Homomorphism Indistinguishability, Multiplicity Automata Equivalence, and Polynomial Identity Testing

Marek Černý
Universiteit Antwerpen, Belgium

Tim Seppelt
IT-Universitetet i København, Denmark

Abstract

Two graphs G and H are *homomorphism indistinguishable* over a graph class $\mathcal{F}$ if they admit the same number of homomorphisms from every graph $F \in \mathcal{F}$. Many graph isomorphism relaxations such as (quantum) isomorphism and cospectrality can be characterised as homomorphism indistinguishability over specific graph classes. Thereby, the problems HomInd($\mathcal{F}$) of deciding homomorphism indistinguishability over $\mathcal{F}$ subsume diverse graph isomorphism relaxations whose complexities range from logspace to undecidable. Establishing the first general result on the complexity of HomInd($\mathcal{F}$), Seppelt (MFCS 2024) showed that HomInd($\mathcal{F}$) is in randomised polynomial time for every graph class $\mathcal{F}$ of bounded treewidth that can be defined in counting monadic second-order logic CMSO_2.

We show that this algorithm is conditionally optimal, i.e. it cannot be derandomised unless polynomial identity testing is in P. For CMSO_2-definable graph classes $\mathcal{F}$ of bounded pathwidth, we improve the previous complexity upper bound for HomInd($\mathcal{F}$) from P to $\mathrm{C}_=\mathrm{L}$ and show that this is tight. Secondarily, we establish a connection between homomorphism indistinguishability and multiplicity automata equivalence which allows us to pinpoint the complexity of the latter problem as $\mathrm{C}_=\mathrm{L}$-complete.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Formal languages and automata theory; Theory of computation → Finite Model Theory

Keywords and phrases treewidth, Courcelle's theorem, logspace, multiplicity automata, polynomial identity testing

Digital Object Identifier 10.4230/LIPIcs.STACS.2026.25

Related Version *Full Version*: `https://arxiv.org/abs/2512.13058` [68]

Funding *Marek Černý*: University of Antwerp (BOF, Doctoral project 47103).
Tim Seppelt: European Union (CountHom, 101077083). 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 We would like to acknowledge fruitful discussions with Mikołaj Bojańczyk and Sam van Gool at the Dagstuhl Seminar 25141 "Categories for Automata and Language Theory". Furthermore, we are grateful for discussions with David E. Roberson and Louis Härtel. Finally, we gratefully acknowledge Floris Geerts for drawing our attention to the link between automata and Specht–Wiegmann-type theorems.

1 Introduction

Graph data is ubiquitous: Graphs may represent social networks, transportation networks, chemical or pharmaceutical molecules, databases or program executions. A central task when presented with graph data is detecting whether two graphs are structurally equivalent

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).
Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thắng
Article No. 25; pp. 25:1–25:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

or *isomorphic*. Graph isomorphism, however, is complexity-theoretically elusive [32] and practically not very robust for example w.r.t. noise or perturbations. These limitations motivate the study of *graph isomorphism relaxations*, i.e. equivalence relations between graphs that are coarser than isomorphism. Although a plethora of graph isomorphism relaxations has been proposed and studied in the past decades, they – to our knowledge – lack a coherent theory which explains e.g. their computational complexity. In recent years, homomorphism indistinguishability has emerged as a framework that provides increasingly comprehensive answers to this end.

Two graphs G and H are *homomorphism indistinguishable* over a graph class $\mathcal{F}$ if, for every $F \in \mathcal{F}$, the graphs G and H admit the same number of homomorphisms from the graph F. Many well-studied graph isomorphism relaxations from a wide range of areas can be characterised as homomorphism indistinguishability relations over natural graph classes. Examples include graph isomorphism corresponding to homomorphism indistinguishability over all graphs [43], quantum isomorphism (planar graphs, [47]), cospectrality (cycles), and equivalence under the k-dimensional Weisfeiler–Leman algorithm (graphs of treewidth at most k, [24, 23]). Further examples draw from notions originating in category theory [22, 49], optimisation [30, 8, 46, 23, 31, 53, 56], quantum information theory [41], machine learning [70, 50, 26, 71], and finite model theory [28, 25, 1, 59], see also the monograph [63].

The central decision problem associated with homomorphism indistinguishability is HomInd($\mathcal{F}$) which asks, given input graphs G and H, whether they are homomorphism indistinguishable over a fixed graph class $\mathcal{F}$. For varying $\mathcal{F}$, the problems HomInd($\mathcal{F}$) subsume diverse graph isomorphism relaxations falling into a wide range of complexity classes. For example, for the class of trees $\mathcal{T}$, HomInd($\mathcal{T}$) is in polynomial time via the Weisfeiler–Leman algorithm, most notably, for the class of all planar graphs $\mathcal{P}$, HomInd($\mathcal{P}$) is undecidable [47, 9, 65], and, for the class of all graphs $\mathcal{G}$, HomInd($\mathcal{G}$) is graph isomorphism and thus in quasipolynomial time [10]. Note that, despite $\mathcal{T} \subseteq \mathcal{P} \subseteq \mathcal{G}$, there is no apparent relation between the complexities of the homomorphism indistinguishability problems over these graph classes, see also [63, Table 9.1].

Crucially, by viewing graph isomorphism relaxations as homomorphism indistinguishability relations HomInd($\mathcal{F}$), one may analyse their complexity in terms of properties of the graph classes $\mathcal{F}$, thus facilitating a principled study of the complexity of graph isomorphism relaxations. Adopting this approach, Seppelt [62] showed that HomInd($\mathcal{F}$) is in randomised polynomial time coRP for every recognisable graph class $\mathcal{F}$ of bounded treewidth. Recognisability is a fairly general property of graph classes introduced by Courcelle [19] which can be thought of as roughly equivalent to definability in the rather powerful counting monadic second-order logic CMSO_2 [12]. By the Robertson–Seymour theorem [57], all minor-closed graph classes are recognisable, the former being central to homomorphism indistinguishability [55, 64]. In [62], it was asked whether the algorithm presented there can be derandomised. Our first result asserts that this would be subject to major complexity-theoretic challenges.

▶ **Theorem 1.** *There exists a CMSO_2-definable graph class $\mathcal{F}$ of bounded treewidth such that* HomInd($\mathcal{F}$)*, MTA equivalence, and PIT are logspace many-one interreducible.*

Here, PIT denotes the polynomial identity testing problem, i.e. the problem of deciding whether a polynomial represented by an arithmetic circuit is the zero polynomial. A deterministic polynomial-time algorithm for PIT would have far-reaching complexity-theoretic repercussions [40]. It was shown in [48] that PIT is logspace many-one interreducible with equivalence testing for multiplicity tree automata (MTA). MTAs [11] are an algebraic model of computation: they assign to every tree a number from a field ($\mathbb{Q}$ in our case). One may, for example, think of the trees as representing XML documents and of the numbers as being probabilities. MTAs have a wide range of applications, see [48].

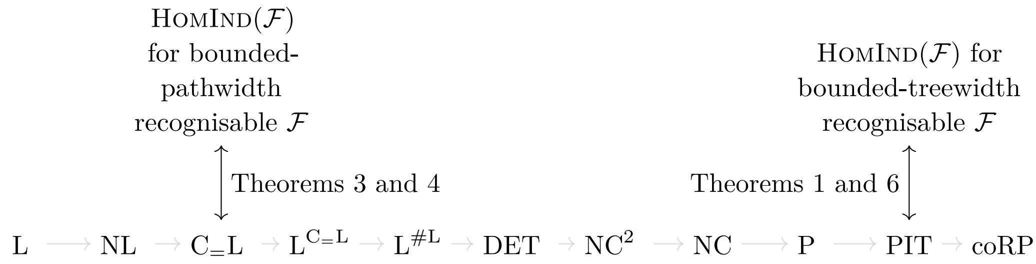


Figure 1 Overview of results relative to some complexity classes. For details on $L^{C_=L}$, $L^{\#L}$, and DET, see [2, 3, 45]. PIT denotes the class of problems logspace many-one reducible to PIT.

As a secondary result, we streamline the reasoning in [62] by reducing HomInd($\mathcal{F}$), for every recognisable graph class $\mathcal{F}$ of bounded treewidth, to MTA equivalence, see Theorem 6. In retrospect, this is quite natural given Courcelle's automata-theoretic motivation for the notion of recognisability [19]. Analogous to the bounded-treewidth case, we show that HomInd($\mathcal{F}$) for recognisable graph classes $\mathcal{F}$ of bounded pathwidth reduces to equivalence of multiplicity word automata (MWA, [61]), see Theorem 6. This does not only simplify the reasoning in [62] but also improves the complexity upper bound for such problems HomInd($\mathcal{F}$), via a classical result of Tzeng [67], from polynomial time to DET, the class of problems that are NC^1-reducible to computing the determinant of integer matrices [18]. Refining Tzeng's analysis, we establish the precise complexity of MWA equivalence. See Figure 1 for an overview of complexity classes.

▶ **Theorem 2.** *MWA equivalence is* $C_=L$*-complete under logspace many-one reductions.*

Combined with the initially described reduction, this yields the following upper bound for the complexity of HomInd($\mathcal{F}$) for recognisable bounded-pathwidth graph classes $\mathcal{F}$.

▶ **Theorem 3.** *For every recognisable graph class* $\mathcal{F}$ *of bounded pathwidth,* HomInd($\mathcal{F}$) $\in C_=L$.

Since $C_=L$ is a subclass of NC^2, Theorem 3 implies that homomorphism indistinguishability over every recognisable bounded-pathwidth graph class can be decided on a parallel computer with polynomially many processors in time $O(\log^2 n)$. Thereby, the theorem establishes a trade-off between distinguishing power and computational complexity. While homomorphism indistinguishability over bounded-pathwidth graph classes is provably weaker than homomorphism indistinguishability over bounded-treewidth graph classes [63, Theorem 6.4.6], it can be decided at a substantially lower computational cost. This is of particular interest for the graph learning community where message-passing graph neural networks are an often-employed paradigm, see [29]. With respect to distinguish power, these architectures correspond to the Weisfeiler–Leman algorithm [70, 50] which cannot be efficiently parallelised unless $P = NC$ [27]. In contrast, our Theorem 3 provides a wealth of efficiently parallisable graph isomorphism relaxations.

In order to get an intuition for the complexity class $C_=L$ as introduced in [5], one may consider its complete problems such as the set of singular integer matrices [3] or the machine characterisation [2, 5] asserting that a language $A \subseteq \{0,1\}^*$ is in $C_=L$ if, and only if, there exists a non-deterministic logspace Turing machine M such that $x \in A$ iff the number of accepting paths equals the number of rejecting paths of M on input x. Many linear-algebraic problems where shown to be complete for $C_=L$ [37, 44], see also [3, 36, 45]. We add to the list of such problems by showing that the upper bound in Theorem 3 is tight.

▶ **Theorem 4.** *There exists a* CMSO_2*-definable graph class* $\mathcal{F}$ *of bounded pathwidth such that* $\textsc{HomInd}(\mathcal{F})$ *is* $\mathrm{C_{=}L}$*-complete under logspace many-one reductions.*

Related Work

Our results provide a complexity-theoretic foundation for the many-faceted connections between systems of equations and homomorphism indistinguishability. Two graphs G and H are isomorphic if, and only if, there exists a permutation matrix X such that $XA_G = A_H X$ where A_G and A_H denote the adjacency matrices of G and H, respectively. When relaxing the constraints on the matrix X, one obtains characterisations of other homomorphism indistinguishability relations. For example, two graphs G and H are homomorphism indistinguishable over all paths iff there exists a pseudo-stochastic matrix X satisfying the above equation [23]. Thus, homomorphism indistinguishability of simple graphs generalises a variety of relaxations of permutation-similarity for symmetric $\{0, 1\}$-matrices. Various relaxations of similarity for integer matrices have been shown [37, 36] to be complete for $\mathrm{C_{=}L}$ and $\mathrm{L^{C_{=}L}}$. Less is known about symmetric $\{0, 1\}$-matrices, see Section 5.3.

The fixed-parameter complexity with respect to logarithmic space of homomorphism counting was studied in [16, 33]. However, this task is only tangentially related to homomorphism indistinguishability. Since the considered graph classes $\mathcal{F}$ are typically infinite, being able to count homomorphisms from $F \in \mathcal{F}$ does not a priori help to decide $\textsc{HomInd}(\mathcal{F})$. In fact, the algorithms in Theorem 3 and [62] compute linear-algebraic invariants rather than homomorphism counts. Conversely, it is not clear how to infer homomorphism counts using $\textsc{HomInd}(\mathcal{F})$.

Finally, we would like to highlight two previous results on the complexity of homomorphism indistinguishability within polynomial time (both results are originally proven in model-theoretic terms but can be recast via [24, 23, 28]): Firstly, Grohe [27] showed that, for every $k \geq 1$, homomorphism indistinguishability over all graphs of treewidth at most k is complete for P under uniform AC^0-reductions. Note that this does not have implications for other graph classes of bounded treewidth, e.g. the class of outerplanar graphs or the class in Theorem 1. Secondly, Raßmann, Schindling, and Schweitzer [52] showed that, for every $k \geq 1$, homomorphism indistinguishability over all graphs of treedepth at most k can be decided in L. Again, this does not have implications for other bounded-treedepth graph classes. Since bounded treedepth implies bounded pathwidth [51], Theorem 3 applies to a much wider class of graph classes while yielding containment in $\mathrm{C_{=}L}$ rather than L.

2 Preliminaries

The natural numbers are $\mathbb{N} = \{0, 1, 2, \dots\}$. In all computational tasks, rationals are encoded as fractions of binary integers.

2.1 Multiplicity automata

A *multiplicity word automaton* (MWA, [61]) is a tuple $\mathcal{A} = (S, \Sigma, M, \alpha, \eta)$, where S is a finite set of states, Σ is a finite alphabet, $\alpha^\top \in \mathbb{Q}^S$ is the *initial vector*, and $\eta \in \mathbb{Q}^S$ is the *final vector*. The map $M \colon \Sigma \to \mathbb{Q}^{S \times S}$ assigns to each letter $a \in \Sigma$ a *transition matrix*. For convenience, we extend M from Σ to all words $w = a_1 \cdots a_t \in \Sigma^*$ by defining $M(w) := M(a_1) \cdots M(a_t)$, so that the empty word ε is mapped to the identity matrix $M(\varepsilon) = I_S$. The automaton $\mathcal{A}$ *recognises* the rational series $[\![\mathcal{A}]\!] \colon \Sigma^* \to \mathbb{Q}$ given by $w \mapsto \alpha^\top M(w)\eta$.

A *finitary type* is a set Ω of symbols with an *arity* map assigning to each symbol $\sigma \in \Omega$ a natural number $|\sigma|$. For $n \in \mathbb{N}$, the set of all n-ary symbols in Ω is denoted by Ω_n. A set of Ω-*trees* denoted by T_Ω is the smallest set such that $\Omega_0 \subseteq T_\Omega$, and if $n \geq 1$, symbol $\sigma \in \Omega_n$ and $t_1, \dots, t_n \in T_\Omega$ then element $\sigma(t_1, \dots, t_n) \in \Omega_n$.

The *Kronecker product* $M \otimes M' \in \mathbb{Q}^{(S\times S')\times(R\times R')}$ of two matrices $M \in \mathbb{Q}^{S\times R}$ and $M' \in \mathbb{Q}^{S'\times R'}$ is given by $(ss', rr') \mapsto M(s,r)\cdot M'(s',r')$. The *direct sum* $M \oplus M' \in \mathbb{Q}^{(S\uplus S')\times(R\uplus R')}$ is given by the original entries of M and M', assigning 0 to the remaining entries in $(R'\times S) \uplus (R \times S')$.

A *multiplicity tree automaton* (MTA, [11]) is a tuple $\mathcal{A} = (S, \Omega, \mu, \eta)$, where S is a finite set of states, Ω is a finitary type, μ is a *tree representation*, i.e. a union of maps $M_n \colon \Omega_n \to \mathbb{Q}^{S^n \times S}$ for each arity n, and $\eta \in \mathbb{Q}^S$ is the *final vector*. For each symbol $\sigma \in \Omega_n$, the matrix $\mu(\sigma) = M_n(\sigma) \in \mathbb{Q}^{S^n\times S}$ is called the *transition matrix*. We extend tree representation μ from Ω to all elements $\sigma(t_1, \dots, t_n) \in T_\Omega$ by defining

$$\mu(\sigma(t_1, \dots, t_n)) \coloneqq (\mu(t_1) \otimes \dots \otimes \mu(t_n)) \cdot \mu(\sigma).$$

The automaton $\mathcal{A}$ *recognises* the series $[\![\mathcal{A}]\!] \colon T_\Omega \to \mathbb{Q}$ given by $t \mapsto \mu(t)\cdot\eta$. Two MTAs $\mathcal{A}$ and $\mathcal{A}'$ over Ω are *equivalent* if $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. The decision problem of MTA equivalence [48] assumes rational entries given as fractions of binary integers. Note that MWAs are a special case of MTAs via $(S, \Omega_0 \cup \Sigma, M_0 \cup M, \eta)$, where $\Omega_0 = \{\sigma_0\}$ and $M_0 \colon \{\sigma_0\} \to \{\alpha\}$.

The following operations on rational word series correspond to operations on MWAs. Let $\mathcal{A}' = (S', \Sigma, M', \alpha', \eta')$ be another MWA over the same alphabet Σ.

Sum lifts to direct sum. The MWA $\mathcal{A} \oplus \mathcal{A}'$ with states $S \uplus S'$ is given by the initial vector $\alpha \oplus \alpha'$, transition matrix $M(a) \oplus M'(a)$ for each $a \in \Sigma$ and the final vector $\eta \oplus \eta'$. MWA $\mathcal{A} \ominus \mathcal{A}'$ is given analogously except for the final vector $\eta \oplus (-\eta')$. It holds that $[\![\mathcal{A} \oplus \mathcal{A}']\!](w) = [\![\mathcal{A}]\!](w) + [\![\mathcal{A}']\!](w)$ and $[\![\mathcal{A} \ominus \mathcal{A}']\!](w) = [\![\mathcal{A}]\!](w) - [\![\mathcal{A}']\!](w)$ for each $w \in \Sigma^*$.

Zero series lifts to the zero automaton. Assume $\mathcal{A}'$ has no states $S' = \emptyset$, hence the underlying vector space is of dimension 0. Then $\mathcal{A}'$ recognizes the zero series, that is $[\![\mathcal{A}']\!](w) = 0$ for each word $w \in \Sigma^*$. We call $\mathcal{A}'$ the *zero automaton* over Σ.

Product lifts to Kronecker product. The MWA $\mathcal{A} \otimes \mathcal{A}'$ with states $S \times S'$ is given by the initial vector $\alpha \otimes \alpha'$, transition matrix $M(a) \otimes M'(a)$ for each $a \in \Sigma$ and the final vector $\eta \otimes \eta'$. It holds that $[\![\mathcal{A} \otimes \mathcal{A}']\!](w) = [\![\mathcal{A}]\!](w) \cdot [\![\mathcal{A}']\!](w)$ for each word $w \in \Sigma^*$.

These operations naturally extend to MTAs [11, 48].

2.2 Relational structures and logic

We assume familiarity with standard notions from finite model theory, see e.g. [39]. All signatures, structures, and graphs are finite. Let τ denote a relational signature. A class $\mathcal{C}$ of τ-structures is L-*definable* for some logic L if there exists a sentence $\varphi \in \mathsf{L}$ such that a τ-structure A satisfies φ if, and only if, $A \in \mathcal{C}$. For some implicit relational signature τ, we write MSO for monadic second-order logic over τ. There are several options for encoding graphs as relational structures in the context of MSO. The less powerful variant MSO_1 encodes a graph (V, E) as relational structure with universe V and the edge relation. In the more powerful variant, MSO_2 the graph is encoded as relational structure with universe $V \uplus E$ and the incidence relation. See [20] for further details.

Let A and B be τ-structures. A *homomorphism* $h\colon A \to B$ is a map from the universe of A to the universe of B such that $h(R^A) \subseteq R^B$ for all relation symbols $R \in \tau$. For example, a homomorphism $h \colon F \to G$ between simple graphs G and H is a map $V(G) \to V(H)$ such that $h(uv) \in E(G)$ for all $uv \in E(F)$. We write $\hom(A, B)$ for the number of homomorphisms from A to B. Two τ-structures A and B are *homomorphism indistinguishable* over a class of τ-structures $\mathcal{F}$, in symbols $A \equiv_{\mathcal{F}} B$, if $\hom(F, A) = \hom(F, B)$ for all $F \in \mathcal{F}$. We will ultimately be interested in homomorphism indistinguishability of simple graphs over classes of simple graphs, which has been the main focus in the literature, see [63, p. 32].

2.3 Logarithmic space and related complexity classes

Let L denote the class of languages decided by a deterministic logarithmic-space Turing machine. See [7] for a definition of logspace many-one reductions. Let #L denote the class of functions $f\colon \{0,1\}^* \to \mathbb{N}$ that count the number of accepting paths of a non-deterministic logarithmic-space Turing machine [6]. Let GapL denote the class of functions of the form $f - g$ for some functions $f, g \in$ #L [5]. Finally, let $\mathrm{C}_=\mathrm{L}$ denote the class of languages of the form $\{x \in \{0,1\}^* \mid f(x) = 0\}$ for some function $f \in$ GapL [5]. It was shown by [21, 69, 66] that GapL coincides with the class of functions which are L-many-one reducible to the determinant, see [2, 45]. A complete problem for $\mathrm{C}_=\mathrm{L}$ is the set of singular integer matrices, cf. [3]. To get a feeling for $\mathrm{C}_=\mathrm{L}$, we make the following observation in full detail.

▶ **Observation 5.** *Homomorphism indistinguishability over directed cycles is in* $\mathrm{C}_=\mathrm{L}$.

Proof. Slightly abusing notation, we write $\vec{C}_0$ for the edge-less one-vertex graph. The cycle $\vec{C}_k$ for $k \geq 1$ has k vertices and k directed edges, i.e. $\vec{C}_1$ is a single vertex with a loop.

First, note that the function $f\colon (G, k) \mapsto \hom(\vec{C}_k, G)$ is in #L. Here, G is a directed graph and $k \geq 0$. A non-deterministic logspace Turing machine operates whose number of accepting paths is $\hom(\vec{C}_k, G)$ operates by guessing a vertex $v_1 \in V(G)$ and, for $2 \leq i \leq k$, guesses a vertex $v_i \in V(G)$ and rejects if $v_k v_1 \notin E(G)$. Finally, it rejects if $v_{i-1} v_0 \notin E(G)$ and accepts otherwise. Clearly, the machine accepts iff $v_1 \dots v_k$ is the homomorphic image of $\vec{C}_k$ Since it suffices to store indices of three vertices vertices, i.e. v_1, v_{i-1}, and v_i, the space requirement is $3\log(|V(G)|) + \log(k)$.

As the difference of two #L-functions [5, Proposition 2], the function $p\colon (G, H, k) \mapsto \hom(\vec{C}_k, G) - \hom(\vec{C}_k, H)$ is in GapL. By [5, Theorem 9], the function $q\colon (G, H) \mapsto \sum_{k=0}^{n} p(G, H, k)^2$ is in GapL where $n \coloneqq \max\{|V(G)|, |V(H)|\}$. It holds that $q(G, H) = 0$ if, and only if, $\hom(\vec{C}_k, G) = \hom(\vec{C}_k, H)$ for all $0 \leq k \leq n$. By Newton's identities [38, 2.4.P10], the latter holds if, and only if, G and H are homomorphism indistinguishable over directed cycles $\vec{C}_k$ for arbitrary length $k \geq 0$. Thus, the claim follows by the definition of $\mathrm{C}_=\mathrm{L}$ [5, Definition 2]. ◀

2.4 Labelled graphs and homomorphism tensors

We recall some of the notation from [62]. Let $k \geq 1$. 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.

Let $k, \ell \geq 1$. 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. Write $\mathcal{D}(k, \ell)$ 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 observed in [47, 31], (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. For $\boldsymbol{F} = (F, \boldsymbol{u}) \in \mathcal{D}(k)$, define $\operatorname{soe}(\boldsymbol{F}) := F$ as the underlying unlabelled graph of $\boldsymbol{F}$. Then for all graphs G, $\hom(\operatorname{soe} \boldsymbol{F}, G) = \sum_{\boldsymbol{v} \in V(G)^k} \boldsymbol{F}_G(\boldsymbol{v}) =: \operatorname{soe}(\boldsymbol{F}_G)$.

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 $\boldsymbol{v} \in V(G)^k$, $(\boldsymbol{F} \odot \boldsymbol{F}')_G(\boldsymbol{v}) = \boldsymbol{F}_G(\boldsymbol{v})\boldsymbol{F}'_G(\boldsymbol{v}) =: (\boldsymbol{F}_G \odot \boldsymbol{F}'_G)(\boldsymbol{v})$. 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 $\boldsymbol{x}, \boldsymbol{z} \in V(G)^k$, $(\boldsymbol{K} \cdot \boldsymbol{K}')_G(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{y} \in V(G)^k} \boldsymbol{K}_G(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{K}'_G(\boldsymbol{y}, \boldsymbol{z}) =: (\boldsymbol{K}_G \cdot \boldsymbol{K}'_G)(\boldsymbol{x}, \boldsymbol{z})$. 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.

3 From homomorphism indistinguishability to multiplicity automata equivalence

In this section, we reduce homomorphism indistinguishability over recognisable graph classes of bounded treewidth to multiplicity automata equivalence.

▶ **Theorem 6.** *For $k \in \mathbb{N}$, let $\mathcal{F}$ be a k-recognisable graph class.*

1. *If $\mathcal{F}$ has treewidth $< k$, $\textsc{HomInd}(\mathcal{F})$ logspace many-one reduces to MTA equivalence.*
2. *If $\mathcal{F}$ has pathwidth $< k$, $\textsc{HomInd}(\mathcal{F})$ logspace many-one reduces to MWA equivalence.*

In order to prove the theorem, we give a formal definition of recognisability, see also [19].

▶ **Definition 7** ([12]). *Let $k \geq 1$. For a 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*

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

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

To parse Definition 7, 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 $\mathcal{F}$. See [62] for examples.

Courcelle [19] 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 [12] proved that, if a recognisable class $\mathcal{F}$ has bounded treewidth, then it is CMSO_2-definable. Furthermore, it holds that every k-recognisable graph class of treewidth $\leq k$ is CMSO_2-definable [13, Section 6]. We use recognisability via the following lemma.

▶ **Lemma 8** ([62, 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 $\boldsymbol{F}_1 \sim^k_{\mathcal{F}} \boldsymbol{F}'_1$ and $\boldsymbol{F}_2 \sim^k_{\mathcal{F}} \boldsymbol{F}'_2$ then $\boldsymbol{F}_1 \odot \boldsymbol{F}_2 \sim^k_{\mathcal{F}} \boldsymbol{F}'_1 \odot \boldsymbol{F}'_2$,*
2. *if $\boldsymbol{F} \sim^k_{\mathcal{F}} \boldsymbol{F}'$ then $\boldsymbol{L} \cdot \boldsymbol{F} \sim^k_{\mathcal{F}} \boldsymbol{L} \cdot \boldsymbol{F}'$.*

The proof of Theorem 6, formally conducted in the full version [68], works by constructing, given a fixed graph class $\mathcal{F}$ and input graphs G and H to HomInd($\mathcal{F}$), three MWAs in the bounded-pathwidth case and three MTAs in the bounded-treewidth case called $\mathcal{A}_\mathcal{F}$, $\mathcal{A}_G$, $\mathcal{A}_H$. All three automata read words/trees over the alphabet comprising distinctly (k,k)-bilabelled graphs representing a single bag of a path/tree decomposition.

The first automaton $\mathcal{A}_\mathcal{F}$ depends only on the graph class $\mathcal{F}$ and does not make use of multiplicities, i.e. it is a deterministic word/tree automaton. Its role is to recognise the graph class $\mathcal{F}$ among all graphs of pathwidth/treewidth less than k. The construction of this automaton dates back to Courcelle [19]. Its states are the equivalence classes of $\sim^k_\mathcal{F}$. Lemma 8 ensures that series and parallel composition, i.e. the operations used to compose labelled graphs, respect these equivalence classes.

The automata $\mathcal{A}_G$ and $\mathcal{A}_H$ depend only on G and H, respectively, and are both constructed in the same way. Their role is to compute the homomorphism tensors of labelled graphs of bounded pathwidth/treewidth and thus make full use of multiplicities. To that end, they assign to a letter of the input alphabet, i.e. a bilabelled graph $\boldsymbol{L} \in \mathcal{D}(k,k)$, the homomorphism matrix $\boldsymbol{L}_G$, respectively $\boldsymbol{L}_H$, as its weight matrix. The correspondence between operations on bilabelled graphs and homomorphism tensors, see Section 2.4, ensures that numbers computed by the automata are homomorphism counts from graphs of bounded pathwidth/treewidth into G and H. Note that $\mathcal{A}_G$ and $\mathcal{A}_H$ are equivalent if, and only if, G and H are homomorphism indistinguishable over *all* graphs of pathwidth/treewidth less than k.

The reduction is completed by testing the equivalence of the product automata $\mathcal{A}_\mathcal{F} \otimes \mathcal{A}_G$ and $\mathcal{A}_\mathcal{F} \otimes \mathcal{A}_H$. For a labelled graph $\boldsymbol{F} \in \mathcal{D}(k)$ of bounded pathwidth/treewidth with underlying unlabelled graph F, it is

$$\begin{aligned}\llbracket \mathcal{A}_\mathcal{F} \otimes \mathcal{A}_G \rrbracket(\boldsymbol{F}) - \llbracket \mathcal{A}_\mathcal{F} \otimes \mathcal{A}_H \rrbracket(\boldsymbol{F}) &= \llbracket \mathcal{A}_\mathcal{F} \rrbracket(\boldsymbol{F}) \cdot (\llbracket \mathcal{A}_G \rrbracket(\boldsymbol{F}) - \llbracket \mathcal{A}_H \rrbracket(\boldsymbol{F})) \\ &= \begin{cases} \hom(F,G) - \hom(F,H), & \text{if } F \in \mathcal{F}, \\ 0, & \text{otherwise.} \end{cases}\end{aligned}$$

Hence, G and H are homomorphism indistinguishable over $\mathcal{F}$ if, and only if, the automata $\mathcal{A}_\mathcal{F} \otimes \mathcal{A}_G$ and $\mathcal{A}_\mathcal{F} \otimes \mathcal{A}_H$ are equivalent.

4 Complexity of multiplicity word automata equivalence

Equipped with the reduction of homomorphism indistinguishability over recognisable bounded-pathwidth graph classes to MTA equivalence from Theorem 6, we now proceed to pinpoint the complexity of the latter problem. Towards Theorem 2, we first show containment in $\mathrm{C}_=\mathrm{L}$, thus improving on DET as shown in [67], see also [42].

▶ **Lemma 9.** *MWA equivalence is in* $\mathrm{C}_=\mathrm{L}$.

Proof. The class $\mathrm{C}_=\mathrm{L}$ is closed under logspace many-one reductions as shown in [5, Theorem 16]. Thus, it suffices to reduce MWA equivalence to a problem in $\mathrm{C}_=\mathrm{L}$. We first reduce the equivalence of two MWAs $\mathcal{A}$ and $\mathcal{A}'$ over Σ to the equivalence of MWA $\mathcal{A} \ominus \mathcal{A}'$ and the zero automaton over Σ. The operation $\ominus$ is clearly computable in logspace [48, Proposition 2]. In the following Claim 10, the last problem further reduces in logspace to a matrix rank bound verification. We use matrices over $\mathbb{Q}$, however, by [2, Section 2, Remark], this reduces in logspace to matrices over $\mathbb{Z}$. By [2, Proposition 2.5], the matrix rank bound verification is $\mathrm{C}_=\mathrm{L}$-complete. Thus, it remains to prove the following Claim 10.

▷ Claim 10. There are logspace-computable functions N and r such that, for each MWA $\mathcal{A}$, it holds that $\mathcal{A}$ is equivalent to the zero automaton if, and only if, the matrix $N(\mathcal{A})$ has rank less than $r(\mathcal{A})$.

Fix an MWA $\mathcal{A} = (S, \Sigma, M, \alpha, \eta)$ with $n := |S|$ states, and denote its square $\mathcal{A} \otimes \mathcal{A}$ by $\mathcal{A}_2 = (S_2, \Sigma, M_2, \alpha_2, \eta_2)$. By [42, Proposition 2.2], The MWA $\mathcal{A}$ is equivalent to the zero automaton if, and only if,

$$\sum_{w \in \Sigma^{\leq n-1}} [\![\mathcal{A}_2]\!](w) = \sum_{w \in \Sigma^{\leq n-1}} \left([\![\mathcal{A}]\!](w)\right)^2 = 0, \tag{1}$$

Let us denote the sum of all transition matrices by

$$T_2 := \sum_{a \in \Sigma} M_2(a) = \sum_{a \in \Sigma} M(a) \otimes M(a).$$

Since we can compute $\mathcal{A}_2$ and sum a constant number of matrices in logspace [17], the matrix T_2 is computable in logspace. We rewrite the expression in Equation (1) as follows

$$\begin{aligned}
\sum_{w} [\![\mathcal{A}_2]\!](w) &= \sum_{k=0}^{n-1} \sum_{a_1 \cdots a_k} \alpha_2 \, M_2(a_1 \cdots a_k) \, \eta_2 \\
&= \sum_{k=0}^{n-1} \alpha_2 \Big(\sum_{a_1 \cdots a_k} M(a_1 \cdots a_k) \otimes M(a_1 \cdots a_k) \Big) \eta_2 \\
&= \sum_{k=0}^{n-1} \alpha_2 \Big(\sum_{a_1 \cdots a_k} (M(a_1) \otimes M(a_1)) \cdots (M(a_k) \otimes M(a_k)) \Big) \eta_2 \\
&= \sum_{k=0}^{n-1} \alpha_2 \, T_2^k \, \eta_2,
\end{aligned} \tag{2}$$

where $a_1 \cdots a_k$ ranges over words in Σ^k. In the third equality, we used the distributivity of the Kronecker product over matrix multiplication. Next, we define a matrix $A \in \mathbb{Q}^{(n^3+1) \times n^3}$ and a vector $b \in \mathbb{Q}^{n^3+1}$ as follows

$$A := \begin{pmatrix} I_{n^2} & 0 & \cdots & \cdots & 0 \\ -T_2 & I_{n^2} & \ddots & & \vdots \\ 0 & -T_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & I_{n^2} & 0 \\ 0 & \cdots & 0 & -T_2 & I_{n^2} \\ \alpha_2 & \cdots & \alpha_2 & \alpha_2 & \alpha_2 \end{pmatrix}, \qquad b := \begin{pmatrix} \eta_2 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

We argue that there exists a solution $x \in \mathbb{Q}^{n^3}$ of $Ax = b$ if, and only if, the term in Equation (2) is equal to zero. To that end, group the n^3 variables into n blocks y_i of n^2 variables. The first n equations are $y_1 = \eta_2$. The subsequent equations yield $y_{i+1} = T_2 y_i$ for all $1 \leq i \leq n-1$. The final n equations simplify to $0 = \alpha_2 y_1 + \cdots + \alpha_2 y_n = \sum_{k=0}^{n-1} \alpha_2 T_2^k \eta_2$, as desired.

We now rephrase the existence of the solution x using matrix rank. Note that feasibility of linear systems of equations is complete for the potentially larger complexity class $\mathrm{L}^{\mathrm{C_=L}}$ [2]. It is therefore crucial that the rank of A is controlled. Since A is lower triangular with identity matrices on the main diagonal, it has rank n^3. Note that the rank of the augmented matrix $[A|b]$ is n^3 if the solution x exists, and n^3+1 otherwise. Finally, it suffices to set the functions to $N(\mathcal{A}) := [A|b]$ and the bound on the rank to be verified to $r(\mathcal{A}) := n^3 + 1$. Both functions can be computed in logspace. ◀

5 Hardness of homomorphism indistinguishability over recognisable graph classes

In this section, we show that homomorphism indistinguishability over recognisable graph classes of bounded pathwidth and of bounded treewidth can be as hard as $C_=L$ and as PIT, respectively.

5.1 Bounded treewidth and polynomial identity testing

Let us formally introduce PIT. An *arithmetic circuit* [3] is a directed acyclic graph whose vertices of in-degree zero are labelled by 0, 1, or by variables $X_1, \dots, X_\ell$. The internal vertices are labelled by $+$, $-$, or $\times$. The *Polynomial Identity Testing (PIT) problem* asks, given a polynomial $f \in \mathbb{Z}[X_1, \dots, X_\ell]$ represented by an arithmetic circuit, whether it is the zero polynomial. By the Schwartz–Zippel lemma [60, 72], PIT can be solved in randomised polynomial time, i.e. lies in coRP. The existence of a deterministic polynomial-time algorithm for PIT would have far-reaching consequences for circuit complexity [40], see also [3]. We show that the same holds for HomInd($\mathcal{F}$) for some recognisable graph class $\mathcal{F}$ of bounded treewidth.

▶ **Theorem 1.** *There exists a* CMSO_2*-definable graph class* $\mathcal{F}$ *of bounded treewidth such that* HomInd($\mathcal{F}$)*, MTA equivalence, and PIT are logspace many-one interreducible.*

Given [48] and Theorem 6, it remains to reduce PIT to homomorphism indistinguishability. To that end, we start with a class of directed vertex-coloured graphs and employ observations made in [48, 4].

▶ **Lemma 11.** *There exists a* MSO*-definable class of directed vertex-coloured graphs* $\mathcal{F}$ *of treewidth* ≤ 2 *with finitely many colours such that PIT logspace many-one reduces to* HomInd($\mathcal{F}$).

Proof. Following [48, 4], we reduce the following variant of PIT to HomInd($\mathcal{F}$). A *variable-free arithmetic circuit* is a finite acyclic vertex-labelled directed multigraph whose vertices have in-degree 0 or 2. The vertices of in-degree 2 are called *internal gates* and are labelled with $+$, $-$, or $\times$. The vertices of in-degree 0 are labelled with 0 or 1. The unique vertex of out-degree 0 is the output gate of the circuit.

Each such circuit computes an integer with the intuitive semantics. By [4, Proposition 2.2] and [48, Proposition 13], PIT logspace many-one reduces to the following problem: Given two variable-free arithmetic circuits C_1, C_2 satisfying the following conditions, decide whether C_1 and C_2 represent the same integer.

1. the internal gates are labelled with $+$ or $\times$, i.e. there are no substraction gates,
2. each gate of height i has precisely two children, which are of height $i - 1$,
3. $+$-gates have even height,
4. $\times$-gates have odd height,
5. the output gate has even height.

Given a circuit C as above, define a graph $G(C)$ with vertex colours $\{0, 1, +, \times, S\}$ by subdividing each outgoing edge of an internal gate and colouring the resulting vertex with S. In case of $\times$-gates, the two S-vertices are connected by an edge. See Figure 2.

Consider the family of graphs F_h as defined in Figure 3. We show the following claim. Here, $\mathrm{val}(g)$ for a gate g denotes the integer computed by g.

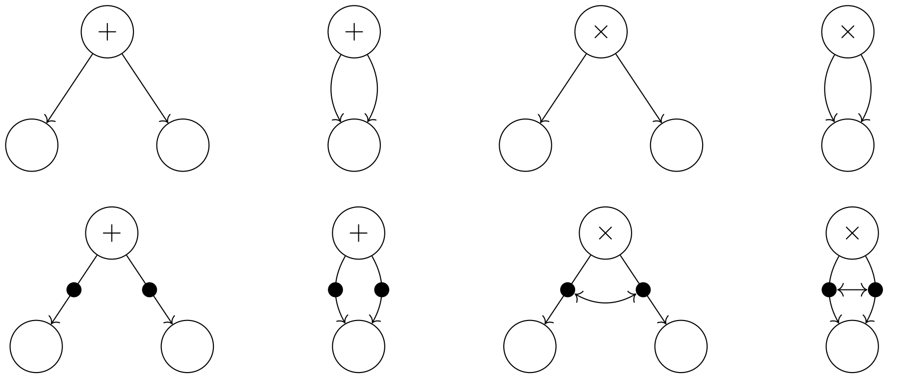
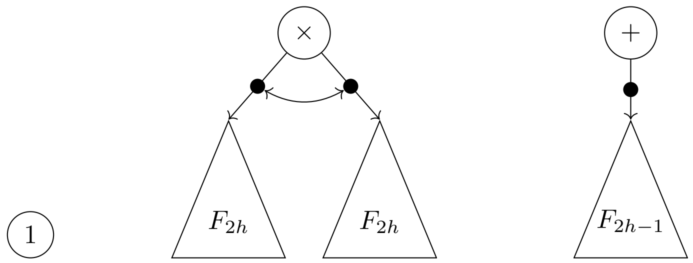


Figure 2 How to transform a circuit C in the top row into a graph $G(C)$ in the bottom row.



(a) F_0. **(b)** F_{2h+1} for $h \geq 0$. **(c)** F_{2h} for $h \geq 1$.

Figure 3 The graphs F_h for $h \geq 0$.

▷ Claim 12. Let C be a circuit. Let $g \in V(C)$ be a gate at height $h \geq 0$. The number of homomorphisms $\hom(F_h, G(C); r \mapsto g)$ from F_h to $G(C)$ which map the root r of F_h to g is equal to $\alpha(h) \cdot \operatorname{val}(g)$ for $\alpha(h) \coloneqq 2^{2^{\lceil h/2 \rceil}-1}$.

Proof. By induction on h. If $h = 0$, then g is either labelled 0 or 1. Since we consider vertex-colour preserving homomorphisms, the claim follows.

For the inductive step, distinguish two cases: If h is odd, then g is a multiplication gate. Write g_1, g_2 for the two children of g. Note that it may be that $g_1 = g_2$. Write r_1 and r_2 for the roots of the two copies of F_{h-1} in F_h. First suppose that $g_1 = g_2 =: g'$. Then, by the inductive hypothesis,

$$\begin{aligned}\hom(F_h, G(C); r \mapsto g) &= 2\hom(F_{h-1}, G(C); r_1 \mapsto g')\hom(F_{h-1}, G(C); r_2 \mapsto g') \\ &= 2\alpha(h-1)^2 \cdot \operatorname{val}(g')^2 \\ &= \alpha(h) \cdot \operatorname{val}(g).\end{aligned}$$

Otherwise, i.e. if $g_1 \neq g_2$, by the inductive hypothesis,

$$\begin{aligned}\hom(F_h, G(C); r \mapsto g) &= \hom(F_{h-1}, G(C); r_1 \mapsto g_1)\hom(F_{h-1}, G(C); r_2 \mapsto g_2) \\ &\quad + \hom(F_{h-1}, G(C); r_1 \mapsto g_2)\hom(F_{h-1}, G(C); r_2 \mapsto g_1) \\ &= 2\alpha(h-1)^2 \cdot \operatorname{val}(g_1)\operatorname{val}(g_2) \\ &= \alpha(h) \cdot \operatorname{val}(g).\end{aligned}$$

It remains to consider the case when h is even, i.e. when g is an addition gate. Write r' for the root of F_{h-1} in F_h. Write g_1, g_2 for the two children of g. If $g_1 = g_2 =: g'$,

$$\begin{aligned}\hom(F_h, G(C); r \mapsto g) &= 2\hom(F_{h-1}, G(C); r' \mapsto g') \\ &= 2\alpha(h-1) \cdot \mathrm{val}(g') \\ &= \alpha(h-1) \cdot (\mathrm{val}(g') + \mathrm{val}(g')) \\ &= \alpha(h) \cdot \mathrm{val}(g).\end{aligned}$$

where the factor 2 is incurred by the two possible images of the subdivision vertex between r and r'. Also note that $\alpha(h) = \alpha(h-1)$ when h is even. Finally, if $g_1 \neq g_2$,

$$\begin{aligned}\hom(F_h, G(C); r \mapsto g) &= \hom(F_{h-1}, G(C); r' \mapsto g_1) + \hom(F_{h-1}, G(C); r' \mapsto g_2) \\ &= \alpha(h-1) \cdot (\mathrm{val}(g_1) + \mathrm{val}(g_2)) \\ &= \alpha(h) \cdot \mathrm{val}(g).\end{aligned}$$ ◁

Define $\widehat{G}(C)$ by making the output gate vertex in $G(C)$ adjacent to a fresh vertex of colour T. Similarly, define $\widehat{F_h}$ by making the output vertex in F_h adjacent to a fresh vertex of colour T. Let $\mathcal{F}' := \{\widehat{F_h} \mid h \geq 0\}$. Finally, define $\mathcal{F}$ analogously but without imposing the restriction that all leaves of the graphs have the same depth. That is, the subgraphs F_{2h} and F_{2h-1} in Figure 3 are replaced with any $F_{h'}$ for even and odd h', respectively.

▷ Claim 13. The graph class $\mathcal{F}$ can be defined in MSO.

Proof. The graphs in $\mathcal{F}$ are precisely those directed graphs with vertex colours 0, 1, +, ×, S, and T that possess the following MSO-definable properties:

1. The underlying undirected graphs are connected and removing any vertex whose overall degree is at least 2 disconnects the directed graph.
2. Every vertex has precisely one of the colours $\{0, 1, +, \times, S, T\}$.
3. There is a unique vertex coloured T.
4. The vertices of out-degree zero are of colours 0 or 1.
5. The vertices coloured + have precisely one out-neighbour coloured S, which has precisely one out-neighbour coloured ×, 0, or 1.
6. The vertices coloured × have precisely two out-neighbours coloured S, which are mutually adjacent and each have precisely one out-neighbour coloured +, 0, or 1.

Note that all but the first property are actually first-order. Only acyclicity and connectedness in Item 1 require second-order resources. ◁

Finally, we claim that C_1 and C_2 represent the same integer if, and only, if $\widehat{G}(C_1)$ and $\widehat{G}(C_2)$ are homomorphism indistinguishable over $\mathcal{F}$. Since C_1 and C_2 have the same height h and the unique T-coloured vertex must be mapped to the unique T-coloured vertex, it follows from Claim 12 that, for $i \in \{1, 2\}$,

$$\hom(\widehat{F_h}, \widehat{G}(C_i)) = \alpha(h) \cdot \mathrm{val}(C_i)$$

Furthermore, $\hom(\widehat{F_{h'}}, \widehat{G}(C_i)) = 0$ for $h' \neq h$. In particular, the graphs in $\mathcal{F} \setminus \mathcal{F}'$ do not admit any homomorphisms to $\widehat{G}(C_i)$. ◀

The following Lemma 14 allows to reduce a homomorphism indistinguishability problem over a directed vertex-coloured graph class to one over a class of undirected graphs.

▶ **Lemma 14.** *Let $\mathcal{C}$ be a class of directed vertex-coloured graphs with finitely many colours. Then there exists a graph class $\mathcal{F}$ such that*
1. $\textsc{HomInd}(\mathcal{C})$ *logspace many-one reduces to* $\textsc{HomInd}(\mathcal{F})$,
2. *if $\mathcal{C}$ has bounded treewidth, then so does $\mathcal{F}$,*
3. *if $\mathcal{C}$ has bounded pathwidth, then so does $\mathcal{F}$, and*
4. *if $\mathcal{C}$ is* MSO*-definable, then $\mathcal{F}$ is* MSO_1*-definable.*

The proof of Lemma 14, which is deferred to the full version [68], is based on a construction of [14, 15] involving Kneser graphs. For integers r and s such that $1 \leq r \leq s/2$, the *Kneser graph* $K(r, s)$ is the graph whose vertices are the r-subsets of $[s]$ and whose edges connect two vertices if, and only if, the corresponding subsets are disjoint. The crucial property of Kneser graphs, which allows them to simulate colours and edge directions, is that there are homomorphically incomparable [34].

Proof of Theorem 1. The logspace many-one interreducibility of MTA equivalence and PIT was established in [48, Propositions 12 and 13]. By Lemmas 11 and 14, PIT logspace many-one reduces to $\textsc{HomInd}(\mathcal{F})$ for some MSO_1-definable graph class of bounded treewidth. By Theorem 6, $\textsc{HomInd}(\mathcal{F})$ logspace many-one reduces to MTA equivalence. ◀

5.2 Bounded pathwidth and $\mathsf{C_=L}$

In this section, we show Theorem 4 and thereby complete the proof of Theorem 2.

▶ **Theorem 4.** *There exists a* CMSO_2*-definable graph class $\mathcal{F}$ of bounded pathwidth such that* $\textsc{HomInd}(\mathcal{F})$ *is* $\mathrm{C_=L}$*-complete under logspace many-one reductions.*

Our reduction is from the problem VCP of verifying the characteristic polynomial χ_A of an integer matrix A, that is the decision problem

$$\left\{ (A, c_0, c_1, \dots, c_{n-1}) \;\middle|\; n \in \mathbb{N}, A \in \mathbb{Z}^{n \times n}, c_0, \dots, c_{n-1} \in \mathbb{Z},\ \chi_A(\lambda) = \lambda^n + \sum_{i=0}^{n-1} c_i \lambda^i \right\}.$$

VCP was shown to be $\mathrm{C_=L}$-complete under logspace many-one reductions in [37, Theorem 3.2]. We start with the following Lemma 15 by which we treat negative entries.

▶ **Lemma 15.** *For every pair of matrices $A, B \in \mathbb{Z}^{n \times n}$, there exist logspace-computable matrices $D, E \in \mathbb{N}^{3n \times 3n}$ such that*
1. *$\chi_A = \chi_B$ if, and only if, $\chi_D = \chi_E$,*
2. *A and B are similar if, and only if, D and E are similar.*

Proof. If $n = 0$, then the statement holds trivially. Assume $n > 0$ and denote the matrix of non-negative and negative elements of A by A^+ and A^-, respectively, so that $A = A^+ - A^-$ and $|A| = A^+ + A^-$. Define B^+, B^-, and $|B|$ analogously. We define matrices D and E in $\mathbb{N}^{3n \times 3n}$ via

$$D := \begin{pmatrix} A^+ & A^- & 0 \\ A^- & A^+ & 0 \\ 0 & 0 & |B| \end{pmatrix}, \quad E := \begin{pmatrix} B^+ & B^- & 0 \\ B^- & B^+ & 0 \\ 0 & 0 & |A| \end{pmatrix}, \quad \text{and} \quad T := \begin{pmatrix} -I & I & 0 \\ I & I & 0 \\ 0 & 0 & I \end{pmatrix}.$$

We consider a similar matrix to D

$$TDT^{-1} := \begin{pmatrix} A & 0 & 0 \\ 0 & |A| & 0 \\ 0 & 0 & |B| \end{pmatrix}, \quad \text{where} \quad T^{-1} := \begin{pmatrix} -\frac{1}{2}I & \frac{1}{2}I & 0 \\ \frac{1}{2}I & \frac{1}{2}I & 0 \\ 0 & 0 & I \end{pmatrix}.$$

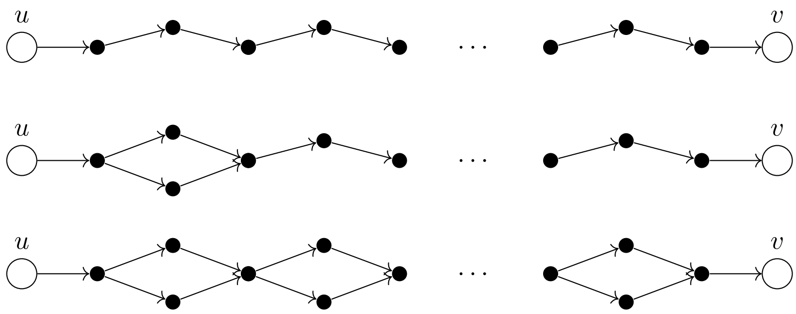


Figure 4 Gadgets $B_{1,b}$, $B_{2,b}$, and $B_{b,b}$ used for replacing the directed edge uv.

and analogously the matrix TET^{-1}, which is similar to E. It follows that $\chi_D = \chi_{TDT^{-1}} = \chi_A \cdot \chi_{|A|} \cdot \chi_{|B|}$ and $\chi_E = \chi_{TET^{-1}} = \chi_B \cdot \chi_{|A|} \cdot \chi_{|B|}$. Since $n > 0$, all polynomials in the product are non-zero, and thus cancelling both polynomials by the term $\chi_{|A|} \cdot \chi_{|B|}$, yields the first statement.

For the second statement, by [58, Theorem 7.7.3] courtesy to [35], TDT^{-1} and TET^{-1} are similar if, and only if, A and B are similar. Clearly, D and E are similar if, and only if, TDT^{-1} and TET^{-1} are similar. ◀

Lemma 15 yields that checking whether two matrices with non-negative integral entries have the same characteristic polynomial is $\mathrm{C_=L}$-complete.

▶ **Theorem 16.** *The set* $\{(A, B) \mid n \in \mathbb{N}, A, B \in \mathbb{N}^{n\times n}, \chi_A = \chi_B\}$ *is* $\mathrm{C_=L}$*-complete under logspace many-one reductions.*

Proof. We show $\mathrm{C_=L}$-hardness and postpone containment in virtue of [5, Theorem 16] to Theorem 17. For hardness, given an instance $(A, c_0, c_1, \dots, c_{n-1})$ of VCP, where $A \in \mathbb{Z}^{n\times n}$, $n \in \mathbb{N}$, we need to decide if $q(\lambda) = \lambda^n + \sum_{i=0}^{n-1} c_i \lambda^i$ is equal to the characteristic polynomial χ_A. For that, take as $B \in \mathbb{Z}^{n\times n}$ the companion matrix of $q(x)$, for which it holds that $\chi_B = q$ [58, Theorem 7.12]. We use Lemma 15 to obtain a pair of non-negative matrices whose characteristic polynomials are equal if, and only if, $\chi_A = \chi_B = q$. ◀

It remains to show that non-negative integral entries can be simulated by $\{0, 1\}$-entries using suitable gadgets. By the following Theorem 17, the decision problem $\{(A, B) \mid n \in \mathbb{N}, A, B \in \{0, 1\}^{n\times n}, \chi_A = \chi_B\}$ is $\mathrm{C_=L}$-complete. We remark that [66, Figure 2.1] describes a similar gadget construction when reducing the problem of computing the determinant of an integer matrix to computing the powers of a $\{-1, 0, 1\}$-matrix.

▶ **Theorem 17.** *Homomorphism indistinguishability over directed cycles is* $\mathrm{C_=L}$*-complete under logspace many-one reductions.*

Proof. Containment was shown in Observation 5. For hardness, let $A, B \in \mathbb{N}^{n\times n}$ be an instance of problem that is shown $\mathrm{C_=L}$-hard in Theorem 16. We consider the following two directed weighted graphs G_w and H_w given as adjacency matrices A and B, respectively.

Choose a bit-length b such that $2^{b+1} - 1$ bounds entries of A and B. We construct a directed simple graph G, by starting with vertices of G_w. Next, for every edge (u, v) of G_w with weight m, we connect u to v in G with the gadgets given in Figure 4 as follows: for the i-th non-zero bit of m we add gadget $B_{i,b}$ to G between u and v. We construct a directed graph H from H_w analogously.

▷ Claim 18. For all k, $\hom(\vec{C}_{kb}, G) = \mathrm{tr}(A^k)$, and $\hom(\vec{C}_\ell, G) = 0$ if b does not divide ℓ.

Proof. Consider a walk between endpoints u and v of a gadget $B_{i,b}$. This walk is of length exactly b, because of the orientation. Each $B_{i,b}$ contributes 2^i distinct walks, so that the collection of gadgets between u and v represents the weight in G_w. On the other hand, every closed walk counted by $\hom(\vec{C}_{kb}, G)$ goes through at least one vertex of G originated in G_w, and thus is also counted in $\operatorname{tr}(A^k)$. The second part follows since every closed walk is necessarily of length kb for some $k \in \mathbb{N}$. ◁

Consequentially, homomorphisms over directed cycles determine the traces of powers and vice versa. By Newton's identities, it holds $\operatorname{tr}(A^k) = \operatorname{tr}(B^k)$ for all $1 \leq k \leq n$ if, and only if, $\chi_A = \chi_B$. ◀

Finally, we apply Lemma 14 to reduce homomorphism indistinguishability over directed cycles to homomorphism indistinguishability over a CMSO_2-definable class of undirected and uncoloured graphs of bounded pathwidth.

Proof of Theorem 4. Containment follows from Theorem 6. Hardness follows from Theorem 17 and Lemma 14 observing that the class of (disjoint unions) of directed cycles contains precisely all directed graphs whose vertices have out-degree 1 and is thus definable in first-order logic. ◀

Proof of Theorem 2. Containment follows from Lemma 9. Hardness follows from Theorems 4 and 6. ◀

5.3 Undirected graphs and symmetric matrices

Even though Theorem 4 pinpoints the complexity of homomorphism indistinguishability over CMSO_2-definable graph classes of bounded pathwidth, it is slightly unsatisfactory in the sense that the constructed graph class is based on a gadget construction. More concretely, in light of Theorem 17, it would be interesting to determine whether homomorphism indistinguishability over undirected cycles is $\mathrm{C}_=\mathrm{L}$-complete. A related question[1] was already posed by Toda in 1984 [66] with little apparent progress since, see [44].

In general, in the realm of logspace computation, discrepancies between problems on directed and undirected graphs are well known. For example, directed connectivity is NL-complete while undirected connectivity is in L [54]. To give another example closer related to homomorphism indistinguishability, we observe invoking [37] that there is a complexity gap between similarity of symmetric and non-symmetric non-negative integer matrices unless the Exact Counting Logspace Hierarchy $\mathrm{L}^{\mathrm{C}_=\mathrm{L}}$ collapses to $\mathrm{C}_=\mathrm{L}$ [2, 5].

▶ **Corollary 19.**

1. *The set of pairs of similar non-negative integer matrices is* $\mathrm{L}^{\mathrm{C}_=\mathrm{L}}$*-complete under logspace many-one reductions.*
2. *The set of pairs of similar symmetric non-negative integer matrices is* $\mathrm{C}_=\mathrm{L}$*-complete under logspace many-one reductions.*

Proof. The first claim follows from [37, Theorem 4.1] and Lemma 15. The second claim follows from Theorem 17 noting that, for symmetric matrices A and B, it holds that $\chi_A = \chi_B$ if, and only if, A and B are similar, see e.g. [31, Theorem 3.1]. ◀

[1] Concretely, Toda [66] asks whether counting the number of not necessarily simple length-n paths between two vertices in an n-vertex undirected input graph is GapL-complete. In the same paper, this is shown for counting directed paths in directed input graphs.

Towards illuminating the discrepancies between the directed and undirected, we show that homomorphism indistinguishability over undirected cycles and undirected cycles and paths is logspace many-one interreducible. Two graphs G and H are homomorphism indistinguishable over cycles / cycles and paths iff there exists an orthogonal / an orthogonal pseudo-stochastic matrix X such that $XA_G = A_HX$, see [63]. The latter graph class is notable since it(s disjoint union closure) is closed under taking minors. Minor-closed graph classes play an important role in homomorphism indistinguishability [55, 64]. Both problems lie in $\mathrm{C_{=}L}$ by Theorem 6. The proof of Theorem 20 is deferred to the full version [68].

▶ **Theorem 20.** *Homomorphism indistinguishability over the class of cycles and paths and over the class of cycles are logspace many-one interreducible.*

6 Conclusion

The objective of this paper was to identify properties of graph classes $\mathcal{F}$ that characterise the complexity of the decision problem HomInd($\mathcal{F}$). We showed that, for recognisable graph classes $\mathcal{F}$ of bounded treewidth, the problem HomInd($\mathcal{F}$) is logspace many-one reducible to PIT and can be PIT-complete. For recognisable graph classes $\mathcal{F}$ of bounded pathwidth, the problem HomInd($\mathcal{F}$) lies in $\mathrm{C_{=}L}$ and can be $\mathrm{C_{=}L}$-complete. In the first case, this shows optimality of the algorithm from [62] while, in the second case, this improves upon [62]. In the process, we show that MWA equivalence is $\mathrm{C_{=}L}$-complete improving upon [67].

Given the role of minor-closed graph classes in homomorphism indistinguishability [55, 64], it would be interesting to obtain analogous results for minor-closed graph classes of bounded treewidth and pathwidth (the graph classes in Theorems 1 and 4 are not closed under taking minors).

For the first case, it was conjectured in [62] that homomorphism indistinguishability is in P, which would be optimal in light of [27]. Proving this, however, seems to require a better understanding of tree automata recognising minor-closed graph classes. The perhaps most tangible minor-closed bounded-treewidth graph class $\mathcal{F}$ for which no deterministic polynomial algorithm for HomInd($\mathcal{F}$) is known is the class of outerplanar graphs [62]. In this case, HomInd($\mathcal{F}$) amounts to deciding exact feasibility of the first level of the Lasserre SDP hierarchy for graph isomorphism [56].

Via Theorem 20, we have reduced the latter case to determining whether there is a complexity gap between homomorphism indistinguishability over directed and undirected cycles, which is related to an old question about GapL-computation [67, 44]. Finally, in light of [52] showing that homomorphism indistinguishability over all graphs of treedepth $\leq k$ is in L, it is conceivable that homomorphism indistinguishability over smaller graph classes, e.g. of bounded treedepth, can be placed into even smaller complexity classes.

References

1 Isolde Adler and Eva Fluck. Monotonicity of the Cops and Robber Game for Bounded Depth Treewidth. In Rastislav Královič and Antonín Kučera, editors, *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2024.6.

2 E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8(2):99–126, November 1999. doi:10.1007/s000370050023.

3 Eric Allender. Arithmetic circuits and counting complexity classes. In *Complexity of computations and proofs*, pages 33–72. Aracne; Rome, 2004. doi:10.7282/00000363.

4 Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the Complexity of Numerical Analysis. *SIAM Journal on Computing*, 38(5):1987–2006, January 2009. doi:10.1137/070697926.

5 Eric Allender and Mitsunori Ogihara. Relationships among PL, #L, and the determinant. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 30(1):1–21, 1996. doi:10.1051/ITA/1996300100011.

6 Carme Álvarez and Birgit Jenner. A very hard log-space counting class. *Theoretical Computer Science*, 107(1):3–30, January 1993. doi:10.1016/0304-3975(93)90252-o.

7 Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, Cambridge; New York, 2009.

8 Albert Atserias and Elitza Maneva. Sherali–Adams Relaxations and Indistinguishability in Counting Logics. *SIAM Journal on Computing*, 42(1):112–137, 2013. doi:10.1137/120867834.

9 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.

10 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.

11 J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18(2):115–148, May 1982. doi:10.1016/0304-3975(82)90019-6.

12 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.

13 Mikołaj Bojańczyk and MichałPilipczuk. Optimizing Tree Decompositions in MSO. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:13, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISSN: 1868-8969. doi:10.4230/LIPIcs.STACS.2017.15.

14 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.

15 Jan Böker, Louis Härtel, Nina Runde, Tim Seppelt, and Christoph Standke. The Complexity of Homomorphism Reconstructibility. *ACM Trans. Comput. Theory*, August 2025. doi:10.1145/3762196.

16 Hubie Chen and Moritz Müller. The Fine Classification of Conjunctive Queries and Parameterized Logarithmic Space. *ACM Transactions on Computation Theory*, 7(2):1–27, May 2015. doi:10.1145/2751316.

17 Andrew Chiu, George Davida, and Bruce Litow. Division in logspace-uniform NC^1. *RAIRO - Theoretical Informatics and Applications*, 35(3):259–275, March 2010. URL: http://eudml.org/doc/222086.

18 Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1-3):2–22, January 1985. doi:10.1016/s0019-9958(85)80041-3.

19 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.

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

21 C. Damm. DET = $L^{\#L}$. *Fachbereich Informatik der Humboldt-Universitat zu, Berlin*, Informatik-Preprint 8, 1991.

22 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.

23 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Leman. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2018.40.

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

25 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.

26 Jingchu Gai, Yiheng Du, Bohang Zhang, Haggai Maron, and Liwei Wang. Homomorphism Expressivity of Spectral Invariant Graph Neural Networks. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, January 2025. URL: https://openreview.net/forum?id=rdv6yeMFpn.

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

28 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.

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

30 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.

31 Martin Grohe, Gaurav Rattan, and Tim Seppelt. Homomorphism Tensors and Linear Equations. *Advances in Combinatorics*, April 2025. doi:10.19086/aic.2025.4.

32 Martin Grohe and Pascal Schweitzer. The Graph Isomorphism Problem. *Commun. ACM*, 63(11):128–134, 2020. Place: New York, NY, USA Publisher: Association for Computing Machinery. doi:10.1145/3372123.

33 Anselm Haak, Arne Meier, Om Prakash, and B. V. Raghavendra Rao. Parameterised Counting in Logspace. *Algorithmica*, 85(10):2923–2961, October 2023. doi:10.1007/s00453-023-01114-2.

34 Geňa Hahn and Claude Tardif. Graph homomorphisms: structure and symmetry. In Geňa Hahn and Gert Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*, pages 107–166. Springer Netherlands, Dordrecht, 1997. doi:10.1007/978-94-015-8937-6_4.

35 Christiaan Hattingh. Show that a block matrix is similar to another block matrix if and only if their blocks are similar. Mathematics Stack Exchange, 2018. URL: https://math.stackexchange.com/q/2845842.

36 Thanh Minh Hoang. *On the complexity of some problems in linear algebra.* PhD thesis, Universität Ulm, 2003. URL: https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Mitarbeiter/hoang/dissertation.pdf.

37 T.M. Hoang and T. Thierauf. The complexity of verifying the characteristic polynomial and testing similarity. In *Proceedings 15th Annual IEEE Conference on Computational Complexity*, pages 87–95, Florence, Italy, 2000. IEEE Comput. Soc. doi:10.1109/ccc.2000.856738.

38 Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge Univ. Press, Cambridge, 23. print edition, 2010.

39 Neil Immerman. *Descriptive Complexity*. Springer New York, New York, NY, 1999. doi:10.1007/978-1-4612-0539-5.

40 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 355–364, San Diego CA USA, June 2003. ACM. doi:10.1145/780542.780595.

41 Prem Nigam Kar, David E. Roberson, Tim Seppelt, and Peter Zeman. NPA Hierarchy for Quantum Isomorphism and Homomorphism Indistinguishability. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 105:1–105:19, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2025.105.

42 Stefan Kiefer. Notes on Equivalence and Minimization of Weighted Automata, September 2020. doi:10.48550/arXiv.2009.01217.

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

44 Meena Mahajan and Jayalal M. N. Sarma. On the Complexity of Matrix Rank and Rigidity. *Theory of Computing Systems*, 46(1):9–26, January 2010. doi:10.1007/s00224-008-9136-8.

45 Meena Mahajan and V Vinay. Determinant: Combinatorics, Algorithms, and Complexity. *Chicago Journal of Theoretical Computer Science*, 1997. URL: https://repository.ias.ac.in/127977/.

46 Peter N. Malkin. Sherali–Adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, May 2014. doi:10.1016/j.disopt.2014.01.004.

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

48 Ines Marušić and James Worrell. Complexity of equivalence and learning for multiplicity tree automata. *J. Mach. Learn. Res.*, 16:2465–2500, 2015. doi:10.5555/2789272.2912078.

49 Yoàv Montacute and Nihil Shah. The Pebble-Relation Comonad in Finite Model Theory. *Logical Methods in Computer Science*, Volume 20, Issue 2, May 2024. doi:10.46298/lmcs-20(2:9)2024.

50 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.

51 Jaroslav Nešetřil and Patrice Ossona De Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, August 2006. doi:10.1016/j.ejc.2005.01.010.

52 Simon Raßmann, Georg Schindling, and Pascal Schweitzer. Finite Variable Counting Logics with Restricted Requantification. In Jörg Endrullis and Sylvain Schmitz, editors, *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*, volume 326 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:23, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2025.14.

53 Gaurav Rattan and Tim Seppelt. Weisfeiler-Leman and Graph Spectra. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2268–2285, 2023. doi:10.1137/1.9781611977554.ch87.

54 Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):1–24, 2008. doi:10.1145/1391289.1391291.

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

56 David E. Roberson and Tim Seppelt. Lasserre Hierarchy for Graph Isomorphism and Homomorphism Indistinguishability. *TheoretiCS*, Volume 3:12321, September 2024. doi:10.46298/theoretics.24.20.

57 Neil Robertson and P.D. Seymour. Graph Minors. XX. Wagner's conjecture. *Special Issue Dedicated to Professor W.T. Tutte*, 92(2):325–357, November 2004. doi:10.1016/j.jctb.2004.08.001.

58 Steven Roman. *Advanced Linear Algebra*, volume 135 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 2008. doi:10.1007/978-0-387-72831-5.

59 Georg Schindling. Homomorphism Indistinguishability and Game Comonads for Restricted Conjunction and Requantification. In Pawel Gawrychowski, Filip Mazowiecki, and Michal Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, August 25-29, 2025, Warsaw, Poland*, volume 345 of *LIPIcs*, pages 89:1–89:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.MFCS.2025.89.

60 J. T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, October 1980. doi:10.1145/322217.322225.

61 M.P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, September 1961. doi:10.1016/s0019-9958(61)80020-x.

62 Tim Seppelt. An Algorithmic Meta Theorem for Homomorphism Indistinguishability. In Rastislav Královič and Antonín Kučera, editors, *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.

63 Tim Seppelt. *Homomorphism Indistinguishability*. Dissertation, RWTH Aachen University, Aachen, 2024. doi:10.18154/RWTH-2024-11629.

64 Tim Seppelt. Logical equivalences, homomorphism indistinguishability, and forbidden minors. *Information and Computation*, 301:105224, December 2024. doi:10.1016/j.ic.2024.105224.

65 William Slofstra. The Set of Quantum Correlations is not Closed. *Forum of Mathematics, Pi*, 7:e1, 2019. Edition: 2019/01/14 Publisher: Cambridge University Press. doi:10.1017/fmp.2018.3.

66 Seinosuke Toda. Counting Problems Computationally Equivalent to Computing the Determinant, 1991. URL: https://people.cs.rutgers.edu/~allender/papers/toda4.pdf.

67 Wen-Guey Tzeng. On path equivalence of nondeterministic finite automata. *Information Processing Letters*, 58(1):43–46, April 1996. doi:10.1016/0020-0190(96)00039-7.

68 Marek Černý and Tim Seppelt. Homomorphism Indistinguishability, Multiplicity Automata Equivalence, and Polynomial Identity Testing, 2025. arXiv:2512.13058v1.

69 V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 270–284, Chicago, IL, USA, 1991. IEEE Comput Soc. Press. doi:10.1109/SCT.1991.160269.

70 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.

71 Bohang Zhang, Jingchu Gai, Yiheng Du, Qiwei Ye, Di He, and Liwei Wang. Beyond Weisfeiler–Lehman: A Quantitative Framework for GNN Expressiveness. In *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=HSKaGOi7Ar.

72 Richard Zippel. Probabilistic algorithms for sparse polynomials. In G. Goos, J. Hartmanis, P. Brinch Hansen, D. Gries, C. Moler, G. Seegmüller, J. Stoer, N. Wirth, and Edward W. Ng, editors, *Symbolic and Algebraic Computation*, volume 72, pages 216–226. Springer Berlin Heidelberg, Berlin, Heidelberg, 1979. doi:10.1007/3-540-09519-5_73.