction along Lemma 9.

Proof of Lemma 15. We argue that $S_d(q) \supseteq S_{d+i}(q)$ for all $i \geq 1$ by induction on i. The base case holds by assumption. The space $S_{d+i+1}(q)$ is spanned by the vectors $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$ where $\boldsymbol{F} \in \mathcal{TW}_{d+i+1}(k)$ is in q. For such $\boldsymbol{F}$, by Lemma 9, there exist $A \subseteq \binom{[k]}{2}$, $L \subseteq [k]$, and $\boldsymbol{F}^\ell \in \mathcal{TW}_{d+i}(k)$ for $\ell \in L$ such that

$$\boldsymbol{F} = \prod_{ij \in A} \boldsymbol{A}^{ij} \cdot \bigodot_{\ell \in L} \boldsymbol{J}^\ell \boldsymbol{F}^\ell.$$

Let q_ℓ denote the state of $\boldsymbol{F}^\ell$. By assumption, there exist $\boldsymbol{K}^{\ell m} \in \mathcal{TW}_d(k)$ in state q_ℓ and $\alpha_m \in \mathbb{R}$ such that $\boldsymbol{F}_G^\ell \oplus \boldsymbol{F}_H^\ell = \sum \alpha_m \boldsymbol{K}_G^{\ell m} \oplus \boldsymbol{K}_H^{\ell m}$. By Lemma 16,

$$\boldsymbol{F} \sim_{\mathcal{F}}^k \prod_{ij \in A} \boldsymbol{A}^{ij} \cdot \bigodot_{\ell \in L} \boldsymbol{J}^\ell \boldsymbol{K}^{\ell m}$$

for all m. Thus, $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$ can be written as linear combination of vectors in $S_{d+i}(q) \subseteq S_d(q)$, by induction. For the final claim, consider the chain of nested subspaces

$$\bigoplus_{q \in Q} S_1(q) \subseteq \bigoplus_{q \in Q} S_2(q) \subseteq \cdots \subseteq \bigoplus_{q \in Q} S(q).$$

By what was just shown, for every $d \geq 1$, either $\bigoplus_{q \in Q} S_d(q)$ is a proper subspace of $\bigoplus_{q \in Q} S_{d+1}(q)$ or $\bigoplus_{q \in Q} S_d(q) = \bigoplus_{q \in Q} S(q)$. Since the dimension of $\bigoplus_{q \in Q} S(q)$ is at most $2Cn^k$, the chain becomes stationary after at most $2Cn^k$ steps. ◀

This concludes the preparations for the proof of Theorem 13.

Proof of Theorem 13. It suffices to prove the backward implication. Since $k \leq k^{2Cn^k}$, it suffices to show that $G \equiv_{\mathcal{F}_{\geq k}} H$ by verifying the condition in Lemma 14. By Lemma 15, $S_d(q) = S(q)$ for $d := 2Cn^k$ and all $q \in Q$. Hence, $S(q)$ is spanned by the $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$ where $\boldsymbol{F} \in \mathcal{TW}_d(k)$ is in state q. By Lemma 8, these graphs have at most $\max\{k^d, d\} = \max\{k^{2Cn^k}, 2Cn^k\}$ vertices. Thus, $\mathbf{1}_G^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H) = \hom(\operatorname{soe} \boldsymbol{F}, G) = \hom(\operatorname{soe} \boldsymbol{F}, H) = \mathbf{1}_H^T(\boldsymbol{F}_G \oplus \boldsymbol{F}_H)$, as desired. ◀

Finally, we adapt the techniques developed so far to prove the following analogue of Theorem 13 for graph classes of bounded pathwidth. In contrast to Theorem 13, the function in Theorem 17 bounding the size of the graphs which need to be considered is polynomial. The proof is deferred to the full version.

▶ **Theorem 17.** *Let $k \geq 1$. Let $\mathcal{F}$ be a graph class of pathwidth $\leq k - 1$ with k-recognisability index C. For graphs G and H on at most n vertices, with $f_{k,C}(n) := 2Cn^k + k - 1$,*

$$G \equiv_{\mathcal{F}} H \iff G \equiv_{\mathcal{F}_{\leq f_{k,C}(n)}} H.$$

4 Modular Homomorphism Indistinguishability in Polynomial Time

The insight that yielded Theorem 13 is that the chain of vector spaces $S_1(q) \subseteq \cdots \subseteq S_d(q) \subseteq S_{d+1}(q) \subseteq \ldots$ reaches a fixed point after polynomially many steps. In this section, we strengthen this result by showing that bases $B(q)$ for the spaces $S(q)$ can be computed efficiently. A technical difficulty arising here is that the numbers produced in the process can be of doubly exponential magnitude. In order to overcome this problem, we first consider homomorphism indistinguishability modulo primes. See [21, 30], for background on modular homomorphism indistinguishability.

ModHomInd($\mathcal{F}$)
Input Graphs G and H, a prime p in binary.
Question Are G and H homomorphism indistinguishable over $\mathcal{F}$ modulo p?

▶ **Theorem 18.** *Let $k \geq 1$. If $\mathcal{F}$ is a k-recognisable graph class of treewidth $\leq k - 1$, then* ModHomInd($\mathcal{F}$) *is in polynomial time.*

The algorithm yielding Theorem 18 is formally stated as Algorithm 1. The idea is to iteratively compute bases $B(q)$ for the spaces

$$S(q) := \langle \{\boldsymbol{F}_G \oplus \boldsymbol{F}_H \mid \boldsymbol{F} \in \mathcal{TW}(k) \text{ in state } q\} \rangle \subseteq \mathbb{F}_p^{V(G)^k \cup V(H)^k}.$$

Initially, all $B(q)$ are empty. Only $B(q_0)$ where q_0 is the state of $\mathbf{1} \in \mathcal{TW}(k)$ from Figure 1a contains the homomorphism vector $\mathbf{1}_G \oplus \mathbf{1}_H$. Subsequently, the operations from Lemma 9 are applied to compute new homomorphism vectors. For every new vector belonging to state q, it is checked whether it is a linear combination of the already computed basis vectors in $B(q)$. If not, it is added to $B(q)$. Analogous to Lemma 15, this process reaches a fixed point after a polynomial number of iterations. At this point, the computed $B(q)$ are bases for the $S(q)$. Finally, Lemma 14 can be invoked to conclude whether the input graphs are homomorphism indistinguishable over $\mathcal{F}$ modulo p.

Algorithm 1 is supplied with a hard-coded description of the graph class $\mathcal{F}$. To that end, consider the following objects. Write $\pi\colon \mathcal{TW}(k) \to Q$ for the map that associates an $\boldsymbol{F} \in \mathcal{TW}(k)$ to its state $q \in Q$. Write q_0 for the state of $\mathbf{1} \in \mathcal{TW}(k)$. Furthermore, write $g\colon Q \times Q \to Q$ and $b_{\boldsymbol{B}}\colon Q \to Q$ for every $\boldsymbol{B} \in \mathcal{B}(k)$ such that

$$g(\pi(\boldsymbol{F}), \pi(\boldsymbol{F}')) = \pi(\boldsymbol{F} \odot \boldsymbol{F}'), \tag{1}$$

$$b_{\boldsymbol{B}}(\pi(\boldsymbol{F})) = \pi(\boldsymbol{B} \cdot \boldsymbol{F}). \tag{2}$$

for every $\boldsymbol{F}, \boldsymbol{F}' \in \mathcal{TW}(k)$ and $\boldsymbol{B} \in \mathcal{B}(k)$. Note that Q, A, g, q_0 and the $b_{\boldsymbol{B}}$, $\boldsymbol{B} \in \mathcal{B}(k)$, are finite objects, which can be hard-coded. The map π does not need to be computable and is only needed for analysing the algorithm.

Algorithm 1 ModHomInd($\mathcal{F}$) for k-recognisable $\mathcal{F}$ of treewidth $\leq k-1$.

Input: graphs G and H, a prime p in binary.
Data: k, Q, A, q_0, g, $b_{\boldsymbol{B}}$ for $\boldsymbol{B} \in \mathcal{B}(k)$.
Output: whether $G \equiv_{\mathcal{F}}^{p} H$.

1 With brute force check whether G and H are homomorphism indistinguishable over the finite graph class $\mathcal{F}_{\leq k}$ modulo p and reject if not;
2 $B(q_0) \leftarrow \{\mathbf{1}_G \oplus \mathbf{1}_H\} \subseteq \mathbb{F}_p^{V(G)^k \cup V(H)^k}$;
3 $B(q) \leftarrow \emptyset \subseteq \mathbb{F}_p^{V(G)^k \cup V(H)^k}$ for all $q \neq q_0$;
4 **repeat**
5   **foreach** $\boldsymbol{B} \in \mathcal{B}(k)$, $q \in Q$, $v \in B(q)$ **do**
6     $w \leftarrow (\boldsymbol{B}_G \oplus \boldsymbol{B}_H)v := \left(\begin{smallmatrix} \boldsymbol{B}_G & 0 \\ 0 & \boldsymbol{B}_H \end{smallmatrix}\right) v$;
7   **if** $w \notin \langle B(b_{\boldsymbol{B}}(q))\rangle$ **then**
8   add w to $B(b_{\boldsymbol{B}}(q))$;
9   **foreach** $q_1, q_2 \in Q$, $v_1 \in B(q_1)$, $v_2 \in B(q_2)$ **do**
10     $w \leftarrow v_1 \odot v_2$;
11   **if** $w \notin \langle B(g(q_1, q_2))\rangle$ **then**
12   add w to $B(g(q_1, q_2))$;
13 **until** *none of the $B(q)$, $q \in Q$, are updated*;
14 **if** $\mathbf{1}_G^T v = \mathbf{1}_H^T v$ *for all $q \in A$ and $v \in B(q)$* **then**
15   accept;
16 **else**
17   reject;

▶ **Lemma 19.** *Write $n := \max\{|V(G)|, |V(H)|\}$ and $C := |Q|$. There exists a computable function f such that Algorithm 1 runs in time $f(k, C)n^{O(k)}(\log p)^{O(1)}$.*

Proof. Consider the following individual runtimes: Counting homomorphisms of a graph on k vertices into a graph on n vertices, can be done in time $O(n^k)$. Hence Line 1 requires time $f(k)n^k$ for some computable function f.

Throughout the execution of Algorithm 1, the vectors in each $B(q)$, $q \in Q$, are linearly independent. Thus, $|B(q)| \leq \dim S(q) \leq 2n^k$ and $\sum_{q \in Q} |B(q)| \leq 2Cn^k$. Hence, the body of the loop in Line 1 is entered at most $2Cn^k$ many times.

The loop in Line 1 iterates over at most $k^2 \cdot C \cdot 2n^k$ many objects. Computing the vector w takes polynomial time in $2n^k \cdot \log p$. The same holds for checking the condition in Line 1, e.g. via Gaussian elimination. The loop in Line 1 iterates over at most $C^2 \cdot (2n^k)^2$ many objects. Finally, checking the condition in Line 1 takes $C \cdot 2n^k \cdot (\log p)^{O(1)}$ many steps. ◀

The following Lemma 20 implies that Algorithm 1 is correct.

▶ **Lemma 20.** *When Algorithm 1 terminates, $B(q)$ spans $S(q)$ for all $q \in Q$.*

Proof. First observe that the invariant $B(q) \subseteq S(q)$ for all $q \in Q$ is preserved throughout Algorithm 1. Indeed, for example in Line 1, since $v \in B(q) \subseteq S(q)$, it can be written as linear combination of $\boldsymbol{F}_G \oplus \boldsymbol{F}_H$ for $\boldsymbol{F} \in \mathcal{TW}(k)$ of state q. Because $\boldsymbol{B} \cdot \boldsymbol{F}$ is in state $b_{\boldsymbol{B}}(q)$ by Equation (2), $(\boldsymbol{B}_G \oplus \boldsymbol{B}_H)v$ is in the span of $\boldsymbol{B}_G\boldsymbol{F}_G \oplus \boldsymbol{B}_H\boldsymbol{F}_H \in S(b_{\boldsymbol{B}}(q))$.

Now consider the converse inclusion. The proof is by induction on the structure in Lemma 9. By initialisation, $\mathbf{1}_G \oplus \mathbf{1}_H$ is in the span of $B(q_0)$.

For the inductive step, suppose that $\boldsymbol{F} \in \mathcal{TW}(k)$ of state $q \in Q$ is such that $\boldsymbol{F}_G \oplus \boldsymbol{F}_H = \sum_{v \in B(q)} \alpha_v v$ for some coefficients $\alpha_v \in \mathbb{F}_p$. Let $\boldsymbol{B} \in \mathcal{B}(k)$ and $\boldsymbol{F}' := \boldsymbol{B} \cdot \boldsymbol{F}$. Then $(\boldsymbol{B}_G \oplus \boldsymbol{B}_H)v$ is in the span of $B(b_{\boldsymbol{B}}(q))$ for all $v \in B(q)$ by the termination condition. Hence, $\boldsymbol{F}'_G \oplus \boldsymbol{F}'_H = \sum_{v \in B(q)} \alpha_v (\boldsymbol{B}_G \oplus \boldsymbol{B}_H)v$ is in the span of $B(b_{\boldsymbol{B}}(q))$.

Let $\boldsymbol{F}^1, \boldsymbol{F}^2 \in \mathcal{TW}(k)$ of states $q_1, q_2 \in Q$ be such that $\boldsymbol{F}^1_G \oplus \boldsymbol{F}^1_H = \sum_{v \in B(q_1)} \alpha_v v$ and $\boldsymbol{F}^2_G \oplus \boldsymbol{F}^2_H = \sum_{w \in B(q_2)} \beta_w w$ for some coefficients $\alpha_v, \beta_w \in \mathbb{F}_p$. Since the algorithm terminated, all $v \odot w$ for $v \in B(q_1)$ and $w \in B(q_2)$ are in the span of $B(g(q_1, q_2))$. Then $(\boldsymbol{F}^1 \odot \boldsymbol{F}^2)_G \oplus (\boldsymbol{F}^1 \odot \boldsymbol{F}^2)_H = (\boldsymbol{F}^1_G \oplus \boldsymbol{F}^1_H) \odot (\boldsymbol{F}^2_G \oplus \boldsymbol{F}^2_H) = \sum_{v \in B(q_1), w \in B(q_2)} \alpha_v \beta_w (v \odot w)$ is in the span of $B(g(q_1, q_2))$. ◀

This concludes the preparations for the proof of Theorem 18:

Proof of Theorem 18. Lemma 20 implies that the conditions in Lemma 14 and Line 1 are equivalent. Thus, G and H are homomorphism indistinguishable over $\mathcal{F}_{\geq k}$ modulo p if and only if the condition in Line 1 holds. The runtime bound is given in Lemma 19. ◀

5 Randomised Polynomial Time

In this section, we give a randomised polynomial-time reduction from HomInd($\mathcal{F}$) to ModHomInd($\mathcal{F}$). Thereby, we prove Theorems 1 and 2. Theorems 13 and 17 give bounds N on the size of the largest graph in $\mathcal{F}$ which needs to be considered in order to conclude whether two graphs on at most n vertices are homomorphism indistinguishable over $\mathcal{F}$. A graph on at most N vertices may have at most n^N homomorphisms to a graph on n vertices. Thus, for graphs on at most n vertices, homomorphism indistinguishability over $\mathcal{F}$ is the same as homomorphism indistinguishability over $\mathcal{F}$ modulo any number greater than n^N. Equipped with the following Lemma 21, which is derived from the Chinese Remainder Theorem and the Prime Number Theorem, we show Theorems 1 and 2. Let log denote the logarithm to base 2.

▶ **Lemma 21.** *Let $N, n \in \mathbb{N}$ be such that $N \log n \geq e^{2000}$. Let $\mathcal{F}$ be a graph class and G and H be graphs on at most n vertices. If $G \not\equiv_{\mathcal{F}_{\leq N}} H$ then the probability that a random prime $N \log n < p \leq (N \log n)^2$ is such that $G \equiv^p_{\mathcal{F}_{\leq N}} H$ is at most $\frac{2}{N \log n}$.*

For graph classes of bounded pathwidth, the bound on N from Theorem 17 is polynomial in n. Thus, one can enumerate all primes $N \log n < p \leq (N \log n)^2$ in polynomial time and invoke ModHomInd($\mathcal{F}$).

▶ **Theorem 2.** *Let $k \geq 1$. If $\mathcal{F}$ is a k-recognisable class of graphs of pathwidth at most $k-1$, then HomInd($\mathcal{F}$) is in polynomial time.*

For graph classes of bounded treewidth, the bound on N from Theorem 13 is exponential in n. The randomised algorithm yielding Theorem 1 stated in the full version samples primes of polynomial size and invokes ModHomInd($\mathcal{F}$). Lemma 21 implies that a random prime p of appropriate size certifies that $G \not\equiv_{\mathcal{F}} H$ with high probability. This yields Theorem 1:

▶ **Theorem 1.** *Let $k \geq 1$. If $\mathcal{F}$ is a k-recognisable class of graphs of treewidth at most $k-1$, then* HomInd($\mathcal{F}$) *is in* coRP.

6 Fixed-Parameter Tractability

In this section, we deduce Theorem 3 from Theorem 1. The challenge is to efficiently compute the data describing the graph class $\mathcal{F}_{\varphi,k}$ for Algorithm 1 from the CMSO_2-sentence φ and k. That this can be done was proven by Courcelle [15]. More precisely, Courcelle proved that for every CMSO_2-sentence φ and integer k one can compute a finite automaton processing expressions which encode (tree decompositions of) graphs of bounded treewidth. It is this automaton from which the data required by Algorithm 1 can be constructed.

▶ **Theorem 3.** *There exists a computable function $f\colon \mathbb{N} \to \mathbb{N}$ and a randomised algorithm for* HomInd *of runtime $f(|\varphi| + k)n^{O(k)}$ for $n := \max\{|V(G)|, |V(H)|\}$ which accepts all YES-instances and accepts NO-instances with probability less than one half.*

For graph classes of bounded pathwidth, the analogous problem can be decided deterministically in the same time, cf. the full version.

7 Lasserre in Polynomial Time

By phrasing graph isomorphism as an integer program, heuristics from integer programming can be used to attempt to solve graph isomorphism. Prominent heuristics are the Sherali–Adams linear programming hierarchy and the Lasserre semidefinite programming hierarchy. While the (approximate) feasibility of each level of these hierarchies can be decided efficiently, it is known that a linear number of levels is required to decide graph isomorphism for all graphs [4, 27, 38].

In [38], for each $t \geq 1$, feasibility of the t-th level of the Lasserre hierarchy was characterised as homomorphism indistinguishability relation over the graph class $\mathcal{L}_t$ which was constructed in the same paper. Moreover, the authors asked whether there is a polynomial-time algorithm for deciding these relations. In this section, we give a randomised algorithm for this problem which is polynomial-time for every level.

The Lasserre semidefinite program can be solved approximately in polynomial time using e.g. the ellipsoid method. How to decide *exact* feasibility is generally unknown [2]. Since the graph class $\mathcal{L}_t$ is a minor-closed and of treewidth at most $3t-1$, Theorem 1 immediately yields a randomised polynomial-time algorithm for each level of the hierarchy. However, it is not clear how to compute the data describing $\mathcal{L}_t$ given t. The following Theorem 22 overcomes this problem by making the dependence on the parameter t effective:

▶ **Theorem 22.** *There exists a computable function $f\colon \mathbb{N} \to \mathbb{N}$ and a randomised algorithm deciding given graphs G and H on at most n vertices and an integer $t \geq 1$ whether the level-t Lasserre relaxation of the integer program for $G \cong H$ has an exact solution. This algorithm always runs in time $f(t)n^{O(t)}$, accepts all YES-instances and accepts NO-instances with probability less than one half.*

8 Lower Bounds

In this final section, we establish two hardness results for the problem HomInd. In both cases, we show hardness for families of minor-closed graph classes. The approaches are orthogonal in the sense that the reduction yielding coNP-hardness in Theorem 5 is from a fixed-parameter tractable problem while the reduction yielding coW[1]-hardness in Theorem 4 is not polynomial-time.

coNP-Hardness. The first hardness result concerns deciding whether two graphs are indistinguishable under the k-dimensional Weisfeiler–Leman algorithm when k is part of the input. By [20, 13, 19], k-WL indistinguishability coincides with homomorphism indistinguishability over the class of graphs of treewidth at most k. Hence, the problem in Theorem 5 is clearly a special case of HomInd, i.e. with φ set to true. Thus, when disregarding the parametrisation, HomInd is coNP-hard under polynomial-time many-one reductions. We obtain Theorem 5 by reducing the NP-complete problem of deciding whether a graph of bounded degree has treewidth $\leq k$ [10]. The reduction is based on the ubiquitous CFI construction [13].

▶ **Theorem 5.** *The problem of deciding given graphs G and H and an integer $k \in \mathbb{N}$ whether G and H are k-*WL *indistinguishable is* coNP*-hard under polynomial-time many-one reductions.*

Towards the proof of Theorem 5, we recall the following version of the classical CFI graphs [13] from [37]. Let G be a connected graph and $U\colon V(G) \to \mathbb{Z}_2$ a function from G to the group on two elements $\mathbb{Z}_2$. For a vertex $v \in V(G)$, write $E(v) \subseteq E(G)$ for the set of edges incident to v. The graph G_U has vertices (v, S) for every $v \in V(G)$ and $S\colon E(v) \to \mathbb{Z}_2$ such that $\sum_{e \in E(v)} S(e) = U(v)$. Two vertices (u, S) and (v, T) are adjacent in G_U if $uv \in E(G)$ and $S(uv) + T(uv) = 0$. Note that $|V(G_U)| = \sum_{v \in V(G)} 2^{\deg(v)-1}$. By [37, Lemma 3.2], if $\sum_{v \in V(G)} U(v) = \sum_{v \in V(G)} U'(v)$ for $U, U'\colon V(G) \to \mathbb{Z}_2$, then $G_U \cong G_{U'}$. We may thus write G_0 and G_1 for the *even* and the *odd* CFI graph of G. We recall the following properties:

▶ **Lemma 23** ([37, Corollary 3.7])**.** *Let G be a connected graph and $U\colon V(G) \to \mathbb{Z}_2$. Then the following are equivalent:*

1. $G_0 \cong G_U$,
2. $\sum_{v \in V(G)} U(v) = 0$,
3. $\hom(G, G_0) = \hom(G, G_U)$.

Proof of Theorem 5. For a graph G, write $\Delta(G)$ for its maximum vertex degree. The following problem is NP-complete by [10, Theorem 11]:

BoundedDegreeTreewidth
Input a graph G with $\Delta(G) \leq 9$, an integer k
Question Is $\operatorname{tw} G \leq k$?

By deleting isolated vertices, we may suppose that every connected component of G contains at least two vertices. If G has multiple connected components, take one vertex from each component and connect them in a pathlike fashion. This increases the maximum degree potentially by one but makes the graph connected. The treewidth is invariant under this operation. Thus, we may suppose that G is connected and $\Delta(G) \leq 10$.

Given such an instance, we produce the instance (G_0, G_1, k) of WL, i.e. the decision problem in Theorem 5. Here, G_0 and G_1 are the even and odd CFI graphs of G.

Then G_0 and G_1 are k-WL indistinguishable if and only if $\operatorname{tw} G \geq k+1$. Indeed, by [13, 20] and [36, Lemma 4.4], since G is connected, if $\operatorname{tw} G \geq k+1$, then G_0 and G_1 are k-WL indistinguishable. Conversely, if $\operatorname{tw} G < k+1$, then G_0 and G_1 are distinguished by k-WL since $\hom(G, G_0) \neq \hom(G, G_1)$ by Lemma 23 and [20]. Hence, (G, k) is a YES-instance of BoundedDegreeTreewidth if and only if (G_0, G_1, k) is a NO-instance of WL. The graphs G_0 and G_1 are of size $\sum_{v \in V(G)} 2^{\deg(v)-1} \leq 2^9 n$, which is polynomial in the input. ◀

coW[1]-Hardness. The second hardness result concerns HomInd as a parametrised problem. Write $\mathcal{G}_{\leq k}$ for the class of all graphs on at most k vertices and consider the following problem:

HomIndSize
Input Graphs G and H, an integer $k \geq 1$.
Parameter k.
Question Are G and H homomorphism indistinguishable over the class $\mathcal{G}_{\leq k}$?

The problem HomIndSize fixed-parameter reduces to HomInd. To that end, consider the first-order formula $\varphi_k \coloneqq \exists x_1 \dots \exists x_k \forall y \bigvee_{i=1}^{k} (y = x_i)$ for $k \in \mathbb{N}$. Then, a graph models φ_k if and only if it has at most k vertices. Furthermore, $|\varphi_k| = O(k)$. Hence, transforming the instance (G, H, k) of HomIndSize to the instance $(G, H, \varphi_k, k-1)$ of HomInd gives the desired reduction. Since $|\varphi_k| + k = O(k)$, Theorem 24 implies Theorem 4.

▶ **Theorem 24.** HomIndSize *is* coW[1]*-hard under fpt-reductions. Unless* ETH *fails, there is no algorithm for* HomIndSize *that runs in time* $f(k)n^{o(k)}$ *for any computable function* $f \colon \mathbb{N} \to \mathbb{N}$.

Proof. The proof is by reduction from the parametrised clique problem Clique, which is well-known to be W[1]-complete and which does not admit an $f(k)n^{o(k)}$-time algorithm for any computable function f unless ETH fails [16, Theorems 13.25, 14.21].

Let K denote the k-vertex complete graph and K_0 and K_1 the even and odd CFI graphs of K. We first observe that $K_0 \equiv_{\mathcal{G}_{\leq k} \setminus \{K\}} K_1$. Indeed, by [37, Theorem 3.13], for every graph F, $\hom(F, K_0) \neq \hom(F, K_1)$ if and only if there exists a *weak oddomorphism* $h \colon F \to K$ as defined in [37, Definition 3.9]. By definition, a weak oddomorphism is a homomorphism which is surjective on edges and vertices, i.e. for every $uv \in E(K)$ there exists $u'v' \in E(F)$ such that $h(u'v') = uv$. Hence, if $\hom(F, K_0) \neq \hom(F, K_1)$, then F has at least k vertices and $\binom{k}{2}$ edges. The only graph in $\mathcal{G}_{\leq k}$ matching this description is K.

The reduction produces given the instance (G, k) of Clique the instance $(G \times K_0, G \times K_1, k)$ of HomIndSize where K is the k-vertex clique and $\times$ denotes the categorical product (also known as tensor product) of two graphs. Producing this instance is fixed-parameter tractable. Furthermore, the parameter k is not affected by this reduction. For correctness, consider the following argument.

If $G \times K_0 \equiv_{\mathcal{G}_{\leq k}} G \times K_1$, then $\hom(K, G) = 0$. Indeed, by assumption and [33, (5.30)], $\hom(K, G) \hom(K, K_0) = \hom(K, G \times K_0) = \hom(K, G \times K_1) = \hom(K, G) \hom(K, K_1)$. However, $\hom(K, K_0) \neq \hom(K, K_1)$ by Lemma 23, and thus $\hom(K, G) = 0$.

Conversely, it holds that $K_0 \equiv_{\mathcal{G}_{\leq k} \setminus \{K\}} K_1$ and hence also $G \times K_0 \equiv_{\mathcal{G}_{\leq k} \setminus \{K\}} G \times K_1$ by the initial observation and [33, (5.30)]. Since $\hom(K, G) = 0$, also $G \times K_0 \equiv_{\mathcal{G}_{\leq k}} G \times K_1$. ◀

9 A Trichotomy for Homomorphism Indistinguishability?

Theorem 1, our central result, asserts that deciding homomorphism indistinguishability is tractable over every recognisable graph class of bounded treewidth. In particular, Theorem 1 shows that $\textsc{HomInd}(\mathcal{F})$ is tractable for every minor-closed graph class of bounded treewidth. Notably, this result does not rely on reformulations of homomorphism indistinguishability relations in terms of logic etc. but operates with the homomorphism counts themselves.

A reasonable next step is to combine Theorem 1 with a hardness result. To that end, we propose the following working hypothesis:

▶ **Conjecture 25.** *Let $\mathcal{F}$ be a minor-closed graph class.*

1. *If $\mathcal{F}$ is the class of all graphs, then $\textsc{HomInd}(\mathcal{F})$ is graph isomorphism.*
2. *If $\mathcal{F}$ has bounded treewidth, then $\textsc{HomInd}(\mathcal{F})$ is in polynomial time.*
3. *If $\mathcal{F}$ is proper and has unbounded treewidth, then $\textsc{HomInd}(\mathcal{F})$ is undecidable.*

The first assertion is implied by [32]. The second assertions amounts to derandomising Theorem 1 and is predicted by the complexity-theoretic hypothesis P = BPP. The third assertion is wide open: The only minor-closed graph class $\mathcal{F}$ for which $\textsc{HomInd}(\mathcal{F})$ is known to be undecidable, is the class $\mathcal{P}$ of planar graphs, as shown by Mančinska and Roberson [34]. Conjecture 25 is inspired by this example and a result from graph minor theory [39] which asserts that every minor-closed graph class is either of bounded treewidth or contains all planar graphs. Intuitively, $\textsc{HomInd}(\mathcal{P})$ is undecidable since the problem amounts to solving an infinite-dimensional system of equations. Roughly speaking, the dimension corresponds to the number of labels needed to generate all planar graphs under operations like series composition. Theorem 1 makes the other direction of this vague argument precise: We show that if the number of labels is bounded (e.g. the graph class has bounded treewidth), then considering finite-dimensional spaces suffices, rendering the problem tractable. That treewidth might be the right parameter in Conjecture 25 is also suggested by the complexity dichotomy for counting homomorphisms [17].

Conjecture 25 implies a weak version of Roberson's conjecture [37, Conjecture 5] asserting that $\equiv_{\mathcal{F}}$ *is not the isomorphism relation* $\cong$ *for every proper minor-closed graph class* $\mathcal{F}$. Towards Conjecture 25, one could devise reductions between $\textsc{HomInd}(\mathcal{F}_1)$ and $\textsc{HomInd}(\mathcal{F}_2)$ for distinct minor-closed graph classes $\mathcal{F}_1$ and $\mathcal{F}_2$. We are not aware of any such reduction.

Another pathway to Conjecture 25 is suggested by Theorems 13 and 17: Instead of proving hardness of the problem $\textsc{HomInd}(\mathcal{F})$, one may attempt to give lower bounds on any function[2] $f \colon \mathbb{N} \to \mathbb{N}$ such that $G \equiv_{\mathcal{F}} H$ if and only if $G \equiv_{\mathcal{F}_{\leq f(n)}} H$ for all graphs G and H on at most n vertices. This problem is purely combinatorial and avoids the intricacies of computation. Theorems 13 and 17 give such functions for every recognisable graph class of bounded treewidth. By [32], for the class $\mathcal{G}$ of all graphs, f can be taken to be the identity $n \mapsto n$. By [34], there exists no such function for the class $\mathcal{P}$ of all planar graphs which is computable. Conjecture 25 implies that no such function is computable for any minor-closed graph class of unbounded treewidth.

▶ **Question 26.** *For a graph class $\mathcal{F}$, what is the least function $f \colon \mathbb{N} \to \mathbb{N}$ such that*

$$G \equiv_{\mathcal{F}} H \iff G \equiv_{\mathcal{F}_{\leq f(n)}} H$$

for all graphs G and H on at most n vertices?

[2] Such a function always exists since there are only finitely many equivalences classes of $\equiv_{\mathcal{F}}$ on graphs on most n vertices, each pair of which is distinguished by homomorphism counts from a single graph $F \in \mathcal{F}$.

References

1 Samson Abramsky, Tomáš Jakl, and Thomas Paine. Discrete Density Comonads and Graph Parameters. In Helle Hvid Hansen and Fabio Zanasi, editors, *Coalgebraic Methods in Computer Science*, pages 23–44, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-10736-8_2.

2 Albert Atserias and Joanna Fijalkow. Definable Ellipsoid Method, Sums-of-Squares Proofs, and the Graph Isomorphism Problem. *SIAM Journal on Computing*, 52(5):1193–1229, 2023. doi:10.1137/20M1338435.

3 Albert Atserias, Phokion G. Kolaitis, and Wei-Lin Wu. On the Expressive Power of Homomorphism Counts. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13, 2021. doi:10.1109/LICS52264.2021.9470543.

4 Albert Atserias and Elitza Maneva. Sherali-Adams Relaxations and Indistinguishability in Counting Logics. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 367–379, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090265.

5 Albert Atserias, Laura Mančinska, David E. Roberson, Robert Šámal, Simone Severini, and Antonios Varvitsiotis. Quantum and non-signalling graph isomorphisms. *J. Comb. Theory, Ser. B*, 136:289–328, 2019. doi:10.1016/j.jctb.2018.11.002.

6 László Babai. Graph Isomorphism in Quasipolynomial Time [Extended Abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 684–697, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897542.

7 Christoph Berkholz. Lower Bounds for Existential Pebble Games and k-Consistency Tests. In *2012 27th Annual IEEE Symposium on Logic in Computer Science*, pages 25–34, Dubrovnik, Croatia, June 2012. IEEE. doi:10.1109/LICS.2012.14.

8 Christoph Berkholz, Paul S. Bonsma, and Martin Grohe. Tight Lower and Upper Bounds for the Complexity of Canonical Colour Refinement. *Theory Comput. Syst.*, 60(4):581–614, 2017. doi:10.1007/s00224-016-9686-0.

9 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, December 1998. doi:10.1016/S0304-3975(97)00228-4.

10 Hans L. Bodlaender and Dimitrios M. Thilikos. Treewidth for graphs with small chordality. *Discrete Applied Mathematics*, 79(1-3):45–61, November 1997. doi:10.1016/S0166-218X(97)00031-0.

11 Mikołaj Bojańczyk and Michał Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 407–416, New York NY USA, July 2016. ACM. doi:10.1145/2933575.2934508.

12 Jan Böker, Yijia Chen, Martin Grohe, and Gaurav Rattan. The Complexity of Homomorphism Indistinguishability. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:13, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2019.54.

13 Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, December 1992. doi:10.1007/BF01305232.

14 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, March 1990. doi:10.1016/0890-5401(90)90043-H.

15 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach.* Cambridge University Press, USA, 1st edition, 2012.

16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-21275-3.

17 Víctor Dalmau and Peter Jonsson. The Complexity of Counting Homomorphisms Seen from the Other Side. *Theoretical Computer Science*, 329(1):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

18 Anuj Dawar, Tomáš Jakl, and Luca Reggio. Lovász-Type Theorems and Game Comonads. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470609.

19 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Leman. *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 40:1–40:14, 2018. doi:10.4230/LIPICS.ICALP.2018.40.

20 Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, August 2010. doi:10.1002/jgt.20461.

21 John Faben and Mark Jerrum. The Complexity of Parity Graph Homomorphism: An Initial Investigation. *Theory of Computing*, 11(2):35–57, 2015. doi:10.4086/toc.2015.v011a002.

22 Eva Fluck, Tim Seppelt, and Gian Luca Spitzer. Going Deep and Going Wide: Counting Logic and Homomorphism Indistinguishability over Graphs of Bounded Treedepth and Treewidth. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:17, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2024.27.

23 Martin Grohe. Equivalence in Finite-Variable Logics is Complete for Polynomial Time. *Combinatorica*, 19(4):507–532, October 1999. doi:10.1007/s004939970004.

24 Martin Grohe. Counting Bounded Tree Depth Homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 507–520, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394739.

25 Martin Grohe. The Logic of Graph Neural Networks. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–17. IEEE, 2021. doi:10.1109/LICS52264.2021.9470677.

26 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/FOCS57990.2023.00052.

27 Martin Grohe and Martin Otto. Pebble Games and Linear Equations. *The Journal of Symbolic Logic*, 80(3):797–844, 2015. doi:10.1017/jsl.2015.28.

28 Martin Grohe, Gaurav Rattan, and Tim Seppelt. Homomorphism Tensors and Linear Equations. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 70:1–70:20, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2022.70.

29 Neil Immerman and Eric Lander. Describing Graphs: A First-Order Approach to Graph Canonization. In Alan L. Selman, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pages 59–81. Springer New York, New York, NY, 1990. doi:10.1007/978-1-4612-4478-3_5.

30 Moritz Lichter, Benedikt Pago, and Tim Seppelt. Limitations of Game Comonads for Invertible-Map Equivalence via Homomorphism Indistinguishability. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2024.36.

31 Moritz Lichter, Simon Raßmann, and Pascal Schweitzer. Computational complexity of the Weisfeiler–Leman dimension. *CoRR*, abs/2402.11531, 2024. doi:10.48550/arXiv.2402.11531.

32 László Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(3):321–328, September 1967. doi:10.1007/BF02280291.

33 László Lovász. *Large networks and graph limits*. Number volume 60 in American Mathematical Society colloquium publications. American Mathematical Society, Providence, Rhode Island, 2012.

34 Laura Mančinska and David E. Roberson. Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–672, 2020. doi:10.1109/FOCS46700.2020.00067.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4602–4609, July 2019. doi:10.1609/aaai.v33i01.33014602.

36 Daniel Neuen. Homomorphism-Distinguishing Closedness for Graphs of Bounded Tree-Width. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:12, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2024.53.

37 David E. Roberson. Oddomorphisms and homomorphism indistinguishability over graphs of bounded degree, June 2022. URL: http://arxiv.org/abs/2206.10321.

38 David E. Roberson and Tim Seppelt. Lasserre Hierarchy for Graph Isomorphism and Homomorphism Indistinguishability. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 101:1–101:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2023.101.

39 Neil Robertson and P.D Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

40 Tim Seppelt. Logical Equivalences, Homomorphism Indistinguishability, and Forbidden Minors. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.82.

41 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2018. URL: https://openreview.net/forum?id=ryGs6iA5Km.