RWTH Aachen University
Chair i7 / Logic and Theory of Discrete Systems
Computer Science (B.Sc.)

Characterising Fragments of First-Order Logic by Counting Homomorphisms

Bachelor Thesis
Submitted for the Degree of
Bachelor of Science

Submitted by
Gian Luca Spitzer
on March 24, 2022

First examiner:	Univ.-Prof. Dr. rer. nat. Martin Grohe
Second examiner:	Priv.-Doz. Dr. Christof Löding

Abstract

Two graphs G, G' are homomorphism-indistinguishable over a class $\mathcal{F}$ of graphs if for each $F \in \mathcal{F}$ the number of homomorphisms from F to G equals the number of homomorphisms from F to G'. Results by Dvořák [9] and Grohe [10] show that homomorphism-indistinguishability over the class of graphs of tree-width at most k coincides with logical equivalence in the $(k+1)$-variable fragment of first-order logic with counting, while homomorphism-indistinguishability over the class of graphs of tree-depth at most q coincides with equivalence in the quantifier-rank-q fragment. We introduce the graph parameter of elimination-order and prove that two graphs are equivalent in the k-variable, quantifier-rank-q fragment of counting first-order logic if and only if they are homomorphism-indistinguishable over the class of graphs of elimination-order (k, q).

Contents

1 Introduction **1**

2 Preliminaries **3**

3 Characterising Logical Equivalence **9**
3.1 Logic on Graphs . . . 9
3.2 The Bijective Pebble Game . . . 10
3.2.1 The m-round game . . . 10
3.2.2 The m-round k-pebble game . . . 11
3.3 Characterising C^k . . . 13
3.3.1 From Logic to Homomorphism Counts . . . 14
3.3.2 Quantum Graphs and Models . . . 16
3.4 Characterising C_q . . . 18
3.4.1 Playing the Bijective Pebble Game . . . 18
3.5 Tying the knot: A characterisation of C_q^k . . . 25
3.5.1 Elimination Order . . . 26
3.5.2 Proving the result . . . 27
3.5.3 Salvaging the conjecture . . . 29

4 Complexity **31**
4.1 Completing the picture . . . 31

5 Conclusion **33**
5.1 Future Work . . . 33

1 Introduction

Let $\hom(G, H)$ be the number of homomorphisms from a graph G to a graph H. It turns out that this parameter captures structural information very well. In fact, an early theorem by Lovász [14] states that two graphs H and H' are isomorphic if and only if $\hom(F, H) = \hom(F, H')$ for all graphs F. A natural question to ask is if there are smaller classes of graphs or, in a similar vein, classes that are more algorithmically tractable that still capture graph isomorphism. We call such classes *distinguishing*.

The existence of a distinguishing class $\mathcal{F}$ for which $\hom(F, \cdot)$ with $F \in \mathcal{F}$ is computable in polynomial time would, for instance, settle the open question of whether the graph isomorphism problem is in coNP [9]. It turns out that such a class does not exist, since the classes for which $\hom(F, \cdot)$ is computable in polynomial time are precisely those of graphs with bounded *tree-width*[1] [7]. As we will see later, two graphs that satisfy the same sentences of the *k-variable fragment of counting first-order logic* C^k cannot be distinguished by a class of graphs with tree-width bounded by k.

Results like this however open the door to the idea of characterising not just isomorphism, but many natural equivalence relations in terms of homomorphism indistinguishability. For a class of graphs $\mathcal{F}$, we call graphs G, G' *homomorphism-indistinguishable over* $\mathcal{F}$ if $\hom(F, G) = \hom(F, G')$ for all $F \in \mathcal{F}$. One can define a weaker notion of graph isomorphism by asking for the existence of a rational solution to a $\{0, 1\}$-linear system of equations encoding the existence of an isomorphism between two graphs [19]. This aptly named *fractional isomorphism* turns out to be equivalent to homomorphism-indistinguishability over the class of all trees. A well-known result states that two graphs are *co-spectral*, that is, their adjacency matrices have the same characteristic polynomial, if and only if they are homomorphism-indistinguishable over the class of cycles. A recent result by Mančinska and Reberson [16] shows that graphs are homomorphism-indistinguishable over the class of planar graphs if and only if they are *quantum isomorphic*.

Especially interesting in this context are classes that characterise logical equivalence. For one, logic is tightly connected to the concept of homomorphism indistinguishability via propositional proof complexity [2, 18]. More importantly, by finding a class $\mathcal{F}$ that captures equivalence in terms of a logic L, we can infer certain properties of L by reasoning about $\mathcal{F}$. One such result, for instance, pertains to the locality of the *quantifier-rank-q fragment of counting first-order logic* C_q [10]. Furthermore, a characterisation by homomorphism indistinguishability often allows us to replace alternating quantifiers by flat homomorphism counts. As Grohe [10] writes, this fits into the common theme in complexity theory of replacing quantifier alternation by counting – a prominent example being Toda's theorem that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$ [20].

One of the results in the domain of logical equivalence is the above-mentioned characterisation of C^k. Dvořák [9] shows that two graphs are C^k-equivalent if and only if they are not distinguished by $\mathcal{TW}_{k-1}$, the class of graphs of tree-width at most $k-1$. Grohe [10] complements this result by showing that graphs are C_q-equivalent if and only if they are homomorphism-indistinguishable over the class $\mathcal{TD}_q$ of graphs of *tree-depth* at most q.

In this thesis, we want to take a closer look at the two characterisations and aim to complete them by giving a characterisation for the k-variable, quantifier-rank-q fragment of counting first-order logic C^k_q. To that end, we introduce the notion of *elimination-order*, which is closely related to both tree-width and tree-depth. We investigate the properties of this new graph

[1] assuming $\#\mathsf{W}[1] \neq \mathsf{FPT}$

parameter and show that, perhaps surprisingly, two graphs are C_q^k-equivalent if and only if they are not distinguished by the class $\mathcal{E}_{k,q}$ of graphs of elimination order (k, q). We conclude by investigating our result in terms of computational complexity. In particular, we consider the family of decision problems $\textsc{HomInd}(\mathcal{F})$, for a class of graphs $\mathcal{F}$: Given two graphs G, H, are G and H homomorphism-indistinguishable over $\mathcal{F}$? Inter alia, we find that $\textsc{HomInd}(\mathcal{E}_{k,q})$ is decidable in polynomial time.

The thesis is divided into five chapters, the first being this introduction. In Chapter 2, we introduce the basic concepts underlying the above-mentioned results. We also introduce notation and conventions specific to this thesis. The reader is assumed to have a rudimentary understanding of first-order logic and to be familiar with basic concepts from linear algebra.

Chapter 3 extensively explores the idea of capturing the logical equivalence of graphs by different means – the focus, of course, being characterisations in terms of homomorphism-indistinguishability. We give proofs for the most important results and put them into context. In the last section of the chapter, we address the characterisation of C_q^k.

In Chapter 4, we explore the complexity of $\textsc{HomInd}$ for the graph classes mentioned in Chapter 3. Finally, in Chapter 5 we discuss our results and outline interesting related- and possible future work.

2 Preliminaries

Notation

For our purposes, the set of natural numbers $\mathbb{N}$ contains 0. The set of positive integers we denote by $\mathbb{N}^+$. By $[k]$ we denote the set $\{1, \dots, k\}$. For a function f, we denote the domain of f by $\mathrm{dom}(f)$. The *image* of f is the set $\mathrm{img}(f) \coloneqq \{f(x) \mid x \in \mathrm{dom}(f)\}$. The *restriction* of a function $f\colon A \to C$ to some set $B \subseteq A$ is the function $f|_B\colon B \to C$ with $f|_B(x) = f(x)$ for $x \in B$.

Tuples are denoted by bold letters. The elements of a tuple $\mathbf{x}$ we denote by the corresponding regular letter and an index, that is, $\mathbf{x} = (x_1, \dots, x_n)$. We denote by $\mathbf{a} \oplus \mathbf{b}$ the *concatenation* $(a_1, \dots, a_n, b_1, \dots, b_m)$ of $\mathbf{a}$ and $\mathbf{b}$. We treat single elements as 1-tuples and sometimes write $\mathbf{a} \oplus c$ for $(a_1, \dots, a_n, c)$. We write $\mathbf{a}(i \to c)$ to denote the tuple $(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n)$ obtained by replacing the i-th element of $\mathbf{a}$ with c. If $n < i$, we let $\mathbf{a}(i \to c) = (a_1, \dots, a_n, c)$. Analogous to sets, we write $|\mathbf{a}|$ to denote the number of elements in $\mathbf{a}$. Finally, by $(a)^\ell$ we denote the ℓ-tuple $(a, \dots, a)$.

Logic on Structures

A signature τ is a finite tuple $(R_1, \dots, R_n, f_1, \dots, f_m)$ of relation symbols R_i and function symbols f_i. With each symbol is associated a fixed arity, denoted by $\mathrm{ar}(\cdot)$. A τ-structure $\mathfrak{A} \coloneqq (A, R_1^{\mathfrak{A}}, \dots, R_n^{\mathfrak{A}}, f_1^{\mathfrak{A}}, \dots, f_m^{\mathfrak{A}})$ is a non-empty set A together with interpretations $R_i^{\mathfrak{A}}$, $f_i^{\mathfrak{A}}$ for the symbols in τ. For each R_i it is $R_i^{\mathfrak{A}} \subseteq A^{n_i}$ and for each f_i we have $f_i^{\mathfrak{A}}\colon A^{m_i} \to A$, where $n_i = \mathrm{ar}(R_i)$ and $m_i = \mathrm{ar}(f_i)$. Unless otherwise stated, we assume structures to be finite.

A *logic* L consists of a mapping that assigns a set of *formulae* $\mathsf{L}(\tau)$ to each signature τ, and a satisfaction relation $\models_{\mathsf{L}}$. If $\mathfrak{A} \models_{\mathsf{L}} \varphi$ for some τ-structure $\mathfrak{A}$ and $\varphi \in \mathsf{L}(\tau)$, we say $\mathfrak{A}$ *satisfies* φ. If it is clear from the context which logic is meant, we omit the subscript and simply write $\mathfrak{A} \models \varphi$. A formula is *atomic* if it does not contain any logical connectives or quantifiers. Two formulae φ, ψ are called *logically equivalent* (we write $\varphi \equiv \psi$) if $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{A} \models \psi$ for all structures $\mathfrak{A}$. Two structures $\mathfrak{A}, \mathfrak{B}$ are L-*equivalent* (we write $\mathfrak{A} \equiv_{\mathsf{L}} \mathfrak{B}$) if $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{B} \models \varphi$ for all $\varphi \in \mathsf{L}$.

Logics with Counting

The *counting extension of first-order logic* C extends regular first-order logic by quantifiers $\exists^{\geq t}$, for $t \in \mathbb{N}$. Consequently, we can build a C-formula in the usual way from atomic formulae, variables $x_1, x_2, \dots$, logical operators $\wedge, \vee, \to, \neg$ and quantifiers $\forall, \exists, \exists^{\geq t}$.

An occurence of a variable x is called *free* if it is not in the scope of any quantifier. The *free variables* $\mathrm{free}(\varphi)$ of a formula φ are precisely those that have a free occurence in φ. We often write $\varphi(x_1, \dots, x_n)$ to denote that the free variables of φ are among $x_1, \dots, x_n$. For a structure $\mathfrak{A}$ and $a_1, \dots, a_n \in A$, we denote by $\mathfrak{A} \models \varphi(a_1, \dots, a_n)$ that $\mathfrak{A}$ satisfies $\varphi(x_1, \dots, x_n)$ if we interpret the free variables x_i by a_i respectively. A formula without free variables is called a *sentence*. To account for formulae with free variables, we generalize the notion of L-equivalence, writing $\mathfrak{A}, a_1, \dots, a_n \equiv_{\mathsf{L}} \mathfrak{B}, b_1, \dots, b_n$ to denote that for all formulae $\varphi(x_1, \dots, x_n) \in \mathsf{L}$ it holds that $\mathfrak{A} \models_{\mathsf{L}} \varphi(a_1, \dots, a_n) \Leftrightarrow \mathfrak{B} \models_{\mathsf{L}} \varphi(b_1, \dots, b_n)$.

The semantics of C are defined like FO, additionally defining that for a formula $\varphi(x_1, \dots, x_n) = \exists^{\geq t} y \psi(x_1, \dots, x_n, y)$, it is $\mathfrak{A} \models \varphi(a_1, \dots, a_n)$ if there exist distinct elements $b_1, \dots, b_t$ such that $\mathfrak{A} \models \psi(b_i, a_1, \dots, a_n)$ for all $i \in [t]$.

We sometimes write $\exists^{=t} x \varphi(x)$ for $\exists^{\geq t} x \varphi(x) \wedge \neg \exists^{\geq t+1} x \varphi(x)$. We also write true for $\forall x (x = x)$ and false for $\neg\, \mathrm{true}$. Formally, we will often restrict ourselves to the equally expressive subset

$\{\neg, \vee, \exists^{\geq t}\}$ of symbols. This is indeed equally expressive by De Morgan's laws and observing that $\exists x\varphi(x) \equiv \exists^{\geq 1} x\varphi(x)$ and $\forall x\varphi(x) \equiv \neg\exists x\neg\varphi(x)$.

The quantifier rank $\mathrm{qr}(\cdot)$ of a formula is defined inductively as follows. It is $\mathrm{qr}(\varphi) = 0$ for atomic formulae φ, $\mathrm{qr}(\neg\varphi) = \mathrm{qr}(\varphi)$, $\mathrm{qr}(\varphi \vee \psi) = \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\}$ and $\mathrm{qr}(\exists^{\geq t} x\varphi) = 1 + \mathrm{qr}(\varphi)$. The *quantifier-rank-q fragment* C_q of counting first order logic consists of all formulae of quantifier rank at most q.

Instead of restricting the quantifer rank, we can also simply restrict the number of distinct variables that are allowed to occur in a formula. By C^k we denote the *k-variable fragment* of C, consisting of all formulae using at most k different variables.

Mappings and Morphisms

A *homomorphism* is a structure-preserving mapping between two structures. This means a mapping $h : \mathfrak{A} \to \mathfrak{B}$ is a homomorphism if

1. for any n-ary relation R it holds that $(a_1, \ldots, a_n) \in R^{\mathfrak{A}} \implies (h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$, and
2. for any n-ary function f it holds that $h(f^{\mathfrak{A}}(a_1, \ldots, a_n)) = f^{\mathfrak{B}}(h(a_1), \ldots, h(a_n))$.

We say that h is a *strong homomorphism* if also the converse of condition (1) holds, that is, $(a_1, \ldots, a_n) \in R^{\mathfrak{A}} \iff (h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$. A homomorphism $h : \mathfrak{A} \to \mathfrak{B}$ is an *embedding* of $\mathfrak{A}$ in $\mathfrak{B}$ if it is injective. An *isomorphism* is a bijective strong homomorphism. If there exists an isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$, we say $\mathfrak{A}$ and $\mathfrak{B}$ are *isomorphic* and write $\mathfrak{A} \cong \mathfrak{B}$.

We denote the set of homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ by $\mathrm{Hom}(\mathfrak{A}, \mathfrak{B})$. The number of homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ we denote by $\hom(\mathfrak{A}, \mathfrak{B}) \coloneqq |\mathrm{Hom}(\mathfrak{A}, \mathfrak{B})|$. We also write $\mathrm{Hom}(\mathfrak{A}, \mathfrak{B}; a_1 \mapsto b_1, \ldots, a_n \mapsto b_n)$ to denote the set of homorphisms $h\colon \mathfrak{A} \to \mathfrak{B}$ satisfying $h(a_i) = b_i$ for $i \in [n]$.

If $\mathcal{A}$ is a class of structures, we denote by $\hom(\mathcal{A}, \mathfrak{B})$ the (potentially infinite) vector

$$\hom(\mathcal{A}, \mathfrak{B}) \coloneqq (\hom(\mathfrak{A}, \mathfrak{B}) \mid \mathfrak{A} \in \mathcal{A}).$$

We say that $\mathfrak{B}$ and $\mathfrak{C}$ are *homomorphism-indistinguishable over* $\mathcal{A}$ if $\hom(\mathcal{A}, \mathfrak{B}) = \hom(\mathcal{A}, \mathfrak{C})$. If $\mathfrak{B}$ and $\mathfrak{C}$ are not homomorphism-indistinguishable over $\mathcal{A}$ we also say that $\mathcal{A}$ *distinguishes* $\mathfrak{B}$ and $\mathfrak{C}$. Similarly, we say $\mathfrak{A}$ distinguishes $\mathfrak{B}$ and $\mathfrak{C}$ if $\hom(\mathfrak{A}, \mathfrak{B}) \neq \hom(\mathfrak{A}, \mathfrak{C})$.

Graphs, Labels and Colours

Unless otherwise specified, all graphs are assumed to be simple: They are undirected, unweighted and contain neither loops nor parallel edges. A graph is a tuple $(V(G), E(G))$, where $V(G)$ is a finite set of *vertices*, and $E(G) \subseteq V(G)^2$ is the set of *edges*. In terms of logic, a graph G is an $\{E\}$-structure with universe $V(G)$.

In this paper, we consider both *labelled* and *coloured* graphs. A coloured graph G is a graph together with a function $\gamma\colon V(G) \to \Gamma$ for some arbitrary set of "colours" Γ. A k-labelled graph G is a graph together with a partial function $\nu\colon [k] \to V(G)$, where $[k]$ is the set of possible labels. Intuitively, a coloured graph assigns each vertex a single colour. A labelled graph, on the other hand, assigns labels to arbitrary nodes, where a single label can be used only once, not all labels have to be used, and each node can be labelled 0 or more times.

For a k-labelled graph G and $l \in [k]$, we write $G(l \to v)$ to denote the graph obtained from G by setting $\nu_{G(l\to v)}(l) = v$. By $L_G \coloneqq \{i \in [k] \mid \nu(i) \text{ is defined}\}$ we denote the *set of labels* of G. The *product* G_1G_2 of two labelled graphs is the graph obtained by taking the disjoint union of G_1 and G_2, identifying the vertices with the same labels, and suppressing any loops or parallel edges that might be created.

For coloured graphs, we allow C-formulae to make statements about the colour of vertices. As such, we consider a coloured graph G also to be a $\{E, \gamma, c_1, \ldots, c_n\}$-structure with universe

$V(G) \cup \Gamma$, where $c_1, \ldots, c_n \in \Gamma$ are constants. This signature allows for atomic formulae of the form $\gamma(v) = c_i$ ("v has colour c_i").

When considering homomorphisms between coloured graphs, we require that they preserve colours as well. For any homomorphism $h\colon G \to G'$ this means $\gamma^G(v) = \gamma^{G'}(h(v))$ for all $v \in V(G)$. For k-labelled graphs, we require homomorphisms to preserve labelling, that is, $\nu_G(i) = v \Rightarrow \nu_{G'}(i) = h(v)$ for all $i \in [k]$ and $v \in V(G)$.

The *order* of a graph $|G|$ is equal to the number of its vertices. A graph H is a *subgraph* of G if H can be obtained from G only by removing edges and vertices. A set $A \subseteq V(G)$ *induces* a subgraph H of G if $V(H) = A$ and for all $a, b \in A$ we have $(a, b) \in E(H) \Leftrightarrow (a, b) \in E(G)$. For labelled graphs, we additionally require that $\nu_H(l) = a \Leftrightarrow \nu_G(l) = a$ for $a \in A$ and $l \in L_G$. For coloured graphs, we require that $\gamma^H = \gamma^G|_A$. Note that by this definition, H is indeed a subgraph of G. By $G - v$ we denote the subgraph of G induced by the set $V(G) \setminus \{v\}$.

A graph is *connected* if there exists a path between any two vertices. A *tree* is a graph where any two vertices are connected by exactly one path. The disjoint union of one or more trees is called a *forest*. A *rooted tree* is a tree T together with some designated vertex $v \in V(T)$, the *root* of T. A *rooted forest* is a disjoint union of rooted trees.

We can view a rooted forest S as a pair $(V(S), \preceq^S)$, where $\preceq^S$ is a partial order on $V(S)$ and for every $v \in V(S)$ the elements of the set $\{u \in V(S) \mid u \preceq^S v\}$ are pairwise comparable: The minimal elements of $\preceq^S$ are precisely the roots of S, and for any rooted tree (T, r) that is part of S we let $v \preceq^S w$ if v is on the unique path from r to w.

The height of a rooted forest S is defined as the length of the longest $\preceq^S$-chain. A rooted tree T' is a *subtree* of a tree T, if $V(T') \subseteq V(T)$ and $\preceq^{T'}$ is the restriction of $\preceq^T$ to $V(T')$. Note that the subgraph of T induced by $V(T')$ might not be a tree, since the vertices of T' can be interleaved with vertices that do not belong to T'. We call a subtree T' of T *connected* if its induced subgraph on T is connected.

Tree-depth and Tree-width

Tree-width[1] is a well-studied structural graph parameter. Intuitively, it measures how close a graph is to a tree. There are many different characterisations of tree-width [3], the most fundamental being the definition via *tree decompositions*.

Definition 2.1. A *tree decomposition* of a graph G is a tree T with $V(T) = \{X_i \mid i \in I\}$ and $X_i \subseteq V(G)$ for $i \in I$, such that

- $\bigcup_{i \in I} X_i = V(G)$,
- for all $(v, w) \in E(G)$, there is an $i \in I$ with $v, w \in X_i$, and
- for all $v \in V(G)$ the set $\{X_i \in V(T) \mid v \in X_i\}$ induces a connected subtree of T.

The sets X_i are often called *bags*. The *width* of a tree decomposition is defined as $\max_{i \in I} |X_i| - 1$. The tree-width $\mathrm{tw}(G)$ of a graph G is the minimum width over all tree decompositions of G.[2]

There is an alternative, recursive definition of tree-width in terms of labelled graphs: A $(k+1)$-labelled graph G has tree-with at most k if it can be constructed by repeatedly applying the following construction steps, starting with a set of $(k+1)$-labelled graphs H with $V(H) = L_H$. (We also say the H must be *fully labelled*).

- Take two graphs H_1 and H_2, and construct the product $H_1 H_2$.
- Take a graph H and remove a label l.

[1] The spelling "treewidth" is far more common in the literature than "tree-width", but since we use the concepts of both tree-depth and tree-width in this thesis, we decided to spell them in a consistent manner.

[2] The rather strange definition of the width of a tree decomposition guarantees that trees are precisely the graphs of tree-width 1 (see Figure 2.1).

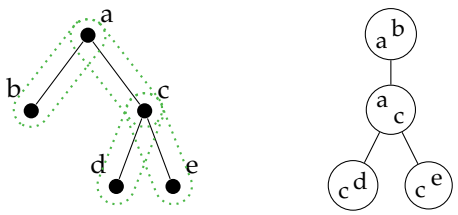


Figure 2.1: Trees have tree-width 1.

Intuitively, the sets $\{v \in V(H) \mid v \in \mathrm{img}(\nu_H)\}$ of labelled vertices of the intermediate graphs H represent the bags of a tree decomposition for G. We denote the class of all graphs of tree-width at most k by $\mathcal{TW}_k$.

Tree-depth, introduced by Nešetřil and Ossona de Mendez [17], is another, more recent, structural graph parameter. Tree-depth generalises the notion of height (or *depth*), normally associated with trees, to arbitrary graphs. We define tree-depth in terms of *elimination forests*.

Definition 2.2. An *elimination forest* of a graph G is a forest S with $V(S) = V(G)$, such that for every edge $(u, v) \in E(G)$ it is either $u \preceq^S v$ or $v \preceq^S u$.

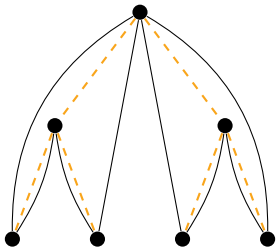


Figure 2.2: A graph G (solid edges) and an elimination tree (dashed edges) for G of height 3.

The tree-depth $\mathrm{td}(G)$ of a graph G is the minimum height of an elimination forest for G. If an elimination forest turns out to be a tree, we also talk about *elimination trees*. We can also give a recursive construction for graphs of tree-depth at most k. A graph G has tree-depth at most k if

- G consists of a single vertex, or
- G contains a vertex v such that all connected components of $G - v$ have tree-depth at most $k - 1$, or
- G is obtained by taking the disjoint union of graphs $G_1, \ldots, G_n$ of tree-depth at most k.

We denote the class of all graphs of tree-depth at most k by $\mathcal{TD}_k$.

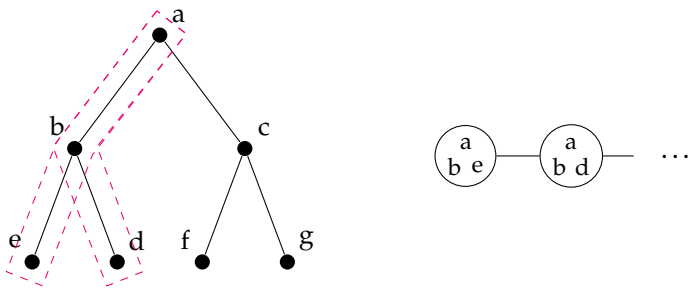


Figure 2.3: Building tree decompositions from elimination trees.

Observe that given an elimination forest S for a graph G, we can construct a tree decomposition of G by creating a bag for every path p from the root to the leaves that contains all the vertices on p. The resulting bags can be arranged as a path $X_1, \ldots, X_n$, such that any vertices that are contained in X_i but not in X_{i+1} are also not contained in X_j for $i < j$.

Proposition 2.3. *For any graph G it holds that* $\mathrm{tw}(G) \leq \mathrm{td}(G) - 1$.

3 Characterising Logical Equivalence

Let L be a logic. The question of whether two structures are L-equivalent, that is, whether they satisfy the same L-sentences, is central to model theory. Answering this question can, for instance, be useful for showing that a class of structures $\mathcal{K}$ is not definable in L – that there is no L-formula φ, such that $\mathfrak{A} \in \mathcal{K} \Leftrightarrow \mathfrak{A} \models \varphi$. If we can find structures $\mathfrak{A} \in \mathcal{K}$ and $\mathfrak{B} \notin \mathcal{K}$, but $\mathfrak{A} \equiv_{\mathsf{L}} \mathfrak{B}$, we have shown that such a defining formula cannot exist.

L-equivalence has logical, algebraic and game-theoretic characterisations that are originally due to Hintikka, Fraïssé, and Ehrenfeucht, respectively. The results by Dvořák and Grohe give us a characterisation by homomorphism counts, at least for the logics C^k and C_q on graphs.

In this chapter, we explore these homomorphism-based characterisations. We also concretise the game-theoretic characterisations of C_q and C_q^k since we use them in several proofs. Although many of the results generalise relatively nicely to arbitrary (relational) structures, we keep our focus on graphs for the sake of simplicity.

3.1 Logic on Graphs

When talking exclusively about labelled graphs, we can give an alternative formalisation of the logic C. Let φ be some formula of regular C with $\mathrm{free}(\varphi) \neq \varnothing$. The question of whether some structure $\mathfrak{A}$ satisfies φ is not really well-defined without some interpretation of the free variables. We can reason about the *existential closure* of φ – $\exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$ for $x_1, \dots, x_n \in \mathrm{free}(\varphi)$ – giving us a notion of general satisfiability. Alternatively, we can answer the question with regard to some *interpretation function* $\mathfrak{I} \colon \mathrm{free}(\varphi) \to A$, mapping the free variables of φ to elements of the universe. Such an interpretation function, however, can become quite tedious in certain situations, for instance in inductive proofs.

When considering labelled graphs, we already have an inherent way to map symbols to vertices, namely the labelling function. This allows us to interpret the free variables of a formula in a more convenient way. For this, we restrict the syntax of C to only include variables of the form x_i for $i \in \mathbb{N}^+$. Intuitively, each subscript i corresponds to a label and each variable x_i to the vertex $\nu(i)$. Let us denote this alternate definition by $\mathcal{C}$ to distinguish it from the standard definition.

Definition 3.1 (Semantics of $\mathcal{C}$)**.** Let $\varphi \in \mathcal{C}$ and let G be a labelled graph such that L_G contains the subscripts of all free variables in φ. (We also say G is *appropriately* labelled).

It is $G \models_{\mathcal{C}} \varphi$ if

- φ is $x_i = x_j$ and $\nu_G(i) = \nu_G(j)$,
- φ is Ex_ix_j and $(\nu_G(i), \nu_G(j)) \in E(G)$,
- φ is $\neg\psi$ and $G \not\models_{\mathcal{C}} \psi$,
- φ is $\psi \vee \vartheta$ and $G \models_{\mathcal{C}} \psi$ or $G \models_{\mathcal{C}} \vartheta$, or
- φ is $\exists^{\geq t} x_l \psi(x_l)$ and there are distinct vertices $v_1, \dots, v_t \in V(G)$, such that $G(l \to v_i) \models \psi(x_l)$ for each $i \in [t]$.

Note that each $\mathcal{C}$ formula is also a C formula, and we can obtain a $\mathcal{C}$-formula from every C formula simply by renaming the variables. It is also not hard to see that for each $\varphi \in \mathcal{C}$ and appropriately labelled graphs G, there exists an interpretation function $\mathfrak{I} \colon \mathrm{free}(\varphi) \to V(G)$, such that $G \models_{\mathcal{C}} \varphi$ if and only if $(G, \mathfrak{I}) \models_{\mathsf{C}} \varphi$.

Lemma 3.2. *Let φ be a $\mathcal{C}$ sentence. Then for appropriately labelled graphs G, the following are equivalent.*

1. $G \models_{\mathcal{C}} \varphi$
2. $G \models_{\mathsf{C}} \varphi$

In the rest of this thesis, we will use the two formalisations interchangeably, depending on which is more convenient.

3.2 The Bijective Pebble Game

— Is this a game or is it real?
— What's the difference?

Matthew Broderick, John Wood, in: WarGames

The bijective pebble game is a game-theoretic approach to determine whether two structures – in our case graphs – are C-equivalent. It was introduced by Hella [11] and is based on the concept of Ehrenfeucht-Fraïssé games for first-order logic.

The bijective pebble game is played by two players, *Spoiler* and *Duplicator*, on two graphs, G and H. Duplicator wants to prove that the graphs are equivalent, whereas Spoiler tries to show that they are different. The game is played in rounds. In each round, Duplicator first chooses a bijection $f_i \colon G \to H$. Spoiler then places a pebble on one of the vertices v_i of G. Duplicator then has to place a corresponding pebble on $f_i(v_i)$. Let v_j denote the vertex chosen by Spoiler in the j-th round and v'_j the corresponding vertex chosen by Duplicator. Spoiler wins a play in i rounds if the mapping $(v_j \mapsto v'_j \mid j \in [i])$ is *not* an isomorphism between the subgraphs induced by $\{v_1, \dots, v_i\}$ and $\{v'_1, \dots, v'_i\}$, respectively.

Formally, *positions* of the game are pairs $(\mathbf{v}, \mathbf{v}')$ where $\mathbf{v} \in V(G)^\ell$ and $\mathbf{v}' \in V(H)^\ell$. A *play* is a sequence of rounds, starting from some initial position $(\mathbf{v_0}, \mathbf{v'_0})$. Unless mentioned otherwise, we assume the initial position to be the empty pair $((), ())$. If in the i-th round the game is in position $(\mathbf{v}, \mathbf{v}')$, Duplicator chooses a bijection f_i, Spoiler chooses a vertex v_i, and the game moves to position $(\mathbf{v} \oplus v_i, \mathbf{v}' \oplus f_i(v_i))$. A position $(\mathbf{v}, \mathbf{v}')$ is *winning* for Duplicator if the mapping $(\mathbf{v}_j \mapsto \mathbf{v}'_j \mid j \in [i])$ is a *local isomorphism* between G and H.

Definition 3.3. A function $f \colon G \to H$ is a *local isomorphism* between G and H if

- f is injective,
- for all $v, w \in \mathrm{dom}(f)$, it is $(v, w) \in E(G) \Leftrightarrow (f(v), f(w)) \in E(H)$,
- for all $v \in \mathrm{dom}(f)$, it is $\gamma^G(v) = \gamma^H(f(v))$ (for coloured graphs).

In other words, a function f is a local isomorphism precisely if it is an isomorphism between the subgraphs induced by $\mathrm{dom}(f)$ and $\mathrm{img}(f)$.

3.2.1 The m-round game

In the m-round bijective pebble game on G and H with initial position $(\mathbf{v_0}, \mathbf{v'_0})$, denoted by $\mathcal{G}_m(G, \mathbf{v_0}; H, \mathbf{v'_0})$, a play ends after m rounds. If by that time the game is in a position that is winning for Duplicator, he wins the play. Otherwise Spoiler wins. As usual with game-theoretic constructions like this, we are not interested in the outcome of any particular play, but instead whether one of the players has a *winning strategy*. A winning strategy for a player P, intuitively, is a function that assigns to each position a responding move that guarantees that the game eventually reaches a position that is winning for P.

The existence of a winning strategy for Duplicator in the m-round game coincides with C_m-equivalence, as the following lemma shows. The lemma itself is a special case of a theorem due to Hella [11].

Lemma 3.4. *Let G and H be graphs of the same order, $\ell, m \in \mathbb{N}^+$, and let $\mathbf{v} \in V(G)^\ell$ and $\mathbf{v}' \in V(H)^\ell$. Then the following are equivalent.*

1. *Duplicator has a winning strategy for $\mathcal{G}_m(G, \mathbf{v}; H, \mathbf{v}')$.*
2. *$G, \mathbf{v} \equiv_{\mathsf{C}_m} H, \mathbf{v}'$.*

The idea behind the proof is that a local isomorphism ι between G and H captures that for the elements of $\mathrm{dom}(\iota)$ and $\mathrm{img}(\iota)$, G and H satisfy the same atomic formulae. As a consequence, they also satisfy the same disjunctions of atomic formulae and their negations. The alternating nature of the game then gives us a characterisation of quantifiers: If G satisfies a formula of the form $\exists^{\geq t} x \varphi(x)$, there are distinct vertices $v_1, \dots, v_t$, such that $G \models \varphi(v_i)$ for $i \in [t]$. By choosing a bijection at the start of the round, Duplicator identifies those vertices of H that he claims correspond to $v_1, v_2, \dots$, respectively, in the sense that they satisfy $\varphi(x)$ in H. If now $H \not\models \exists^{\geq t} x \varphi(x)$, then there are only vertices $v'_1, \dots, v'_k$ for $k < t$, such that $H \models \varphi(v'_i)$. As such, regardless which bijection f Duplicator picks, Spoiler can always choose a vertex $v \in \{v_1, \dots, v_t\}$ in a way that $f(v) \notin \{v'_1, \dots, v'_k\}$, and thus show that Duplicator's initial claim was false.

3.2.2 The m-round k-pebble game

We obtained the logic C_m by restricting the quantifier-rank of C-formulae. Instead of only restricting quantifier-rank, we can additionally restrict the number of distinct vertices that are allowed to occur in a formula, resulting in the k-variable, quantifier-rank-m fragment C^k_m. It seems natural to ask whether can tweak the definition of the m-round bijective pebble game to obtain a characterisation of C^k_m-equivalence. And indeed, this turns out to be the case.

Observe that the truth value of a disjunction of (negated) atomic formulae that uses only k variables can only depend on k vertices. As such, it seems intuitive that Duplicator only has to show the existence of local isomorphisms ι with $|\mathrm{dom}(\iota)| = |\mathrm{img}(\iota)| \leq k$. We can translate this constraint to the pebble game by only allowing Spoiler to use k different pebbles. We thus obtain the following modification of the m-round bijective pebble game.

The m-round bijective k-pebble game on G and H, denoted by $\mathcal{G}^k_m(G, H)$, is played like the regular m-round game, only with k pairs of pebbles. As long as there are still pebbles off the board, Spoiler may proceed as usual and Duplicator answers accordingly. Once there are no pebbles left, she must move one of the already placed ones between vertices of G. Duplicator then has to respond by moving the corresponding pebble in H.

Formally, positions in the k-pebble game are pairs $(\mathbf{v}, \mathbf{v}')$ where $\mathbf{v} \in (V(G) \cup \{\bot\})^k$ and $\mathbf{v}' \in (V(H) \cup \{\bot\})^k$. We assume the default initial position to be $((\bot, \dots, \bot), (\bot, \dots, \bot))$. If in the i-th round the game is in position $(\mathbf{v}, \mathbf{v}')$, Duplicator chooses a bijection f_i and Spoiler chooses a vertex v_i, as well as an index $j \in [k]$. The game then moves to position $(\mathbf{v}(j \to v_i), \mathbf{v}'(j \to f_i(v_i)))$. A position $(\mathbf{v}, \mathbf{v}')$ is winning for Duplicator if the mapping $(v_j \mapsto v'_j \mid j \in [i], v_j \neq \bot, v'_j \neq \bot)$ is a local isomorphism. To properly relate tuples of vertices to positions of the k-pebble game, let us denote by $\mathbf{v}^\bot$ the tuple $\mathbf{v} \oplus (\bot)^{k-|\mathbf{v}|}$.

The following lemma shows that this modification indeed captures C^k_m-equivalence.[1]

Lemma 3.5. *Let G and H be graphs of the same order, $k, \ell, m \in \mathbb{N}^+$, and $\ell < k$. Moreover let $\mathbf{v} \in V(G)^\ell$ and $\mathbf{w} \in V(H)^\ell$. Then the following are equivalent.*

1. *Duplicator has a winning strategy for $\mathcal{G}^k_m(G, \mathbf{v}^\bot; H, \mathbf{w}^\bot)$.*
2. *$G, \mathbf{v} \equiv_{\mathsf{C}^k_m} H, \mathbf{w}$.*

[1] The idea of restricting the number of pebbles to characterise k-variable fragments is not new [12], but since to our knowledge there is no reference for an explicit characterisation in the case of counting first-order logic, we give a formal proof.

Proof. (1) $\implies$ (2). We show the result by contraposition: Let $\varphi(\mathbf{x}) \in \mathsf{C}^k_m$ be a formula such that $G \models \varphi(\mathbf{v})$ and $H \not\models \varphi(\mathbf{w})$. Then Spoiler has a winning strategy for $\mathcal{G}^k_m(G, \mathbf{v}^\perp; H, \mathbf{w}^\perp)$. The proof is per induction over m.

$m = 0$: In this case, $\varphi(\mathbf{x})$ is a boolean combination of atomic formulae. In particular, this means that $G, \mathbf{v}$ and $H, \mathbf{w}$ are already distinguished by an atom. Consequently, the mapping $(v_j \mapsto w_j \mid j \in [i])$ cannot be a local isomorphism and Spoiler wins $\mathcal{G}^k_0(G, \mathbf{v}^\perp; H, \mathbf{w}^\perp)$.

$m \to m+1$: In this case, $\varphi(\mathbf{x})$ is a boolean combination of formulae ψ with $\mathrm{qr}(\psi) \leq m+1$. Consequently, there has to be a formula ϑ among them that already distinguishes $G, \mathbf{v}$ and $H, \mathbf{w}$. If $\mathrm{qr}(\vartheta) < m+1$, then the statement follows immediately from the induction hypothesis, so assume w.l.o.g. that $\vartheta(\mathbf{x}) = \exists^{\geq t} x_j \psi(\mathbf{x}, x_j)$,[2] for some $j \in [k]$, with $\mathrm{qr}(\psi) = m$. It is thus either

1. $G \models \vartheta(\mathbf{v})$ and $H \not\models \vartheta(\mathbf{w})$, or
2. $G \not\models \vartheta(\mathbf{v})$ and $H \models \vartheta(\mathbf{w})$.

In case (1) there are t distinct elements $u_1, \dots, u_t \in V(G)$, such that $G \models \psi(\mathbf{v}, u_i)$ for $i \in [t]$. Since $H \not\models \vartheta(\mathbf{w})$, there are only $s < t$ distinct elements $u'_1, \dots, u'_s \in V(H)$ with $H \models \psi(\mathbf{w}, u'_i)$ for $i \in [k]$. Now suppose Duplicator chose some bijection f at the beginning of the round. Spoiler can always choose a u_i in a way that $f(u_i) \notin \{u'_1, \dots, u'_s\}$. She then places the j-th pebble on u_i. It is $G \models \psi(\mathbf{v}, u_i)$ and $H \not\models \psi(\mathbf{w}, f(u_i))$, so per induction hypothesis she wins the game $\mathcal{G}^k_m(G, \mathbf{v}^\perp(j \to u_i); H, \mathbf{w}^\perp(j \to f(u_i))$ and thus $\mathcal{G}^k_{m+1}(G, \mathbf{v}^\perp; H, \mathbf{w}^\perp)$.

The argument for case (2) is analogous, except Spoiler now chooses a $u' \in V(H)$, such that $H \models \psi(\mathbf{w}, u')$ and $G \not\models \psi(\mathbf{v}, f^{-1}(u'))$.

(2) $\implies$ (1). Let $n \coloneqq |G| = |H|$. We define a modified notion of logical equivalence of formulae by letting $\varphi \equiv_n \psi$ if $F \models \varphi \iff F \models \psi$ for all graphs F with $|F| \leq n$. In the following, when constructing sets of formulae we tacitly assume that they contain at most one representative from each class. In particular, this implies that all these sets are finite. We further assume without loss of generality that for any $\varphi \in \mathsf{C}^k$, the variables of φ are among $x_1, \dots, x_k$.

The proof is now again per induction over m.

$m = 0$: Suppose $G, \mathbf{v} \equiv_{\mathsf{C}^k_0} H, \mathbf{w}$ and assume for contradiction that $h \coloneqq (v_i \mapsto w_i \mid i \in [\ell])$ is not a local isomorphism. First assume h were not injective. Then there are indices $i, j \in [\ell]$, such that $v_i \neq v_j$ but $w_i = w_j$. But then for $\varphi(\mathbf{x}) \coloneqq [x_i \neq x_j] \in \mathsf{C}^k_0$, we have $G \models \varphi(\mathbf{v})$ and $H \not\models \varphi(\mathbf{w})$. This contradicts our assumption that $G, \mathbf{v} \equiv_{\mathsf{C}^k_0} H, \mathbf{w}$, so h must be injective. That means there must be indices $i, j \in [\ell]$, such that $(v_i, v_j) \in E(G) \iff (w_i, w_j) \notin E(H)$. But then for $\varphi(\mathbf{x}) \coloneqq Ex_ix_j$, it holds that $G \models \varphi(\mathbf{v}) \iff H \not\models \varphi(\mathbf{w})$, again contradicting the assumption.

$m \to m+1$: For a graph F and a tuple $\mathbf{a}$ over $V(F)$, we define the following formula with free variables among $x_1, \dots, x_{|\mathbf{a}|}$.

$$\psi^{m+1}_{F,\mathbf{a}}(\mathbf{x}) = \bigwedge_{i \in [k]} \bigwedge_{n \leq |F|} \bigwedge_{\substack{D \subseteq V(F) \\ |D| = n}} \exists^{\geq n} x_i \bigvee_{a \in D} \bigwedge \left\{ \varphi(\mathbf{x}(i \to x_i)) \in \mathsf{C}^k_m \mid F \models \varphi(\mathbf{a}(i \to a)) \right\}. \tag{E1}$$

It holds that $G \models \psi^{m+1}_{G,\mathbf{v}}(\mathbf{v})$: Note that the second and third outer conjunctions together range over all subsets $D \subseteq V(G)$. The crucial observation is that for each clause of this conjunction, we can use D itself as a witness for the satisfiability of the counting quantifier $\exists^{\geq n} x_i$. For an appropriate choice of $a \in D$, the innermost conjunction then simplifies to

$$\bigwedge \left\{ \varphi(\mathbf{v}(i \to a)) \in \mathsf{C}^k_m \mid G \models \varphi(\mathbf{v}(i \to a)) \right\},$$

which G trivially satisfies.

[2] It might be that $x_j \in \mathbf{x}$. For brevity, when writing $\psi(\mathbf{x}, x_j)$ and $G \models \psi(\mathbf{v}, v)$ we assume that later occurences of the same variable overwrite previous occurences. In this example, x_j would be interpreted by v and not by $v_j \in \mathbf{v}$.

Since $G \models \psi^{m+1}_{G,\mathbf{v}}(\mathbf{v})$ and $\psi^{m+1}_{F,\mathbf{a}} \in \mathsf{C}^k_{m+1}$, it holds per assumption that also $H \models \psi^{m+1}_{G,\mathbf{v}}(\mathbf{w})$. Observe that this implies that for every $i \in [k]$ and every subset $D \subseteq V(G)$, there is a subset $D' \subseteq V(H)$ of the same cardinality such that for every $d' \in D'$ there is a $d \in D$, in a way that $G, \mathbf{v}(i \to d) \equiv_{\mathsf{C}^k_m} H, \mathbf{w}(i \to d')$. A symmetric result follows from $H \models \psi^{m+1}_{H,\mathbf{w}}(\mathbf{w})$ and $G \models \psi^{m+1}_{H,\mathbf{w}}(\mathbf{v})$.

Claim 1. *There exists a bijection $f\colon G \to H$, such that for all $i \in [k]$ and $v \in V(G)$ it holds that $G, \mathbf{v}(i \to v) \equiv_{\mathsf{C}^k_m} H, \mathbf{w}(i \to f(v))$.*

Proof of the claim. We define an equivalence relation $\sim$ as follows. We let $x \sim y$ if and only if $X, \mathbf{a}_X(i \to x) \equiv_{\mathsf{C}^k_m} Y, \mathbf{a}_Y(i \to y)$ for all $i \in [k]$, where $X, Y \in \{G, H\}$, $x \in V(X), y \in V(Y)$, $\mathbf{a}_G = \mathbf{v}$ and $\mathbf{a}_H = \mathbf{w}$. For $j \in [n]$, let $\mathcal{K}_j$ denote the finitely many $\sim$-equivalence classes. Further let $p_j = \mathcal{K}_j \cap V(G)$ and $p'_j = \mathcal{K}_j \cap V(H)$. We show that

$$|p_j| = |p'_j| \text{ for } j \in [n],$$

which implies the claim. It is $p_j \subseteq V(G)$, so by $H \models \psi^{m+1}_{G,\mathbf{v}}(\mathbf{w})$ and our previous observation there is a corresponding $q_j \subseteq V(H)$ of the same cardinality, such that for all $w \in q_j$ there is a $v \in p_j$, in a way that $G, \mathbf{v}(i \to v) \equiv_{\mathsf{C}^k_m} H, \mathbf{w}(i \to w)$. Per transitivity, that implies $q_j \subseteq \mathcal{K}_j$ and thus $q_j \subseteq p'_j$. Consequently $|p_i| = |q_j| \leq |p'_i|$. The converse follows from a symmetric argument. ■

Duplicator now chooses this bijection. Suppose Spoiler chooses a vertex $v \in V(G)$ and index $i \in [k]$. Then $G, \mathbf{v}(i \to v) \equiv_{\mathsf{C}^k_m} H, \mathbf{w}(i \to f(v))$, so per induction hypothesis Duplicator wins the game $\mathcal{G}^k_m(G, \mathbf{v}^\perp(i \to v); H, \mathbf{w}^\perp(i \to f(v))$ and thus $\mathcal{G}^k_{m+1}(G, \mathbf{v}^\perp; H, \mathbf{w}^\perp)$. □

We can generalise the two games to obtain characterisations for C and C^k, respectively. The key observation is that two graphs G, H are C-equivalent (C^k-equivalent) if and only if they are C_m-equivalent (C^k_m-equivalent) for all $m \in \mathbb{N}$. Using this fact, we obtain the games $\mathcal{G}(G, H)$ and $\mathcal{G}^k(G, H)$. Here Spoiler chooses an arbitrary $m \in \mathbb{N}$ at the beginning of each play. Spoiler and Duplicator then proceed to play the corresponding game $\mathcal{G}_m(G, H)$ or $\mathcal{G}^k_m(G, H)$. Consequently, Duplicator has a winning strategy for $\mathcal{G}(G, H)$ if and only if he has a winning strategy for all $\mathcal{G}_m(G, H)$.

3.3 Characterising C^k

There are several reasons why we are particularly interested in the k-variable and quantifier-rank-q fragments of C. As the astute reader might have already noticed, C and FO have the same expressive power, since each formula $\varphi = \exists^{\geq t} x \vartheta(x)$ is equivalent to a formula

$$\psi = \exists x_1 \dots \exists x_t \left(\bigwedge_{1 \leq i < j \leq t} x_i \neq x_j \wedge \bigwedge_{1 \leq i \leq t} \vartheta(x_i) \right).$$

Therefore, if we wanted to reason about C, we could rather consider FO and avoid the added complexity of the counting quantifiers. The translation from C- to FO-formulae however increases both the quantifier rank and the number of required variables. In fact, both the k-variable fragment C^k and the quantifier-rank-q fragment C_q are strictly more expressive than their first-order logic counterparts FO^k and FO_q, respectively.

Particularly when considering characterisations in terms of homomorphism indistinguishability, there is another reason for considering less expressive fragments of C. Namely, there is an FO sentence – and thus a C sentence – capturing a finite graph (any finite structure, really) up to isomorphism.

Proposition 3.6. *For every graph G there is a* FO *sentence φ_G, such that for all graphs H*

$$G \cong H \iff H \models \varphi_G.$$

Proof. Suppose $V(G) = \{a_1, \dots, a_n\}$ and let Ψ denote the set of atomic formulae and their negations. Then

$$\varphi_G = \exists x_1 \dots \exists x_n \left(\bigwedge_{1 \le i < j \le n} x_i \neq x_j \wedge \forall y \bigvee_{1 \le i \le n} y = x_i \wedge \bigwedge \{ \psi(\mathbf{x}) \in \Psi \mid G \models \psi(\mathbf{a}) \} \right).$$

□

Consequently, in the case of finite graphs C-equivalence coincides with isomorphism, which is already characterised by the class of all graphs [14].[3] When trying to expand the big picture of graph classes and the equivalence relations they induce in terms of homomorphism-counts, it might be more interesting to investigate fragments of C that are not yet known to have such a characterisation.

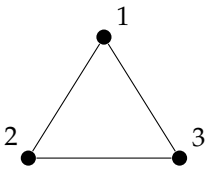
The k-variable fragment C^k specifically turns out to be especially meaningful, as it quite naturally characterises the $(k-1)$-dimensional Weisfeiler-Leman algorithm, a well-known heuristic for graph-isomorphism testing [6]. In the following, we want to explore the characterisation of C^k-equivalence in terms of homomorphism-indistinguishability. We show that for any two graphs G, H, it holds that $G \equiv_{\mathsf{C}^{k+1}} H$ if and only if $\hom(\mathcal{TW}_k, G) = \hom(\mathcal{TW}_k, H)$. We roughly follow the presentation of Dvořák's original paper.

3.3.1 From Logic to Homomorphism Counts

For the rest of the section we will mainly consider labelled graphs. Recall that we require homomorphisms between labelled graphs to preserve the labelling. Furthermore, note that every graph without labels is also an ℓ-labelled graph, for any $\ell \in \mathbb{N}$, since we do not require the labels to actually occur in the graph. This means that the following results generalise immediately to normal graphs.

Let us begin with an example to illustrate how we might relate homomorphism counts to formulae of C^k.

Example 3.7. Consider the following labelled graph G.



Note that every homomorphism from G to another graph H is completely determined by the position of the labels in H. More precisely, if $h \colon G \to H$ is a homomorphism from G to H, then $h = \{\nu^G(i) \mapsto \nu^H(i) \mid i \in [3]\}$. This means that there is exactly one homomorphism from G to H if h is a homomorphism between the underlying graphs. Otherwise, $\hom(G, H) = 0$. Consequently, the $\mathcal{C}$-formula

$$\varphi_m = \begin{cases} Ex_1x_2 \wedge Ex_2x_3 \wedge Ex_3x_1 & \text{if } m = 1 \\ \neg Ex_1x_2 \vee \neg Ex_2x_3 \vee \neg Ex_3x_1 & \text{if } m = 0 \\ \text{false} & \text{otherwise} \end{cases}$$

[3] There is definitely merit in finding smaller classes that characterise graph isomorphism. The characterisation of isomorphism in terms of C-equivalence seems – at least prima facie – like a promising approach to that end. We discuss this briefly in Chapter 5.

encodes that there are precisely m homomorphisms from G to H, that is, $H \models \varphi_m$ if and only if $\hom(G, H) = m$. $\triangleleft$

It turns out that we can generalise this idea of expressing homomorphism counts as C^k-formulae to arbitrary graphs G. Moreover, if $\mathrm{tw}(G) = k$, we only need $k+1$ different variables. Recall the inductive characterisation of graphs of tree-width at most k. A $(k+1)$-labelled graph has tree-width at most k if it can be constructed from a set of fully labelled graphs by repeatedly

1. Taking two graphs G_1, G_2 and constructing their product, or
2. Taking a single graph and removing one of its labels.

We will use this characterisation to show our desired result. To help with case (2), we prove the following helpful decomposition lemma.

Lemma 3.8. *Let G, H be labelled graphs and $m \in \mathbb{N}^+$. Furthermore, let G' be the graph obtained from G by removing a label l. Then $\hom(G', H) = m$ if and only if there is a decomposition $m = \sum_{i=1}^{t} c_i m_i$ for $c_i, m_i \in \mathbb{N}^+$, such that*

1. *$m_1, \dots, m_t$ are mutually distinct*
2. *$\hom(G, H(l \to v)) \neq 0$ for exactly $\sum_{i=1}^{t} c_i$ vertices $v \in V(H)$*
3. *$\hom(G, H(l \to v)) = m_i$ for exactly c_i vertices $v \in V(H)$*

Proof. Suppose there is such a decomposition. For $i \in [t]$, let C_i be the set of vertices v satisfying

$$\hom(G, H(l \to v)) = m_i. \tag{E1}$$

Per definition, $|C_i| = c_i$. Furthermore, let $\Phi_i \coloneqq \bigcup_{v \in C_i} \mathrm{Hom}(G, H(l \to v))$ be the set of the corresponding $c_i m_i$ homomorphisms. Assume there were indices $1 \leq i < j \leq t$ such that $\Phi_i \cap \Phi_j \neq \varnothing$. Let $f \in \Phi_i \cap \Phi_j$. Per definition, this means there is a $v \in C_i$ and a $v' \in C_j$, such that f is both a homomorphism from G to $H(l \to v)$ and from G to $H(l \to v')$. Since f preserves labels, it is $v = \nu_{H(l \to v)}(l) = f(\nu_G(l)) = \nu_{H(l \to v')}(l) = v'$. It follows that there is a $v \in C_i \cap C_j$. But then (E1) implies $m_i = \hom(G, H(l \to v)) = m_j$, contradicting (1). Consequently, all C_i and Φ_i are disjoint. Since every homomorphism from G to H is also a homomorphism from G' to H, it follows that $\hom(G', H) \geq \bigcup_{i=1}^{t} \Phi_i = \sum_{i=1}^{t} c_i m_i = m$.

Finally, assume for contradiction that $\hom(G', H) > m$. Then there exists a homomorphism f from G' to H, that is not contained in any Φ_i. Let $v = f(\nu_G(l))$. Then f is also a homomorphism from G to $H(l \to v)$. Observe that $v \notin \bigcup_i C_i$. Since the C_i are disjoint, this implies that $\hom(G, H(l \to v)) \neq 0$ for at least $1 + \sum_{i=1}^{t} |C_i| = 1 + \sum_{i+1}^{t} c_i > \sum_{i+1}^{t} c_i$ vertices $v \in V(H)$, contradicting (2).

Now suppose $\hom(G', H) = m$. Consider some arbitrary $f \in \mathrm{Hom}(G', H)$ and let $w_G = \nu_G(l) \in V(G')$. Then f is a homomorphism from G to $H(l \to f(w_G))$. Consider the sets

$$\Phi_v \coloneqq \{f \in \mathrm{Hom}(G', H) \mid f(w_G) = v\}$$

for $v \in V(H)$. We define an equivalence relation $\sim$ by letting $\Phi_v \sim \Phi_{v'}$ if $|\Phi_v| = |\Phi_{v'}|$. Let $\mathcal{K}_i$ denote the $\sim$-equivalence classes. We now let $c_i = |\mathcal{K}_i|$ and $m_i = |\Phi_v|$ for some $\Phi_v \in \mathcal{K}_i$.

It is easy to see that $\sum_{i=1}^{t} c_i m_i = m$ by observing that the Φ_v form a partition of $\mathrm{Hom}(G', H)$. Properties (1)-(3) follow immediately from the definitions and the fact that $\sim$ is an equivalence relation. $\square$

We now have all the tools we need to prove the following central lemma.

Lemma 3.9. *Let $G \in \mathcal{TW}_k$ be a $(k+1)$-labelled graph, and let $m \in \mathbb{N}$. Then there exists a formula $\varphi_m \in \mathcal{C}^{k+1}$ such that for each $(k+1)$-labelled graph H with $L_G \subseteq L_H$, $H \models \varphi_m$ if and only if $\hom(G, H) = m$.*

Proof. We prove the lemma inductively over the recursive construction of G. First assume $V(G) = L(G)$ and w.l.o.g. $k \geq 1$. This case is similar to Example 3.7: Since $L_G \subseteq L_H$, there is at most one homomorphism from G to H. We thus let $\varphi_1 = \bigwedge_{i,j \in L_G} (\psi_{i,j} \wedge \vartheta_{i,j})$, where

$$\psi_{i,j} = \begin{cases} x_i = x_j & \text{if } \nu_G(i) = \nu_G(j) \\ \text{true} & \text{otherwise} \end{cases}$$

captures that the labelling can be preserved, while

$$\vartheta_{i,j} = \begin{cases} Ex_ix_j & \text{if } (\nu_G(i), \nu_G(j)) \in E(G) \\ \text{true} & \text{otherwise} \end{cases}$$

captures the edge relation. We then let $\varphi_0 = \neg\varphi_1$ and $\varphi_m = \text{false}$ if $m > 1$.

Now suppose G is the product of two graphs G_1, G_2 of tree-width at most k. Per induction hypothesis there are formulae φ^1_m and φ^2_m, such that $\hom(G_i, H) = m$ if and only if $H \models \varphi^i_m$ for appropriately labelled graphs H and $i \in \{1, 2\}$. If $m \geq 1$ we let φ_m be the disjunction over all m_1, m_2 with $m = m_1 m_2$ of formulae $\varphi^1_{m_1} \wedge \varphi^2_{m_2}$. Otherwise, we let $\varphi_m = \varphi^1_0 \vee \varphi^2_0$.

Finally, suppose we obtain G from a graph G' of tree-width at most k by removing some label l. Again per induction hypothesis there is a formula φ'_m for each $m \in \mathbb{N}$, such that for each appropriately labelled graph H, it holds that $H \models \varphi'_m$ if and only if $\hom(G, H) = m$. By Lemma 3.8, it is $\hom(G, H) = m$ if and only if there is a decomposition $m = \sum_{i=1}^{t} c_i m_i$, such that $\hom(G', H(l \to v)) = m_i$ for exactly c_i vertices, and $\hom(G', H(l \to v)) \neq 0$ for exactly $c = \sum_{i=1}^{t} c_i$ vertices $v \in V(H)$. Note that the operation $H(l \to v)$ – assigning label l to vertex v – coincides with the $\mathcal{C}$ semantics of the counting quantifier. Consequently, we can simply define φ_m to be the disjunction over all such decompositions of formulae

$$\psi_{c_1,m_1,\ldots,c_t,m_t} = \exists^{=c} x_l \, \neg\varphi'_0 \wedge \bigwedge_{i \in [t]} \exists^{=c_i} x_l \, \varphi'_{m_i}.$$

Since in all cases we only use variables x_i for $i \in [k+1]$, we indeed have $\varphi_m \in \mathcal{C}^{k+1}$. □

Observe that for graphs G that do not contain labels, every label used in the recursive construction of G needs to be removed at some point. Per our construction, this implies that every variable x_i in φ_m is bound by a quantifier.

Corollary 3.10. *Let $G \in \mathcal{TW}_k$ be a graph and let $m \in \mathbb{N}$. Then there exists a sentence $\varphi_m \in \mathcal{C}^{k+1}$, such that for each graph H it holds that $H \models \varphi_m$ if and only if $\hom(G, H) = m$.*

3.3.2 Quantum Graphs and Models

We have shown that $\mathcal{C}^{k+1}$-equivalence is strong enough to capture homomorphism counts over $\mathcal{TW}_k$. The converse direction is slightly more involved. The idea is to somehow encode a $\mathcal{C}$ sentence φ as a graph G, such that for some graph H, $\hom(G, H)$ differs based on whether $H \models \varphi$ or not.

A very useful tool to build such an encoding will be what we call *quantum graphs*.[4]

Definition 3.11. A (labelled) *quantum graph* $|G\rangle = \sum_{i=1}^{t} a_i G_i$ is a finite linear combination of (labelled) graphs G_i with coefficients $a_i \in \mathbb{R}$. For a class of graphs $\mathcal{F}$, we say $|G\rangle$ is a *quantum graph over* $\mathcal{F}$ if $G_i \in \mathcal{F}$ for all $i \in [t]$.

We extend the $\hom(\cdot, \cdot)$ function to quantum graphs in a linear way by letting $\hom(|G\rangle, H) = \sum_{i=1} a_i \hom(G_i, H)$. Note that this definition implies that if $\hom(|G\rangle, H) \neq \hom(|G\rangle, H')$, then there is already a G_i, such that $\hom(G_i, H) \neq \hom(G_i, H')$.

[4] To our knowledge, quantum graphs were first introduced by Lovsz and Szegedy [15]. Most mentions of quantum graphs in the literature however refer to a different concept.

Proposition 3.12. *Let H, H' be graphs and let $\mathcal{F}$ be a class of graphs. If a quantum graph over $\mathcal{F}$ distinguishes H and H', then so does $\mathcal{F}$.*

We define the product $|G\rangle\,|G'\rangle$ of two quantum graphs $|G\rangle = \sum_{i=1}^{s} a_i G_i$ and $|G'\rangle = \sum_{i=1}^{t} b_i G'_i$ to be $\sum_{i,j} a_i b_j G_i G'_j$. In intuitive terms, the product operation simply distributes over the linear combination.

Quantum graphs allow us to work with homomorphism counts in a quite elegant way. The following construction, for instance, will be very useful to prove the result of this section.

Lemma 3.13. *Let $\mathcal{F}$ be a class of graphs and let $|G\rangle$ be a (labelled) quantum graph over $\mathcal{F}$. If S_+, S_- are disjoint finite sets of real numbers, then there exists a (labelled) quantum graph $|G[S_-; S_+]\rangle$, such that for any graph H*

- $\hom(|G[S_-; S_+]\rangle, H) = 1$ *if* $\hom(|G\rangle, H) \in S_+$, *and*
- $\hom(|G[S_-; S_+]\rangle, H) = 0$ *if* $\hom(|G\rangle, H) \in S_-$.

Moreover, if $\mathcal{F}$ is closed under products then $|G[S_-; S_+]\rangle$ is also a quantum graph over $\mathcal{F}$.

Proof. Fix some polynomial $p(x) = \sum_{i=1}^{t} a_i x^i$ with $p(x) = 1$ for $x \in S_+$ and $p(x) = 0$ for $x \in S_-$. We let $|G[S_-; S_+]\rangle = \sum_{i=1}^{t} a_i\,|G\rangle^i$. Observe that for two quantum graphs $|G\rangle, |G'\rangle$ and a graph H, it holds that $\hom(|G\rangle\,|G'\rangle, H) = \hom(|G\rangle, H)\hom(|G'\rangle, H)$. We thus obtain

$$\hom(|G[S_-; S_+]\rangle, H) = \sum_{i=1}^{s} a_i \hom(|G\rangle^i, H) = \sum_{i=1}^{s} a_i \hom(|G\rangle, H)^i = p(\hom(|G\rangle, H)),$$

which proves the lemma. □

Let us now concretise our idea of the "encoding" of a $\mathcal{C}$ sentence.

Definition 3.14. Let φ be a C formula. We say a quantum graph $|G\rangle$ *models φ for graphs of size n*, if for all graphs H of order n it holds that

- $\hom(|G\rangle, H) = 1$ if $H \models \varphi$, and
- $\hom(|G\rangle, H) = 0$ otherwise.

The following lemma completes our picture of $\mathcal{C}^{k+1}$-equivalence.

Lemma 3.15. *Let $k \geq 1$ and let φ be a $\mathcal{C}^{k+1}$ formula. Then for every $n \in \mathbb{N}^+$ there exists a quantum graph $|G\rangle$ over $\mathcal{TW}_k$ modelling φ for graphs of size n.*

Proof. The proof is per induction over the structure of φ. If $\varphi = [x_i = x_j]$, we let $|G\rangle = G$ be the graph consisting of a single vertex v with $\nu_G(i) = \nu_G(j) = v$. If $\varphi = Ex_ix_j$, we let $|G\rangle = G$ be the graph consisting of two adjacent vertices v_1, v_2 with $\nu_G(i) = v_1$ and $\nu_G(j) = v_2$. If $i = j$, we let $|G\rangle = 0$, the empty linear combination, since our graphs are loopless and thus Ex_ix_i is unsatisfiable.

If $\varphi = \neg\psi$ then per induction hypothesis there exists a quantum graph $|G_\psi\rangle$ modelling ψ for graphs of size n. Using the construction from Lemma 3.13 we let $|G\rangle = |G_\psi[1; 0]\rangle$.

If $\varphi = \psi \vee \vartheta$ then again there exist quantum graphs $|G_\psi\rangle$ and $|G_\vartheta\rangle$ modelling ψ and ϑ, respectively. We let $|H\rangle = |G_\psi\rangle + |G_\vartheta\rangle$ and set $|G\rangle = |H[0; 1, 2]\rangle$.

Finally, suppose $\varphi = \exists^{\geq t} x_l \psi$ and let $|G_\psi\rangle = \sum_{i=1}^{t} a_i G_{\psi,i}$ be the quantum graph modelling ψ. We construct the graph $|G'_\psi\rangle$ by removing the label l from all $G_{\psi,i}$. We then have

$$\hom(|G'_\psi\rangle, H) = \sum_{v \in V(H)} \hom(|G_\psi\rangle, H(l \to v)).$$

Consequently, $|G\rangle = |G'_\psi[1,\dots,t-1;t,\dots,n]\rangle$ models φ for graphs of size n.

Note that while $\mathcal{TW}_k$ in general is not closed under products, the components of $|G\rangle$ are obtained using only label deletions and products of $(k+1)$-labelled graphs, starting with fully labelled graphs. Such a class is per definition closed under products, and by the recursive definition of tree-width a subclass of $\mathcal{TW}_k$. Consequently, $|G\rangle$ is still a quantum graph over $\mathcal{TW}_k$. □

We can now prove the main result of this section.

Theorem 3.16. *For any two graphs G, G', the following are equivalent:*

1. *G and G' satisfy the same C^{k+1} sentences.*
2. $\hom(\mathcal{TW}_k, G) = \hom(\mathcal{TW}_k, G')$

Proof. (1) $\implies$ (2): Assume $G \equiv_{\mathsf{C}^{k+1}} G'$. Let $F \in \mathcal{TW}_k$ be arbitrary and let $m = \hom(F, G)$. By Corollary 3.10 there is a sentence $\varphi \in \mathsf{C}^{k+1}$ such that for any graph H, it holds that $H \models \varphi$ if and only if $\hom(F, H) = m$. Consequently, $G \models \varphi$ and since $G \equiv_{\mathsf{C}^{k+1}} G'$ also $G' \models \varphi$. This implies $\hom(H, G') = m = \hom(H, G)$.

(2) $\implies$ (1): Assume $G \not\equiv_{\mathsf{C}^{k+1}} G'$. Then there is a C^{k+1} sentence φ such that w.l.o.g. $G \models \varphi$ and $G' \not\models \varphi$. Note that if G and G' were not of the same order, then already $\hom(\mathcal{TW}_0, G) \neq \hom(\mathcal{TW}_0, G')$. Let $n = |V(G)| = |V(G')|$. By Lemma 3.15 there is a quantum graph $|G\rangle$ over $\mathcal{TW}_k$ modelling φ for graphs of size n. Thus, $\hom(|G\rangle, G) \neq \hom(|G\rangle, G')$ and by Proposition 3.12 $\hom(\mathcal{TW}_k, G) \neq \hom(\mathcal{TW}_k, G')$. □

3.4 Characterising C_q

Arguably more fundamental than restricting the number of variables is restricting the quantifier rank. We have already seen that fragments of bounded quantifier rank emerge quite naturally when trying to characterise C-equivalence using games.[5] In general, quantifier rank can be seen as an intuitive measure of the "complexity" of a formula.

In this section, we want to take a closer look at how C_q-equivalence may be characterised in terms of homomorphism-indistinguishability. We retrace Grohe's result [10] that C_q-equivalence is characterised precisely by the class of graphs of tree-depth at most q, that is, for any two graphs G, H, it is $G \equiv_{\mathsf{C}_q} H$ if and only if $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, H)$.

3.4.1 Playing the Bijective Pebble Game

Instead of considering individual C formulae directly, like we did in the previous section, our goal is to show that two graphs G, H are homomorphism-indistinguishable over $\mathcal{TD}_q$ if and only if Duplicator wins the q-round bijective pebble game $\mathcal{G}_q(G, H)$. Unless otherwise specified, we consider all graphs in this section to be coloured. This will allow us to preserve adjacency information in the form of colours when removing vertices. Note that we can do this without loss of generality, since we can view normal graphs as coloured graphs whose vertices all have the same colour, that is, $|\Gamma| = 1$.

Let us first assume that G and H are two graphs that are homomorphism-indistinguishable over $\mathcal{TD}_q$ and ponder how a winning strategy for Duplicator in $\mathcal{G}_q(G, H)$ could look. We might observe that for Duplicator to win the 1-round game $\mathcal{G}_1(G, H)$, it suffices if for each colour $c \in \Gamma$, G and H have the same number of vertices of colour c. Then he can choose a bijection f with $\gamma^G(v) = \gamma^H(f(v))$ for all $v \in V(G)$, making $(v \mapsto f(v))$ a partial isomorphism between G and

[5] In fact, for any logic that admits similar Ehrenfeucht-type games, the number of rounds played corresponds to the quantifier rank.

H. Conveniently, it is not hard to see that this condition already follows from homomorphism-indistinguishability over $\mathcal{TD}_1$: For each colour $c \in \Gamma$, the single vertex graph F_c with $V(F_c) = \{v\}$ and $\gamma^{F_c}(v) = c$ has tree-depth 1, and for any graph X, $\hom(F_c, X)$ is equal to the number of vertices in X with colour c.

We might now hope that, similar to the proof of the characterisation of C^k, we can somehow leverage the inductive definition of tree-depth to reduce the question of whether Duplicator wins the game $\mathcal{G}_q(G, H)$ to the question of whether he wins the game $\mathcal{G}_{q-1}(G', H')$, for some G' and H' that are only homorphism-indistinguishable over $\mathcal{TD}_{q-1}$. We already know that Duplicator wins $\mathcal{G}_q(G, H)$ if there is a bijection f, such that for any $v \in V(G)$ he wins the game $\mathcal{G}_{q-1}(G, v; H, f(v))$. Using a clever construction, we can get rid of this initial position.

For a graph G and $v \in V(G)$, we let $G \wr v := G - v$ with colours given by

$$\gamma^{G \wr v}(u) = \begin{cases} (\gamma^G(u), 1) & \text{if } (u, v) \in E(G), \\ (\gamma^G(u), 0) & \text{if } (u, v) \notin E(G). \end{cases}$$

The following lemma is now straightforward.

Lemma 3.17. *Let G, H be graphs with $|G| = |H| \geq 2$, and $v \in V(G)$, $w \in V(H)$, such that $\gamma^G(v) = \gamma^H(w)$. Then the following are equivalent.*

1. *Duplicator wins $\mathcal{G}_q(G, v; H, w)$.*
2. *Duplicator wins $\mathcal{G}_q(G \wr v, H \wr w)$.*

Again, let G, H be graphs with $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, H)$. The lemma implies that for Duplicator to win the game $\mathcal{G}_q(G, H)$, it would suffice to find a bijection f, such that for all $v \in V(G)$, he wins $\mathcal{G}_{q-1}(G \wr v, H \wr f(v))$. If we could further show that $G \wr v$ and $H \wr f(v)$ are homomorphism-indistinguishable over $\mathcal{TD}_{q-1}$, we could inductively construct a winning strategy for the rest of the game. Let us illustrate with an example what such a bijection could look like.

Example 3.18. Consider the following graphs G and H.

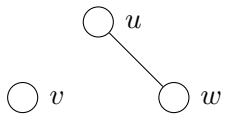


The reader may verify that $\hom(\mathcal{TD}_2, G) = \hom(\mathcal{TD}_2, H)$. Now consider the bijection $f = \{a \mapsto 1, b \mapsto 2, \ldots, e \mapsto 5\}$. Say Spoiler chooses $b \in V(G)$. Then the resulting graphs $G \wr b$ and $H \wr f(b)$ look as follows.



The two graphs are homomorphism-indistinguishable over $\mathcal{TD}_1$. This is immediately obvious from the fact that for each colour c they have the same number of vertices of colour c. Let us take a look at how we might prove this more systematically.

Let $F \in \mathcal{TD}_1$ be arbitrary, say $F = ⓪ \ \ ①$. Let us build a graph F^+ by adding a new vertex u to F, with $\gamma^{F^+}(u) = \gamma^G(b) = \gamma^H(f(b))$. We then make u adjacent to all vertices with colour $(\cdot, 1)$ and subsequently replace colours $(c, \cdot)$ by c. In our case, F^+ thus looks as follows.



It seems like there is now a one-to-one correspondence between the homomorphisms from F^+ to G mapping u to b and the homomorphisms from F to $G \wr b$, as well as between the homomorphisms from F^+ to H mapping u to $f(b)$ and the homomorphisms from F to $H \wr f(b)$.

If we take an elimination forest S for F (in our specific case S is just F itself), and make all roots of S adjacent to u, we always obtain an elimination tree for F^+ whose height is one larger than the height of S. In our case, this implies $F^+ \in \mathcal{TD}_2$. Since G and H are homomorphism-indistinguishable over $\mathcal{TD}_2$, this means $\hom(F^+, G) = \hom(F^+, H)$. For our choice of f, we moreover have $\hom(F^+, G; u \mapsto b) = \hom(F^+, H; u \mapsto f(b))$, implying

$$\hom(F, G \wr b) = \hom(F^+, G; u \mapsto b) = \hom(F^+, H; u \mapsto f(b)) = \hom(F, H \wr f(b)).$$

Since we chose $F \in \mathcal{TD}_1$ arbitrarily, we have $\hom(\mathcal{TD}_1, G \wr b) = \hom(\mathcal{TD}_1, H \wr f(b))$.

It seems like we reduced the question of the existence of a winning strategy for Duplicator to the question of whether we can always find a bijection f, such that $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, H)$ implies $\hom(D, G; r \mapsto v) = \hom(D, H; r \mapsto f(v))$ for any $D \in \mathcal{TD}_q$, where $v \in V(G)$ and r is the root of an elimination tree for D of height q.

Unfortunately, our initial assumption that there is a one-to-one correspondence between $\mathrm{Hom}(F^+, G; u \mapsto b)$ and $\mathrm{Hom}(F, G \wr b)$ is false in general. For a counterexample, consider the homomorphism $h\colon F^+ \to G$ given by $h = \{u \mapsto b, v \mapsto b, w \mapsto c\}$, which does not correspond to a homomorphism from F to $G \wr b$. A possible approach could be to restrict ourselves to a subset of *well-behaved* homomorphisms, for which the correspondence holds. We then have to hope that our construction does not break when we instead require that for any $D \in \mathcal{TD}_q$, the number of *well-behaved* homomorphisms from D to G equal the number of *well-behaved* homomorphisms from D to H. ◁

In the following, we want to tackle the questions raised in Example 3.18. It will be convenient to consider graphs together with a corresponding elimination tree. We denote by $\mathcal{D}$ the class of all pairs (F, T) where F is a graph and T an elimination tree of F. For $D = (F, T) \in \mathcal{D}$, we let $F^D := F$ and $T^D := T$. Moreover, we let $V(D) := V(F) = V(T)$, $E(D) := E(F)$, $\gamma^D := \gamma^F$, $\preceq^D := \preceq^T$ and denote by r^D the root of T. We further define $\mathcal{D}_q$ as the class of all $D \in \mathcal{D}$ where T^D has height at most q. Note that $D \in \mathcal{D}_q$ implies $F^D \in \mathcal{TD}_q$.

It turns out that our notion of *well-behaved homomorphisms* from Example 3.18 can be adequately formalised by so-called *past-preserving* homomorphisms.

Definition 3.19. Let $D \in \mathcal{D}$. A homomorphism $f\colon D \to G$ is *past-preserving* if for all $u, v \in V(D)$ with $u \prec^D v$, it is $f(u) \neq f(v)$; and additionally for all $u, v \in V(D)$ with $u \preceq^D v$, it is $(u, v) \in E(D) \iff (f(u), f(v)) \in E(G)$.

Analogously to $\hom$, we denote by $\text{pp-hom}(D, G)$ the number of past-preserving homomorphisms from D to G. For some class $\mathcal{F} \subseteq \mathcal{D}$, we denote by $\text{pp-hom}(\mathcal{F}, G)$ the (possibly infinite dimensional) vector

$$\text{pp-hom}(\mathcal{F}, G) := (\text{pp-hom}(F, G) \mid F \in \mathcal{F}).$$

Perhaps surprisingly, it turns out that this restriction of homomorphism counts remains equally expressive in terms of homomorphism-indistinguishability. More precisely, the following lemma holds.

Lemma 3.20 (Theorem 3.12 in [10])**.** *For $q \geq 0$ and graphs G, H, the following are equivalent.*

1. $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, H)$

2. $\text{pp-hom}(\mathcal{D}_q, G) = \text{pp-hom}(\mathcal{D}_q, H)$

The proof is quite technical and does not actually provide any additional insight within the scope of this thesis, so we just refer to Grohe's original paper. The result, however, means that our reasoning in Example 3.18 remains valid even when we restrict ourselves to past-preserving homomorphisms. Thus only the question remains if there always exists a bijection as described in the example. And indeed, this is the case.

Lemma 3.21. *Let G, H be graphs of the same order, such that*

$$\text{pp-hom}(\mathcal{D}_q, G) = \text{pp-hom}(\mathcal{D}_q, H).$$

Then there is a bijection $f\colon V(G) \to V(H)$, such that

$$\text{pp-hom}(D, G; r^D \mapsto v) = \text{pp-hom}(D, H; r^D \mapsto f(v))$$

for all $D \in \mathcal{D}_q$ and $v \in V(G)$.

Proof. Let $n = |G| = |H|$. We define an equivalence relation $\sim$ on $V(G) \cup V(H)$, where $x \sim y$ if for all $D \in \mathcal{D}_q$

$$\text{pp-hom}(D, X; r^D \mapsto x) = \text{pp-hom}(D, Y; r^D \mapsto y),$$

where $X, Y \in \{G, H\}$, $x \in V(X)$ and $y \in V(Y)$. Let $\mathcal{K}_i$ for $i \in [n]$ denote the finitely many $\sim$-equivalence classes. Further let $p_i = |\mathcal{K}_i \cap V(G)|$ and $p'_i = |\mathcal{K}_i \cap V(H)|$. The lemma now follows immediately from the following claim.

Claim 1. *Let $\mathbf{p} = (p_1, \dots, p_n)$ and $\mathbf{p}' = (p'_1, \dots, p'_n)$. Then $\mathbf{p} = \mathbf{p}'$.*

Proof. Observe that since there are only finitely many $\sim$-equivalence classes, we do not have to consider the entire set $\mathcal{D}_q$. Instead, for all distinct $i, j \in [n]$ we choose a $D_{ij} \in \mathcal{D}_q$ such that for $x \in \mathcal{K}_i$, $y \in \mathcal{K}_j$ and $X, Y \in \{G, H\}$ with $x \in X$ and $y \in Y$ it holds that

$$\text{pp-hom}(D_{ij}, X; r^{D_{ij}} \mapsto x) \neq \text{pp-hom}(D_{ij}, Y; r^{D_{ij}} \mapsto y).$$

We can then work with the finite class $\mathcal{F}$ of all these D_{ij} without changing the equivalence relation.

Suppose $\mathcal{F} = \{D_1, \dots, D_m\}$. For every $i \in [m]$, let $r_i = r^{D_i}$. We let $a_{ij} \coloneqq \text{pp-hom}(D_j, X; r_j \mapsto x)$ for all $x \in \mathcal{K}_i \cap V(X)$. Note that this is well-defined, since $\sim$ is an equivalence relation. We can view the a_{ij} as a $(n \times m)$-matrix $A \coloneqq (a_{ij})$. We denote by $\mathbf{a}_i$ the n row vectors and by $\mathbf{a}^j$ the m column vectors of A. Again, since $\sim$ is an equivalence relation, the $\mathbf{a}_i$ are mutually distinct.

For all $j \in [m]$ it holds that

$$\begin{aligned}\text{pp-hom}(D_j, G) &= \sum_{v \in V(G)} \text{pp-hom}(D_j, G; r_j \mapsto v) \\ &= \sum_{i=1}^{n} p_i a_{ij} \\ &= \langle \mathbf{p}, \mathbf{a}^j \rangle,\end{aligned}$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the standard scalar product of $\mathbf{x}$ and $\mathbf{y}$. Similarly, we obtain

$$\text{pp-hom}(D_j, H) = \langle \mathbf{p}', \mathbf{a}^j \rangle.$$

Before we continue, let us get some necessary notation out of the way. Let $\mathbf{d} = (d_1, \dots, d_n)$ be some vector. We call the set $\text{supp}(\mathbf{d}) \coloneqq \{i \in [n] \mid d_i \neq 0\}$ the *support* of $\mathbf{d}$. Similarly to products of labelled graphs, let us define a product operation for coloured graphs. Let $G_1, G_2 \in \mathcal{D}$ be coloured graphs with $\gamma^{G_1}(r^{G_1}) = \gamma^{G_2}(r^{G_2})$; we say they are *compatible*. The *root product* $G_1 G_2$

is the tuple (F, T), where F is the disjoint union of F^{G_1} and F^{G_2}, identifying r^{G_1} and r^{G_2}, and T is the disjoint union of T^{G_1} and T^{G_2}, again identifying the roots. For a set $\{G_1, \dots, G_k\}$ of mutually compatible graphs, we also write $\prod_{i\in[k]} G_i$ instead of $G_1 G_2 \dots G_k$. We further write G_1^k for $\prod_{i\in[k]} G_1$.

Let $\mathbf{d} \in \mathbb{N}^m$ be a non-zero vector. If for all $i \in \operatorname{supp}(\mathbf{d})$ the D_i are mutually compatible, we let

$$D^{(\mathbf{d})} \coloneqq \prod_{i\in\operatorname{supp}(\mathbf{d})} D_i^{d_i}.$$

Let us denote the root of $D^{(\mathbf{d})}$ by $r^{(\mathbf{d})}$. Observe that since the D_i are all in $\mathcal{D}_q$, so is $D^{(\mathbf{d})}$. Further observe that for $X \in \{G, H\}$ and $x \in V(X)$ we have

$$\text{pp-hom}(D^{(\mathbf{d})}, X; r^{(\mathbf{d})} \mapsto x) = \prod_{i=1}^{m} \text{pp-hom}(D_i, X; r_j \mapsto x)^{d_i}.$$

It follows that for every $i \in [n]$, $X \in \{G, H\}$ and $x \in \mathcal{K}_i \cap V(X)$ we have

$$\text{pp-hom}(D^{(\mathbf{d})}, X; r^{(\mathbf{d})} \mapsto x) = \prod_{j=1}^{m} a_{ij}^{d_j} =: \mathbf{a}_i^{(\mathbf{d})}. \tag{E2}$$

Let $\mathbf{a}^{(\mathbf{d})} = (\mathbf{a}_1^{(\mathbf{d})}, \dots, \mathbf{a}_n^{(\mathbf{d})})$. Then

$$\begin{aligned}\text{pp-hom}(D^{(\mathbf{d})}, G) &= \sum_{v\in V(G)} \text{pp-hom}(D^{(\mathbf{d})}, G; r^{(\mathbf{d})} \mapsto v) \\ &= \sum_{i=1}^{n} p_i \mathbf{a}_i^{(\mathbf{d})} \\ &= \langle \mathbf{p}, \mathbf{a}^{(\mathbf{d})} \rangle.\end{aligned}$$

Again, we analogously obtain

$$\text{pp-hom}(D^{(\mathbf{d})}, H) = \langle \mathbf{p}', \mathbf{a}^{(\mathbf{d})} \rangle.$$

Moreover, since $D^{(\mathbf{d})} \in \mathcal{D}_q$, we have

$$\langle \mathbf{p}, \mathbf{a}^{(\mathbf{d})} \rangle = \text{pp-hom}(D^{(\mathbf{d})}, G) = \text{pp-hom}(D^{(\mathbf{d})}, H) = \langle \mathbf{p}', \mathbf{a}^{(\mathbf{d})} \rangle. \tag{E3}$$

For our purposes we can let $0^0 = 1$, so for every $i \in [n]$ and every $\mathbf{d} \in \mathbb{N}^m$ we have

$$\mathbf{a}_i^{(\mathbf{d})} \neq 0 \iff \operatorname{supp}(\mathbf{d}) \subseteq \operatorname{supp}(\mathbf{a}_i). \tag{E4}$$

With all these assertions out of the way, let us now assume for contradiction that $\mathbf{p} \neq \mathbf{p}'$. Then there exists an index $k \in [n]$, such that $p_k \neq p'_k$. We choose k in a way that $\operatorname{supp}(\mathbf{a}_k)$ is inclusionwise maximal.

If $\operatorname{supp}(\mathbf{a}_k) = \varnothing$, then $\mathbf{a}_k = \mathbf{0}$. Since the vectors $\mathbf{a}_i$ are mutually distinct, we have $\mathbf{a}_i \neq \mathbf{0}$ for all $i \neq k$ and thus $\operatorname{supp}(\mathbf{a}_i) \neq \varnothing$. The maximality of $\operatorname{supp}(\mathbf{a}_k)$ now implies that it must be $p_i = p'_i$ for all $i \neq k$. But this implies $p_k = n - \sum_{i\neq k} p_i = n - \sum_{i\neq k} p'_i = p'_k$, which contradicts our initial assumption.

Suppose then $\operatorname{supp}(\mathbf{a}_k) \neq \varnothing$ and let $I = \{i \in [n] \mid \operatorname{supp}(\mathbf{a}_i) = \operatorname{supp}(\mathbf{a}_k)\}$. For every $i \in I$, let $\hat{\mathbf{a}}_i \coloneqq (a_{ij} \mid j \in S)$. Then the vectors $\hat{\mathbf{a}}_i$ have only positive entries and are mutually distinct. A known, but quite technical result now implies that there is a vector $\hat{\mathbf{d}} = (\hat{d}_j \mid j \in \operatorname{supp}(\mathbf{a}_k))$, such that the numbers $\hat{\mathbf{a}}_i^{(\hat{\mathbf{d}})}$ for $i \in I$ are mutually distinct. For a proof of said result, see Lemma 4.1 in [10].

Let $\mathbf{d} = (d_1, \dots, d_m)$ with $d_i = \hat{d}_i$ for $i \in \operatorname{supp}(\mathbf{a}_k)$ and $d_i = 0$ otherwise. Then also the numbers $\mathbf{a}_i^{(\mathbf{d})}$ for $i \in I$ are mutually distinct. Let $j\mathbf{d} = (jd_1, \dots, jd_m)$. Observe that for every $j \in \mathbb{N}$, we have $\mathbf{a}_i^{(j\mathbf{d})} = (\mathbf{a}_i^{(\mathbf{d})})^j$. This means the $|I| \times |I|$ matrix A with entries $A_{ij} \coloneqq \mathbf{a}_i^{(j\mathbf{d})}$ is a Vandermonde matrix and thus – with the $\mathbf{a}_i^{(\mathbf{d})}$ being mutually distinct – invertible (see, for instance, [13]).

Let $\mathbf{p}_I = (p_i \mid i \in I)$ and $\mathbf{p}'_I = (p'_i \mid i \in I)$, and let $S \coloneqq \operatorname{supp}(\mathbf{a}_k)$. For every j, the jth entry of $\mathbf{p}_I \cdot A$ is

$$\begin{aligned}\sum_{i \in I} p_i \mathbf{a}_i^{(j\mathbf{d})} &= \sum_{i=1}^{m} p_i \mathbf{a}_i^{(j\mathbf{d})} - \sum_{\substack{i \in [m] \\ S \not\subseteq \operatorname{supp}(\mathbf{a}_i)}} p_i \mathbf{a}_i^{(j\mathbf{d})} - \sum_{\substack{i \in [m] \\ S \subsetneq \operatorname{supp}(\mathbf{a}_i)}} p_i \mathbf{a}_i^{(j\mathbf{d})} \\ &= \langle \mathbf{p}, \mathbf{a}^{(j\mathbf{d})} \rangle - \sum_{\substack{i \in [m] \\ S \subsetneq \operatorname{supp}(\mathbf{a}_i)}} p_i \mathbf{a}_i^{(j\mathbf{d})}.\end{aligned}$$

This follows since by (E4) we have $\mathbf{a}_i^{(j\mathbf{d})} = 0$ if $S = \operatorname{supp}(j\mathbf{d}) \not\subseteq \operatorname{supp}(\mathbf{a}_i)$. Similarly, the j-th entry of $\mathbf{p}'_I \cdot A$ is

$$\langle \mathbf{p}', \mathbf{a}^{(j\mathbf{d})} \rangle - \sum_{\substack{i \in [m] \\ S \subsetneq \operatorname{supp}(\mathbf{a}_i)}} p'_i \mathbf{a}_i^{(j\mathbf{d})}.$$

The maximality of S now implies that for all i with $S \subsetneq \operatorname{supp}(\mathbf{a}_i)$ we have $p_i = p'_i$. Consequently,

$$\sum_{\substack{i \in [m] \\ S \subsetneq \operatorname{supp}(\mathbf{a}_i)}} p_i \mathbf{a}_i^{(j\mathbf{d})} = \sum_{\substack{i \in [m] \\ S \subsetneq \operatorname{supp}(\mathbf{a}_i)}} p'_i \mathbf{a}_i^{(j\mathbf{d})}.$$

Finally, it follows with (E3) that

$$\sum_{i \in I} p_i \mathbf{a}_i^{(j\mathbf{d})} = \sum_{i \in I} p'_i \mathbf{a}_i^{(j\mathbf{d})}$$

for every j, so $\mathbf{p}_I \cdot A = \mathbf{p}'_I \cdot A$. Since A is invertible, it follows that $\mathbf{p}_I = \mathbf{p}'_I$. In particular, this implies $p_k = p'_k$. This is a contradiction, so it must hold that $\mathbf{p} = \mathbf{p}'$. ■

Recall that we defined $\mathbf{p} = (p_1, \dots, p_n)$ and $\mathbf{p}' = (p'_1, \dots, p'_n)$, where $p_i = |\mathcal{K}_i \cap V(G)|$ and $p_i = |\mathcal{K}_i \cap V(H)|$ for $i \in [n]$ and $\sim$-equivalence classes $\mathcal{K}_i$. This implies that there is a bijection $f \colon V(G) \to V(H)$, with $v \sim f(v)$ for all $v \in V(G)$, which completes the proof. □

We can now use the ideas from Example 3.18 to prove the following central lemma. The converse direction, that is, that Duplicator winning the game $\mathcal{G}_q(G, H)$ implies $\text{pp-hom}(\mathcal{D}_q, G) = \text{pp-hom}(\mathcal{D}_q, H)$ is fortunately more straightforward.

Lemma 3.22. *Let G, H be graphs of the same order. Then the following are equivalent.*

1. *Duplicator wins $\mathcal{G}_q(G, H)$.*
2. $\text{pp-hom}(\mathcal{D}_q, G) = \text{pp-hom}(\mathcal{D}_q, H)$

Proof. The proof is per induction on q.

$q = 1$: First assume Duplicator wins $\mathcal{G}_1(G, H)$. Then there exists a bijection f, such that for every $v \in V(G)$ the mapping $(v \mapsto f(v))$ is a local isomorphism between G and H, that is, $\gamma^G(v) = \gamma^H(f(v))$. This means for each colour c, G and H have the same number of vertices of colour c, which implies $\text{pp-hom}(\mathcal{D}_1, G) = \text{pp-hom}(\mathcal{D}_1, H)$. Conversely, if $\text{pp-hom}(\mathcal{D}_1, G) = \text{pp-hom}(\mathcal{D}_1)$ then G and H have the same number of vertices of colour c for each c. This in turn

means that there is a bijection mapping each vertex in G to a vertex in H of the same colour. Duplicator chooses this bijection in the first and only round of $\mathcal{G}_1(G, H)$ and wins.

$q \to q+1$: We first show (2) $\implies$ (1), so assume $\text{pp-hom}(\mathcal{D}_{q+1}, G) = \text{pp-hom}(\mathcal{D}_{q+1}, H)$. By Lemma 3.21, this means that there is a bijection f, such that $\text{pp-hom}(D, G; r^D \mapsto v) = \text{pp-hom}(D, H; r^D \mapsto f(v))$ for all $D \in \mathcal{D}_{q+1}$. Observe that this implies $\gamma^G(v) = \gamma^H(f(v))$. Duplicator now chooses this bijection. Say Spoiler chooses $v \in V(G)$. By Lemma 3.17 it suffices to show that Duplicator wins the game $\mathcal{G}_q(G \wr v, H \wr f(v))$. This follows from the induction hypothesis and the following claim.

Claim 1.

$$\text{pp-hom}(\mathcal{D}_q, G \wr v) = \text{pp-hom}(\mathcal{D}_q, H \wr f(v))$$

Proof. Let $D \in \mathcal{D}_q$ be arbitrary. We can assume without loss of generality that all vertices of D have a colour of the form (c, i) for $i \in \{0, 1\}$, since otherwise $\text{pp-hom}(D, G \wr v) = 0 = \text{pp-hom}(D, H \wr f(v))$.

Let $D = (F, T)$. We construct a graph F^+ by introducing a new vertex r^+ and letting $V(F^+) \coloneqq V(F) \cup \{r^+\}$, as well as

$$E(F^+) \coloneqq E(F) \cup \{(r^+, u) \mid u \in V(F), \gamma^F(u) = (\cdot, 1)\}^s.$$

We let $\gamma^{F^+}(r^+) \coloneqq \gamma^G(v) = \gamma^H(f(v))$ and $\gamma^{F^+}(u) = c$ if $\gamma^F(u) = (c, \cdot)$ for all $u \in V(F)$. We further construct a tree T^+ with $V(T^+) \coloneqq V(F^+)$ and

$$\preceq^{T^+} \coloneqq \preceq^T \cup \{(r^+, u) \mid u \in V(T)\}.$$

Observe that T^+ is an elimination tree for F^+ of height $q+1$, so $D^+ \coloneqq (F^+, T^+) \in \mathcal{D}_{q+1}$. Moreover, note that now there *really* is a one-to-one correspondence between the *past-preserving* homomorphisms from D^+ to G mapping r^+ to v and the past-preserving homomorphisms from D to $G \wr v$, as well as between the past-preserving homomorphisms from D^+ to G mapping r^+ to $f(v)$ and the past-preserving homomorphisms from D to $H \wr f(v)$. Lemma 3.21 furthermore implies

$$\text{pp-hom}(D^+, G; r^+ \mapsto v) = \text{pp-hom}(D^+, H; r^+ \mapsto f(v)).$$

All together, we obtain

$$\begin{aligned} \text{pp-hom}(D, G \wr v) &= \text{pp-hom}(D^+, G; r^+ \mapsto v) \\ &= \text{pp-hom}(D^+, H; r^+ \mapsto f(v)) \\ &= \text{pp-hom}(D, H \wr f(v)), \end{aligned}$$

which proves the claim. ∎

Let us now prove (1) $\implies$ (2). Assume that Duplicator has a winning strategy for $\mathcal{G}_{q+1}(G, H)$. By Lemma 3.17 there is a bijection f, such that for each $v \in V(G)$ we have $\gamma^G(v) = \gamma^H(f(v))$ and Duplicator wins $\mathcal{G}_q(G \wr v, H \wr f(v))$. Per induction hypothesis, this implies

$$\text{pp-hom}(\mathcal{D}_q, G \wr v) = \text{pp-hom}(\mathcal{D}_q, H \wr f(v)). \tag{E5}$$

Now let $D = (F, T) \in \mathcal{D}_{q+1}$. By removing r^D from D, we obtain a family $D_1, \dots, D_n \in \mathcal{D}_q$. We construct for each $i \in [n]$ a graph D_i' by recolouring the vertices of D_i. We let

$$\gamma^{D_i'}(u) = \begin{cases} (\gamma^{D_i}(u), 1) & \text{if } (r^D, u) \in E(D_i), \\ (\gamma^{D_i}(u), 0) & \text{if } (r^D, u) \notin E(D_i). \end{cases}$$

The crucial observation now is that for all $v \in V(G)$ with $\gamma^G(v) = \gamma^D(r^D)$, we have

$$\text{pp-hom}(D, G; r^D \mapsto v) = \prod_{i=1}^{n} \text{pp-hom}(D_i', G \wr v)$$

and analogously

$$\text{pp-hom}(D, H; r^D \mapsto f(v)) = \prod_{i=1}^{n} \text{pp-hom}(D'_i, H \wr f(v)).$$

By (E5) this implies

$$\text{pp-hom}(D, G; r^D \mapsto v) = \text{pp-hom}(D, H; r^D \mapsto f(v))$$

and thus

$$\begin{aligned}\text{pp-hom}(D, G) &= \sum_{v \in V(G)} \text{pp-hom}(D, G; r^D \mapsto v) \\ &= \sum_{v \in V(H)} \text{pp-hom}(D, H; r^D \mapsto f(v)) \\ &= \text{pp-hom}(D, H),\end{aligned}$$

which completes the proof. □

The main result of this section now follows from Lemma 3.20, Lemma 3.22, and the game-theoretic characterisations investigated in Section 3.2.

Theorem 3.23. *For any two graphs G, G', the following are equivalent.*

1. *G and G' satisfy the same C_q sentences.*
2. $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, G')$

Proof. Note that $|G| \neq |G'|$ implies that G and G' are neither homomorphism-indistinguishable over $\mathcal{TD}_q$ nor C_q-equivalent, for $q \geq 1$. This means we can assume without loss of generality that G and G' are of the same order. It follows that G and G' satisfy the same C_q sentences
$\iff$ Duplicator wins $\mathcal{G}_q(G, G')$
$\iff$ $\text{pp-hom}(\mathcal{D}_q, G) = \text{pp-hom}(\mathcal{D}_q, G')$
$\iff$ $\hom(\mathcal{TD}_q, G) = \hom(\mathcal{TD}_q, G')$. □

3.5 Tying the knot: A characterisation of C_q^k

> Hell is a place where nothing connects with nothing.
>
> — *T.S. Eliot*

Given the characterisations for C^{k+1} and C_q, a natural question to ask is if there is a class of graphs that characterises C_q^{k+1}. It seems intuitive to conjecture that C_q^{k+1}, the intersection of C^{k+1} and C_q, is characterised by the intersection of $\mathcal{TW}_k$ and $\mathcal{TD}_q$ – namely the class $\mathcal{T}_q^k$ of graphs of tree-width at most k and tree-depth at most q.

Conjecture 3.24. *For any two graphs G, G', the following are equivalent.*

1. *G and G' satisfy the same C_q^{k+1} sentences.*
2. $\hom(\mathcal{T}_q^k, G) = \hom(\mathcal{T}_q^k, G')$

Surprisingly, we will be able to prove a result that renders the conjecture very unlikely. Namely, we prove that C_q^{k+1}-equivalence is characterised by the class of graphs of *elimination-order* $(k+1, q)$. We then investigate the notion of elimination-order further and find that while the classes $\mathcal{T}_q^k$ and $\mathcal{E}_{k+1,q}$ coincide when restricting them to certain important graph classes, it is highly unlikely that the two notions are equivalent in general.

3.5.1 Elimination Order

Recall again the recursive construction of a graph G of tree-width at most k. We can partition the construction into *product steps* and (*label*) *elimination steps*. In each product step we take two graphs and compute their product, while in every elimination step we take one graph and remove one of its labels. The inductive definition of graphs of tree-width k now implies that there is a set A of graphs $\{G_1, G_2, \ldots, G_n\}$ with $V(G_i) = L_{G_i}$ and $|G_i| \leq k+1$ for $i \in [n]$, such that G can be constructed from A in a finite number of product- and elimination steps.

It seems natural to formalise such a construction as a tree with leaves in A, whose inner vertices can be partitioned into product nodes and elimination nodes.

Definition 3.25. Let G be a graph. A *k-construction tree* for G is a rooted tree (T, r) whose vertices are k-labelled graphs, and additionally it holds that

- $r = G$,
- for all leaves H we have $V(H) = L_H$,
- for all vertices $H \in V(T)$ with more than one child, H is the product of its children, and
- for all vertices $H \in V(T)$ with one child H', H can be obtained from H' by removing a single label.

It follows that G has a $(k+1)$-construction tree if and only if $\mathrm{tw}(G) \leq k$. Observe that since the product operation is associative, we can combine multiple consecutive product steps into one. This however means that construction trees that represent the same underlying construction can differ in height. To obtain a meaningful height measure, we only consider the elimination steps. We define the *elimination-depth* of a construction tree T as the maximum number of elimination steps on any path from the root to a leaf. We can now give a measure of how shallow a construction tree for G can be, depending on the number of labels we are allowed to use.

Definition 3.26 (Elimination-Order)**.** Let G be a graph. We say that G has *elimination-order* (k, q) if there exists a k-construction tree for G of elimination-depth at most q.

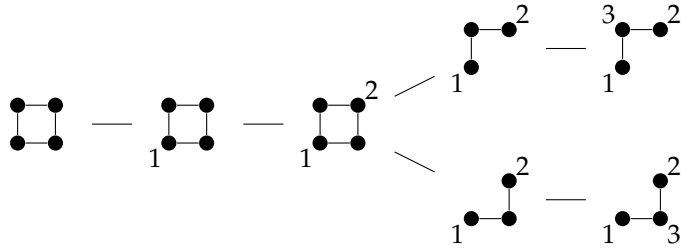


Figure 3.1: A 3-construction tree for the 2×2 grid graph.

To avoid ambiguity when examining construction trees, tree decompositions, and elimination trees in the same context, we from now on refer to the vertices of a construction tree as *nodes* instead of vertices.

Let us investigate how elimination-order fits into the landscape of graph parameters. The following two lemmas are straightforward consequences of the inductive definition of tree-width.

Lemma 3.27. *There is no graph G without labels of elimination-order $(\mathrm{tw}(G), \cdot)$.*

Proof. Assume otherwise and let $k := \mathrm{tw}(G)$. Per definition, there is a construction tree for G using only k labels. But this implies by the recursive definition of tree-width that $\mathrm{tw}(G) \leq k - 1 < \mathrm{tw}(G)$. □

Lemma 3.28. *There is no graph G without labels of elimination-order $(\mathrm{tw}(G)+1, \mathrm{tw}(G))$.*

Proof. Assume there was a graph G of elimination-order $(\mathrm{tw}(G)+1, \mathrm{tw}(G))$. Let $k \coloneqq \mathrm{tw}(G)$ and let T be a $(k+1)$-construction tree for G of elimination-depth at most k. Then T must contain a path p from the root to one of the leaves on which every label $l \in [k+1]$ occurs in some node H on p. Otherwise, we could restrict ourselves to a strictly smaller set of labels without changing the recursive construction of G, and thus obtain a k-construction tree – contradicting Lemma 3.27. Since G contains no labels, we need to remove each of these $k+1$ labels on p at least once, implying that T has elimination-depth at least $k+1$. This is a contradiction. □

Using a clever construction, we can also relate elimination-order to tree-depth. The crucial observation is that in the recursive construction of a graph G, once we remove a label from a vertex v, we can no longer add any neighbours to v. This implies that any statement about the adjacency of v made for this intermediate graph G' also holds for G. We can use this fact to build an elimination forest for G from a construction tree.

Lemma 3.29. *For all graphs G without labels of elimination-order (k, q), it holds that $\mathrm{td}(G) \leq q$.*

Proof. Let us fix a k-construction tree T for G of elimination-depth at most q and let $\varepsilon(T) \subseteq V(T)$ denote the set of nodes that represent elimination steps. For a graph $H \in \varepsilon(T)$, we denote by $\nu(H)$ the vertex v whose label was removed in the corresponding elimination step.

Let $H \in V(T)$. Observe that each vertex $v \in V(H)$ corresponds to a vertex in $V(G)$, since we only combine graphs and never remove vertices. Sometimes, however, multiple vertices can correspond to the same vertex in $V(G)$, considering we identify vertices with the same labels during product steps. In the following we will tacitly identify vertices $v \in V(H)$ with their corresponding vertex in $V(G)$ and vice versa. Note that by that assumption we have $\nu(F) \in V(G)$ for all $F \in \varepsilon(T)$.

We now construct a rooted forest S with vertex set $V(S) = \{\nu(H) \mid H \in \varepsilon(T)\}$ by letting $\nu(H) \preceq^S \nu(H') \iff H \preceq^T H'$. Then S is an elimination forest for G:

Suppose $v \in V(G)$. Since v has no labels in G and all vertices start labelled, there must be an elimination step in T that removes the label from v. It follows that $v \in V(S)$ and thus $V(S) = V(G)$.

Now let $(u, v) \in E(G)$. Assume for contradiction that neither $u \preceq^S v$ nor $v \preceq^S u$. This implies that u and v can never be part of the same leaf H, otherwise the labels of u and v must necessarily be removed on the unique path from H to the root, implying $u \preceq^S v$ or $v \preceq^S u$, depending on which label is removed first. But since product steps cannot create edges that are not present in at least one of the leaves, u and v cannot be adjacent in G. Contradiction.

Finally, since $\preceq^S$ is essentially the restriction of $\preceq^T$ to elimination steps, the size of the longest $\preceq^S$-chain is equal to the longest $\preceq^T$-chain containing only elimination steps. It follows that the height of S is at most q, and thus $\mathrm{td}(G) \leq q$. □

In the following, let us denote the class of all graphs of elimination-order (k, q) by $\mathcal{E}_{k,q}$.

3.5.2 Proving the result

Now that we have a grasp on what elimination-order is and how it relates to tree-depth and tree-width, let us investigate how the class of graphs of elimination-order (k, q) characterises C_q^k-equivalence. The result follows from slightly stronger versions of Lemma 3.9 and Lemma 3.15, respectively. We proceed analogously to Section 3.3 and first show that C_q^k is strong enough to capture homomorphism counts over $\mathcal{E}_{k,q}$.

Lemma 3.30. *Let $G \in \mathcal{E}_{k,q}$ be a k-labelled graph, and let $m \in \mathbb{N}$. Then there exists a formula $\varphi_m \in \mathcal{C}_q^k$ such that for each k-labelled graph H with $L_G \subseteq L_H$, $H \models \varphi_m$ if and only if $\mathrm{hom}(G, H) = m$.*

Proof. Since G has elimination-order (k, q), there is a k-construction tree T for G of elimination-depth at most q. We construct φ_m per induction on the structure of T. Without loss of generality, we assume that T is a binary tree.

Let $F \in V(T)$ be a leaf of T. We proceed similar to Lemma 3.9. We let φ_1^F be the conjunction of formulae $x_i = x_j$ for $\nu_F(i) = \nu_F(j)$ and Ex_ix_j for $(\nu_F(i), \nu_F(j)) \in E(G)$. We then let $\varphi_0^F = \neg\varphi_1^F$ and $\varphi_m^F = \text{false}$ for $m > 1$. Note that φ_m^F uses at most k distinct variables and has quantifier-rank 0, so $\varphi_m^F \in \mathcal{C}_q^k$.

Now let $F \in V(T)$ be a node with two children F_1, F_2. Since T is a construction tree, we have $F = F_1F_2$. Per induction hypothesis, there are formulae $\varphi_m^1, \varphi_m^2 \in \mathcal{C}_q^k$, such that $\hom(F_i, H) = m$ if and only if $H \models \varphi_m^i$ for appropriately labelled graphs H and $i \in \{1, 2\}$. If $m \geq 1$ we let φ_m be the disjunction of formulae $\varphi_{m_1}^1 \wedge \varphi_{m_2}^2$ for all m_1, m_2 with $m = m_1m_2$. For $m = 0$ we let $\varphi_m = \varphi_0^1 \vee \varphi_0^2$. This boolean combination does not alter the quantifier rank, so φ_m still has quantifier-rank at most q. Moreover, φ_m^1 and and φ_m^2 both have variables among $x_1, \ldots, x_k$, so $\varphi_m \in \mathcal{C}_q^k$.

Finally, suppose $F \in V(T)$ has only one child F'. This implies that we can obtain F from F' by removing a label. Per induction hypothesis, there is a formula $\varphi_m \in \mathcal{C}_q^k$ that captures that there are m homomorphisms from F' to an appropriately labelled graph H. Recall that we can use Lemma 3.8 to define φ_m as the disjunction over all decompositions $m = \sum_{i=1}^t c_i m_i$, for $c_i, m_i \in \mathbb{N}^+$, of formulae

$$\psi_{c_1,m_1,\ldots,c_t,m_t} = \exists^{=c} x_l \, \neg\varphi_0' \wedge \bigwedge_{i\in[t]} \exists^{=c_i} x_l \, \varphi_{m_i}'.$$

If $F = G$ is the root of T, we let $\varphi_m = \varphi_m^F$. Observe that each elimination step increases the quantifier-rank of φ_m by one. Since T has elimination-depth at most q, it holds that $\varphi_m \in \mathcal{C}_q^k$. □

Similarly to Lemma 3.9, the proof implies the following corollary for graphs without labels.

Corollary 3.31. *Let $G \in \mathcal{E}_{k,q}$ with $L_G = \varnothing$, and let $m \in \mathbb{N}$. Then there exists a sentence $\varphi_m \in \mathcal{C}_q^k$, such that for each graph H it holds that $H \models \varphi_m$ if and only if $\hom(G, H) = m$.*

To prove the converse direction, that is, to show that homomorphism-indistinguishability over $\mathcal{E}_{k,q}$ implies C_q^k-equivalence, we once more make use of quantum graphs. Recall that a quantum graph $|G\rangle$ *models* a formula φ for graphs of size n, if for all graphs H with $|H| = n$ it holds that $\hom(|G\rangle, H) = 1$ whenever $H \models \varphi$, and $\hom(|G\rangle, H) = 0$ otherwise. Furthermore, recall that for a quantum graph $|G\rangle$, we denote by $|G[S_-; S_+]\rangle$ a quantum graph such that $\hom(|G[S_+; S_-]\rangle, H) = 1$ if $\hom(|G\rangle, H) \in S_+$ and $\hom(|G[S_+; S_-]\rangle) = 0$ if $\hom(|G\rangle, H) \in S_-$ for any graph H and disjoint sets of natural numbers S_-, S_+. Moreover, if $|G\rangle$ was a quantum graph over a class of graph $\mathcal{F}$ that is closed under the product operation, then $|G[S_+; S_-]\rangle$ is also a quantum graph over $\mathcal{F}$.

Lemma 3.32. *The class $\mathcal{E}_{k,q}$ is closed under taking products.*

Proof. Let $G_1, G_2 \in \mathcal{E}_{k,q}$. Then there exist k-construction trees (T_{G_1}, r_1) and (T_{G_2}, r_2) of elimination-depth at most q and with leaves in $\mathcal{TW}_k$. Consequently, the construction tree S, obtained by adding a new root r adjacent to r_1 and r_2 with $r = r_1r_2 = G_1G_2$, is a k-construction tree for G_1G_2. Moreover, the elimination-depth of S is equal to the maximum elimination-depth of T_{G_1} and T_{G_2}, so in particular at most q. It follows that $G_1G_2 \in \mathcal{E}_{k,q}$. □

This result implies that we can prove the following lemma along the same lines as Lemma 3.15.

Lemma 3.33. *Let $k \geq 1$ and let φ be a C_q^k formula. Then for every $n \in \mathbb{N}^+$ there exists a quantum graph $|G\rangle$ over $\mathcal{E}_{k,q}$ modelling φ for graphs of size n.*

Proof. The proof is per induction over the structure of φ. If $\varphi = [x_i = x_j]$, we let $|G\rangle = G$ be the graph consisting of a single vertex v with $\nu_G(i) = \nu_G(j) = v$. If $\varphi = Ex_ix_j$, we let $|G\rangle = G$ be the graph consisting of two adjacent vertices v_1, v_2 with $\nu_G(i) = v_1$ and $\nu_G(j) = v_2$, unless $i = j$, in which case we let $|G\rangle = 0$. In all these cases $\mathrm{qr}(\varphi) = 0$, and since G is always fully labelled, it holds that G has elimination-order $(k, 0)$. Consequently, $|G\rangle$ is a quantum graph over $\mathcal{E}_{k,0}$.

If $\varphi = \neg\psi$, then there exists per induction hypothesis a quantum graph G_ψ over $\mathcal{E}_{k,q}$ modelling ψ for graphs of size n. We let $|G\rangle = |G_\psi[1;0]\rangle$, which is a quantum graph over $\mathcal{E}_{k,q}$ by Lemma 3.32. The case $\varphi = \psi \vee \vartheta$ we handle similarly.

Finally, consider the case $\varphi = \exists^{\geq t} x_l \psi$. Let $|G_\psi\rangle = \sum_{i=1}^{t} a_i G_{\psi,i}$ be a quantum graph modelling ψ for graphs of size n. Since $\mathrm{qr}(\psi) = \mathrm{qr}(\varphi) - 1$, we may assume that $|G_\psi\rangle$ is a quantum graph over $\mathcal{E}_{k,q-1}$. We let $|G'_\psi\rangle$ be the graph obtained from $|G_\psi\rangle$ by removing the label l from all $G_{\psi,i}$. Then $|G\rangle = |G'_\psi[1, \ldots, t-1; t, \ldots, n]\rangle$ models φ for graphs of size n. Moreover, it is easy to see that the $G'_{\psi,i}$, obtained by removing a label from $G_{\psi,i}$, have elimination-order (k, q). Consequently, $|G\rangle$ is a quantum graph over $\mathcal{E}_{k,q}$. □

The following theorem can now be proved completely analogously to Theorem 3.16.

Theorem 3.34. *For any two graphs G, G', the following are equivalent.*

1. *G and G' satisfy the same C_q^k sentences.*
2. $\hom(\mathcal{E}_{k,q}, G) = \hom(\mathcal{E}_{k,q}, G')$

3.5.3 Salvaging the conjecture

If we wanted to somehow save our initial conjecture, we could hope that $\mathcal{E}_{k+1,q}$ turns out to coincide with $\mathcal{T}_q^k$, the class of all graphs of tree-width at most k and tree-depth at most q.

Conjecture. $\mathcal{E}_{k+1,q} = \mathcal{T}_q^k$.

The observation that elimination-order provides us with an upper bound for tree-depth certainly lets us hope. In fact, it is not hard to verify that $\mathcal{E}_{k+1,q} \subseteq \mathcal{T}_q^k$. Moreover, in certain extremal cases our previously established bounds already imply the conjecture.

Lemma 3.35. *For all complete graphs K_n it holds that $K_n \in \mathcal{E}_{k+1,q} \iff K_n \in \mathcal{T}_q^k$.*

Proof. The forward direction already follows from the fact that $\mathcal{E}_{k+1,q} \subseteq \mathcal{T}_q^k$. Let $K_n \in \mathcal{T}_q^k$. It is a well-known result that the complete graph of n vertices K_n has tree-width $n-1$. Proposition 2.3 then implies that $\mathrm{td}(K_n) \geq \mathrm{tw}(K_n) + 1 = n$. Consequently, $k \geq n-1$ and $q \geq n$. A valid $(k+1)$-construction tree is the path starting with a fully labelled graph K_n, followed by n elimination steps. This implies $K_n \in \mathcal{E}_{k+1,q}$. □

Lemma 3.36. *For all complete bipartite graphs $K_{m,n}$ it holds that $K_{m,n} \in \mathcal{E}_{k+1,q} \iff K_{m,n} \in \mathcal{T}_q^k$.*

Proof. We again only have to show the backward direction, so assume $K_{m,n} \in \mathcal{T}_q^k$. It is known that $\mathrm{tw}(K_{m,n}) = \min(m, n)$ (this follows for instance from the characterisation of tree-width in terms of chordal graphs, see [3]), and thus by Proposition 2.3 $\mathrm{td}(K_{m,n}) \geq \min(m, n) + 1$. The latter fact can also be seen by noting that each leaf in an elimination forest of $K_{m,n}$ has to have at least $\min(m, n)$ ancestors. It follows that $k \geq \min(m, n) =: \ell$ and $q \geq \ell + 1$.

It thus suffices to give an $(\ell+1)$-construction tree of height at most $\ell + 1$. We can construct such a tree as follows. Let us refer to the smaller set of the partition as A and to the larger set as B. It is $A, B \subseteq V(K_{m,n})$ and $|A| = \ell$. We now take fully labelled subgraphs $K_{1,\ell}$ of $K_{m,n}$, induced by $A \cup \{v\}$ for all $v \in B$. For each of these graphs, we then remove the label from v and subsequently build their product. It remains to remove the labels of the vertices in A, which results in ℓ additional elimination steps. Since $|K_{1,\ell}| = \ell + 1$, we only have to use $\ell + 1$ different labels. The construction is thus a valid $(\ell+1)$-construction tree of height $\ell + 1$, so $K_{m,n} \in \mathcal{E}_{k+1,q}$. □

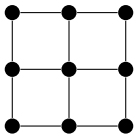


Figure 3.2: The graph M_2.

There is, however, little hope of saving the conjecture in general. Consider the class of *Mersenne grids* M_n, for $n \in \mathbb{N}$, which we define as the $(2^n - 1) \times (2^n - 1)$ grid graphs. We can quite conveniently construct an elimination tree for M_n by repeatedly cutting the grid in half (see Figure 3.3). This leads to an elimination tree of height

$$\sum_{i=0}^{n-1} \left\lfloor \frac{2^n - 1}{2^i} \right\rfloor + \left\lfloor \frac{2^n - 2}{2^{i+1}} \right\rfloor .$$

Consequently, $\mathrm{td}(M_n) \in \mathcal{O}(n \cdot 2^n)$. Meanwhile, M_n is a grid graph and thus has tree-width $2^n - 1$. The canonical tree decomposition that achieves this width (illustrated in Figure 3.4) is a path of length $t := (2^n - 1)^2 - (2^n - 1)$. By choosing the middle bag as the root, we can obtain a tree of height $t/2$, and by handling edge cases smartly, we can reduce this height somewhat. In general, however, it appears very unlikely that we can do better than $\Theta(2^{2n})$ in terms of height.

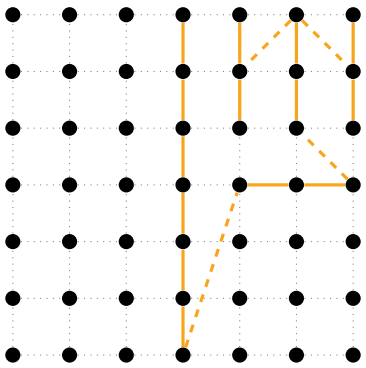


Figure 3.3: Construction of an elimination tree (orange lines) for M_3.

Now let G be some graph. Observe that we can build from each $(k+1)$-construction tree T for G a tree decomposition of G of width k, whose height is at most the elimination-depth of T. Now if there were a $(\mathrm{tw}(M_n) + 1)$-construction tree for M_n with elimination-depth not in $\Omega(2^{2n})$, then there would also be a tree decomposition of optimal width for M_n with height not in $\Omega(2^{2n})$. By our previous observations, this is very unlikely. Thus, it is reasonable to assume that all $(\mathrm{tw}(M_n) + 1)$-construction trees for M_n have elimination-depth in $\Omega(2^{2n})$. In particular, this means that if M_n has elimination-order $(\mathrm{tw}(M_n) + 1, q)$, then $q \in \Omega(2^{2n})$ and thus $q \in \omega(\mathrm{td}(M_n))$.

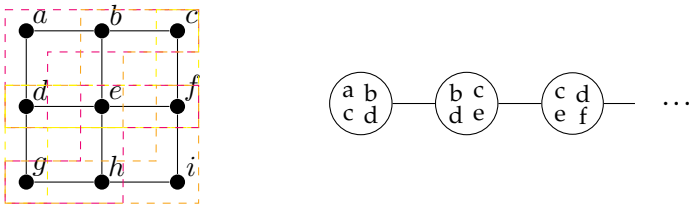


Figure 3.4: A tree decomposition of M_2.

We thus feel comfortable to pose the following conjecture.

Conjecture 3.37. *In general, it holds that* $\mathcal{E}_{k+1,q} \neq \mathcal{T}_q^k$.

4 Complexity

In this final chapter, we want to shed some light on these results in terms of their complexity. For a class of graphs $\mathcal{F}$, let $\textsc{HomInd}(\mathcal{F})$ be the problem of deciding whether two graphs G and H are homomorphism-indistinguishable over $\mathcal{F}$, that is, whether $\hom(\mathcal{F}, G) = \hom(\mathcal{F}, H)$. The connection between different natural equivalence relations and homomorphism-indistinguishability means that $\textsc{HomInd}$ behaves quite interestingly from a complexity-theoretic standpoint. Böker, Chen, Grohe and Rattan extensively investigated the problem in [5] and found classes $\mathcal{F}$ for which $\textsc{HomInd}(\mathcal{F})$ ranges from polynomial-time decidable to undecidable.

4.1 Completing the picture

The classes $\mathcal{TW}_k$, $\mathcal{TD}_q$ and $\mathcal{E}_{k,q}$ characterising equivalence in the different fragments of C turn out to be at the lower end of this spectrum. Let us first consider $\mathcal{TW}_k$. As we have seen, two graphs G and H are homomorphism-indistinguishable over $\mathcal{TW}_k$ if and only if they are C^{k+1}-equivalent. As also mentioned in Section 3.3, C^{k+1}-equivalence corresponds precisely to equivalence with respect to the k-dimensional Weisfeiler-Leman algorithm [6].

The one-dimensional Weisfeiler-Leman algorithm, also known as *colour refinement*, computes a colouring of the vertices of the input graph that is invariant under isomorphism. The k-dimensional Weisfeiler-Leman algorithm instead colours k-tuples of vertices. The algorithm is often used for graph-isomorphism testing by comparing the computed colourings of two graphs. If they are different, the graphs cannot be isomorphic. In that case, we say the algorithm *distinguishes* the graphs.

Lemma 4.1. *Let G, H be graphs. Then the following are equivalent.*

1. $\hom(\mathcal{TW}_k, G) = \hom(\mathcal{TW}_k, H)$
2. *G and H are not distinguished by the k-dimensional Weisfeiler-Leman algorithm.*

A direct proof of this result can also be found in [8]. Crucially, the algorithm runs in polynomial time in the size of the input graph. The lemma thus immediately implies the following result.

Theorem 4.2. $\textsc{HomInd}(\mathcal{TW}_k)$ *is in* P.

The characterisation of C_q-equivalence in terms of graphs of tree-depth at most q gives us a similar result for $\textsc{HomInd}(\mathcal{TD}_q)$. Namely, we can compute the winning regions for duplicator in the q-round bijective pebble game in polynomial time. We thus obtain the following theorem, first noticed by Grohe [10].

Theorem 4.3. $\textsc{HomInd}(\mathcal{TD}_q)$ *is in* P.

Similarly, we can use the q-round bijective k-pebble game to decide in polynomial time whether two graphs are C^k_q-equivalent. Let us further concretise this. Let G, H be graphs and $k, q \geq 1$.

Proposition 4.4. *We can inductively compute, in polynomial time, an equivalence relation $\sim_i$ on $(V(G) \cup \{\bot\})^k \cup (V(H) \cup \{\bot\})^k$, such that Duplicator wins the i-round game $\mathcal{G}^k_i(G, \mathbf{v}^\bot; H, \mathbf{w}^\bot)$ if and only if $\mathbf{v}^\bot \sim_i \mathbf{w}^\bot$.*

Proof. Let us define the equivalence relation $\sim_0$ on $(V(G) \cup \{\bot\})^k \cup (V(H) \cup \{\bot\})^k$ by letting $\mathbf{x}^\bot \sim_0 \mathbf{y}^\bot$ if $x_i^\bot \neq \bot \iff y_i^\bot \neq \bot$ and $(x_i^\bot \mapsto y_i^\bot \mid i \in [k])$ is a local isomorphism between X and Y, where $X, Y \in \{G, H\}$, $\mathbf{x}^\bot \in (V(X) \cup \{\bot\})^k$ and $\mathbf{y}^\bot \in (V(Y) \cup \{\bot\})^k$.

Note that there are only $O(n^k)$ many such tuples, and for each pair of tuples we can check if they induce a local isomorphism in time $O(k!) = O(1)$. Consequently, we can compute the $\sim_0$-equivalence classes in polynomial time.

We now obtain $\sim_{i+1}$ from $\sim_i$ as follows. Let $\mathbf{x}^\bot, \mathbf{y}^\bot, X, Y$ as above. It is $\mathbf{x}^\bot \sim_{i+1} \mathbf{y}^\bot$ if there is a bijection f such that for all indices $j \in [k]$ and vertices $v \in V(G)$, it is $\mathbf{x}^\bot(j \to v) \sim_i \mathbf{y}^\bot(j \to f(v))$. We can verify this in polynomial time:

For every $i \in [m]$ let $\mathcal{K}_i$ denote the equivalence classes of $\sim_i$. For every $j \in [k]$, let us denote by $\mathcal{S}_j^{\mathbf{x}}$ the set of tuples $\mathbf{x}'$ for which there exists an $x \in X$ such that $\mathbf{x}^\bot(j \to x) = \mathbf{x}'$. To check if $\mathbf{x}^\bot \sim_{i+1} \mathbf{y}^\bot$ we only have to verify by counting that

$$|\mathcal{K}_i \cap \mathcal{S}_j^{\mathbf{x}} \cap V(X)^k| = |\mathcal{K}_i \cap \mathcal{S}_j^{\mathbf{y}} \cap V(Y)^k|$$

for all $j \in [k]$ and $i \in [m]$. □

The characterisation of C_q^k-equivalence in terms of homomorphism-indistinguishability over $\mathcal{E}_{k,q}$ now implies the following final theorem.

Theorem 4.5. $\textsc{HomInd}(\mathcal{E}_{k,q})$ *is in* P.

5 Conclusion

> In case I don't see ya, good afternoon, good evening, and good night!
>
> *Jim Carrey, in: The Truman Show*

In this thesis, we extensively explored the idea of characterising the equivalence of graphs with respect to certain fragments of first-order logic with counting C by comparing homomorphism counts. In particular, we investigated how one might characterise equivalence in the k-variable, quantifier-rank-q fragment in terms of homomorphism-indistinguishability. We found that two graphs are C_q^k-equivalent if and only if there are homomorphism-indistinguishable over the class $\mathcal{E}_{k,q}$ of all graphs of elimination-order (k, q). This closes a gap between two previous results, that are due to Dvořák and Grohe, respectively.

We introduced elimination-order as a new structural graph parameter that is closely related to tree-width. Among other properties, we showed that elimination-order provides us with a tight upper bound for tree-depth. We stopped, however, short of proving Conjecture 3.37 that $\mathcal{E}_{k,q}$ is not equivalent to the class $\mathcal{T}_q^{k-1}$ of graphs of tree-width at most $k-1$ and tree-depth at most q, since a proof would go beyond the scope of this thesis.

Finally, we looked at homomorphism-indistinguishability from a complexity-theoretic point of view. Namely, we investigated how efficiently we can decide whether two graphs are homomorphism-indistinguishable over a fixed class of graphs $\mathcal{F}$. We found that for $\mathcal{TW}_k$, $\mathcal{TD}_q$, and $\mathcal{E}_{k,q}$, the problem is decidable in polynomial time.

5.1 Future Work

Apart from proving Conjecture 3.37, there are a few directions we touched on in this thesis that could be interesting to further look into. One of them pertains to the idea of finding smaller, or somehow more tractable graph classes that are sufficient to capture isomorphism than the class of all graphs. Finding such a class $\mathcal{F}$ for which $\textsc{HomInd}(\mathcal{F})$ is in P would, for instance, settle the open question of whether the graph isomorphism problem is solvable in polynomial time. Results in this domain include Dvořák's theorem that the class of 2-degenerate graphs is sufficient to distinguish any two non-isomorphic graphs [9]. Böker [4] proved that the class of all directed acyclic graphs is also sufficient to capture isomorphism. Since the logic C is strong enough to capture finite structures up to isomorphism, it might pay in this regard to investigate whether there is a natural class characterising C-equivalence. The naïve approach – noting that two graphs are C-equivalent if and only if they are C_q-equivalent for all $q \in \mathbb{N}$ – obviously just yields the class of all graphs, but perhaps there is a smarter approach.

Another question one might ask is what the limitations of characterisations by homomorphism-indistinguishability are. Do there exist equivalence relations that cannot be captured in terms of homomorphism-indistinguishability? An initial analysis of this question can be found in [1].

Finally, it might be interesting to take a look at other logics. Are there more logics that are captured by homomorphism counts over well-studied graph classes? Which properties of a logic can we infer from its characterisation in terms of homomorphism-indistinguishability? Or regarding the previous point, are there logics that cannot be captured by homomorphism counts at all?

In general, the concept of capturing structural information by homomorphism counts promises a broad theory connecting many different fields that definitely warrants further exploration.

Bibliography

[1] Albert Atserias, Phokion G Kolaitis, and Wei-Lin Wu. On the expressive power of homomorphism counts. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2021.

[2] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In *International Colloquium on Automata, Languages, and Programming*, pages 155–166. Springer, 2015.

[3] Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical computer science*, 209(1-2):1–45, 1998.

[4] Jan Böker. Color refinement, homomorphisms, and hypergraphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 338–350. Springer, 2019.

[5] Jan Böker, Yijia Chen, Martin Grohe, and Gaurav Rattan. The complexity of homomorphism indistinguishability. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[6] 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, 1992.

[7] Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoretical Computer Science*, 329(1-3):315–323, 2004.

[8] 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. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

[9] Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010.

[10] Martin Grohe. Counting bounded tree depth homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 507–520, 2020.

[11] Lauri Hella. Logical hierarchies in ptime. *Information and Computation*, 129(1):1–19, 1996.

[12] Neil Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25(1):76–98, 1982.

[13] Allen Klinger. The vandermonde matrix. *The American Mathematical Monthly*, 74(5):571–574, 1967.

[14] László Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(3-4):321–328, 1967.

[15] László Lovász and Balázs Szegedy. Contractors and connectors of graph algebras. *Journal of Graph Theory*, 60(1):11–30, 2009.

[16] 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.

[17] Jaroslav Nešetřil and Patrice Ossona de Mendez. Linear time low tree-width partitions and algorithmic consequences. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of Computing*, pages 391–400, 2006.

[18] Wied Pakusa, Benedikt Pago, Martin Grohe, and Erich Grädel. A finite-model-theoretic view on propositional proof complexity. *Logical Methods in Computer Science*, 15, 2019.

[19] Motakuri V. Ramana, Edward R. Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1):247–265, 1994.

[20] Seinosuke Toda. Pp is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.