Title

A Study of the Expressive Power of Homomorphism Counts

Author
Wu, Wei-Lin

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

A STUDY OF THE EXPRESSIVE POWER OF HOMOMORPHISM COUNTS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Wei-Lin Wu

December 2023

The Dissertation of Wei-Lin Wu
is approved:

Professor Phokion G. Kolaitis, Chair

Professor Seshadhri Comandur

Professor Albert Atserias

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

A Study of the Expressive Power of Homomorphism Counts

by

Wei-Lin Wu

The Lovász Theorem asserts that two graphs $\mathbf{G}$ and $\mathbf{H}$ are isomorphic if and only if $\hom(\mathbf{F},\mathbf{G}) = \hom(\mathbf{F},\mathbf{H})$ for all graphs $\mathbf{F}$, here $\hom(\mathbf{A},\mathbf{B})$ denotes the number of homomorphisms from $\mathbf{A}$ to $\mathbf{B}$. This characterization of graph isomorphism in terms of graph homomorphism counts motivated a wealth of research that seeks to characterize different *relaxations of isomorphism* – equivalence relations that are coarser than isomorphism – in terms of the numbers of homomorphisms "from" certain graphs $\mathbf{F}$. Symmetrically, the Chaudhuri-Vardi Theorem says that two graphs $\mathbf{G}$ and $\mathbf{H}$ are isomorphic if and only if $\hom(\mathbf{G},\mathbf{F}) = \hom(\mathbf{H},\mathbf{F})$ for all graphs $\mathbf{F}$. While this dual characterization prompts to characterize relaxations of isomorphism in terms of the numbers of homomorphisms "to" certain graphs $\mathbf{F}$, it received relatively little attention, and is investigated in depth in this dissertation. The notions of isomorphism and homomorphism as well as these theorems also apply to relational structures. A different view of these theorems is that they give rise to *query algorithms* for testing membership in a class (in the case here the isomorphism type of a fixed graph or relational structure) that answer "yes" or "no" by making a bounded number of *homomorphism-count* queries. A variant of such an algorithm makes *homomorphism-existence* queries (whose values are 0 or 1) instead. An analysis is conducted in this dissertation for various classes regarding certain variants of query algorithms.

To my dear family and friends

Acknowledgments

First and foremost I want to express my gratitude to my advisor, Professor Phokion G. Kolaitis for his encouragement, patience and support during the course of my Ph.D. degree and for introducing me to the beautiful field of homomorphisms. This dissertation would have been impossible without his guidance. My gratitude extends to Professors Seshadhri Comandur and Albert Atserias for serving on the committee of my dissertation and providing helpful suggestions for revisions and future work. I also want to thank the coauthors of my research papers, Professors Albert Atserias, Balder ten Cate and Víctor Dalmau, in addition to my advisor, from whom I learned so much in our collaboration.

A special thanks goes to Tim Seppelt, who brought to my attention an excellent paper on query algorithms that motivated me to embark on an investigation of this topic, and to Chih-Duo Hong, a long-time friend who read an early manuscript of this dissertation and gave me valuable feedback.

Next, I would like to thank my lab members, Vishal Chakraborty, Zehui Cheng and Akhil Dixit, as well as a friend, Andreas Gittis, for offering their help during my study at UC Santa Cruz. I would also like to thank my housemates in an on-campus apartment, Shishir Kapoor and Taisuke Kameda, with whom I have had a good time.

I am deeply indebted to my dear friends, Gloria Huang, Christine Pan and Ava Tang, who always give me emotional support during my hard times, and especially to Jeffrey Chang, who has always been there for me.

Last, but not least, I am very grateful to my dear family for encouraging me and giving me strength all along.

Chapter 1

Introduction

The decision problem `GRAPH-ISOMORPHISM` is the following: Given two graphs $\mathbf{G}$ and $\mathbf{H}$, is there an isomorphism from $\mathbf{G}$ to $\mathbf{H}$? A bijective mapping π from the vertices of $\mathbf{G}$ to the vertices of $\mathbf{H}$ is an *isomorphism from* $\mathbf{G}$ *to* $\mathbf{H}$ if for all vertices u, v in $\mathbf{G}$, they are adjacent in $\mathbf{G}$ if and only if their images $\pi(u), \pi(v)$ are adjacent in $\mathbf{H}$, and we say $\mathbf{G}$ is *isomorphic* to $\mathbf{H}$ if such a mapping π exists (obviously, being isomorphic is an equivalence relation over graphs).

This problem has found applications in various areas such as computer vision, chemoinformatics and circuit design [32]. Despite important progress [4, 31], the exact status of its computational complexity remains unknown today: While this problem obviously belongs to NP, the class of all decision problems that admit a nondeterministic polynomial-time algorithm, it is an important open question whether `GRAPH-ISOMORPHISM` is NP-complete or admits a polynomial-time algorithm in general.

A wider decision problem is `GRAPH-HOMOMORPHISM`: Given two graphs $\mathbf{G}$ and $\mathbf{H}$, is there a homomorphism from $\mathbf{G}$ to $\mathbf{H}$? A mapping (not necessarily bijective) from the vertices of $\mathbf{G}$ to those of $\mathbf{H}$ is a *homomorphism from* $\mathbf{G}$ *to* $\mathbf{H}$ if for all vertices u, v in $\mathbf{G}$, the adjacency between them in $\mathbf{G}$ implies the adjacency between their images $h(u), h(v)$ in $\mathbf{H}$. It is obvious that all instances of `GRAPH-HOMOMORPHISM` are in NP.

There are two special cases of `GRAPH-HOMOMORPHISM` that are common decision problems themselves, derived by fixing either $\mathbf{G}$ or $\mathbf{H}$. For example, when $\mathbf{G}$ is fixed to the triangle (a complete graph of three vertices), then the problem amounts to asking, given a graph $\mathbf{H}$ as input, whether it contains a triangle as a subgraph (since graphs are loopless). Symmetrically, when $\mathbf{H}$ is fixed to the triangle, then the problem is the same as the `3-COLORABILITY` problem, well known to be NP-complete. The latter example indicates that `GRAPH-HOMOMORPHISM`, in general, is NP-complete.

Clearly, an isomorphism from a graph to another is also a homomorphism, but the converse may not be true. There is a deeper connection between the two notions, indeed. In the following, we write $\hom(\mathbf{F}, \mathbf{G})$ for the number of homomorphisms from $\mathbf{F}$ to $\mathbf{G}$ for arbitrary graphs $\mathbf{F}$ and $\mathbf{G}$. It turns out that *two graphs* $\mathbf{G}$ *and* $\mathbf{H}$ *are isomorphic if and only if* $\hom(\mathbf{F}, \mathbf{G}) = \hom(\mathbf{F}, \mathbf{H})$ *for all graphs* $\mathbf{F}$. Fix a graph $\mathbf{G}$, the homomorphism counts $\hom(\mathbf{F}, \mathbf{G})$ for all graphs $\mathbf{F}$ collectively are called the (*full*) *left profile of* $\mathbf{G}$. Stated otherwise, the left profile of $\mathbf{G}$ reveals its complete structural information (i.e., isomorphism type). This assertion is known as Lovász Theorem [40] (see Theorem 4.1). Clearly, by restricting the graphs $\mathbf{F}$ in the left profiles of graphs to be drawn from a certain nonempty class, a relaxation of isomorphism (i.e., an equivalence relation coarser than isomorphism) arises from pairs of graphs $\mathbf{G}$ and $\mathbf{H}$ that share the same restricted left profile. This consideration has inspired an extensive body of research investigating what interesting equivalence relations over graphs are captured in this way.

For example, two graphs are said to be C^k-*equivalent*, where $k \geq 1$, if they satisfy the same C^k-sentences, i.e., first-order sentences with counting in at most k variables. In [16], it is shown that two graphs are C^k-equivalent if and only if they share the same restricted left profile where the graphs $\mathbf{F}$ all have *treewidth* at most $k-1$. (This, together with Lovász Theorem, strengthens and reproves a known fact that two graphs are isomorphic if and only if they satisfy the same first-order sentences, with or without counting.)

As a second example, let us recall `GRAPH-ISOMORPHISM` stated earlier. While it remains open today whether this problem can be solved in polynomial-time in general, some heuristic polynomial-time algorithms have been devised for it. The *k-dimensional Weisfeiler-Leman algorithm* [50], where $k \geq 1$, is a well-known example of this; when $k = 1$, it is also known as the *color refinement algorithm*. Let k be fixed. For a given graph $\mathbf{G}$, this algorithm initially assigns the default color to every k-tuple of vertices of $\mathbf{G}$ that matches the isomorphism type of the subgraph of $\mathbf{G}$ induced by that k-tuple, and then runs in iterations, assigning a new color to every k-tuple that matches its old color and the number of adjacent k-tuples of each old color assigned in the previous iteration (or based on their default colors if in the first iteration). It can be easily observed that, two k-tuples that are assigned different colors in an iteration (or are assigned different default colors prior to the first iteration) will never be assigned the same new color in a later iteration. That is to say, during the execution of the algorithm, the groups of k-tuples of all colors split into smaller and smaller groups in the iterations, until no more split is possible (the total number of iterations is obviously bounded by the total number of k-tuples of vertices of $\mathbf{G}$), for which we say the coloring *stabilizes*. Intuitively, this algorithm computes local structural information (i.e., the color) of each k-tuple in $\mathbf{G}$, and returns the global coloring of the entire graph $\mathbf{G}$. Therefore, if two graphs are isomorphic, then they must share the same global coloring. However, the converse may not be true: In case of $k = 1$, this algorithm computes the same global coloring for nonisomorphic regular graphs of the same degree that have the same number of vertices; in case of $k > 1$, it is known that there are infinitely many pairs of nonisomorphic graphs that share the same global coloring [7]. In other words, this algorithm, while computationally efficient, is not a complete solution for `GRAPH-ISOMORPHISM`. In [7], it is also shown that for $k \geq 1$, two graphs are C^{k+1}-equivalent if and only if they share the same global coloring computed by the k-dimensional Weisfeiler-Leman algorithm. This, together with the previous example, implies that two graphs indistinguishable by the k-dimensional Weisfeiler-Leman algorithm (in terms of the global colorings they receive) if and only if they share the same restricted left profile in which all graphs $\mathbf{F}$ have treewidth at most k.

As a third example, two graphs are said to be C_k*-equivalent*, where $k \geq 1$, if they satisfy the same C_k-sentences, the first-order sentences with counting whose quantifier rank is most k. It is shown in [29] that two graphs are C_k-equivalent if and only if they share the same left profile where the graphs $\mathbf{F}$ all have *treedepth* at most k.

Interestingly, there is a dual statement to the above Lovász Theorem, known as Chaudhuri-Vardi Theorem [9] (see Theorem 4.2), which says that *two graphs* $\mathbf{G}$ *and* $\mathbf{H}$ *are isomorphic if and only if* $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$ *for all graphs* $\mathbf{F}$. Fix a graph $\mathbf{G}$, the homomorphism counts $\hom(\mathbf{G}, \mathbf{F})$ for all graphs $\mathbf{F}$ collectively are called the (*full*) *right profile of* $\mathbf{G}$. In a likewise manner, pairs of graphs $\mathbf{G}$ and $\mathbf{H}$ that share the same right profile in which the graphs $\mathbf{F}$ come from a restricted nonempty class, also give rise to an equivalence relation coarser than isomorphism. However, little attention was paid in this regard, compared to that received by the consideration mentioned in the previous paragraphs.

This dissertation gives an in-depth discussion of what interesting equivalence relations over graphs (coarser than isomorphism) can or cannot be captured by restricting the left profile or by restricting the right profile in Chapter 4. It turns out that some equivalence relations known to be captured by restricting one profile cannot be captured by restricting the other profile. Besides tailor-made arguments for specific equivalence relations, a more general condition of when an equivalence relation cannot be captured by restricting the right profile is given (see Theorem 4.17). Furthermore, for $k \geq 1$, both FO^k-equivalence and FO_k-equivalence cannot be captured by restricting the left profile nor by restricting the right profile (see Proposition 4.24), where FO^k-equivalence and FO_k-equivalence are indistinguishability in terms of satisfaction of FO^k-sentences (first-order sentences in at most k variables) and FO_k-sentences (first-order sentences of quantifier rank at most k), respectively.

A different perspective on Lovász Theorem tells that the class of all graphs isomorphic to a fixed graph $\mathbf{H}$ shares the same left profile, thus the membership problem corresponding to this class can be solved by evaluating the homomorphism counts $\hom(\mathbf{F}, \mathbf{G})$ of an input graph $\mathbf{G}$ (for all graphs $\mathbf{F}$) and checking these numbers against those in the left profile of $\mathbf{H}$. The astute reader might instantly notice an issue: The full left profile of $\mathbf{H}$ involves infinitely many numbers, and it is not realistic to evaluate "all" homomorphism counts $\hom(\mathbf{F}, \mathbf{G})$ before concluding whether $\mathbf{G}$ is isomorphic to $\mathbf{H}$. Indeed, an examination of the proof of Lovász Theorem shows that $\mathbf{G}$ *is isomorphic to* $\mathbf{H}$ *if and only if* $\hom(\mathbf{F}, \mathbf{G}) = \hom(\mathbf{F}, \mathbf{H})$ *for all graphs* $\mathbf{F}$ *having at most* n *vertices, provided that* $\mathbf{H}$ *has* n *vertices* (cf. Corollary 4.5). That is to say, it suffices to check a "finite" number of homomorphism counts to see if $\mathbf{G}$ is isomorphic to $\mathbf{H}$. This leads to an algorithm for checking a (structural) property that is based on the evaluation of the finitely many fixed queries that are homomorphism counts, and in this case the property refers to a graph being isomorphic to a fixed graph. Since an arbitrary (structural) property can be identified with the class of all graphs having the property, the aforementioned algorithm is indeed one to decide membership in a class. It is then natural to ask what interesting classes (equivalently, what interesting properties) admit an algorithm that answers yes or no by making queries that are homomorphism counts $\hom(\mathbf{F}, \mathbf{G})$ for finitely many predetermined graphs $\mathbf{F}$.

In [10, 11], it is shown that every Boolean combination of universal first-order sentences admits such an algorithm and, particularly in [11], that some common classes, including the class of graphs having an isolated vertex, the class of planar graphs and the class of k-colorable graphs ($k \geq 2$), do not admit this kind of algorithms.

In addition, a nontrivial enhanced version of Chaudhuri-Vardi Theorem is given in [11] that says that $\mathbf{G}$ *is isomorphic to* $\mathbf{H}$ *if and only if* $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$ *for all graphs* $\mathbf{F}$ *having at most* m *vertices, where* $m = \max\{2, n^3\}$*, provided that* $\mathbf{H}$ *has* n *vertices* (cf. Corollary 4.5). In other words, the class of graphs isomorphic to a fixed graph $\mathbf{H}$ admits an algorithm for deciding membership based on the evaluation of the homomorphism-count queries $\hom(\mathbf{G}, \mathbf{F})$ against the input graph $\mathbf{G}$ for all graphs $\mathbf{F}$ of a bounded number of vertices. Therefore, one can likewise ask what interesting classes admit an algorithm in this way. In [11], there is essentially only one "positive" result in this regard, however: *If there is a bound on the number of edges in the graphs of a class, then the class admits such an algorithm.*

In that same paper, it is also shown that, by allowing the graphs $\mathbf{F}$ in the queries to vary according to the values of the previously made queries but with a predetermined construction (that is, the graphs $\mathbf{F}$ are adaptive), it suffices to make at most three queries of the form $\hom(\mathbf{F}, \mathbf{G})$ to determine whether the input graph $\mathbf{G}$ is isomorphic to the fixed graph $\mathbf{H}$; moreover, three adaptive queries are necessary in some cases and hence the result is optimal in terms of the number of queries made. As a "negative" result, it is proved that there is no algorithm that makes a bounded number of adaptive queries of the form $\hom(\mathbf{G}, \mathbf{F})$ for the same task.

For brevity, we call algorithms that make queries of the form $\hom(\mathbf{F}, \mathbf{G})$ for input $\mathbf{G}$, with $\mathbf{F}$ fixed or adaptive, the *left query algorithms* (since the graphs $\mathbf{F}$ appear as the left argument of $\hom(*, *)$) and, symmetrically, those that make queries of the dual form $\hom(\mathbf{G}, \mathbf{F})$ for input $\mathbf{G}$, with $\mathbf{F}$ fixed or adaptive, the *right query algorithms* (since the graphs $\mathbf{F}$ appear as the right argument of $\hom(*, *)$). The paper [11] mostly gives positive results of left query algorithms and negative results of right query algorithms.

This dissertation gives some counterbalancing examples in Chapter 5: An important family of classes of graphs – those that correspond to a (nonuniform) constraint satisfaction problem, while admitting a straightforward right query algorithm (of a fixed query), do not admit any left query algorithm (of fixed queries) except for the trivial ones (see Lemma 5.12). Moreover, if adaptive queries of either form $\hom(\mathbf{F}, \mathbf{G})$ or $\hom(\mathbf{G}, \mathbf{F})$ are allowed for the input graph $\mathbf{G}$, then there is a *hybrid query algorithm* that combines the best feature of either form and requires at most two adaptive queries for determining whether the input $\mathbf{G}$ is isomorphic to a fixed $\mathbf{H}$ (cf. Figure 5.4).

As a matter of fact, the aforementioned notions of isomorphism and homomorphism, as well as left and right profiles and the statements of Lovász and Chaudhuri-Vardi Theorems, also apply to relational structures (which are essentially equivalent to *relational databases* in database theory). Thus, the same questions whether a class admits a left or a right query algorithm can be asked of relational structures, and these are also discussed in Chapter 5, where homomorphism counts are evaluated over either the bag-set semiring N (the actual numbers) or the Boolean semiring B (indicators of existence of homomorphisms, 0 or 1). We will call a left query algorithm in which the homomorphism counts are evaluated over N a *left query algorithm over* N and one in which these counts are evaluated over B a *left query algorithm algorithm over* B, analogously for *a right query algorithm over* N and *a right query algorithm over* B. Intuitively, homomorphism counts over N carry more information than those over B and, in general, query algorithms over N are more capable in deciding membership in a class than are query algorithms over B (see Example 5.3). Thus, whenever a class admits a left (or right) query algorithm over B, it must also admit a left (or right, respectively) query algorithm over N (see Proposition 5.1). The classes of relational structures that correspond to constraint satisfaction problems naturally admit a right query algorithm over B and hence one over N, and they may or may not admit a left query algorithm over B or over N. Surprisingly, however, *such classes admit a left query algorithm over* B *precisely when they admit one over* N*, and precisely when they are definable in first-order logic* (see Theorem 5.15), which means that left query algorithms over N are not more capable than their counterparts over B when it comes to deciding membership in such classes. This statement extends to unions of classes that correspond to constraint satisfaction problems (see Theorem 5.16), which are closed under homomorphic equivalence (i.e., for arbitrary relational structures **A** and **B** which admit a homomorphism to each other, **A** is in a class $\mathcal{D}$ that is a union of such classes if and only if **B** is in $\mathcal{D}$). Furthermore, it is shown in [8] that in general this statement is valid for classes that are closed under homomorphic equivalence.

The organization of this dissertation is as follows. In Chapter 2, we go over some fundamental concepts and introduce the terms used throughout the dissertation, followed by a separate section for graphs and relational structures each.[1] In Chapter 3, we give a brief introduction to mathematical logic, where first-order logic and its augmentation with counting quantifiers (and their fragments) are covered, and then we study the basics of homomorphisms and their counts over the semirings B and N. These are the two frameworks to investigate various problems and they play a central role in this dissertation. In Chapter 4, we present a simultaneous generalization of Lovász Theorem and Chaudhuri-Vardi Theorem with a detailed proof, and then proceed to investigate the problems of characterizing equivalence relations over graphs by restricted left or right profiles. In Chapter 5, we

[1] Although it is common practice to define graphs as relational structures, we view them as distinct objects. This facilitates a clearer exposition of graph-theoretic notions concerning relational structures, and it allows us to present the results in Chapters 4 and 5 in a more proper context.

turn our attention to query algorithms and study some characterizations of when a class (of graphs or of relational structures) admits a left or a right query algorithm over B or over N. Finally, we conclude this dissertation in Chapter 6.

Chapter 2

Preliminaries

This chapter serves to introduce and clarify the basic terms on which more sophisticated notions are to be developed and to bring in some basic results that will be referred to repeatedly in later chapters. We will go over some fundamental concepts including sets (and classes), graphs and relational structures, and isomorphism. The reader should be advised that the materials presented in this introductory chapter are by no means comprehensive as each of these topics deserves a whole treatise.

2.1 Fundamentals

As a widely accepted foundation of mathematics, *set theory* provides a simple yet flexible framework in which to build various mathematical theories. The most prominent objects of study in set theory are of course *sets*, based on which other more involved objects are defined. A *class* of objects is informally a collection of such things and may or may not be a set itself. In this dissertation, however, we shall not concern ourselves with such technical issues but rather treat classes as sets, using the two terms interchangeably. We assume our sets to have been built on a standard system of axioms (e.g., ZFC) and common objects such as *functions* (also known as *mappings*), *sequences* (also known as *tuples*) and *relations* to have been defined as special sets. For the rest of this section, we clarify some terms and notations of these objects that will be used throughout this dissertation.

Sets

The *empty set*, i.e., the set containing no elements, is denoted $\emptyset$. Let A and B be sets. We write $A \subseteq B$ if A is a *subset* of B. If U is the *universe* of objects of discourse (i.e., the *context*) and if $A \subseteq U$, then we write $\overline{A}$ for the *complement* of A in U. The *difference* of A from B, denoted $A \setminus B$, is the set $A \cap \overline{B}$. We say A and B are *disjoint* if $A \cap B = \emptyset$; when this is the case, we sometimes write $A \uplus B$, called the *disjoint union* of A and B, for $A \cup B$ to emphasize that A and B are disjoint. The *Cartesian product* of A and B is denoted $A \times B$. For $n \geq 1$, the *n-th Cartesian power* of A is the set $A^n := \underbrace{A \times \cdots \times A}_{n\text{-times}}$.

The three operations $\overline{}$ (*complementation*), $\cap$ (*intersection*) and $\cup$ (*union*) are *Boolean operations.*

Let $A_1, \ldots, A_n$ and A be sets. We say A is a *Boolean combination* of the sets $A_1, \ldots, A_n$ if A can be obtained by successively applying the Boolean operations, starting from the sets $A_1, \ldots, A_n$.

The set of natural numbers $0, 1, 2, \ldots$ is denoted $\mathbb{N}$, and the set of positive integers $1, 2, 3, \ldots$ is denoted $\mathbb{Z}^+$. Given $m, n \in \mathbb{N}$, we let $[m, n]$ denote the set $\{m, \ldots, n\}$ if $m \leq n$ and the empty set $\emptyset$ otherwise; in particular, we let $[n]$ abbreviate $[1, n]$.

Let A be a set. The *size* (or *cardinality*) of A, denoted $|A|$, is the number of elements in A. We say A is *finite* if $|A|$ is a natural number and *infinite* otherwise.

Let $n \geq 2$. We say a list of objects $a_1, \ldots, a_n$ are *pairwise distinct* if $i \neq j$ implies $a_i \neq a_j$, for all $i, j \in [n]$. We say a list of sets $A_1, \ldots, A_n$ are *pairwise disjoint* if $i \neq j$ implies $A_i \cap A_j = \emptyset$, for all $i, j \in [n]$; when this is the case, we sometimes write $\biguplus_{i=1}^n A_i$ for $\bigcup_{i=1}^n A_i$ to emphasize that $A_1, \ldots, A_n$ are pairwise disjoint.

Functions

Let $f : A \to B$ be a function. For $a \in A$, we call $f(a)$ the *image of a under f*. Let S be a subset of A. The *image of S under f* is the set $f(S) = \{f(a) \mid a \in S\}$. We write $f|_S$ for the *restriction* of f to S. We write $a \mapsto f(a)$ for the *mapping rule* of f, i.e., the set of pairs $\{(a, f(a)) \mid a \in A\}$. We say

- f is *injective* if it is one-to-one,
- f is *surjective* if it is onto,
- f is *bijective* if it is one-to-one and onto.

If f is bijective, we sometimes say f is a *one-to-one correspondence* from A to B and we write $f^{-1} : B \to A$ for its *inverse*. Abusing the notation f^{-1}, for all $b \in B$ we write $f^{-1}(b)$ for the set $\{a \in A \mid f(a) = b\}$ (regardless whether f is bijective or not), called the *preimage of b*. For every function $g : B \to C$, we write $f \circ g$ for the *composition* of f with g: For all $a \in A$, we have $(f \circ g)(a) = f(g(a))$.

For every statement S, we write $\mathrm{I}(S)$ for the *indicator function* of S, which takes value 1 if S is true and 0 otherwise.

Let A and B be two sets. For every $n \in \mathbb{Z}^+$, we say $f : A^n \to B$ is an *n-ary* *function*; if $B = A$, we say it is an *n-ary operation on A*. It is customary to write a binary operator in between its operands. Let $\Box$ be a binary operation on A. We say

- $\Box$ is *associative* if $(a \Box b) \Box c = a \Box (b \Box c)$ for all $a, b, c \in A$ (and we write $a \Box b \Box c$ for $(a \Box b) \Box c$ and equivalently for $a \Box (b \Box c)$),
- $\Box$ is *commutative* if $a \Box b = b \Box a$ for all $a, b \in A$.

If $A \subseteq B$, we write $i : A \to B$ for the *inclusion mapping*: For all $a \in A$, we have $i(a) = a$.

Remark 2.1. Fix sets A, B and T, and let $f : A \to B$ be a function.

(a) If $B \subseteq T$, then there is a unique function $g : A \to T$ whose mapping rule coincides with $a \mapsto f(a)$. Indeed, $g = i \circ f$ where $i : B \to T$ is the inclusion mapping.

(b) If $f(A) \subseteq T \subseteq B$, then there is a unique function $g : A \to T$ whose mapping rule coincides with $a \mapsto f(a)$. In particular, there is a unique surjective function $g : A \to f(A)$ whose mapping rule is $a \mapsto f(a)$.

Sequences (Tuples) and Relations

Let $f : A \to B$ be a function. The *sequence* $(f(a) \mid a \in A)$ is an alternative view of f, in which the *entries*, $f(a)$, are indexed by $a \in A$, and the *size* of this sequence is $|A|$ (we refrain from the more common term *length* to avoid ambiguity when we define the length of a walk (as a sequence of vertices) in a graph later in Section 2.2).

Let $n \in \mathbb{Z}^+$. For objects $a_1, \dots, a_n$ (not necessarily pairwise distinct), we write $(a_1, \dots, a_n)$ in place of $(f(i) \mid i \in [n])$ where $f(i) = a_i$ for all $i \in [n]$. For all $i, j \in [n]$ with $i \leq j$, the sequence $(a_i, \dots, a_j)$ is a *subsequence* of $(a_1, \dots, a_n)$. A sequence of size n is also called an *n-tuple*. In particular, a 2-tuple is also called an (*ordered*) *pair*. When the size n of an n-tuple $(a_1, \ldots, a_n)$ is irrelevant or understood from the context, we often simply say a *tuple* and let $\overline{a}$ abbreviate $(a_1, \ldots, a_n)$.

Let A be a set, and let $n \in \mathbb{Z}^+$. The elements in A^n are *n-tuples in A*. A subset $R \subseteq A^n$ is an *n-ary relation over A*, whose *arity* is n. For every $\overline{a} \in A^n$, every $a \in A$ and every $i \in [n]$, we write $\overline{a}(i/a)$ for the n-tuple $(a_1, \ldots, a_{i-1}, a, a_{i+1}, a_n)$ and write $\overline{a}a$ for the $(n+1)$-tuple $(a_1, \ldots, a_n, a)$.

Binary Relations and Equivalence

Let A be a set and P be a binary relation over A. Whenever possible, we write P as a predicate as opposed to a set, i.e., we prefer the notation $a\ P\ b$ to $(a, b) \in P$; and we write $a \not P\ b$ if it is not true that $a\ P\ b$. We say P is

- *reflexive* if for all $a \in A$, it holds that $a\ P\ a$,
- *symmetric* if for all $a, b \in A$, it holds that $a\ P\ b$ implies $b\ P\ a$,
- *transitive* if for all $a, b, c \in A$, it holds that $a\ P\ b$ and $b\ P\ c$ together imply $a\ P\ c$.

If P is symmetric, then $a\ P\ b$ precisely when $b\ P\ a$; thus, we sometimes say "a and b are P" alternatively to "a is P to b." We say P is an *equivalence relation* over A if it is reflexive, symmetric and transitive; in this case, we often say "a and b are P (equivalent)" alternatively to "a is P (equivalent) to b," by the symmetry.

Let A be a set and P and P' be equivalence relations over A. We say P is *finer* than P' or, reciprocally, P' is *coarser* than P, if for all $a, b \in A$, it holds that $a\ P\ b$ implies $a\ P'\ b$; we also say P' is a *relaxation* of P in this case. We say P *coincides with* P' or, equivalently, P *and* P' *coincide* when both P is finer than P' and P' is finer than P.

We often use the symbol $\equiv$ when referring to an (arbitrary) equivalence relation over a set and, if given a term T (such as first-order logic, see Subsection 3.1.1), the symbol $\equiv_T$ when referring to the equivalence relation arising from being equal or indistinguishable in (the sense of) T (the precise meaning of this statement will be made clear when T is introduced). When $a \equiv_T b$, we say *a is T-equivalent to b* or *a is indistinguishable from b in* (*the sense of*) T.

Let A be a set and $\equiv$ be an equivalence relation over A.

- For $a \in A$, we write $[a]_{\equiv} := \{b \in A \mid a \equiv b\}$ for the *equivalence class of a induced by* $\equiv$, and we assume that there is a distinguished member of $[a]_{\equiv}$, called the *representative*.
- We say "up to the equivalence $\equiv$" in a statement when the statement being made is true within the exceptions due to the possible differences between a and b, for all $a, b \in A$ such that $a \equiv b$.
- Let T be a term, we say that $\equiv$ *is characterized by* $\equiv_T$ (*over* A) or $\equiv$ *is characterized in* T (*over* A) if $\equiv$ coincides with $\equiv_T$ over A.
- Let T be a term and $a \in A$.
  - For every expression t in T, we say *a is characterized by t* (*over* A) if a can be distinguished from other elements in A by t. We say *a is characterized in T* (*over* A) if there is such an expression t,
  - For every expression t in T, we say *a is characterized by t up to* $\equiv$ (*over* A) if a can be distinguished by t from all other elements $b \in A$ such that $a \not\equiv b$. We say *a is characterized in T up to* $\equiv$ (*over* A) if there is such an expression t.

- We say a set $B \subseteq A$ is *closed under* $\equiv$ if for all a and b in A such that $a \equiv b$, we have $a \in B$ implies $b \in B$; in this case, we have $B = \bigcup_{b \in B} [b]_\equiv$.
- We say an n-ary function f of domain A is *invariant under* $\equiv$ if $f(a_1, \ldots, a_n) = f(b_1, \ldots, b_n)$ for all $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ such that $a_1 \equiv b_1, \ldots, a_n \equiv b_n$.
- We say an n-ary relation R over A is *invariant under* $\equiv$ if for all $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ such that $a_1 \equiv b_1, \ldots, a_n \equiv b_n$, it holds that $(a_1, \ldots, a_n) \in R$ if and only if $(b_1, \ldots, b_n) \in R$.

Multisets and Placeholder

A *multiset* is a collection of objects in which the members can occur multiple times. We use braces $\boldsymbol{\{\}}$ in boldface for the notation of multisets in contrast to $\{\}$ for sets.

We adopt the *wildcard symbol* $*$ as a placeholder, usually within function or relation (predicate) symbols, in a statement to make the statement universally applicable for everything that is appropriate where $*$ is.

2.2 Graphs

Definition 2.1. A *graph* is a pair $\mathbf{G} = (V(\mathbf{G}), E(\mathbf{G}))$, where the *vertex set* $V(\mathbf{G})$ *of* $\mathbf{G}$ is a finite nonempty set of elements, called the *vertices* (or *nodes*) in $\mathbf{G}$, and the *edge set* $E(\mathbf{G})$ *of* $\mathbf{G}$ is a set of subsets of $V(\mathbf{G})$ of size 2, called the *edges* in $\mathbf{G}$. The *size of* $\mathbf{G}$ refers to $|V(\mathbf{G})|$.

Definition 2.2. Let $\mathbf{G}$ be a graph. The *degree of a vertex* v *in* $\mathbf{G}$ is

$$\deg_{\mathbf{G}}(v) := |\{u \in V(\mathbf{G}) \mid \{u, v\} \in E(\mathbf{G})\}|,$$

and the *degree of* $\mathbf{G}$ is $\deg(\mathbf{G}) := \max\{\deg_{\mathbf{G}}(v) \mid v \in V(\mathbf{G})\}$.

Let $\mathbf{G}$ be a graph. For every graph $\mathbf{H}$, we say that $\mathbf{G}$ *is a subgraph of* $\mathbf{H}$ or $\mathbf{H}$ *contains* $\mathbf{G}$ (*as a subgraph*) (written: $\mathbf{G} \subseteq \mathbf{H}$) if $V(\mathbf{G}) \subseteq V(\mathbf{H})$ and $E(\mathbf{G}) \subseteq E(\mathbf{H})$. For every nonempty subset $S \subseteq V(\mathbf{G})$, we write $\mathbf{G}[S]$ for the *subgraph of* $\mathbf{G}$ *induced by* S, namely, the graph with S as its vertex set and $\{\{u, v\} \in E(\mathbf{G}) \mid \{u, v\} \subseteq S\}$ as its edge set.

Definition 2.3. Let $\mathbf{G}$ and $\mathbf{H}$ be graphs.

(a) The *direct sum of* $\mathbf{G}$ *and* $\mathbf{H}$, denoted $\mathbf{G} \oplus \mathbf{H}$, is the graph with

- vertex set $V(\mathbf{G} \oplus \mathbf{H}) := V(\mathbf{G}) \uplus V(\mathbf{H})$ and edge set $E(\mathbf{G} \oplus \mathbf{H}) := E(\mathbf{G}) \uplus E(\mathbf{H})$ if $V(\mathbf{G})$ and $V(\mathbf{H})$ are disjoint, otherwise
- vertex set $V(\mathbf{G} \oplus \mathbf{H}) := \{(v, 1) \mid v \in V(\mathbf{G})\} \uplus \{(v, 2) \mid v \in V(\mathbf{H})\}$ and edge set $E(\mathbf{G} \oplus \mathbf{H}) := \{\{(u, 1), (v, 1)\} \mid \{u, v\} \in E(\mathbf{G})\} \uplus \{\{(u, 2), (v, 2)\} \mid \{u, v\} \in E(\mathbf{H})\}$.

(b) The *direct product of* $\mathbf{G}$ *and* $\mathbf{H}$, denoted $\mathbf{G} \otimes \mathbf{H}$, is the graph with vertex set $V(\mathbf{G} \otimes \mathbf{H}) := V(\mathbf{G}) \times V(\mathbf{H})$ and edge set

$$E(\mathbf{G} \otimes \mathbf{H}) := \{\{(u_1, v_1), (u_2, v_2)\} \mid \{u_1, u_2\} \in E(\mathbf{G}) \text{ and } \{v_1, v_2\} \in E(\mathbf{H})\}.$$

Let $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ be arbitrary graphs. We have that $\mathbf{G} \oplus \mathbf{H}$ and $\mathbf{H} \oplus \mathbf{G}$ are *isomorphic* (informally speaking, structurally identical, a notion to be formally introduced later) and so are $(\mathbf{F} \oplus \mathbf{G}) \oplus \mathbf{H}$ and $\mathbf{F} \oplus (\mathbf{G} \oplus \mathbf{H})$. That is to say, the binary operation $\oplus$ is commutative and associative in graphs *up to isomorphism*, which allows us to write $\bigoplus_{i=1}^{n} \mathbf{G}_i$ or write $\bigoplus_n \mathbf{G}$ particularly when $\mathbf{G}_i = \mathbf{G}$ for all $i \in [n]$. The above can be said about $\otimes$ by analogy. Note that both $\oplus$ and $\otimes$ lack an identity element, and in particular we may run into an issue with $n = 0$ in $\bigoplus_{i=1}^{n} \mathbf{G}_i$ in Subsection 4.2.5, for which we devise an ad hoc identity element as a resolution.

Definition 2.4. Let $\mathbf{G}$ be a graph. A *partition of* $V(\mathbf{G})$ is a set $\theta = \{V_1, \ldots, V_n\}$, $n \in \mathbb{Z}^+$, of pairwise disjoint nonempty subsets of $V(\mathbf{G})$ such that

- $e \not\subseteq V_i$ for $i \in [n]$ and for $e \in E(\mathbf{G})$, and
- $\biguplus_{i=1}^{n} V_i = V(\mathbf{G})$.

The *quotient of* $\mathbf{G}$ *by* θ is the graph $\mathbf{G}/\theta$ with vertex set $V(\mathbf{G}/\theta) := \theta$ and edge set $E(\mathbf{G}/\theta) := \{\{V_i, V_j\} \subseteq V(\mathbf{G}/\theta) \mid \{v_i, v_j\} \in E(\mathbf{G}) \text{ for some } v_i \in V_i \text{ and } v_j \in V_j\}$.

We say a partition θ is *trivial* if every set $V \in \theta$ has size 1; in this case, $\mathbf{G}/\theta$ is isomorphic to $\mathbf{G}$ and thus we set $\mathbf{G}/\theta := \mathbf{G}$ for brevity.

Let $\mathbf{G}$ be a graph.

- For $u, v \in V(\mathbf{G})$, we say u *is adjacent to* v if $\{u, v\} \in E(\mathbf{G})$.
- For $v \in V(\mathbf{G})$, we say v is *isolated* if it is not adjacent to any vertex.
- Let $n \geq 1$. A *walk of length* n in $\mathbf{G}$ is a sequence $(v_0, \ldots, v_n)$ of vertices $v_0, \ldots, v_n \in V(\mathbf{G})$ in which $(v_i, v_{i+1}) \in E(\mathbf{G})$ for $i \in [0, n-1]$.[1] A *path of length* n in $\mathbf{G}$ is a walk of length n in $\mathbf{G}$ in which $v_0, \ldots, v_n$ are pairwise distinct.
- For $n \geq 2$, a *closed walk of length* n in $\mathbf{G}$ is a walk $(v_0, \ldots, v_n)$ in $\mathbf{G}$ such that $v_n = v_0$.
- For $n \geq 3$, a *cycle of length* n in $\mathbf{G}$ is a closed walk $(v_0, \ldots, v_n)$ in $\mathbf{G}$ in which $v_0, \ldots, v_{n-1}$ are pairwise distinct.

[1] Note that while the walk $(v_0, \ldots, v_n)$ has *length* n, it has *size* $n + 1$ when viewed as a sequence.

When the length n is irrelevant or can be understood from the context, we often simply say a (*closed*) *walk*, *path* or *cycle*.

- We say $\mathbf{G}$ is *acyclic* if it contains no cycles as a subgraph.
- For distinct $u, v \in V(\mathbf{G})$, we say $(v_0, \ldots, v_n)$ is a *walk* (or *path*) *from* u *to* v *in* $\mathbf{G}$ if $(v_0, \ldots, v_n)$ is a walk (or path, respectively) in $\mathbf{G}$ with $v_0 = u$ and $v_n = v$.
- For $u, v \in V(\mathbf{G})$, the *distance between* u *and* v (*in* $\mathbf{G}$), denoted $d_{\mathbf{G}}(u, v)$, is 0 if $u = v$, is the minimum length of a path from u to v if $u \neq v$ and such a path exists, and is ∞ otherwise.
- The *diameter of* $\mathbf{G}$ is $\delta(\mathbf{G}) := \max\{d_{\mathbf{G}}(u, v) \mid u, v \in V(\mathbf{G})\}$.
- For $u, v \in V(\mathbf{G})$, we say v *is reachable from* u (written: $u \sim v$) if $d_{\mathbf{G}}(u, v) < \infty$. (Note that $\sim$ is a reflexive, symmetric and transitive binary relation and hence is an equivalence relation over $V(\mathbf{G})$.)
- A *connected component* of $\mathbf{G}$ is a subgraph of $\mathbf{G}$ induced by a set of vertices in $\mathbf{G}$ that is an equivalence class induced by $\sim$. (Note that if $\mathbf{G}$ has $\mathbf{G}_1, \ldots, \mathbf{G}_n$ as its connected components, then $\mathbf{G} = \mathbf{G}_1 \oplus \cdots \oplus \mathbf{G}_n$.)
- We say $\mathbf{G}$ is *connected* if it is a (and hence the only) connected component of itself. (Note that $\mathbf{G}$ is connected if and only if $\delta(\mathbf{G}) < \infty$.)

The next is a useful property about closed walks of odd length and cycles of odd length in a graph.

Proposition 2.1. *Let* $\mathbf{G}$ *be a graph. If* $\mathbf{G}$ *contains a closed walk of odd length* $n \geq 3$*, then* $\mathbf{G}$ *contains a cycle of odd length* m *such that* $n \geq m \geq 3$*.* $\square$

Note that Proposition 2.1 does not hold for closed walks and cycles of even lengths in general. In fact, for the graph $\mathbf{G}$ with $V(\mathbf{G}) = \{u, v\}$ and $E(\mathbf{G}) = \{\{u, v\}\}$, it is obvious that (u, v, u) is a closed walk of length 2 in $\mathbf{G}$ but there is no cycle (hence no cycle of even length) in $\mathbf{G}$.

Special Graphs

Definition 2.5. Let $n \in \mathbb{Z}^+$.

(a) The *independent set* (or *empty graph*) *of size* n is denoted $\mathbf{I}_n$ and is the graph with vertex set $V(\mathbf{I}_n) = \{v_1, \ldots, v_n\}$ and edge set $E(\mathbf{I}_n) = \emptyset$.

(b) The *clique* (or *complete graph*) *of size* n is denote $\mathbf{K}_n$; $\mathbf{K}_1 := \mathbf{I}_1$ and for $n \geq 2$, $\mathbf{K}_n$ is the graph with vertex set $V(\mathbf{K}_n) = \{v_1, \ldots, v_n\}$ and edge set

$$E(\mathbf{K}_n) = \{\{v_i, v_j\} \mid i, j \in [n] \text{ and } i < j\}.$$

(c) The *path graph of size* n is denoted $\mathbf{P}_n$; $\mathbf{P}_1 := \mathbf{I}_1$ and for $n \geq 2$, $\mathbf{P}_n$ is the graph[2] with vertex set $V(\mathbf{P}_n) = \{v_1, \ldots, v_n\}$ and edge set $E(\mathbf{P}_n) = \{\{v_i, v_{i+1}\} \mid i \in [n-1]\}$.

(d) The *cycle graph of size* n is denoted $\mathbf{C}_n$; $\mathbf{C}_1 := \mathbf{K}_1$, $\mathbf{C}_2 := \mathbf{K}_2$ and for $n \geq 3$, $\mathbf{C}_n$ is the graph[3] with vertex set $V(\mathbf{C}_n) = \{v_1, \ldots, v_n\}$ and edge set

$$E(\mathbf{C}_n) = \{\{v_i, v_{i+1}\} \mid i \in [n-1]\} \cup \{\{v_n, v_1\}\}.$$

By definition, a graph $\mathbf{G}$ is acyclic if and only if $\mathbf{G}$ does not contain $\mathbf{C}_n$ as a subgraph for any $n \geq 3$. We say a graph is a *forest* when it is acyclic, and we say that it is

[2] Note that path graphs $\mathbf{P}_n$ are all paths themselves except for $n = 1$.

[3] Note that cycle graphs $\mathbf{C}_n$ are all cycles themselves except for $n = 1$ and $n = 2$.

a *free tree*[4] if it is connected in addition.

The list below enumerates notations for some common classes of graphs. We assume that all classes of graphs to be *closed under isomorphism* (to be explained next).

Notations. (a) The class of all graphs is denoted $\mathcal{G}$.

(b) The class of all independent sets is denoted $\mathcal{I}$.

(c) The class of all cliques is denoted $\mathcal{K}$.

(d) The class of all path graphs is denoted $\mathcal{P}$.

(e) The class of all cycle graphs is denoted $\mathcal{C}$.

(f) The class of all free trees is denoted $\mathcal{T}$.

Isomorphism: Structural Identity

In *vector algebra*, vectors in the Euclidean plane are often represented as arrows, in which the additive inverse of a vector is simply the same arrow with the reversed direction and addition of two vectors is done by joining the head of one to the tail of the other to obtain a new straight arrow. An alternative representation of vectors in the Euclidean plain is by pairs of coordinates (i.e., pairs of real numbers), in which the additive inverse of a vector is the pair of real numbers that are the additive inverses of the respective coordinates and addition of two vectors is done by coordinate-wise addition. It turns out that there is a one-to-one correspondence between the set of arrows and the set of pairs of coordinates

[4]A *rooted tree* is the same as a free tree, except that it contains a distinguished node called the *root*, and is often depicted in a top-down manner where the root is at the top and the child nodes of a given node are below it. The nodes in a rooted tree that have no child nodes are *leaves*. A subtle difference in the terminology of free trees and rooted trees is that, while nodes and vertices are used interchangeably for free trees (since they are a special kind of graphs), it is often preferable to say nodes instead of vertices for rooted trees (and to differentiate rooted trees from graphs).

under which "the individuals in one world behave in exactly the same way as those in the other." Hence, the system of arrows and the system of pairs of coordinates are "structurally identical" and any difference between them is deemed superficial, which suggests that the two systems should be treated as synonymous to each other, as long as vector algebra is concerned.

This brings us to *isomorphism*, an important notion in mathematics, especially in areas such as graph theory, database theory, mathematical logic and category theory.[5] In particular, `GRAPH-ISOMORPHISM` is a well-known decision problem in graph theory that asks whether two given graphs are isomorphic. Its computational complexity is obviously NP, and yet it remains unknown whether it is in P or is NP-complete, although it is believed by many to be NP-intermediate (assuming $\mathsf{P} \neq \mathsf{NP}$). The notion of isomorphism among graphs is of interest to us, and we formalize it as follows.

Definition 2.6. Let $\mathbf{G}$ and $\mathbf{H}$ be graphs. A function $\pi : V(\mathbf{G}) \to V(\mathbf{H})$ *is an isomorphism from* $\mathbf{G}$ *to* $\mathbf{H}$ (written: $\pi : \mathbf{G} \cong \mathbf{H}$) if π is a bijection and if for all $u, v \in V(\mathbf{G})$, it is true that $\{u, v\} \in E(\mathbf{G})$ if and only if $\{\pi(u), \pi(v)\} \in E(\mathbf{H})$. We say that $\mathbf{G}$ *is isomorphic to* $\mathbf{H}$ (written: $\mathbf{G} \cong \mathbf{H}$) if there is an isomorphism from $\mathbf{G}$ to $\mathbf{H}$.

It is easy to see that $\cong$ is an equivalence relation over graphs:

Reflexivity. For $\mathbf{G}$ and the identify mapping $\iota : V(\mathbf{G}) \to V(\mathbf{G})$ (such that $\iota(v) = v$ for all $v \in V(\mathbf{G})$), we have $\iota : \mathbf{G} \cong \mathbf{G}$.

[5] The spirit in which mathematicians regard isomorphic systems as one and the same while ignoring any distinctions between them is perhaps best manifested by the famous quote by Henri Poincaré, translated to English as "Mathematicians study not objects, but relations between objects; the replacement of these objects by others is therefore indifferent to them, provided the relations do not change. The matter is for them unimportant, the form alone interests them."

Symmetry. For $\mathbf{G}$ and $\mathbf{H}$, if $\pi : \mathbf{G} \cong \mathbf{H}$, then $\pi^{-1} : \mathbf{H} \cong \mathbf{G}$.

Transitivity. For $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$, if $\pi_1 : \mathbf{F} \cong \mathbf{G}$ and $\pi_2 : \mathbf{G} \cong \mathbf{H}$, then $(\pi_2 \circ \pi_1) : \mathbf{F} \cong \mathbf{H}$.

The equivalence class of a graph $\mathbf{G}$ induced by $\cong$ is the class $[\mathbf{G}]_{\cong} := \{\mathbf{H} \in \mathcal{G} \mid \mathbf{G} \cong \mathbf{H}\}$, called the *isomorphism type of* $\mathbf{G}$. An isomorphism from a graph $\mathbf{G}$ to itself is called an *automorphism of* $\mathbf{G}$, and the number of automorphisms of $\mathbf{G}$ is denoted $\mathrm{aut}(\mathbf{G})$.

Remark 2.2. Every graph $\mathbf{G}$ with vertex set $V(\mathbf{G}) = \{v_1, \ldots, v_n\}$ is completely described by its *adjacency matrix*, an $(n \times n)$-matrix $M^{\mathbf{G}}$ whose entries are

$$M_{ij}^{\mathbf{G}} := \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E(\mathbf{G}), \\ 0 & \text{otherwise.} \end{cases}$$

In terms of adjacency matrices, $\mathbf{G}$ is isomorphic to a graph $\mathbf{H}$ if and only if $M^{\mathbf{H}}$ can be obtained from $M^{\mathbf{G}}$ by simultaneously permuting the rows and columns.

Throughout this dissertation, the focus of study is on structural properties (and they are invariant under isomorphism). Therefore, most often *we mean* $\mathbf{G} \cong \mathbf{H}$ *when we write* $\mathbf{G} = \mathbf{H}$, and *we mean* $\mathbf{G}$ *is isomorphic to a subgraph of* $\mathbf{H}$ *when we write* $\mathbf{G} \subseteq \mathbf{H}$. Furthermore, we assume that *classes of graphs are closed under isomorphism*, i.e, for graphs $\mathbf{G}$ and $\mathbf{H}$ and for classes $\mathcal{D}$ of graphs, if $\mathbf{G} \in \mathcal{D}$ and if $\mathbf{G} \cong \mathbf{H}$, then $\mathbf{H} \in \mathcal{D}$. The *size* of a class $\mathcal{D}$ of graphs is the total number of distinct isomorphism types among the graphs in $\mathcal{D}$, and we say $\mathcal{D}$ is *finite* if its size is in $\mathbb{N}$, otherwise we say $\mathcal{D}$ is *infinite*. If $\mathcal{D}$ is finite with the isomorphism types $[\mathbf{G}_1]_{\cong}, \ldots, [\mathbf{G}_n]_{\cong}$ among the graphs in it, then we often write $\mathcal{D} = \{\mathbf{G}_1, \ldots, \mathbf{G}_n\}$ in place of $\mathcal{D} = [\mathbf{G}_1]_{\cong} \cup \cdots \cup [\mathbf{G}_n]_{\cong}$ for brevity.

For classes $\mathcal{D}$ of graphs, we write

$$\mathcal{D}/\cong \;:=\; \{\mathbf{G} \in \mathcal{D} \mid \mathbf{G} \text{ is the representative of the isomorphism type } [\mathbf{G}]_{\cong}\}$$

for the *set* of graphs in $\mathcal{D}$ that are also the representatives of their respective isomorphism types, which is needed in the condition of set descriptions, sequence descriptions, enumerations or big operators such as $\sum$ or $\biguplus$, to avoid redundancy.

Graph Parameters

A *graph parameter* is a function $f : \mathcal{G} \to \mathbb{R}$ that is invariant under isomorphism, i.e., $f(\mathbf{G}) = f(\mathbf{H})$ for all graphs $\mathbf{G}$ and $\mathbf{H}$ such that $\mathbf{G} \cong \mathbf{H}$ (see [41] for a thorough reference). For example, the diameter $\delta(\mathbf{G})$ of a graph $\mathbf{G}$ introduced earlier is an example of graph parameter.

Definition 2.7. Let $\mathbf{G}$ be a graph.

(a) For $n \in \mathbb{Z}^+$, a function $f : V(\mathbf{G}) \to \{0, \ldots, n-1\}$ is an *n-coloring of* $\mathbf{G}$, with the n integers $0, \ldots, n-1$ as *colors*, if for every two distinct vertices $u, v \in V(\mathbf{G})$, we have that $\{u, v\} \in E(\mathbf{G})$ implies $f(u) \neq f(v)$. The graph $\mathbf{G}$ is *n-colorable* if there is an n-coloring of $\mathbf{G}$.

(b) The *chromatic number of* $\mathbf{G}$, denoted $\chi(\mathbf{G})$, is the minimum $n \in \mathbb{Z}^+$ such that $\mathbf{G}$ is n-colorable.

Example 2.1. Let $n \in \mathbb{Z}^+$ and let $\mathbf{G}$ be a graph.

(a) If $|V(\mathbf{G})| = n$, then $\mathbf{G}$ is n-colorable: Assume $V(\mathbf{G}) = \{0, \ldots, n-1\}$, then the identity function $f : \{0, \ldots, n-1\} \to \{0, \ldots, n-1\}$, (i.e., $f(k) = k$), is an n-coloring of $\mathbf{G}$. Hence, the notion of chromatic number of a graph is well-defined.

(b) *If* $\mathbf{G}$ *is* n*-colorable and if* $m \in \mathbb{Z}^+$ *such that* $m > n$*, then it is* m*-colorable.* In fact, if $f : V(\mathbf{G}) \to \{0, \ldots, n-1\}$ is an n-coloring of $\mathbf{G}$, then the function $g : V(\mathbf{G}) \to \{0, \ldots, m-1\}$ with the same mapping rule as f is an m-coloring of $\mathbf{G}$ (see Remark 2.1(a)).

The next proposition is immediate from Definition 2.1.

Proposition 2.2. *Let* $n \in \mathbb{Z}^+$*, and let* $\mathbf{G}$ *be a graph.*

(a) *If* $\mathbf{G}$ *is* n*-colorable, then every subgraph* $\mathbf{H}$ *of* $\mathbf{G}$ *is also* n*-colorable.*

(b) *Let* $\mathbf{G}_1, \ldots, \mathbf{G}_k$ *be the connected components of* $\mathbf{G}$ *for some* $k \in \mathbb{Z}^+$*. Then,* $\mathbf{G}$ *is* n*-colorable if and only if* $\mathbf{G}_i$ *is* n*-colorable for all* $i \in [k]$*.* $\square$

For $n \in \mathbb{Z}^+$, the decision problem of n-`COLORABILITY` asks whether a given graph is n-colorable. While 3-`COLORABILITY` is obviously in NP (the class of decision problems that admit a nondeterministic polynomial-time algorithm) and indeed is known to be NP-complete, 2-`COLORABILITY`, in contrast, is a member of P (the class of decision problems that admit a deterministic polynomial-time algorithm).

Proposition 2.3. *For every connected graph* $\mathbf{G}$ *that is* 2*-colorable, the number of* 2*-colorings of* $\mathbf{G}$ *is* 2*.* $\square$

Proposition 2.4. *For every graph* $\mathbf{G}$*, we have that* $\mathbf{G}$ *is* 2*-colorable if and only if* $\mathbf{G}$ *contains no cycle of odd length* ≥ 3*.* $\square$

The next lemma, known as (a variant of) the *Sparse Incomparability Lemma* due to P. Erdős in graph theory, asserts the existence of a graph that has a specific chromatic number and an arbitrarily large girth (see Corollary 3.13 in [33] for reference). The relevant definitions are as follows.

Definition 2.8. Let $\mathbf{G}$ be a graph.

(a) The *girth of* $\mathbf{G}$, denoted $\gamma(\mathbf{G})$, is the minimum $n \geq 3$ such that $\mathbf{C}_n$ is contained in $\mathbf{G}$ as a subgraph if any, and is ∞ otherwise.

(b) The *odd girth of* $\mathbf{G}$, denoted $\gamma_{\mathrm{odd}}(\mathbf{G})$, is the minimum odd integer $n \geq 3$ such that $\mathbf{C}_n$ is contained in $\mathbf{G}$ as a subgraph if any, and is ∞ otherwise.

Lemma 2.5 (Sparse Incomparability Lemma)**.** [21] *For all $m, n \in \mathbb{Z}^+$, there exists a graph $\mathbf{G}$ with $\chi(\mathbf{G}) = m$ and $\gamma(\mathbf{G}) \geq n$.* $\square$

Variants of Graphs

There are numerous different variants of graphs in graph theory, and the graphs introduced previously are of the simplest type. We shall introduce two variants subsequently.

Definition 2.9. A *labeled bipartite multigraph* is a triple $\mathbf{G} = (V_1(\mathbf{G}), V_2(\mathbf{G}), E(\mathbf{G}))$, where

- the *first vertex set* $V_1(\mathbf{G})$ *of* $\mathbf{G}$ is a finite nonempty set,
- the *second vertex set* $V_2(\mathbf{G})$ *of* $\mathbf{G}$ is a finite nonempty set,

for which $V_1(\mathbf{G}) \cap V_2(\mathbf{G}) = \emptyset$, and the elements in $V_1(\mathbf{G}) \uplus V_2(\mathbf{G})$ are called the *vertices in* $\mathbf{G}$, and

- the *edge set* $E(\mathbf{G})$ *of* $\mathbf{G}$ is a finite subset of

$$\{(\{u, v\}, n) \mid u \in V_1(\mathbf{G}),\, v \in V_2(\mathbf{G}) \text{ and } n \in \mathbb{Z}^+\},$$

in which n in $(\{u, v\}, n) \in E(\mathbf{G})$ is called a *label*, and the elements in $E(\mathbf{G})$ are called the (*labeled*) *edges in* $\mathbf{G}$.

We say that two vertices u *and* v *are adjacent* (or u *is adjacent to* v) in a labeled bipartite multigraph $\mathbf{G}$ if $(\{u, v\}, n) \in E(\mathbf{G})$ for some label n. Note that if $u, v \in V_1(\mathbf{G})$ or if $u, v \in V_2(\mathbf{G})$, then u and v cannot be adjacent. The notions of *isolated vertices*, *walks*, *paths*, *closed walks*, *cycles*, *acyclicity*, *reachability from one vertex to another*, *connected components* and *connectedness*, respectively, can all be said by analogy to those of graphs. One thing to note, however, is that now closed walks and cycles all have even lengths ≥ 2 and, in particular, the shortest length possible for cycles is 2.

If we change the definition of graphs so that edges are no longer 2-vertex sets but rather pairs of two (not necessarily distinct) vertices, then the resulting objects are called *directed graphs*. It is customary in graph theory to refer to these pairs of vertices in digraphs as *arcs* rather than edges.

Definition 2.10. A *directed graph* (or *digraph* for short) is a pair $\mathbf{G} = (V(\mathbf{G}), E(\mathbf{G}))$, where the *vertex set* $V(\mathbf{G})$ *of* $\mathbf{G}$ is a finite nonempty set of elements, called the *vertices* (or *nodes*) in $\mathbf{G}$, and the *arc set* $E(\mathbf{G})$ *of* $\mathbf{G}$ is a set of pairs of vertices in $V(\mathbf{G})$, called the *arcs* in $\mathbf{G}$.

The graphs introduced in Definition 2.1 are occasionally referred to as *undirected graphs* to distinguish them from directed graphs. Note that a digraph may contain (u, v) but not (v, u) as an arc, and may contain (v, v) as an arc (also called a *loop*). A digraph is called *symmetric* if (v, u) is contained as an arc whenever (u, v) is, and is called *irreflexive* if it contains no loops.

Remark 2.3. In fact, digraphs are legitimate *structures*, a type of objects to be introduced in the next section, while graphs are not. Nevertheless, there is a one-to-one correspondence between all graphs and all irreflexive and symmetric digraphs: For every graph $\mathbf{G} = (V(\mathbf{G}), E(\mathbf{G}))$, the corresponding irreflexive and symmetric digraph

$$\mathbf{G}^{\sigma(\mathrm{G})} = (V(\mathbf{G}^{\sigma(\mathrm{G})}), E(\mathbf{G}^{\sigma(\mathrm{G})}))$$

is defined so that

- $V(\mathbf{G}^{\sigma(\mathrm{G})}) := V(\mathbf{G})$,
- $E(\mathbf{G}^{\sigma(\mathrm{G})}) := \{(u, v) \in V(\mathbf{G}^{\sigma(\mathrm{G})}) \times V(\mathbf{G}^{\sigma(\mathrm{G})}) \mid \{u, v\} \in E(\mathbf{G})\}$.

This one-to-one correspondence $\mathbf{G} \mapsto \mathbf{G}^{\sigma(\mathrm{G})}$ allows us to study (undirected) graphs in the framework of mathematical logic and the symbol $\sigma(\mathrm{G})$ will be named the *vocabulary for graph theory*, more details in the next section.

2.3 Structures

The notion of structures is prevalent in areas such as mathematical logic, abstract algebra and database theory.

Recall that in the previous section, we mentioned that digraphs are an example of structures. Informally, a *structure* is a nonempty set that may be endowed with functions and relations over this set in which some elements may be distinguished (called *constants*). Examples of structures include $(\mathbb{Q}, \leq^{\mathbb{Q}})$ and $(\mathbb{R}, \leq^{\mathbb{R}})$, which are the set $\mathbb{Q}$ of rational numbers and the set $\mathbb{R}$ of real numbers endowed with the (ordering) relations $\leq^{\mathbb{Q}}$ and $\leq^{\mathbb{R}}$ over $\mathbb{Q}$ and $\mathbb{R}$, respectively. The symbol $\leq$ is the "name" shared by both relations. A *relational* *structure* is one that can only be endowed with relations (no functions and constants). Note that $(\mathbb{Q}, \leq^{\mathbb{Q}})$ and $(\mathbb{R}, \leq^{\mathbb{R}})$ are both relational structures. We say a structure is *finite* if the underlying set (i.e., the domain, see Definition 2.12) is finite, and we say it is *infinite* otherwise.

Proviso 1. Throughout this dissertation, we only ever study relational structures and so we shall omit the qualifier "relational" altogether and simply say "structures." Moreover, we mean "finite structures" whenever we say "structures" (without the qualifiers "finite" or "infinite"), i.e., we tacitly assume that structures are by default the finite ones, unless the qualifier "infinite" is stated explicitly.

We now formalize the relevant notions as follows.

Definition 2.11. A *vocabulary* (or *signature*) is a finite set of *relation symbols*, each of which has an associated positive integer called its *arity*.

Throughout this section, unless stated otherwise, we assume a fixed vocabulary $\sigma = \{R_1, \ldots, R_m\}$, and let r_i denote the arity of the relation symbol R_i.

Definition 2.12. A *σ-structure* (or *structure of the vocabulary σ*), finite or infinite, is a tuple $\mathbf{A} = (\mathrm{dom}(\mathbf{A}), R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$, where $\mathrm{dom}(\mathbf{A})$ is a nonempty set called the *domain* (or *universe*) *of* $\mathbf{A}$ and each $R_i^{\mathbf{A}}$ is an r_i-ary relation over $\mathrm{dom}(\mathbf{A})$. The elements in the domain $\mathrm{dom}(\mathbf{A})$ are called the *elements of* $\mathbf{A}$. The *size of* $\mathbf{A}$ refers to $|\mathrm{dom}(\mathbf{A})|$.

In the literature of database theory, a vocabulary σ is more commonly referred to as a *relational schema*, and a σ-structure as a *relational database* (*instance*) *of* σ.

As in Section 2.2, we will assume that classes of σ-structures are *closed under isomorphism*, and we ask the reader to bear with us mentioning these terms informally until we introduce them later on.

Notation. The class of all σ-structures is denoted $\mathcal{A}[\sigma]$.

Example 2.2. The *vocabulary of graph theory* is $\sigma(\mathrm{G}) = \{E\}$, where E is a binary relation symbol (that is, a relation symbol of arity 2). As noted in Remark 2.3, *digraphs are structures*, and in fact they only differ in notations: Every digraph $\mathbf{G} = (V(\mathbf{G}), E(\mathbf{G}))$ is a $\sigma(\mathrm{G})$-structure $\mathbf{A} = (\mathrm{dom}(\mathbf{A}), E^{\mathbf{A}})$, and vice versa, where $\mathrm{dom}(\mathbf{A}) = V(\mathbf{G})$ and $E^{\mathbf{A}} = E(\mathbf{G})$. The class $\mathcal{A}[\sigma(\mathrm{G})]$ of all $\sigma(\mathrm{G})$-structures then is indeed the class of all digraphs. For coherence, we shall henceforth say $\sigma(\mathrm{G})$-structures in place of digraphs.

Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures. We say that $\mathbf{A}$ *is a substructure of* $\mathbf{B}$ or that $\mathbf{B}$ *contains* $\mathbf{A}$ (*as a substructure*) (written: $\mathbf{A} \subseteq \mathbf{B}$) when $\mathrm{dom}(\mathbf{A}) \subseteq \mathrm{dom}(\mathbf{B})$ and $R_i^{\mathbf{A}} \subseteq R_i^{\mathbf{B}}$ for all $i \in [m]$.

Let $\mathbf{A}$ be a σ-structure and S be a nonempty subset of $\mathrm{dom}(\mathbf{A})$. The *substructure of* $\mathbf{A}$ *induced by* S is the σ-structure $\mathbf{A}' \subseteq \mathbf{A}$ such that $\mathrm{dom}(\mathbf{A}') = S$ and $R_i^{\mathbf{A}'} = R_i^{\mathbf{A}} \cap S^{r_i}$ for all $i \in [m]$, which is denoted $\mathbf{A}[S]$.

Definition 2.13. Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures.

(a) The *direct sum of* $\mathbf{A}$ *and* $\mathbf{B}$, denoted $\mathbf{A} \oplus \mathbf{B}$, is the σ-structure with

- domain $\mathrm{dom}(\mathbf{A} \oplus \mathbf{B}) := \mathrm{dom}(\mathbf{A}) \uplus \mathrm{dom}(\mathbf{B})$ and relations $R_i^{\mathbf{A}\oplus\mathbf{B}} := R_i^{\mathbf{A}} \uplus R_i^{\mathbf{B}}$ for all $i \in [m]$ if $\mathrm{dom}(\mathbf{A})$ and $\mathrm{dom}(\mathbf{B})$ are disjoint, otherwise

- domain $\mathrm{dom}(\mathbf{A}\oplus\mathbf{B}) := \{(a,1) \mid a \in \mathrm{dom}(\mathbf{A})\} \cup \{(b,2) \mid b \in \mathrm{dom}(\mathbf{B})\}$ and relations

$$\begin{aligned} R_i^{\mathbf{A}\oplus\mathbf{B}} &:= \{((a_1,1),\ldots,(a_{r_i},1)) \mid (a_1,\ldots,a_{r_i}) \in R_i^{\mathbf{A}}\} \cup \\ &\qquad \{(b_1,2),\ldots,(b_{r_i},2) \mid (b_1,\ldots,b_{r_i}) \in R_i^{\mathbf{B}}\} \end{aligned}$$

  for all $i \in [m]$.

(b) The *direct product of* $\mathbf{A}$ *and* $\mathbf{B}$, denoted $\mathbf{A}\otimes\mathbf{B}$, is the σ-structure with domain $\mathrm{dom}(\mathbf{A}\otimes\mathbf{B}) := \mathrm{dom}(\mathbf{A}) \times \mathrm{dom}(\mathbf{B})$ and relations

$$R_i^{\mathbf{A}\otimes\mathbf{B}} := \{((a_1,b_1),\ldots,(a_{r_i},b_{r_i})) \mid (a_1,\ldots,a_{r_i}) \in R_i^{\mathbf{A}} \text{ and } (b_1,\ldots,b_{r_i}) \in R_i^{\mathbf{B}}\}$$

for all $i \in [m]$.

(c) The $\mathbf{A}$*-th power of* $\mathbf{B}$ (or *the exponential of* $\mathbf{B}$ *to* $\mathbf{A}$), denoted $\mathbf{B}\uparrow\mathbf{A}$, is the σ-structure with domain $\mathrm{dom}(\mathbf{B}\uparrow\mathbf{A}) := \{f \mid f : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B}) \text{ is a function}\}$ and relations

$$\begin{aligned} R_i^{\mathbf{B}\uparrow\mathbf{A}} &:= \{(f_1,\ldots,f_{r_i}) \mid (a_1,\ldots,a_{r_i}) \in R_i^{\mathbf{A}} \text{ implies } (f_1(a_1),\ldots,f_{r_i}(a_{r_i})) \in R_i^{\mathbf{B}} \\ &\qquad \text{for all } a_1,\ldots,a_{r_i} \in \mathrm{dom}(\mathbf{A})\} \end{aligned}$$

for all $i \in [m]$.

Recall the one-to-one correspondence in Remark 2.3, and note that $(\mathbf{G}\oplus\mathbf{H})^{\sigma(\mathrm{G})} = \mathbf{G}^{\sigma(\mathrm{G})}\oplus\mathbf{H}^{\sigma(\mathrm{G})}$ for all graphs $\mathbf{G}$ and $\mathbf{H}$; this explains why we use $\oplus$ uniformly for graphs and for σ-structures. As is for graphs, the binary operation $\oplus$ for σ-structures is commutative and associative *up to isomorphism* (the notion of isomorphism for σ-structures is to be introduced later), which allows us to write $\bigoplus_{i=1}^{n}\mathbf{A}_i$ or write $\bigoplus_n \mathbf{A}$ particularly when $\mathbf{A}_i = \mathbf{A}$ for all $i \in [n]$. The above can be repeated for $\otimes$.

Definition 2.14. Let $\mathbf{A}$ be a σ-structure. A *partition of* $\mathrm{dom}(\mathbf{A})$ is a set $\theta = \{A_1,\ldots,A_n\}$, $n \in \mathbb{Z}^+$, of pairwise disjoint nonempty subsets of $\mathrm{dom}(\mathbf{A})$ such that $\biguplus_{i=1}^{n} A_i = \mathrm{dom}(\mathbf{A})$.

The *quotient of* $\mathbf{A}$ *by* θ is the σ-structure $\mathbf{A}/\theta$ with domain $\mathrm{dom}(\mathbf{A}/\theta) := \theta$ and relations

$$R_i^{\mathbf{A}/\theta} := \{(A_{j_1}, \ldots, A_{j_{r_i}}) \mid (a_{j_1}, \ldots, a_{j_{r_i}}) \in R_i^{\mathbf{A}} \text{ for some } a_{j_1} \in A_{j_1}, \ldots, a_{j_{r_i}} \in A_{j_{r_i}}\}$$

for all $i \in [m]$.

As in Section 2.2, we say a partition θ is *trivial* if every set $A \in \theta$ has size 1; in this case, $\mathbf{A}/\theta$ is isomorphic to $\mathbf{A}$ and thus we set $\mathbf{A}/\theta := \mathbf{A}$ for brevity.

Graph-Theoretic Properties

The following definition allows us to study (some aspects of) σ-structures in the framework of graph theory, specifically from the viewpoint of labeled bipartite multigraphs (see Definition 2.9).

Definition 2.15. Let $\mathbf{A}$ be a σ-structure. The *incidence multigraph of* $\mathbf{A}$ is the labeled bipartite multigraph $\mathrm{Inc}(\mathbf{A}) = (V_1(\mathrm{Inc}(\mathbf{A})), V_2(\mathrm{Inc}(\mathbf{A})), E(\mathrm{Inc}(\mathbf{A})))$ with first vertex set $V_1(\mathrm{Inc}(\mathbf{A})) := \mathrm{dom}(\mathbf{A})$, second vertex set $V_2(\mathrm{Inc}(\mathbf{A})) := \{(R_i, t) \mid i \in [m] \text{ and } t \in R_i^{\mathbf{A}}\}$, and edge set

$$\begin{aligned} E(\mathrm{Inc}(\mathbf{A})) \;\; := \;\; & \{(\{a, (R_i, t)\}, j) \mid a \in V_1(\mathrm{Inc}(\mathbf{A})) \text{ and } (R_i, t) \in V_2(\mathrm{Inc}(\mathbf{A})) \\ & \qquad \text{and } a \text{ is the } j\text{-th entry of } t\}. \end{aligned}$$

With the mapping $\mathbf{A} \mapsto \mathrm{Inc}(\mathbf{A})$ for all σ-structures, the notions of *adjacency between two elements in* $\mathrm{dom}(\mathbf{A})$, *isolated element in* $\mathrm{dom}(\mathbf{A})$, *walks*, *paths*, *closed walks*, *cycles*, *acyclicity*, *distance between two elements*, *diameter of* $\mathbf{A}$, *reachability from one element to another in* $\mathrm{dom}(\mathbf{A})$, *connected components* and *connectedness*, respectively, can all be said of $\mathbf{A}$ in view of the corresponding incidence multigraph $\mathrm{Inc}(\mathbf{A})$. In particular, we say a sequence $w = (a_0, \ldots, a_n)$ of elements $a_0, \ldots, a_n \in \mathrm{dom}(\mathbf{A}) = V_1(\mathrm{Inc}(\mathbf{A}))$ is a *walk* (or *path*, *closed walk*, *cycle*) in $\mathbf{A}$ if $w' = (a_0, b_0, \ldots, a_{n-1}, b_{n-1}, a_n)$ is a walk (or path, closed walk, cycle, respectively) in $\mathrm{Inc}(\mathbf{A})$ for some $b_0, \ldots, b_{n-1} \in V_2(\mathrm{Inc}(\mathbf{A}))$, and we say the *length* of the walk (or path, closed walk, cycle, respectively) w in $\mathbf{A}$ is n, while the length of w' in $\mathrm{Inc}(\mathbf{A})$ is $2n$. Moreover, if (a, b, a) is a closed walk in $\mathrm{Inc}(\mathbf{A})$ with $a \in V_1(\mathrm{Inc}(\mathbf{A}))$ and $b \in V_2(\mathrm{Inc}(\mathbf{A}))$, then we say that (a, a) is a closed walk and also a cycle, both of length 1, in $\mathbf{A}$.

The next example justifies our choice not to identify (undirected) graphs with irreflexive and symmetric $\sigma(\mathrm{G})$-structures, due to the incompatibility of some graph-theoretic properties.

Example 2.3. Consider the graph $\mathbf{K}_2$, with $V(\mathbf{K}_2) = \{v_1, v_2\}$ and $E(\mathbf{K}_2) = \{\{v_1, v_2\}\}$, and its corresponding $\sigma(\mathrm{G})$-structure, $\mathbf{K}_2^{\sigma(\mathrm{G})}$, with $\mathrm{dom}(\mathbf{K}_2^{\sigma(\mathrm{G})}) = \{v_1, v_2\}$ and $E^{\mathbf{K}_2^{\sigma(\mathrm{G})}} = \{(v_1, v_2), (v_2, v_1)\}$. Note that $\mathbf{K}_2$ is acyclic while $\mathbf{K}_2^{\sigma(\mathrm{G})}$ contains a cycle of length 2, i.e., (v_1, v_2, v_1).

Isomorphism

In Section 2.2, we discussed isomorphism among graphs and graph parameters, which are structural properties and hence invariant under isomorphism. The notion of isomorphism can likewise be extended to σ-structures.

Definition 2.16. Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures, finite or infinite. A function $\pi : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$ *is an isomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$ (written: $\pi : \mathbf{A} \cong \mathbf{B}$) if π is a bijection and if for all $i \in [m]$, and for all elements $a_1, \ldots, a_{r_i} \in \mathrm{dom}(\mathbf{A})$, it is true that $(a_1, \ldots, a_{r_i}) \in R_i^{\mathbf{A}}$ if and only if $(\pi(a_1), \ldots, \pi(a_{r_i})) \in R_i^{\mathbf{B}}$. We say that $\mathbf{A}$ *is isomorphic to* $\mathbf{B}$ (written: $\mathbf{A} \cong \mathbf{B}$) if there is an isomorphism from $\mathbf{A}$ to $\mathbf{B}$.

As in Section 2.2, $\cong$ is an equivalence relation among structures of the same vocabulary. For all structures $\mathbf{A}$, the definitions of the *isomorphism type* $[\mathbf{A}]_{\cong}$ *of* $\mathbf{A}$ and an *automorphism* $\pi : \mathbf{A} \cong \mathbf{A}$ *of* $\mathbf{A}$ as well as the notation $\mathrm{aut}(\mathbf{A})$ for the number of automorphisms of $\mathbf{A}$, and for all classes $\mathcal{D}$ of structures, the definitions of the *size* of $\mathcal{D}$ and finiteness of $\mathcal{D}$ as well as the notation $\mathcal{D} = \{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$, can all be repeated by analogy. Throughout this dissertation, most often *we mean* $\mathbf{A} \cong \mathbf{B}$ *when we write* $\mathbf{A} = \mathbf{B}$, and *we mean* $\mathbf{A}$ *is isomorphic to a substructure of* $\mathbf{B}$ *when we write* $\mathbf{A} \subseteq \mathbf{B}$. We also assume that *classes of structures are closed under isomorphism* and, likewise, for classes $\mathcal{D}$ of structures we write

$$\mathcal{D}/\cong \;\; := \;\; \{\mathbf{A} \in \mathcal{D} \mid \mathbf{A} \text{ is the representative of the isomorphism type } [\mathbf{A}]_{\cong}\}$$

for the *set* of structures in $\mathcal{D}$ that are the representatives of their respective isomorphism types, for the same purpose of avoiding redundancy in set descriptions, enumerations, etc.

Remark 2.4. For every two graphs $\mathbf{G}$ and $\mathbf{H}$, and for every function $f : V(\mathbf{G}) \to V(\mathbf{H})$, which indeed is also $f : \mathrm{dom}(\mathbf{G}^{\sigma(\mathrm{G})}) \to \mathrm{dom}(\mathbf{H}^{\sigma(\mathrm{G})})$, we have $f : \mathbf{G} \cong \mathbf{H}$ if and only if $f : \mathbf{G}^{\sigma(\mathrm{G})} \cong \mathbf{H}^{\sigma(\mathrm{G})}$. This is useful when translating results concerning isomorphism among irreflexive and symmetric $\sigma(\mathrm{G})$-structures to graphs.

Due to the fact that two distinct types of objects, namely graphs and structures, are studied in this dissertation, it is hard to give a perfectly uniform discourse for both types of objects in terms of statements that convey essentially the same meaning (see Chapters 3, 4 and 5). We overcome this difficulty by resorting to a versatile symbol, $\mathcal{U}$, for the *universal class* of the objects of the respective type.

Proviso 2. Throughout this dissertation, we let $\mathcal{U}$ denote the class $\mathcal{G}$ or $\mathcal{A}[\sigma]$ for some vocabulary σ (or $\mathcal{A}$ when σ is understood from the context) on various occasions to signify that the statement being made applies (inherently) equally to both graphs and structures.

Chapter 3

Two Frameworks: Logic and Homomorphism Counts

In this chapter, we will introduce the two frameworks, mathematical logic and homomorphism (counts), in which to study graphs and structures. While the investigation of graphs and finite structures through the lens of mathematical logic has been a standard part of a well-established field known as *finite model theory*, the research on them via homomorphism counts is relatively new. The theme of this dissertation is mostly about the latter and about the relation between the two frameworks.

3.1 Mathematical Logic

Recall in Section 2.3 that a vocabulary σ is a finite nonempty set of relation symbols, which are the names of the relations in a σ-structure. The language of a logic consists of common logical symbols and is parameterized by a specific vocabulary.

Throughout this section, we assume a fixed vocabulary $\sigma = \{R_1, \ldots, R_m\}$ for some $m \in \mathbb{Z}^+$, unless otherwise specified.

3.1.1 First-Order Logic and Its Fragments

We shall give a brief introduction to first-order logic, arguably the most common and well-studied logic in mathematics, computer science, philosophy, and so on.

As with any computer programming language, first-order logic is a *formalism* that comprises syntax and semantics. The syntax refers to the grammar of the language, while the semantics gives meanings to grammatical strings of the language.

Syntax

In *automata theory* and the *theory of formal languages*, a *string* is a finite sequence of *characters* from a fixed finite nonempty set called an *alphabet*. We adopt these terms that suit well our purpose of introducing the syntax of first-order logic.

The alphabet for first-order logic parameterized by the vocabulary σ consists of, in addition to the relation symbols from σ, the following list of symbols:

- (*formal*) *variables*: $z_0, z_1, z_2, \ldots$
- (*formal*) *equality*: $=$
- *negation*: $\neg$
- *conjunction*: $\wedge$
- *disjunction*: $\vee$
- *implication*: $\rightarrow$

- *bi-implication*: $\leftrightarrow$
- *universal quantifier*: $\forall$
- *existential quantifier*: $\exists$

and the auxiliary symbols

- *parentheses*: $), ($

to optionally form enclosing parentheses to avoid ambiguity. The symbols $z_0, z_1, z_2, \ldots$ are called (*formal*) *variables* because they are part of first-order logic. In contrast, we use the so-called *meta variables* such as $x, y, z, \ldots$ (for formal variables), $\varphi, \psi, \chi, \ldots$ (for (first-order) σ-formulas, to be defined below), $u, v, w, \ldots$ (for vertices in a graph) and $i, j, k, m, n, p, q, r, s, \ldots$ (for numbers) as placeholders in statements made in English. The *grammar* of a logic formalism (first-order logic or its extensions) consists of a finite number of *formation rules* that either define strings of a particular form,

$$\frac{}{\varphi}$$

or generate new strings from old ones,

$$\frac{\varphi}{\psi} \qquad \text{or} \qquad \frac{\varphi, \psi}{\chi}.$$

Definition 3.1. The formalism of *first-order logic* (*parameterized by* σ), denoted $\mathrm{FO}[\sigma]$, is the set of strings, called *first-order σ-formulas*, over the alphabet

$$\sigma \cup \{z_0, z_1, z_2, \ldots, =, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists,), (\}$$

whose grammar consists of the formation rules:

$$\frac{}{z_i = z_j} \text{ where } i, j \in \mathbb{N}; \qquad \frac{}{R_i(z_{j_1}, \ldots, z_{j_{r_i}})} \text{ where } i \in [m] \text{ and } j_1, \ldots, j_{r_i} \in \mathbb{N};$$

$$\frac{\varphi}{\neg\varphi}; \quad \frac{\varphi, \psi}{\varphi \wedge \psi}; \quad \frac{\varphi, \psi}{\varphi \vee \psi}; \quad \frac{\varphi, \psi}{\varphi \to \psi}; \quad \frac{\varphi, \psi}{\varphi \leftrightarrow \psi};$$

$$\frac{\varphi}{\forall z_i \varphi} \text{ where } i \in \mathbb{N}; \qquad \frac{\varphi}{\exists z_i \varphi} \text{ where } i \in \mathbb{N}.$$

In the above definition, the formation rules in the first row give rise to *equational formulas* (on the left) and *relational formulas* (on the right), which are collectively called *atomic formulas* (or *atoms*). The formation rules in the second row are for *connectives* (i.e., $\neg, \wedge, \vee, \to, \leftrightarrow$). The first-order σ-formula $\neg\varphi$ is called the *negation* of φ, while the first-order σ-formulas $\varphi \wedge \psi$ and $\varphi \vee \psi$ are called the *conjunction* and the *disjunction* of φ and ψ, respectively, and $\varphi \to \psi$ is called the *implication* of ψ from φ, while $\varphi \leftrightarrow \psi$ is called the *bi-implication* of φ and ψ. Finally, the formation rules in the third row involve the two quantifiers $\forall, \exists$, and the first-order σ-formulas $\forall z_i \varphi$ and $\exists z_i \varphi$ are the *universal quantification* and the *existential quantification* of φ in z_i, respectively.

Occasionally, we write $\overline{x}$ for the sequence of meta variables $x_1, \ldots, x_n$ and $\overline{z}$ for the sequence of formal variables $z_1, \ldots, z_n$ when the length n is understood or irrelevant; we also write $\forall \overline{x} \varphi$ for $\forall x_1 \cdots \forall x_n \varphi$ and $\exists \overline{z} \psi$ for $\exists z_1 \cdots \exists z_n \psi$, etc.

We let FO$[\sigma]$*-formulas* abbreviate “first-order σ-formulas” and let FO abbreviate “first-order logic” for succinctness. Moreover, when the underlying vocabulary σ is understood or irrelevant, we often drop the reference to σ in the terminology; in other words, we will often use the terms or notations such as *structures*, $\mathcal{A}$, *first-order formulas*, FO-*formulas* and FO in place of "σ-structures", "$\mathcal{A}[\sigma]$", "first-order σ-formulas", "FO[σ]-formulas" and "FO[σ]", respectively. Moreover, when the underlying logic is known to be first-order logic, we often simply say *formulas*.

Oftentimes concepts defined for a logic formalism and the proofs of theorems about it are given by induction that reflects its grammar, and first-order logic is no exception. An example is given in the definition below. An occurrence of a variable x in a formula φ is *bound* if it is quantified by an occurrence of "$\exists x$" or "$\forall x$" in φ, and is *free* otherwise; x is a *free variable in* φ if it has a free occurrence in φ, and is a *bound variable in* φ otherwise.

Definition 3.2. The set of free variables in a FO-formula φ, denoted free(φ), is defined inductively as follows:

$$
\begin{array}{lcl}
\mathrm{free}(x = y) & := & \{x, y\}, \\
\mathrm{free}(R_i(x_1, \ldots, x_{r_i})) & := & \{x_1, \ldots, x_{r_i}\}, \\
\mathrm{free}(\neg\varphi) & := & \mathrm{free}(\varphi), \\
\mathrm{free}(\varphi \wedge \psi) & := & \mathrm{free}(\varphi) \cup \mathrm{free}(\psi), \\
\mathrm{free}(\varphi \vee \psi) & := & \mathrm{free}(\varphi) \cup \mathrm{free}(\psi), \\
\mathrm{free}(\varphi \to \psi) & := & \mathrm{free}(\varphi) \cup \mathrm{free}(\psi), \\
\mathrm{free}(\varphi \leftrightarrow \psi) & := & \mathrm{free}(\varphi) \cup \mathrm{free}(\psi), \\
\mathrm{free}(\forall x\varphi) & := & \mathrm{free}(\varphi) \setminus \{x\}, \\
\mathrm{free}(\exists x\varphi) & := & \mathrm{free}(\varphi) \setminus \{x\}.
\end{array}
$$

We say φ is a (FO-)*sentence* if free(φ) $= \emptyset$.

Example 3.1. The following two are first-order σ(G)-sentences known as the *axioms of* *graph theory*:

$$\begin{aligned}\varphi_{\text{irreflx}} &:= \forall x \neg E(x,x), \\ \varphi_{\text{sym}} &:= \forall x \forall y (E(x,y) \to E(y,x)).\end{aligned}$$

Semantics

Let $\mathbf{A}$ be a structure. A (*variable*) *assignment* for $\mathbf{A}$ is a function

$$\alpha : \{z_0, z_1, z_2, \ldots\} \to \operatorname{dom}(\mathbf{A}).$$

Given an element $a \in \operatorname{dom}(\mathbf{A})$, the assignment

$$\alpha(z_i/a) : \{z_0, z_1, z_2, \ldots\} \to \operatorname{dom}(\mathbf{A})$$

satisfies $(\alpha(z_i/a))(z_i) = a$ and otherwise agrees with α.

Now we are ready to state formally one of the most important notions in mathematical logic, the concept of *truth in a structure*. For a pair $(\mathbf{A}, \alpha)$ in which $\mathbf{A}$ is a σ-structure and α is an assignment for $\mathbf{A}$, and for a first-order σ-formula φ, it can be determined by induction on φ whether φ is a truth in $(\mathbf{A}, \alpha)$. This is known as *Tarski semantics*.

Definition 3.3. Let $\mathbf{A}$ be a σ-structure, and let α be an assignment for $\mathbf{A}$. For all first-order σ-formulas, the notion that φ *is true in* $(\mathbf{A}, \alpha)$ or that $(\mathbf{A}, \alpha)$ *satisfies* φ or that $(\mathbf{A}, \alpha)$ *is a model of* φ, denoted $(\mathbf{A}, \alpha) \models \varphi$, is defined inductively on φ as follows:

$$
\begin{array}{lll}
(\mathbf{A},\alpha) \models x = y & \text{:if} & \alpha(x) = \alpha(y), \\
(\mathbf{A},\alpha) \models R_i(x_1,\ldots,x_{r_i}) & \text{:if} & (\alpha(x_1),\ldots,\alpha(x_{r_i})) \in R_i^{\mathbf{A}}, \\
(\mathbf{A},\alpha) \models \neg\varphi & \text{:if} & \text{not } (\mathbf{A},\alpha) \models \varphi, \\
(\mathbf{A},\alpha) \models (\varphi \vee \psi) & \text{:if} & (\mathbf{A},\alpha) \models \varphi \text{ or } (\mathbf{A},\alpha) \models \psi, \\
(\mathbf{A},\alpha) \models (\varphi \wedge \psi) & \text{:if} & (\mathbf{A},\alpha) \models \varphi \text{ and } (\mathbf{A},\alpha) \models \psi, \\
(\mathbf{A},\alpha) \models (\varphi \to \psi) & \text{:if} & (\mathbf{A},\alpha) \models \psi \text{ when } (\mathbf{A},\alpha) \models \varphi, \\
(\mathbf{A},\alpha) \models (\varphi \leftrightarrow \psi) & \text{:if} & (\mathbf{A},\alpha) \models \psi \text{ precisely when } (\mathbf{A},\alpha) \models \varphi, \\
(\mathbf{A},\alpha) \models \forall x\varphi & \text{:if} & (\mathbf{A},\alpha(x/a)) \models \varphi \text{ for every element } a \in \mathrm{dom}(\mathbf{A}), \\
(\mathbf{A},\alpha) \models \exists x\varphi & \text{:if} & (\mathbf{A},\alpha(x/a)) \models \varphi \text{ for some element } a \in \mathrm{dom}(\mathbf{A}).
\end{array}
$$

We say that two formulas φ and ψ are *logically equivalent* when it holds that $(\mathbf{A},\alpha) \models \varphi$ if and only if $(\mathbf{A},\alpha) \models \psi$ (or, equivalently, $(\mathbf{A},\alpha) \models (\varphi \leftrightarrow \psi)$) for every structure $\mathbf{A}$ and every assignment α for $\mathbf{A}$. In fact, we have the following list of pairs of logically equivalent formulas:

(1) $\varphi \wedge \psi$ and $\neg(\neg\varphi \vee \neg\psi)$, for all formulas φ and ψ,

(2) $\varphi \vee \psi$ and $\neg(\neg\varphi \wedge \neg\psi)$, for all formulas φ and ψ,

(3) $\varphi \to \psi$ and $\neg\varphi \vee \psi$, for all formulas φ and ψ,

(4) $\varphi \leftrightarrow \psi$ and $\neg(\neg(\neg\varphi \vee \psi) \vee \neg(\neg\psi \vee \varphi))$, for all formulas φ and ψ,

(5) $\forall x\varphi$ and $\neg\exists x\neg\varphi$, for all formulas φ,

(6) $\exists x\varphi$ and $\neg\forall x\neg\varphi$, for all formulas φ.

The above (1) and (2) are known as *De Morgan's Laws*. Due to (1), (3), (4) and (5), we shall view $\varphi \wedge \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$ and $\forall x\varphi$ as abbreviations for $\neg(\neg\varphi \vee \neg\psi)$, $\neg\varphi \vee \psi$,

$\neg(\neg(\neg\varphi\vee\psi)\vee\neg(\neg\psi\vee\varphi))$ and $\neg\exists x\neg\varphi$, respectively, and regard the respective formation rules as redundant ones *when proving properties of all formulas by induction on the formation rules*, because it results in a more succinct proof.

Lemma 3.1 (Coincidence Lemma)**.** *For every set V of variables, every first-order formula φ with $\mathrm{free}(\varphi) \subseteq V$, every structure $\mathbf{A}$ and every two assignments α_1 and α_2 for $\mathbf{A}$ that agree on their values of the variables in V, we have $(\mathbf{A}, \alpha_1) \models \varphi$ if and only if $(\mathbf{A}, \alpha_2) \models \varphi$.* $\square$

This lemma is also known as *Relevance Lemma* and has a straightforward induction proof on φ (omitted), and it can be generalized to any reasonable logic formalism developed in the area of mathematical logic. Due to this lemma, it is customary to write $\varphi(x_1, \ldots, x_n)$ for φ when $\mathrm{free}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ to emphasize that the variables $x_1, \ldots, x_n$ behave as parameters in the satisfaction relation $\models$. Thus, for variables $y_1, \ldots, y_n$, we write $\varphi(y_1, \ldots, y_n)$ for the formula obtained from φ by substituting the occurrences of y_i for the *free* occurrences of x_i, for all $i \in [n]$ (formally introduced in Definition 3.4). The formula $\varphi(y_1, \ldots, y_n)$ asserts for $y_1, \ldots, y_n$ what $\varphi(x_1, \ldots, x_n)$ asserts for $x_1, \ldots, x_n$.

For all formulas $\varphi = \varphi(x_1, \ldots, x_n)$, all structures $\mathbf{A}$ and all elements $a_1, \ldots, a_n \in \mathrm{dom}(\mathbf{A})$, we shall write $\mathbf{A} \models \varphi(a_1, \ldots, a_n)$ to mean $(\mathbf{A}, \alpha) \models \varphi$ for an arbitrary assignment α for $\mathbf{A}$ such that $\alpha(x_1) = a_1, \ldots, \alpha(x_n) = a_n$. In particular, if φ is a sentence, i.e., if $\mathrm{free}(\varphi) = \emptyset$, then we can omit the reference to any assigned values entirely.

Notation. Let $\mathbf{A}$ be a σ-structure and φ be a first-order σ-sentence. We write $\mathbf{A} \models \varphi$ if $(\mathbf{A}, \alpha) \models \varphi$ for an arbitrary assignment α for $\mathbf{A}$.

Next, we delve deeper into the issue of free variables as parameters in a formula. As stated previously, we can substitute the occurrences of a variable y for the free occurrences of a variable x in a formula φ to obtain another formula ψ so that ψ asserts about y what φ asserts about x. However, we need to exercise caution in some situations, detailed below.

Definition 3.4. Let φ be a FO[σ]-formula, and let x and y refer to formal variables. The *substitution of* y *for* x *in* φ, denoted $\varphi(x/y)$, is a formula defined in two cases below:

(a) If $x \notin \text{free}(\varphi)$ or if y and x refer to the same formal variable, then $\varphi(x/y) := \varphi$.

(b) Otherwise, $\varphi(x/y)$ is defined inductively:

$(x_1 = x_2)(x/y)$	$:=$	$(y_1 = y_2)$, where y_j and y refer to the same formal variable if x_j and x do, otherwise y_j and x_j refer to the same formal variable,
$(R_i(x_1, \ldots, x_{r_i}))(x/y)$	$:=$	$R_i(y_1, \ldots, y_{r_i})$, where y_j and y refer to the same formal variable if x_j and x do, otherwise y_j and x_j refer to the same formal variable,
$(\neg\varphi)(x/y)$	$:=$	$\neg(\varphi(x/y))$,
$(\varphi \vee \psi)(x/y)$	$:=$	$(\varphi(x/y)) \vee (\psi(x/y))$,
$(\exists z\varphi)(x/y)$	$:=$	$\exists u(\psi(x/y))$, where u refers to the first formal variable among $z_0, z_1, \ldots$ that does not appear in $\exists z\varphi$, and $\psi := \varphi(z/u)$, if y and z refer to the same formal variable, otherwise u and z refer to the same formal variable, and $\psi := \varphi$.

Example 3.2. In Example 2.2, we have seen clearly that digraphs are exactly the $\sigma(\mathrm{G})$-structures. Moreover, for the two FO[$\sigma(\mathrm{G})$]-sentences φ_{irreflx} and φ_{sym} (axioms of graph theory) introduced in Example 3.1, we have, for every $\sigma(\mathrm{G})$-structure $\mathbf{A}$, that

- $\mathbf{A} \models \varphi_{\text{irreflx}}$ if and only if $\mathbf{A}$ is irreflexive,
- $\mathbf{A} \models \varphi_{\text{sym}}$ if and only if $\mathbf{A}$ is symmetric.

Therefore, for every $\sigma(\mathrm{G})$-structure $\mathbf{A}$, it holds that $\mathbf{A}$ satisfies the axioms of graph theory precisely when $\mathbf{A}$ corresponds to a graph in the one-to-one correspondence $\mathbf{G} \mapsto \mathbf{G}^{\sigma(\mathrm{G})}$ introduced in Remark 2.3.

Although we distinguish between a graph $\mathbf{G}$ and its corresponding $\sigma(\mathrm{G})$-structure $\mathbf{G}^{\sigma(\mathrm{G})}$ due to a compatibility issue with graph-theoretic properties (see Example 2.3), it is standard to identify $\mathbf{G}$ with $\mathbf{G}^{\sigma(\mathrm{G})}$ when defining the satisfaction relation $\models$ for graphs.

Definition 3.5. Let $\mathbf{G}$ be a graph. A (*variable*) *assignment* for $\mathbf{G}$ is a function $\alpha : \{z_0, z_1, z_2, \ldots\} \to V(\mathbf{G})$, which is also a (variable) assignment for $\mathbf{G}^{\sigma(\mathrm{G})}$, and for every first-order $\sigma(\mathrm{G})$-formula φ, we write $(\mathbf{G}, \alpha) \models \varphi$ if $(\mathbf{G}^{\sigma(\mathrm{G})}, \alpha) \models \varphi$.

The Coincidence Lemma 3.1 also holds for graphs with the one-to-one correspondence $\mathbf{G} \mapsto \mathbf{G}^{\sigma(\mathrm{G})}$. Hence, it justifies the simpler notations $\mathbf{G} \models \varphi(v_1, \ldots, v_n)$ and $\mathbf{G} \models \varphi$.

Definition 3.6. Let $\mathbf{G}$ be a graph and let φ be a first-order $\sigma(\mathrm{G})$-formula.

(a) If $\mathrm{free}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ and if α is an assignment for $\mathbf{G}$ with $\alpha(x_1) = v_1, \ldots, \alpha(x_n) = v_n$, then we write $\mathbf{G} \models \varphi(v_1, \ldots, v_n)$ for $(\mathbf{G}, \alpha) \models \varphi$.

(b) If $\mathrm{free}(\varphi) = \emptyset$, then we write $\mathbf{G} \models \varphi$ for $(\mathbf{G}, \alpha) \models \varphi$ where α is an arbitrary assignment for $\mathbf{G}$.

Characterizing Isomorphism in First-Order Logic

One of the central topics in finite model theory is the issue of identifying an equivalence relation $\equiv$ with the indistinguishability by the sentences in a logic formalism (namely the equivalence relation among graphs or among structures that arises from satisfying the same sentences in a logic formalism), for which we say to *characterize* $\equiv$ *in a logic formalism*.

Recall the versatile symbol $\mathcal{U}$ introduced in Proviso 2 for the class $\mathcal{G}$ of graphs or the class $\mathcal{A}$ of structures.

Definition 3.7. Let $\mathbf{A}, \mathbf{B} \in \mathcal{U}$. We say $\mathbf{A}$ *is first-order equivalent to* $\mathbf{B}$ or $\mathbf{A}$ *is indistinguishable from* $\mathbf{B}$ *in first-order logic* (written: $\mathbf{A} \equiv_{\mathrm{FO}} \mathbf{B}$) if for every first-order sentence φ, it holds that $\mathbf{A} \models \varphi$ if and only $\mathbf{B} \models \varphi$.

The above definition of first-order equivalence extends straightforwardly to infinite structures, and it is also known as *elementary equivalence* in the literature of mathematical logic. It turns out that first-order equivalence is coarser than isomorphism.

Lemma 3.2 (Isomorphism Lemma)**.** *For all structures* $\mathbf{A}$ *and* $\mathbf{B}$*, finite or infinite, if* $\mathbf{A} \cong \mathbf{B}$*, then* $\mathbf{A} \equiv_{\mathrm{FO}} \mathbf{B}$*.* $\square$

This lemma follows from a straightforward induction proof on *formulas* (omitted) for given isomorphic structures and a given variable assignment, and it can be generalized to any reasonable logic formalism developed in the area of mathematical logic.

An immediate consequence of Lemma 3.2 is that for $\sigma(\mathrm{G})$-structures $\mathbf{A}$ and $\mathbf{B}$, if $\mathbf{A}$ satisfies the axioms of graph theory (see Example 3.1) and if $\mathbf{A} \cong \mathbf{B}$, then $\mathbf{B}$ also satisfies the axioms of graph theory. Thus, *for every graph* $\mathbf{G}$*, we have* $[\mathbf{G}^{\sigma(\mathrm{G})}]_{\cong} = \{\mathbf{H}^{\sigma(\mathrm{G})} \mid \mathbf{H} \in [\mathbf{G}]_{\cong}\}$, a fact already implicit in Remark 2.4 which, in view of Definitions 3.5 and 3.6, implies the corresponding version of Isomorphism Lemma 3.2 for graphs.

Corollary 3.3. *For all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, if* $\mathbf{G} \cong \mathbf{H}$*, then* $\mathbf{G} \equiv_{\mathrm{FO}} \mathbf{H}$*.* □

Remark 3.1. Lemma 3.2 can be recast as that *satisfaction of first-order sentences by structures (or by graphs) is invariant under isomorphism* and hence depends only on the isomorphism type of structures (or graphs, respectively) rather than the individual ones themselves.

For graphs or finite structures, it turns out that first-order equivalence is also finer than isomorphism, i.e., the converse of Lemma 3.2 also holds, as stated formally below.

Proposition 3.4 (Characterization of Isomorphism in First-Order Logic)**.** *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$*, we have* $\mathbf{A} \cong \mathbf{B}$ *if and only if* $\mathbf{A} \equiv_{\mathrm{FO}} \mathbf{B}$*.* □

In other words, *isomorphism coincides with first-order equivalence among graphs or among finite structures.* This proposition is a direct consequence of the next, which says that for every graph or finite structure $\mathbf{A}$, there is a first-order sentence of which $\mathbf{A}$ is the only model up to isomorphism or, more concisely, *graphs or finite structures are characterized in first-order logic up to isomorphism.*

Proposition 3.5 (Characterization of Graphs or Finite Structures in First-Order Logic Up to Isomorphism)**.** *For every* $\mathbf{A}$ *in* $\mathcal{U}$*, there is a first-order sentence* $\chi^{\mathbf{A}}$ *such that for every* $\mathbf{B}$ *in* $\mathcal{U}$*, we have* $\mathbf{A} \cong \mathbf{B}$ *if and only if* $\mathbf{B} \models \chi^{\mathbf{A}}$*.*

Proof. We first consider $\mathcal{U} = \mathcal{A}$. Let $\mathbf{A}$ be a structure whose domain is $\mathrm{dom}(\mathbf{A}) = \{a_1, \ldots, a_n\}$. We let $\chi^{\mathbf{A}}$ formalize how a structure $\mathbf{B}$ isomorphic to $\mathbf{A}$ should look like:

There are elements $b_1, \ldots, b_n$ in the domain, $b_1, \ldots, b_n$ are related or unrelated (i.e., pairwise distinct and whether or not in a relation) as $a_1, \ldots, a_n$ are in $\mathbf{A}$, and $b_1, \ldots, b_n$ are the only elements in the domain, i.e., for every element b_{n+1}, it must be among $b_1, \ldots, b_n$. It is clear that for every structure $\mathbf{B}$, it satisfies this description precisely when it is isomorphic to $\mathbf{A}$. In the formalization, let the variables $z_1, \ldots, z_n$ play the role of $b_1, \ldots, b_n$, respectively:

$$
\begin{aligned}
\chi^{\mathbf{A}} \quad := \quad & \exists z_1 \cdots \exists z_n (\\
& \bigwedge \{\varphi(z_1, \ldots, z_n) \mid \varphi \text{ is atomic and } \mathbf{A} \models \varphi(a_1, \ldots, a_n)\} \wedge \\
& \bigwedge \{\neg\varphi(z_1, \ldots, z_n) \mid \varphi \text{ is atomic and not } \mathbf{A} \models \varphi(a_1, \ldots, a_n)\} \wedge \\
& \forall z_{n+1} \bigvee_{1 \leq i \leq n} z_{n+1} = z_i).
\end{aligned}
$$

Now, for $\mathcal{U} = \mathcal{G}$ and for every graph $\mathbf{A}$, we apply the version of this proposition for $\sigma(\mathrm{G})$-structures to $\mathbf{A}^{\sigma(\mathrm{G})}$ to obtain $\chi^{\mathbf{A}^{\sigma(\mathrm{G})}}$ and set $\chi^{\mathbf{A}} := \chi^{\mathbf{A}^{\sigma(\mathrm{G})}}$. By Remark 2.4, the version of this proposition for graphs immediately follows. $\square$

One thing to note is that, as opposed to finite structures, while Isomorphism Lemma 3.2 holds also in infinite structures, its converse in general does not; for example, $(\mathbb{Q}, \leq^{\mathbb{Q}})$ and $(\mathbb{R}, \leq^{\mathbb{R}})$ are first-order equivalent but not isomorphic. This fact is known to be a consequence of *Löwenheim-Skolem-Tarski Theorem*, which in turn is a consequence of *Compactness Theorem* (for first-order logic).

Remark 3.2. Using the notation of indicator function (see Section 2.1 for its definition), every graph or finite structure $\mathbf{A}$ gives rise to the *bit-sequence* (i.e., a sequence whose entries are 0 or 1):

$$\mathrm{FO}(\mathbf{A}) := (\mathrm{I}(\mathbf{A} \models \varphi) \mid \varphi \text{ is a first-order sentence}).$$

We can then restate Proposition 3.4: *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$*, we have* $\mathbf{A} \cong \mathbf{B}$ *if and only if* $\mathrm{FO}(\mathbf{A}) = \mathrm{FO}(\mathbf{B})$.

If we enumerate the isomorphism types $[\mathbf{A}_1]_{\cong}, [\mathbf{A}_2]_{\cong}, \ldots$ of graphs or of finite structures vertically and first-order sentences $\varphi_1, \varphi_2, \ldots$ horizontally in the following visualization of the mapping $\mathrm{I}(* \models *)$ as a 2-dimensional infinite matrix

$$\begin{array}{c|cccccc}
\mathrm{I}(* \models *) & \varphi_1 & \varphi_2 & \cdots & \varphi_j & \cdots \\
\hline
[\mathbf{A}_1]_{\cong} & \mathrm{I}(\mathbf{A}_1 \models \varphi_1) & \mathrm{I}(\mathbf{A}_1 \models \varphi_2) & \cdots & \mathrm{I}(\mathbf{A}_1 \models \varphi_j) & \cdots \\
[\mathbf{A}_2]_{\cong} & \mathrm{I}(\mathbf{A}_2 \models \varphi_1) & \mathrm{I}(\mathbf{A}_2 \models \varphi_2) & \cdots & \mathrm{I}(\mathbf{A}_2 \models \varphi_j) & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\
[\mathbf{A}_i]_{\cong} & \mathrm{I}(\mathbf{A}_i \models \varphi_1) & \mathrm{I}(\mathbf{A}_i \models \varphi_2) & \cdots & \mathrm{I}(\mathbf{A}_i \models \varphi_j) & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots
\end{array}$$

then the above restatement of Proposition 3.4 says that *every row in the matrix is unique.*

Defining Classes of Graphs or of Structures in First-Order Logic

Another central topic in finite model theory is the issue of defining a class of graphs or of structures in terms of the satisfaction of a sentence in a logic formalism. Recall the class $\mathcal{U}$ from Proviso 2.

Definition 3.8. (a) For every first-order sentence φ, the *model class of* φ is the class $\mathrm{Mod}(\varphi) := \{\mathbf{A} \in \mathcal{U} \mid \mathbf{A} \models \varphi\}$.

(b) Let $\mathcal{D} \subseteq \mathcal{U}$ be a class. We say that $\mathcal{D}$ is *definable by a first-order sentence* φ if $\mathcal{D} = \mathrm{Mod}(\varphi)$, and that $\mathcal{D}$ is *definable in first-order logic* (or FO-*definable* for short) if such a sentence exists.

By Proposition 3.5, it holds that the (equivalence) class $[\mathbf{A}]_{\cong}$ is definable in first-order logic since $[\mathbf{A}]_{\cong} = \mathrm{Mod}(\chi^{\mathbf{A}})$, for every graph or structure $\mathbf{A}$. Moreover, as the following example shows, the class of all graphs or all structures of a bounded size or a given size is definable in first-order logic. The sentences in the example are derived in a similar way as the one derived in the proof of Proposition 3.5.

Example 3.3. Let $n \in \mathbb{Z}^{+}$. The class $\{\mathbf{A} \in \mathcal{U} \mid |\mathrm{dom}(\mathbf{A})| \leq n\}$ is equal to $\mathrm{Mod}(\varphi_{\leq n})$, where

$$\varphi_{\leq n} := \exists z_1 \cdots \exists z_n \forall z_{n+1} \bigvee_{1 \leq i \leq n} z_{n+1} = z_i.$$

Moreover, the class $\{\mathbf{A} \in \mathcal{U} \mid |\mathrm{dom}(\mathbf{A})| = n\}$ is equal to $\mathrm{Mod}(\varphi_{=n})$, where

$$\varphi_{=n} := \exists z_1 \cdots \exists z_n ((\bigwedge_{1 \leq i < j \leq n} \neg z_i = z_j) \wedge \forall z_{n+1} \bigvee_{1 \leq i \leq n} z_{n+1} = z_i).$$

Note that $\varphi_{=1}$ coincides with $\varphi_{\leq 1}$ (since the vertex set of a graph or the domain of a structure is nonempty), which is logically equivalent to $\forall z_1 \forall z_2\ z_1 = z_2$ and that for $n \geq 2$, we have that $\varphi_{=n}$ is logically equivalent to $(\varphi_{\leq n} \wedge \neg \varphi_{\leq n-1})$.

Next, we consider examples of graphs, referring to the one-to-one correspondence between graphs and irreflexive and symmetric $\sigma(\mathrm{G})$-structures in Remark 2.3. The examples below illustrate that the class of graphs containing a specific graph as a subgraph are definable in first-order logic.

Example 3.4. For every graph $\mathbf{G}$, the binary relation $E^{\mathbf{G}}$ in $\mathbf{G}^{\sigma(\mathrm{G})}$ is symmetric. Thus, to assert that $\mathbf{G}$ contains the triangle $\mathbf{K}_3$ as a subgraph, it suffices to formalize it as $\varphi_{\Delta} := \exists z_0 \exists z_1 \exists z_2 (E(z_0, z_1) \wedge E(z_1, z_2) \wedge E(z_2, z_0))$. Therefore, $\{\mathbf{G} \in \mathcal{G} \mid \mathbf{K}_3 \subseteq \mathbf{G}\} = \mathrm{Mod}(\varphi_{\Delta})$.

Example 3.5. The class $\mathcal{I}$ of independent sets introduced in Section 2.2 consists of graphs that do not contain $\mathbf{K}_2$ as a subgraph. By a direct formalization, this class is definable by the first-order sentence $\neg\exists z_0 \exists z_1\, E(z_0, z_1)$.

Ehrenfeucht-Fraïssé Games and Fragments of First-Order Logic

The *model-checking problem for first-order logic* is the decision problem: Given a structure $\mathbf{A}$ and a first-order sentence φ, does $\mathbf{A}$ satisfy φ? Typically, it is assumed that φ is in *prenex normal form*, in which all quantifications, universal or existential, altogether appear nested at the front of φ, collectively called the *prefix*, while the remaining (quantifier-free) part of φ is sometimes called the *matrix* of φ.

As one can imagine, an intuitive algorithm for such a problem checks the matrix against all suitable variable assignments (and runs in loops, say). Of course, only the fraction of an assignment that is relevant to the variables in φ is considered, by Coincidence Lemma 3.1. However, if there are a total of n nested quantifiers, equivalently, if the prefix contains n (not necessarily distinct) variables, then the total number of all suitable variable assignments is $O(|\mathrm{dom}(\mathbf{A})|^n)$. These nested quantifiers and variables take up computational resources – time and space, respectively (e.g., the time required for such an intuitive algorithm grows exponentially in n).

This brings up an optimization issue: Is it possible to reduce the number of (nested) quantifiers or variables in φ? More precisely, is there a first-order sentence ψ logically equivalent to φ with a fewer number of (nested) quantifiers or variables? Regrettably, the answer is "no" in general, although in some special cases of φ such a ψ exists and more efficient algorithms can be devised.

Definition 3.9. The *quantifier rank* of a first-order σ-formula φ, denoted $\mathrm{qr}(\varphi)$, is defined inductively as follows:

$$\begin{aligned}
\mathrm{qr}(x = y) &:= 0, \\
\mathrm{qr}(R_i(x_1, \ldots, x_{r_i})) &:= 0, \\
\mathrm{qr}(\neg\varphi) &:= \mathrm{qr}(\varphi), \\
\mathrm{qr}(\varphi \vee \psi) &:= \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\}, \\
\mathrm{qr}(\exists x \varphi) &:= \mathrm{qr}(\varphi) + 1.
\end{aligned}$$

Definition 3.10. Let $n \in \mathbb{Z}^+$. The *fragment* of FO that consists of σ-formulas φ for which $\mathrm{qr}(\varphi) \leq n$ is denoted $\mathrm{FO}_n[\sigma]$, or FO_n when the vocabulary σ is understood from the context.

The sentence $\chi^{\mathbf{A}}$ given in the proof of Proposition 3.5 characterizes $\mathbf{A}$ up to isomorphism. When $|\mathrm{dom}(\mathbf{A})| = n$, we have $\chi^{\mathbf{A}} \in \mathrm{FO}_{n+1}$ (although it is not in prenex normal form due to the position of the universal quantifier "$\forall$"). We will argue that this sentence is in general optimal in terms of the number of nested quantifiers in it.

Definition 3.11. Let L be a logic formalism, and let φ be an L-formula. For every graph $\mathbf{G}$ and every assignment α for $\mathbf{G}$, we write $(\mathbf{G}, \alpha) \models \varphi$ if $(\mathbf{G}^{\sigma(\mathrm{G})}, \alpha) \models \varphi$. In particular, when φ is an L-sentence, then we write $\mathbf{G} \models \varphi$ if $\mathbf{G}^{\sigma(\mathrm{G})} \models \varphi$.

The *expressive power* of a logic formalism L refers to its ability for us to formalize in it a concept or notion in English as an L-sentence φ such that for every structure $\mathbf{A}$, the given description in English fits $\mathbf{A}$ if and only if $\mathbf{A} \models \varphi$. In fact, we have already done a formalization in the proof of Proposition 3.5 and in Example 3.4.

The measure of the expressive power of a logic formalism and the comparison of logic formalisms in terms of expressive power is one central topic in a subarea of mathematical logic known as (*classical*) *model theory* and also in finite model theory. There have been tools developed for this purpose in the former – e.g., Compactness Theorem and the method of ultraproducts – and while they found important applications predominantly in axiomatizability of classes of general structures (mainly infinite ones) there, they do not turn out to be as successful in the latter, where the focus is primarily on finite structures. That being said, one tool survives in both theories – the method of *Ehrenfeucht-Fraïssé games* – and we introduce it next.

Let $n \in \mathbb{Z}^+$, and let $\mathbf{A}$ and $\mathbf{B}$ be structures. The Ehrenfeucht-Fraïssé game $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$ of n moves parameterized by $\mathbf{A}$ and $\mathbf{B}$ consists of:

- a "chessboard" with the structures $\mathbf{A}$ and $\mathbf{B}$ on it called the *board*,
- infinitely many pairs of "pebbles" $\alpha_0, \beta_0, \alpha_1, \beta_1, \ldots$ (although, obviously, finitely many pairs of pebbles will be used during every round of the game since the number of moves is limited to n for each player),

and is played by two players, the *spoiler* and the *duplicator*. The pebbles $\alpha_0, \alpha_1, \ldots$ are to be placed on elements in $\mathbf{A}$, and similarly the pebbles $\beta_0, \beta_1, \ldots$ on elements in $\mathbf{B}$. Initially, the pebbles are all off the board. During a *round* of the game, multiple pebbles can be put on the same elements. The two players take turn in making moves, the spoiler first. In the k-th move, the spoiler chooses a structure $\mathbf{A}$ or $\mathbf{B}$, then picks a pebble α_j or β_j (on or off the board) that matches the structure chosen, and places it onto an element; the duplicator responds by picking the other pebble β_j or α_j (on or off the board), respectively, and places it onto an element of the opposite structure. After the k-th move, let $a_1, \ldots, a_p$ be the elements in $\mathbf{A}$ (not necessarily distinct) and $b_1, \ldots, b_p$ be the elements in $\mathbf{B}$ (not necessarily distinct) on which a pebble is placed such that α_j is placed on a_i if and only if β_j is placed on b_i. If the mapping $\pi : \{a_1, \ldots, a_p\} \to \{b_1, \ldots, b_p\}$ with $\pi(a_i) = \pi(b_i)$ for $i \in [p]$ is an isomorphism $\pi : \mathbf{A}[\{a_1, \ldots, a_p\}] \cong \mathbf{B}[\{b_1, \ldots, b_p\}]$ and

- if $k < n$, then the players proceed to make their $(k+1)$-st moves;
- if $k = n$, then *the duplicator wins the round* and the game finishes.

Otherwise, π is not an isomorphism, *the spoiler wins the round* and the game finishes (possibly prematurely, with $k < n$).

Several things to note in the following.

(a) The game has *no tie*: Either the spoiler or the duplicator wins a round.

(b) Both players are eager to win: If there is a chance to win, then they surely will win.

(c) The game is of *complete information*: Either the spoiler or the duplicator *has a winning strategy* to ensure winning “every” round.

Intuitively, the duplicator's objective is to maintain the structural similitude between $\mathbf{A}$ and $\mathbf{B}$ by preserving an isomorphism from the substructure of $\mathbf{A}$ to the substructure of $\mathbf{B}$ both of which are induced by the elements where a pebble is placed (to *duplicate*), whereas the spoiler's objective is to reveal the structural distinction between $\mathbf{A}$ and $\mathbf{B}$ by destroying such an isomorphism (to *spoil*), which, in the game $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$, amounts to finding a FO_n-sentence that distinguishes $\mathbf{A}$ from $\mathbf{B}$ in terms of satisfaction.

All the above can be said and defined with graphs in place of structures, in view of Remark 2.4 and Definitions 3.5 and 3.6. There is an important link between the indistinguishability of two structures (or graphs) $\mathbf{A}$ and $\mathbf{B}$ in FO_n and the existence of a winning strategy of the duplicator for the game $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$. We first introduce the general notion of indistinguishability in a logic formalism. Recall the class $\mathcal{U}$ from Proviso 2.

Definition 3.12. Let L be a logic formalism. For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, we say $\mathbf{A}$ *is L-equivalent to* $\mathbf{B}$ or $\mathbf{A}$ *is indistinguishable from* $\mathbf{B}$ *in* L (written: $\mathbf{A} \equiv_L \mathbf{B}$) if for every L-sentence φ, it holds that $\mathbf{A} \models \varphi$ if and only $\mathbf{B} \models \varphi$.

Theorem 3.6. [20, 25] *Let $n \in \mathbb{Z}^+$. For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, the duplicator has a winning strategy for $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$ if and only if $\mathbf{A}$ is FO_n-equivalent to $\mathbf{B}$.* □

The next example demonstrates that the number of (nested) quantifiers used in a sentence characterizing a structure (or a graph) $\mathbf{A}$ up to isomorphism like $\chi^{\mathbf{A}}$ cannot be reduced.

Example 3.6. Let us consider two independent sets $\mathbf{I}_3$ and $\mathbf{I}_4$. By Proposition 3.5, $\mathbf{I}_3$ can be characterized up to isomorphism by a $\mathrm{FO}_4[\sigma(\mathrm{G})]$-sentence $\chi^{\mathbf{I}_3}$. However, by Theorem 3.6, $\mathbf{I}_3$ and $\mathbf{I}_4$ are $\mathrm{FO}_3[\sigma(\mathrm{G})]$-equivalent since there is a simple winning strategy of the duplicator for the game $\mathrm{G}_3(\mathbf{I}_3, \mathbf{I}_4)$: In the first move, any arbitrary element in the corresponding structure will do. In the second or third move, if a pebble with index j has just been placed by the spoiler on an element of a structure where a pebble with a different index k was already there, then place the other pebble with index j on the element of the opposite structure where the pebble of index k is; otherwise, the pebble with index j placed by the spoiler does not share the element with other pebbles, and the duplicator is free to choose any arbitrary element in the opposite structure where there is no pebble placed and place the pebble with index j on it (such an element is guaranteed to exist due to the sizes of $\mathbf{I}_3$ and $\mathbf{I}_4$). The above discussion applies to $\mathbf{K}_3$ and $\mathbf{K}_4$ as well.

Remark 3.3. Let $n \in \mathbb{Z}^+$. Note that FO_{n+1}*-equivalence implies* FO_n*-equivalence, but the converse does not hold.* The former statement is trivially true because a FO_n-sentence is also a FO_{n+1}-sentence. The latter statement can be verified by considering the game $\mathrm{G}_n(\mathbf{I}_n, \mathbf{I}_{n+1})$, for which the duplicator has a winning strategy (cf. Example 3.6), and the game $\mathrm{G}_{n+1}(\mathbf{I}_n, \mathbf{I}_{n+1})$, for which there is a simple winning strategy for the spoiler: In each move place a new pebble on a different element of $\mathbf{I}_{n+1}$. Therefore, $\mathbf{I}_n \equiv_{\mathrm{FO}_n} \mathbf{I}_{n+1}$ *while* $\mathbf{I}_n \not\equiv_{\mathrm{FO}_{n+1}} \mathbf{I}_{n+1}$. The same can be said with $\mathbf{K}_n$ and $\mathbf{K}_{n+1}$ in place of $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$, respectively. By induction, it follows that *for all $m > n$, we have* FO_m*-equivalence implies* FO_n*-equivalence, but not vice versa.*

Next, we turn our attention to the issue of the number of variables in a first-order sentence.

Definition 3.13. Let $n \in \mathbb{Z}^+$. The *fragment* of FO that consists of σ-formulas φ such that the variables, free or bound, appearing in φ are among $z_0, \ldots, z_{n-1}$ is denoted $\mathrm{FO}^n[\sigma]$, or FO^n when the vocabulary σ is understood from the context.

There is also an important link analogous to Theorem 3.6, this time connecting the FO^n-indistinguishability between two structures (or two graphs) $\mathbf{A}$ and $\mathbf{B}$ and the existence of a winning strategy of the duplicator for a variant of the Ehrenfeucht-Fraïssé game $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$ called a *pebble game*, which we introduce next.

Let $n \in \mathbb{Z}^+$, and let $\mathbf{A}$ and $\mathbf{B}$ be structures. The *pebble game* $\mathrm{G}^n(\mathbf{A}, \mathbf{B})$ of n pebbles parameterized by $\mathbf{A}$ and $\mathbf{B}$ is the same as $\mathrm{G}_n(\mathbf{A}, \mathbf{B})$ except that now

- there are only n pairs of pebbles $\alpha_0, \beta_0, \ldots, \alpha_{n-1}, \beta_{n-1}$, and

- the spoiler starts the game by announcing an $m \in \mathbb{Z}^+$ to be the maximum number of moves to be made by each player, before proceeding to make the first move.

All the rest including the conditions of winning a round and the notion of a winning strategy are the same and we do not repeat them here.

The duplicator's objective is to maintain an isomorphism between the substructures induced by the elements in either structures where a pebble is placed, as before, and the spoiler's objective is to find a FO^n-sentence that would distinguish $\mathbf{A}$ from $\mathbf{B}$ in terms of satisfaction.

Likewise, all the above can be said and defined for graphs in place of structures, in view of Remark 2.4 and Definitions 3.5 and 3.6.

Theorem 3.7. [34] *Let $n \in \mathbb{Z}^+$. For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, the duplicator has a winning strategy for $\mathrm{G}^n(\mathbf{A}, \mathbf{B})$ if and only if $\mathbf{A}$ is FO^n-equivalent to $\mathbf{B}$.* $\square$

Remark 3.4. As a matter of fact, N. Immerman discusses in [34], for all m and n in $\mathbb{Z}^+$,

- the fragment $\mathrm{FO}^n_m := (\mathrm{FO}^n \cap \mathrm{FO}_m)$ of formulas of quantifier rank at most m whose variables are among $z_0, \ldots, z_{n-1}$, and
- the variant game G^n_m that differs from G^n in that the maximum number of moves announced by the spoiler is (fixed to) the given m,

and shows that

(1) for all m and n in $\mathbb{Z}^+$ and all structures $\mathbf{A}$ and $\mathbf{B}$, the duplicator has a winning strategy for the game $\mathrm{G}^n_m(\mathbf{A}, \mathbf{B})$ if and only if $\mathbf{A}$ is FO^n_m-equivalent to $\mathbf{B}$.

For $n \in \mathbb{Z}^+$, it is clear that

(2) $\mathrm{FO}^n = \bigcup_{m \in \mathbb{Z}^+} \mathrm{FO}^n_m$, and

(3) the spoiler has a winning strategy for the game $\mathrm{G}^n(\mathbf{A}, \mathbf{B})$ if and only if the spoiler has a winning strategy for the game $\mathrm{G}^n_m(\mathbf{A}, \mathbf{B})$ for some $m \in \mathbb{Z}^+$ (cf. Exercise XII.4.3 in [19]).

We claim that Theorem 3.7 immediately follows from these considerations for the case $\mathcal{U} = \mathcal{A}$: For all structures $\mathbf{A}$ and $\mathbf{B}$,

the spoiler has a winning strategy for the game $\mathrm{G}^n(\mathbf{A}, \mathbf{B})$

iff the spoiler has a winning strategy for the game $\mathrm{G}^n_m(\mathbf{A}, \mathbf{B})$ for some $m \in \mathbb{Z}^+$ (by (3))

iff $\mathbf{A}$ is not FO^n_m-equivalent to $\mathbf{B}$ for some $m \in \mathbb{Z}^+$ (by (1))

iff $\mathbf{A}$ is not FO^n-equivalent to $\mathbf{B}$ (by (2)).

The case $\mathcal{U} = \mathcal{G}$ immediately follows by considering $\mathbf{A}^{\sigma(\mathrm{G})}$ and $\mathbf{B}^{\sigma(\mathrm{G})}$ for graphs $\mathbf{A}$ and $\mathbf{B}$, in view of Remark 2.4 and Definitions 3.5 and 3.6.

Example 3.7. In Example 3.6, the FO_4-sentence $\chi^{\mathbf{I}_3}$ characterizing $\mathbf{I}_3$ up to isomorphism is also a FO^4-sentence. However, $\mathbf{I}_3$ is $\mathrm{FO}^3[\sigma(\mathrm{G})]$-equivalent to $\mathbf{I}_4$ by Theorem 3.7: A winning strategy of the game $\mathrm{G}_3(\mathbf{I}_3, \mathbf{I}_4)$ for the duplicator can be easily adapted from the one in Example 3.6 (hence omitted). The above discussion applies to $\mathbf{K}_3$ and $\mathbf{K}_4$ as well.

Remark 3.5. Let $n \in \mathbb{Z}^+$. Obviously, every FO^n-sentence is also a FO^{n+1}-sentence. Moreover, the duplicator has a winning strategy for the game $\mathrm{G}^n(\mathbf{I}_n, \mathbf{I}_{n+1})$, while the spoiler has a winning strategy for the game $\mathrm{G}^{n+1}(\mathbf{I}_n, \mathbf{I}_{n+1})$ (see Example 3.7). Hence, $\mathbf{I}_n \equiv_{\mathrm{FO}^n} \mathbf{I}_{n+1}$ *while* $\mathbf{I}_n \not\equiv_{\mathrm{FO}^{n+1}} \mathbf{I}_{n+1}$. The same can be said with $\mathbf{K}_n$ and $\mathbf{K}_{n+1}$ in place of $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$, respectively. As in Remark 3.3, therefore, it holds that FO^{n+1}*-equivalence implies* FO^n*-equivalence, but the converse does not hold.* Furthermore, by induction it immediately follows that *for all* $m > n$*, we have* FO^m*-equivalence implies* FO^n*-equivalence.*

Example 3.8. For every graph $\mathbf{G}$ and every subset $S \subseteq V(\mathbf{G})$ of size $|S| = 1$, the induced subgraph $\mathbf{G}[S]$ is isomorphic to $\mathbf{I}_1$. Thus, for all graphs $\mathbf{G}$ and $\mathbf{H}$, the duplicator has a straightforward winning strategy for the games $\mathrm{G}_1(\mathbf{G}, \mathbf{H})$ and $\mathrm{G}^1(\mathbf{G}, \mathbf{H})$. Therefore, *all graphs are* FO_1*-equivalent and* FO^1*-equivalent.*

We have seen in Theorems 3.6 and 3.7 the characterizations of indistinguishability in the two fragments of first-order logic, namely FO_n and FO^n. We conclude this subsection with a related issue: definability of a class of structures (or of graphs) in FO_n or FO^n.

Definition 3.14. Let L be a logic formalism.

(a) For every L-sentence φ, the *model class of* φ is the class $\mathrm{Mod}(\varphi) := \{\mathbf{A} \in \mathcal{U} \mid \mathbf{A} \models \varphi\}$.

(b) Let $\mathcal{D} \subseteq \mathcal{U}$ be a class. We say that $\mathcal{D}$ is *definable by an L-sentence* φ if $\mathcal{D} = \mathrm{Mod}(\varphi)$, and that $\mathcal{D}$ is *definable in L* (or *L-definable* for short) if such a sentence exists.

Example 3.9. The class $\mathcal{I}$ of independent sets introduced in Section 2.2 is both $\mathrm{FO}_2[\sigma(\mathrm{G})]$-definable and $\mathrm{FO}^2[\sigma(\mathrm{G})]$-definable, because $\mathcal{I} = \mathrm{Mod}(\varphi)$ for $\varphi := \forall x \forall y \neg E(x, y)$.

Observe that

$$\mathrm{FO} = \bigcup_{n \in \mathbb{Z}^+} \mathrm{FO}_n = \bigcup_{n \in \mathbb{Z}^+} \mathrm{FO}^n.$$

Hence, a class $\mathcal{D}$ is first-order definable if and only if $\mathcal{D} = \mathrm{Mod}(\varphi)$ for some $n \in \mathbb{Z}^+$ and some formula φ in FO_n or FO^n.

Example 3.10. Let $n \geq 2$ and consider the two graphs $\mathbf{C}_{2^n} \oplus \mathbf{C}_{2^n}$ and $\mathbf{C}_{2^{n+1}}$. It is straightforward to formalize as a $\mathrm{FO}^3[\sigma(\mathrm{G})]$-sentence the statement "for all distinct vertices u and v, there is a walk of length at most $n+1$ from u to v" (cf. Example 3.3.1(b) in [18]).

Thus, these two graphs are not $\mathrm{FO}^3[\sigma(\mathrm{G})]$-equivalent. However, there is a winning strategy of the duplicator for the game $\mathrm{G}_n(\mathbf{C}_{2^n} \oplus \mathbf{C}_{2^n}, \mathbf{C}_{2^{n+1}})$ (cf. Example 2.3.8 in [18]).

Therefore, it holds for every $n \geq 2$ that the duplicator has a winning strategy for the game $\mathrm{G}_n(\mathbf{C}_{2^n} \oplus \mathbf{C}_{2^n}, \mathbf{C}_{2^{n+1}})$ and hence $\mathbf{C}_{2^n} \oplus \mathbf{C}_{2^n}$ and $\mathbf{C}_{2^{n+1}}$ are indistinguishable in $\mathrm{FO}_n[\sigma(\mathrm{G})]$, by Theorem 3.6. As a result, *the class of connected graphs is not* FO-*definable* (also see Proposition 2.3.28 in [27]).

By Remarks 3.3 and 3.5, we have for every $n \in \mathbb{Z}^+$ that the class of graphs of size at most n is not definable in FO_n or FO^n. The reader is on the right track if speculating that the inadequate expressive power of FO_n and FO^n in these situations is due to their lack of ability for *counting*. This brings us to the next topic – enhancing these fragments with the ability to count.

3.1.2 First-Order Logic Augmented with Counting and Its Fragments

To conclude this section, we consider the extension of first-order logic by a mechanism of counting and its fragments.

Syntax

Definition 3.15. The formalism of *first-order logic with counting* or simply *counting logic* (*parameterized by* σ), denoted $\mathrm{C}[\sigma]$, is the set of strings, called *first-order σ-formulas with counting*, over the alphabet

$$\sigma \cup \{z_0, z_1, z_2, \ldots, =, \neg, \wedge, \vee, \rightarrow, \forall, \exists, \exists^{\geq 1}, \exists^{\geq 2}, \exists^{\geq 3}, \ldots,), (\}$$

whose grammar consists of the formation rule in addition to those in Definition 3.1:

$$\frac{\varphi}{\exists^{\geq n} z_i \varphi} \quad \text{where } n \in \mathbb{Z}^+ \text{ and } i \in \mathbb{N}.$$

The new symbols $\exists^{\geq n}$ are called *counting quantifiers*. By analogy, we adopt the abbreviation C$[\sigma]$-*formulas* for "first-order σ-formulas with counting" and, when the underlying vocabulary σ is irrelevant or can be understood from the context, omit the explicit reference to σ in all our terms concerning C, and further omit "C" or "counting" when it is known the underlying logic is counting logic.

Definition 3.16. The set of free variables in a C-formula φ, denoted free(φ), is defined inductively with those cases in Definition 3.2 and the case:

$$\text{free}(\exists^{\geq n} x\varphi) \quad := \quad \text{free}(\varphi) \setminus \{x\}, \quad \text{for all } n \in \mathbb{Z}^+.$$

Semantics

Definition 3.17. Let $\mathbf{A}$ be a σ-structure, and let α be an assignment for $\mathbf{A}$. For all C$[\sigma]$-formulas, the notion that φ *is true in* $(\mathbf{A}, \alpha)$ or that $(\mathbf{A}, \alpha)$ *satisfies* φ or that $(\mathbf{A}, \alpha)$ *is a model of* φ, denoted $(\mathbf{A}, \alpha) \models \varphi$, is defined inductively on φ with the clauses from Definition 3.3 and the one below:

$$(\mathbf{A}, \alpha) \models \exists^{\geq n} x\varphi \quad \text{:if} \quad (\mathbf{A}, \alpha(x/a)) \models \varphi \text{ for at least } n \text{ distinct elements } a \in \text{dom}(\mathbf{A}).$$

The Coincidence Lemma 3.1 also holds for counting logic. As a result, we let the convention of writing $\varphi(x_1, \ldots, x_n)$ for φ when free$(\varphi) \subseteq \{x_1, \ldots, x_n\}$ and the term *sentence* extends to counting logic; the notation $\mathbf{A} \models \varphi(a_1, \ldots, a_n)$ is analogous. When φ is a sentence, we also write $\mathbf{A} \models \varphi$ to mean $(\mathbf{A}, \alpha) \models \varphi$ for an arbitrary assignment α

for $\mathbf{A}$. The convention of defining notations involving $\models$ for graphs $\mathbf{G}$ in terms of $\mathbf{G}^{\sigma(\mathrm{G})}$ is also analogous (e.g., $\mathbf{G} \models \varphi$ means $\mathbf{G}^{\sigma(\mathrm{G})} \models \varphi$). The notion of *logical equivalence* reads the same for counting logic. We have, in addition to the list following Definition 3.3, the subsequent pair of logically equivalent formulas:

(7) $\exists x\varphi$ and $\exists^{\geq 1} x\varphi$, for all formulas φ.

Abbreviation. For $n \in \mathbb{Z}^+$ and for $\varphi \in \mathrm{C}$, let $\exists^{=n} x\varphi$ abbreviate $\exists^{\geq n} x\varphi \wedge \neg\exists^{\geq n+1} x\varphi$.

Example 3.11. For every $s \in \mathbb{Z}^+$ and every structure $\mathbf{A}$, we have that $\mathbf{A}$ satisfies $\exists^{=s} x\, x = x$ if and only if $|\mathrm{dom}(\mathbf{A})| = s$.

Remark 3.6. Note that first-order logic is itself a fragment of counting logic by nature of the syntax and semantics. However, every C-formula of the form $\exists^{\geq n} x\varphi$ is logically equivalent to the FO-formula

$$\exists x_1 \cdots \exists x_n \Big(\bigwedge_{1 \leq i \leq n} \varphi(x/x_i) \wedge \bigwedge_{1 \leq i < j \leq n} \neg x_i = x_j \Big).$$

Hence, C *has the same expressive power as* FO and it turns out that augmenting FO with counting quantifiers does not help us significantly in any aspect, except for the number of variables and quantifier ranks. Furthermore, C-*equivalence coincides with* FO-*equivalence* (cf. Definition 3.12) and consequently the Isomorphism Lemma 3.2 holds for C as well.

Expressive Power of Fragments of Counting Logic

In the preceding remark, we saw that no extra expressive power was gained by enhancing first-order logic with the mechanism of counting. That said, things can change if we limit the number of variables used in a formula.

Definition 3.18. Let $n \in \mathbb{Z}^+$. The *fragment* of C that consists of σ-formulas φ such that the variables, free or bound, appearing in φ are among $z_0, \ldots, z_{n-1}$ is denoted $\mathrm{C}^n[\sigma]$, or C^n when the vocabulary σ is understood from the context.

Example 3.12. In Example 3.7, we saw that $\mathbf{I}_3$ and $\mathbf{I}_4$ are FO^3-equivalent. However, now that we are equipped with counting ability in the new logic formalism C, the graph $\mathbf{I}_3$ can even be characterized up to isomorphism in C^2 by $\exists^{=3} z_0\, z_0 = z_0 \wedge \forall z_0 \forall z_1 \neg E(z_0, z_1)$, which asserts that there are exactly three vertices each of which has degree 0.

Remark 3.7. Fix an $s \in \mathbb{Z}^+$.

(a) We saw in Example 3.11 that for all structures $\mathbf{A}$, it holds that $\mathbf{A}$ is a model of the C^1-sentence $\exists^{=s} x\, x = x$ if and only if $|\mathrm{dom}(\mathbf{A})| = s$. Moreover, for every graph $\mathbf{G}$, the adjacency relation $E^{\mathbf{G}^{\sigma(\mathrm{G})}}$ of $\mathbf{G}^{\sigma(\mathrm{G})}$ is irreflexive (see Remark 2.3), i.e., $\mathbf{G}^{\sigma(\mathrm{G})} \models \forall x \neg E(x, x)$. Using a variant of Ehrenfeucht-Fraïssé game for counting logic (see, e.g., [18, 35, 44]), it can be proved that *for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, they are* C^1*-equivalent if and only if they have the same size*. Consequently, *for every graph* $\mathbf{G}$*, the sentence* $\exists^{=|V(\mathbf{G})|} x\, x = x$ *characterizes* $\mathbf{G}$ *up to* C^1*-equivalence*, i.e., for every graph $\mathbf{H}$, it is a model of this sentence if and only if $\mathbf{G} \equiv_{\mathrm{C}^1} \mathbf{H}$.

(b) There are finitely many isomorphism types of structures of size s and hence, for every $n \in \mathbb{Z}^+$, there are finitely many equivalence classes, say, $\mathcal{D}_1, \ldots, \mathcal{D}_k$, induced by C^n-equivalence among all structures of size s. Let $\mathbf{A}_1, \ldots, \mathbf{A}_k$ be the representatives of the respective equivalence classes and, for distinct $i, j \in [k]$, let $\varphi_{i,j}$ be a C^n-sentence such that $\mathbf{A}_i \models \varphi_{i,j}$ while $\mathbf{A}_j \models \neg\varphi_{i,j}$. By part (a), it follows that for every $i \in [k]$, the C^n-sentence $\chi^{\mathbf{A}_i}_{\mathrm{C}^n} := (\exists^{=s} x\, x = x \wedge \varphi_{i,j})$ characterizes $\mathbf{A}_i$ up to C^n-equivalence, i.e., for every structure $\mathbf{A}$, it is a model of $\chi_{\mathrm{C}^n}^{\mathbf{A}_i}$ if and only if $\mathbf{A} \equiv_{\mathrm{C}^n} \mathbf{A}_i$. Hence, *for every structure* $\mathbf{A}$ *and every* $n \in \mathbb{Z}^+$*, there is a* C^n*-sentence* $\chi_{\mathrm{C}^n}^{\mathbf{A}}$ *that characterizes* $\mathbf{A}$ *up to* C^n*-equivalence.* (This statement is mentioned in [28] as Fact 3.4.17, and the argument provided here is adapted from the proof of Lemma 1.33 for part (i) in [44].)

Example 3.13. The two graphs $\mathbf{C}_n \oplus \mathbf{C}_n$ and $\mathbf{C}_{2n}$ are C^2-equivalent. A proof for this can be obtained by adapting the discussion in Example 12.19 of [35].

Let $n \in \mathbb{Z}^+$. Since every C^n-sentence is also a C^{n+1}-sentence, we have C^{n+1}*-equivalence implies* C^n*-equivalence, but the converse does not holds.* The latter statement is proved in [7].

Augmenting Counting Logic with Infinite (Conjunctions and) Disjunctions

The *infinitary logic with counting*, denoted $\mathrm{C}_{\infty\omega}$, enhances C by allowing the "formation rule" besides Definition 3.15 in its syntax,

$$\frac{}{\bigvee \Phi} \quad \text{where } \Phi \text{ is a set of formulas,}$$

and by adding, for every structure $\mathbf{A}$, every assignment α for $\mathbf{A}$, and every set Φ of $\mathrm{C}_{\infty\omega}$-formulas, the following clause to Definition 3.17 in its semantics,

$$(\mathbf{A}, \alpha) \models \bigvee \Phi \quad \text{:if} \quad (\mathbf{A}, \alpha) \models \varphi \text{ for some } \varphi \in \Phi.$$

That is to say, in $\mathrm{C}_{\infty\omega}$ we allow disjunctions over a set of arbitrarily many – even infinitely many – formulas, hence *infinitary* in its name. As a result, a "$\mathrm{C}_{\infty\omega}$-formula" is not necessarily a finite string, and properties about these formulas may require *transfinite induction* in the proofs. We shall not address the technical issues here, and we note that the notions of the *set of free variables in an* $\mathrm{C}_{\infty\omega}$*-formula*, $\mathrm{C}_{\infty\omega}$*-sentences*, and *logically equivalent* $\mathrm{C}_{\infty\omega}$-

formulas can all be extended in a straightforward manner. In particular, a $\mathrm{C}_{\infty\omega}$-formula can have infinitely many free variables, the *conjunction* $\bigwedge \Phi$ over a set Φ of $\mathrm{C}_{\infty\omega}$-formulas is taken as an abbreviation for $\neg \bigvee \{\neg\varphi \mid \varphi \in \Phi\}$, and $\varphi \vee \psi$ is logically equivalent to $\bigvee \{\varphi, \psi\}$.

However, the logic $\mathrm{C}_{\infty\omega}$ has too strong expressive power to produce any insightful results when it comes to finite structures, e.g., for every class $\mathcal{D}$ of finite structures there is a $\mathrm{C}_{\infty\omega}$-sentence φ such that $\mathcal{D} = \mathrm{Mod}(\varphi)$. In fact, the fragment $\mathrm{L}_{\infty\omega}$ of $\mathrm{C}_{\infty\omega}$ that consists of formulas without counting quantifiers already possesses such expressive power (see [18,27]).

Remark 3.8. Let $n \in \mathbb{Z}^+$. The fragment of $\mathrm{C}_{\infty\omega}$ that consists of formulas φ such that the variables, free or bound, appearing in φ are among $z_0, \ldots, z_{n-1}$ is denoted $\mathrm{C}^n_{\infty\omega}$. The fragment $\mathrm{C}^n_{\infty\omega}$, in contrast to $\mathrm{C}_{\infty\omega}$, has some interesting aspects in finite model theory:

(a) It can be proved by induction on φ that *for all structures* $\mathbf{A}$ *and* $\mathbf{B}$*, and for all* $\mathrm{C}^n_{\infty\omega}$*-formulas* $\varphi(\overline{x})$*, there is a* C^n*-formula* $\psi(\overline{x})$ *with* $\mathrm{free}(\psi) \subseteq \mathrm{free}(\varphi)$ *such that* $\mathbf{A} \models \forall\overline{x}(\varphi(\overline{x}) \leftrightarrow \psi(\overline{x}))$ *and* $\mathbf{B} \models \forall\overline{x}(\varphi(\overline{x}) \leftrightarrow \psi(\overline{x}))$. Thus, *for all structures* $\mathbf{A}$ *and* $\mathbf{B}$*, it holds that* $\mathbf{A} \equiv_{\mathrm{C}^n} \mathbf{B}$ *if and only if* $\mathbf{A} \equiv_{\mathrm{C}^n_{\infty\omega}} \mathbf{B}$ (the "if" direction is trivial). (Arguments for these can be obtained by adapting the ones for Proposition 3.2.2 and Corollary 3.3.3 in [18]. Moreover, the latter result immediately follows from Corollary 2.4 in [44].)

(b) Let $\mathcal{D}$ be a class of structures. It holds that $\mathcal{D}$ *is definable in* $\mathrm{C}^n_{\infty\omega}$ *if and only if for all structures* $\mathbf{A}$ *and* $\mathbf{B}$*, we have that* $\mathbf{A} \in \mathcal{D}$ *and* $\mathbf{A} \equiv_{\mathrm{C}^n} \mathbf{B}$ *imply* $\mathbf{B} \in \mathcal{D}$. For the "if" direction, consider the C^n-sentence $\chi^{\mathbf{A}}_{\mathrm{C}^n}$ in Remark 3.7(b) that characterizes a given structure $\mathbf{A}$ up to C^n-equivalence, and as a consequence we have $\mathcal{D} = \mathrm{Mod}(\varphi)$ for the $\mathrm{C}^n_{\infty\omega}$-sentence $\varphi := \bigvee \{\chi^{\mathbf{A}}_{\mathrm{C}^n} \mid \mathbf{A} \in (\mathcal{D}/{\cong})\}$, where $\mathcal{D}/{\cong}$ is the *set* of representatives of the isomorphism types of structures in $\mathcal{D}$ (see Section 2.3). The "only if" direction follows immediately from the above part (a).

Remark 3.9. The various logic formalisms L considered in Section 3.1 are closed under the connectives $\neg$, $\vee$ and $\wedge$. Thus, we have that $\overline{\mathrm{Mod}(\varphi)} = \mathrm{Mod}(\neg\varphi)$ for all L-sentences φ and that $\mathrm{Mod}(\varphi) \cup \mathrm{Mod}(\psi) = \mathrm{Mod}(\varphi \vee \psi)$ and $\mathrm{Mod}(\varphi) \cap \mathrm{Mod}(\psi) = \mathrm{Mod}(\varphi \wedge \psi)$ for all L-sentences φ and ψ. Moreover, it is clear that $\mathrm{Mod}(\varphi) \subseteq \mathrm{Mod}(\psi)$ if and only if $\mathbf{A} \models (\varphi \to \psi)$ for all structures $\mathbf{A}$.

3.2 Homomorphism

The notion of *homomorphism* captures the idea that, when comparing two graphs or two structures, some structural properties are preserved (similar or even the same) in both. As the reader may have guessed, isomorphism is among those structural properties. Furthermore, homomorphism is also a notion prevalent in various areas such as graph theory, database theory, abstract algebra, mathematical logic and category theory.

Throughout this section, we will often use the versatile symbol $\mathcal{U}$ to represent the class $\mathcal{G}$ of graphs or the class $\mathcal{A}$ of structures (see Proviso 2) to make a uniform statement.

3.2.1 Basic Definitions and Properties

Definition 3.19. (a) Let $\mathbf{G}$ and $\mathbf{H}$ be graphs. A function $h : V(\mathbf{G}) \to V(\mathbf{H})$ *is a homomorphism from* $\mathbf{G}$ *to* $\mathbf{H}$ (written: $h : \mathbf{G} \to \mathbf{H}$) if for all $u, v \in V(\mathbf{G})$, it is true that $\{u, v\} \in E(\mathbf{G})$ implies $\{h(u), h(v)\} \in E(\mathbf{H})$. We say that $\mathbf{G}$ *is homomorphic to* $\mathbf{H}$ (written: $\mathbf{G} \to \mathbf{H}$) if there is a homomorphism from $\mathbf{G}$ to $\mathbf{H}$.

(b) Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures of some vocabulary σ. A function $h : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$

is a homomorphism from $\mathbf{A}$ *to* $\mathbf{B}$ (written: $h : \mathbf{A} \to \mathbf{B}$) if for all $n \in \mathbb{Z}^+$, all n-ary relation symbols $R \in \sigma$ and all elements $a_1, \ldots, a_n \in \mathrm{dom}(\mathbf{A})$, it is true that $(a_1, \ldots, a_n) \in R^{\mathbf{A}}$ implies $(h(a_1), \ldots, h(a_n)) \in R^{\mathbf{B}}$. We say that $\mathbf{A}$ *is homomorphic to* $\mathbf{B}$ (written: $\mathbf{A} \to \mathbf{B}$) if there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Example 3.14. Consider the cycle graph $\mathbf{C}_6$ and the path graph $\mathbf{P}_3$, where $V(\mathbf{C}_6) = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ and $E(\mathbf{C}_6) = \{(u_1, u_2), (u_2, u_3), (u_3, u_4), (u_4, u_5), (u_5, u_6), (u_6, u_1)\}$, $V(\mathbf{P}_3) = \{v_1, v_2, v_3\}$ and $E(\mathbf{P}_3) = \{(v_1, v_2), (v_2, v_3)\}$. We have $h : \mathbf{C}_6 \to \mathbf{P}_3$ with $h(u_1) = h(u_3) = h(u_5) = v_1, h(u_2) = h(u_4) = h(u_6) = v_2$.

Remark 3.10. (a) By definition, an isomorphism is also a homomorphism.

(b) The binary relation $\to$ is reflexive since for all $\mathbf{A}$ in $\mathcal{U}$, we have $\mathbf{A} \to \mathbf{A}$. It is also transitive because for all $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ in $\mathcal{U}$, if $h_1 : \mathbf{A} \to \mathbf{B}$ and $h_2 : \mathbf{B} \to \mathbf{C}$, then $(h_2 \circ h_1) : \mathbf{A} \to \mathbf{C}$. Therefore, $\to$ is a *preorder* over $\mathcal{U}$. In the case of $\mathcal{U} = \mathcal{G}$, the independent sets $\mathbf{I}_1, \mathbf{I}_2, \ldots$ are the $\to$-minimals; when restricted to the class of graphs of size at most n, the graphs $\mathbf{I}_1, \ldots, \mathbf{I}_n$ are the $\to$-minimals while $\mathbf{K}_n$ is the $\to$-maximal.

Remark 3.11. Similar to Remark 2.4, for all graphs $\mathbf{G}$ and $\mathbf{H}$ and all functions $f : V(\mathbf{G}) \to V(\mathbf{H})$, it holds that $f : \mathbf{G} \to \mathbf{H}$ is a homomorphism if and only if $f : \mathbf{G}^{\sigma(\mathrm{G})} \to \mathbf{H}^{\sigma(\mathrm{G})}$ is a homomorphism.

Homomorphism and Subgraphs or Substructures

Definition 3.20. (a) Let $\mathbf{G}$ and $\mathbf{H}$ be graphs, and let $h : \mathbf{G} \to \mathbf{H}$ be a homomorphism. For every subgraph $\mathbf{F} \subseteq \mathbf{G}$, the *homomorphic image of* $\mathbf{F}$ *under* h is the subgraph $h(\mathbf{F}) \subseteq \mathbf{H}$ with vertex set $V(h(\mathbf{F})) := h(V(\mathbf{F}))$ and edge set $E(h(\mathbf{F})) := \{\{h(u), h(v)\} \mid \{u, v\} \in E(\mathbf{F})\}$.

(b) Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures for some vocabulary σ, and let $h : \mathbf{A} \to \mathbf{B}$ be a homomorphism. For every substructure $\mathbf{C} \subseteq \mathbf{A}$, the *homomorphic image of* $\mathbf{C}$ *under* h is the substructure $h(\mathbf{C}) \subseteq \mathbf{B}$ with domain $\mathrm{dom}(h(\mathbf{C})) := h(\mathrm{dom}(\mathbf{C}))$ and relations $R^{h(\mathbf{C})} := \{(h(a_1), \ldots, h(a_n)) \mid (a_1, \ldots, a_n) \in R^{\mathbf{C}}\}$ for all $n \in \mathbb{Z}^+$ and all n-ary relation symbols $R \in \sigma$.

The above definition of the image of a substructure $\mathbf{C} \subseteq \mathbf{A}$ under a homomorphism $h : \mathbf{A} \to \mathbf{B}$ implies that $h(\mathbf{C})$ is a substructure of $\mathbf{B}$: If $(a_1, \ldots, a_n) \in R^{\mathbf{C}} \subseteq R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_n)) \in R^{\mathbf{B}}$. Likewise for the definition of the image of a subgraph under a homomorphism (in view of Remark 3.11).

Remark 3.12. Let $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathcal{U}$. By Remark 2.1(a), if $\mathbf{A} \to \mathbf{B}$ and $\mathbf{B} \subseteq \mathbf{C}$, then $\mathbf{A} \to \mathbf{C}$.

Homomorphism Counts and a Decomposition Equation

Notation. For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, let $\hom(\mathbf{A}, \mathbf{B})$ denote the number of homomorphisms from $\mathbf{A}$ to $\mathbf{B}$.

Example 3.15. Consider the independent set $\mathbf{I}_1$ with $V(\mathbf{I}_1) = \{v_1\}$ and $E(\mathbf{I}_1) = \emptyset$. For all graphs $\mathbf{G}$, we have $\mathbf{I}_1 \to \mathbf{G}$ (see Remark 3.10(b)) and, in fact, $\hom(\mathbf{I}_1, \mathbf{G}) = |V(\mathbf{G})|$ because every function $h : V(\mathbf{I}_1) \to V(\mathbf{G})$ is a homomorphism $h : \mathbf{I}_1 \to \mathbf{G}$. In general, for every $n \in \mathbb{Z}^+$, the independent set $\mathbf{I}_n$ has n vertices and no edges, and we have $\hom(\mathbf{I}_n, \mathbf{G}) = |V(\mathbf{G})|^n$.

Example 3.16. Consider the clique $\mathbf{K}_2$ with $V(\mathbf{K}_2) = \{v_1, v_2\}$ and $E(\mathbf{K}_2) = \{(v_1, v_2)\}$. For all graphs $\mathbf{G}$, every function $h : V(\mathbf{K}_2) \to V(\mathbf{G})$ is a homomorphism $h : \mathbf{K}_2 \to \mathbf{G}$ precisely when $\{h(v_1), h(v_2)\} \in E(\mathbf{G})$, and every function $g : V(\mathbf{K}_2) \to V(\mathbf{G})$ such that $g(v_1) = f(v_2)$ and $g(v_2) = f(v_1)$ is a homomorphism $g : \mathbf{K}_2 \to \mathbf{G}$ precisely when f is a homomorphism. Thus, $\hom(\mathbf{K}_2, \mathbf{G}) = 2 \times |E(\mathbf{G})|$.

Example 3.17. Let $n \in \mathbb{Z}^+$. Assume that the clique $\mathbf{K}_n$ has vertex set $V(\mathbf{K}_n) = \{0, \ldots, n-1\}$ and edge set $E(\mathbf{K}_n) = \{\{i, j\} \mid i, j \in [0, n-1] \text{ and } i \neq j\}$.

Let $\mathbf{G}$ be a graph. It is n-colorable (see Definition 2.7) when there is an n-coloring of $\mathbf{G}$, and a function $h : V(\mathbf{G}) \to \{0, \ldots, n-1\}$ is an n-coloring of $\mathbf{G}$ exactly when it is true that for all distinct vertices $u, v \in V(\mathbf{G})$, if $\{u, v\} \in E(\mathbf{G})$, then $h(u) \neq h(v)$, i.e., $\{h(u), h(v)\} \in E(\mathbf{K}_n)$. Therefore, a function $h : V(\mathbf{G}) \to \{0, \ldots, n-1\}$ is an n-coloring of $\mathbf{G}$ if and only if it is a homomorphism $h : \mathbf{G} \to \mathbf{K}_n$. Hence, $\hom(\mathbf{G}, \mathbf{K}_n)$ *is the number of n-colorings of* $\mathbf{G}$ and, furthermore, $\mathbf{G}$ *is n-colorable if and only if* $\hom(\mathbf{G}, \mathbf{K}_n) > 0$.

Example 3.18. (a) Let $\mathbf{J}_1$ be a σ-structure with $|\mathrm{dom}(\mathbf{J}_1)| = 1$ and $R^{\mathbf{J}_1} = \emptyset$ for all r-ary relation symbols $R \in \sigma$, that is, $\mathbf{J}_1$ is an *empty* σ-structure of size 1. For all σ-structures $\mathbf{A}$, we have $\mathbf{J}_1 \to \mathbf{A}$ and, indeed, $\hom(\mathbf{J}_1, \mathbf{A}) = |\mathrm{dom}(\mathbf{A})|$ because every function $h : \mathrm{dom}(\mathbf{J}_1) \to \mathrm{dom}(\mathbf{A})$ is a homomorphism $h : \mathbf{J}_1 \to \mathbf{A}$. In general, for every σ-structure $\mathbf{J}_n$ of size $n \in \mathbb{Z}^+$ that is empty (i.e., $R^{\mathbf{J}_n} = \emptyset$), we have $\hom(\mathbf{J}_n, \mathbf{A}) = |\mathrm{dom}(\mathbf{A})|^n$.

(b) Let $\mathbf{S}_2$ be a σ-structure with $|\mathrm{dom}(\mathbf{S}_2)| = 2$ and $R^{\mathbf{S}_2} = \mathrm{dom}(\mathbf{S}_2)^r$ for all r-ary relation symbols $R \in \sigma$, that is, $\mathbf{S}_2$ is a *complete* σ-structure of size 2. For all σ-structures $\mathbf{A}$, we have $\mathbf{A} \to \mathbf{S}_2$ and, indeed, $\hom(\mathbf{A}, \mathbf{S}_2) = 2^{|\mathrm{dom}(\mathbf{A})|}$ because every function $h : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{S}_2)$ is a homomorphism $h : \mathbf{A} \to \mathbf{S}_2$. In general, for every σ-structure

$\mathbf{S}_n$ of size $n \in \mathbb{Z}^+$ that is complete (i.e., $R^{\mathbf{S}_n} = \mathrm{dom}(\mathbf{S}_n)^r$), we have $\hom(\mathbf{A}, \mathbf{S}_n) = n^{|\mathrm{dom}(\mathbf{A})|}$.

Remark 3.13. As noted in Remark 3.1, satisfaction of first-order sentences in structures or in graphs is invariant under isomorphism. Indeed, homomorphism counts are also invariant under isomorphism: For all $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ in $\mathcal{U}$ with $\pi_1 : \mathbf{A} \cong \mathbf{C}$ and $\pi_2 : \mathbf{B} \cong \mathbf{D}$, if $h : \mathbf{A} \to \mathbf{B}$, then we have $(\pi_2 \circ (h \circ \pi_1^{-1})) : \mathbf{C} \to \mathbf{D}$. Thus, the homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ can be put into a one-to-one correspondence with the homomorphisms from $\mathbf{C}$ to $\mathbf{D}$ via the mapping $h \mapsto (\pi_2 \circ (h \circ \pi_1^{-1}))$.

Next, we shall derive a useful equation that is a decomposition of homomorphism counts involving surjective homomorphism counts, injective homomorphism counts and automorphism counts in Proposition 3.8. Let us start with relevant definitions.

Definition 3.21. Let $\mathbf{A}, \mathbf{B} \in \mathcal{U}$.

(a) A homomorphism $h : \mathbf{A} \to \mathbf{B}$ is *injective* if the underlying mapping h is injective. The set of injective homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ is denoted $\mathrm{Inj}(\mathbf{A}, \mathbf{B})$, with $\mathrm{inj}(\mathbf{A}, \mathbf{B}) := |\mathrm{Inj}(\mathbf{A}, \mathbf{B})|$.

(b) A homomorphism $h : \mathbf{A} \to \mathbf{B}$ is *surjective* if $h(\mathbf{A}) = \mathbf{B}$. The set of surjective homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ is denoted $\mathrm{Sur}(\mathbf{A}, \mathbf{B})$, with $\mathrm{sur}(\mathbf{A}, \mathbf{B}) := |\mathrm{Sur}(\mathbf{A}, \mathbf{B})|$.

Notations. Let $\mathbf{A}, \mathbf{B} \in \mathcal{U}$. The set of homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ is denoted $\mathrm{Hom}(\mathbf{A}, \mathbf{B})$, and the set of isomorphisms from $\mathbf{A}$ to $\mathbf{B}$ is denoted $\mathrm{Iso}(\mathbf{A}, \mathbf{B})$, with $\mathrm{iso}(\mathbf{A}, \mathbf{B}) := |\mathrm{Iso}(\mathbf{A}, \mathbf{B})|$.

Remark 3.14. Let $\mathbf{A}$ be a structure (or a graph, the following applies as well with "graph" in place of "structure" and with "vertex set, V" in place of "domain, dom" accordingly).

(a) If $\mathbf{A} \subseteq \mathbf{B}$ for some structure $\mathbf{B}$, then there is an injective homomorphism $i : \mathbf{A} \to \mathbf{B}$ such that the underlying mapping $i : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$ is the inclusion mapping (see Section 2.1), i.e., $i(a) = a$ for all $a \in \mathrm{dom}(\mathbf{A})$.

(b) If θ is a partition of $\mathrm{dom}(\mathbf{A})$, then there is a surjective homomorphism $s : \mathbf{A} \to \mathbf{A}/\theta$ from $\mathbf{A}$ to the quotient $\mathbf{A}/\theta$, in which the underlying mapping $s : \mathrm{dom}(\mathbf{A}) \to \theta$ maps every $a \in \mathrm{dom}(\mathbf{A})$ to the set $A \in \theta$ such that $a \in A$.

(c) For all structures $\mathbf{B}$, if $h : \mathbf{A} \to \mathbf{B}$ is a homomorphism, then $h' : \mathbf{A} \to h(\mathbf{A})$ is a surjective homomorphism where $h' : \mathrm{dom}(\mathbf{A}) \to h(\mathrm{dom}(\mathbf{A}))$ is the surjective function with the same mapping rule as $h : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$ (see Remark 2.1(b)); in particular, if h is an injective homomorphism, then $h' : \mathbf{A} \cong h(\mathbf{A})$ is an isomorphism.

Remark 3.15. (a) Immediate by definition, the binary relation $\to$ over $\mathcal{U}$ restricted to the existence of injective or of surjective homomorphisms is transitive: For all $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ in $\mathcal{U}$, if $f : \mathbf{A} \to \mathbf{B}$ and $g : \mathbf{B} \to \mathbf{C}$ are both injective (or surjective) homomorphisms, then $(g \circ f) : \mathbf{A} \to \mathbf{C}$ is also an injective (or surjective, respectively) homomorphism.

(b) Let $\mathbf{A}$ and $\mathbf{B}$ be both σ-structures (or both graphs, the following applies with "graph" in place of "σ-structure" and accordingly, with "vertex set V" in place of "domain, dom" and with "edge set, E" in place of "relation, R" for all $R \in \sigma$, respectively), and let $f : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$ be a function. If $f \in \mathrm{Inj}(\mathbf{A}, \mathbf{B})$, then by Remark 3.14(c), we have $|\mathrm{dom}(\mathbf{A})| = |\mathrm{dom}(h(\mathbf{A}))| \leq |\mathrm{dom}(\mathbf{B})|$ and $|R^{\mathbf{A}}| = |R^{h(\mathbf{A})}| \leq |R^{\mathbf{B}}|$ for all $R \in \sigma$, and the equalities hold in both "$\leq$" precisely when $f \in \mathrm{Sur}(\mathbf{A}, \mathbf{B})$, i.e., when $f \in \mathrm{Iso}(\mathbf{A}, \mathbf{B})$. If $f \in \mathrm{Sur}(\mathbf{A}, \mathbf{B})$, then by Remark 3.14(c), we have $|\mathrm{dom}(\mathbf{B})| = |h(\mathrm{dom}(\mathbf{A}))| \leq |\mathrm{dom}(\mathbf{A})|$

and $|R^{\mathbf{B}}| = |R^{h(\mathbf{A})}| \leq |R^{\mathbf{A}}|$ for all $R \in \sigma$, and the equalities hold in both "$\leq$" precisely when $f \in \mathrm{Inj}(\mathbf{A}, \mathbf{B})$, i.e., when $f \in \mathrm{Iso}(\mathbf{A}, \mathbf{B})$.

The subsequent result is often informally alluded to in the research of (graph) homomorphisms, and yet, to my best knowledge, has not been formally derived, although relevant discussion and exercises are given in [33] for $\sigma(\mathrm{G})$-structures (i.e., directed graphs) and can immediately be generalized to arbitrary vocabularies σ. For *classes* $\mathcal{F} \subseteq \mathcal{U}$, we will use the symbol $\mathcal{F}/\cong$ introduced in Sections 2.2 and 2.3 for the *set* of the representatives of the respective isomorphism types of the members in $\mathcal{F}$.

Proposition 3.8. *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$*,*

$$\hom(\mathbf{A}, \mathbf{B}) = \sum_{\mathbf{C} \in (\mathcal{U}/\cong)} \mathrm{sur}(\mathbf{A}, \mathbf{C}) \times \mathrm{inj}(\mathbf{C}, \mathbf{B}) / \mathrm{aut}(\mathbf{C}).$$

Before proving Proposition 3.8, we give the following discussion that elaborates on that given in Section 1.5 of [33].

Remark 3.16. Let $\mathbf{A}$ be a structure (or a graph, the following applies as well with "graph" in place of "structure" and accordingly "vertex set, V" in place of "domain, dom").

(a) By Remark 3.15(b), for all structures $\mathbf{B}$, we have $\mathrm{Iso}(\mathbf{A}, \mathbf{B}) = \mathrm{Inj}(\mathbf{A}, \mathbf{B}) \cap \mathrm{Sur}(\mathbf{A}, \mathbf{B})$.

(b) Let $f : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{A})$ be a function. By Remark 3.15(b), f is an injective homomorphism if and only if f is a surjective homomorphism, and if and only if f is an isomorphism. Thus, $\mathrm{inj}(\mathbf{A}, \mathbf{A}) = \mathrm{sur}(\mathbf{A}, \mathbf{A}) = \mathrm{iso}(\mathbf{A}, \mathbf{A}) = \mathrm{aut}(\mathbf{A}) > 0$.

(c) By the same argument based on functional composition presented in Remark 3.13, we have that $\mathrm{inj}(*, *)$, $\mathrm{sur}(*, *)$, $\mathrm{iso}(*, *)$ and hence $\mathrm{aut}(*)$ are all invariant under isomorphism, as is $\hom(*, *)$.

Proof of Proposition 3.8. We will mainly concern ourselves with the case for $\mathcal{U} = \mathcal{A}$. In the following discussion, we consider arbitrary structures $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{A}$.

Let $h : \mathbf{A} \to \mathbf{B}$ be a homomorphism. Note that h can be associated with

- a natural partition θ induced by the equivalence relation $\sim$ over $\mathrm{dom}(\mathbf{A})$ for which $a \sim b$ if and only if $h(a) = h(b)$ (i.e., $\theta = \{[a]_\sim \mid a \in \mathrm{dom}(\mathbf{A})\}$), such that h is the composition $i \circ s_\theta$ in which $s_\theta : \mathbf{A} \to (\mathbf{A}/\theta)$ is the surjective homomorphism that maps $a \in \mathrm{dom}(\mathbf{A})$ to $[a]_\sim$ (see Remark 3.14(b)) and $i : (\mathbf{A}/\theta) \to \mathbf{B}$ is the injective homomorphism that maps $[a]_\sim$ to $h(a)$ for all $a \in \mathrm{dom}(\mathbf{A})$, and
- a natural substructure of $\mathbf{B}$, namely the image $h(\mathbf{A})$ of $\mathbf{A}$ under h, such that h is the composition $i_{h(\mathbf{A})} \circ s$ in which $s : \mathbf{A} \to h(\mathbf{A})$ is the surjective homomorphism whose underlying mapping has the same mapping rule as h and $i_{h(\mathbf{A})} : h(\mathbf{A}) \to \mathbf{B}$ is the injective homomorphism whose underlying mapping is the inclusion mapping (see Remark 3.14(a)).

An important observation to make is that *there is an isomorphism π from $\mathbf{A}/\theta$ to $h(\mathbf{A})$* and, furthermore, *the underlying mapping of h is the composition $i_{h(\mathbf{A})} \circ (\pi \circ s_\theta)$*. Thus, h is uniquely determined by the following items (note that the items in a row depend on those in the previous row),

- an isomorphism type $[\mathbf{C}]_\cong$,
- a partition θ of $\mathrm{dom}(\mathbf{A})$ such that $(\mathbf{A}/\theta) \in [\mathbf{C}]_\cong$ as well as a substructure $\mathbf{B}'$ of $\mathbf{B}$ such that $\mathbf{B}' \in [\mathbf{C}]_\cong$ (the respective choices of θ and $\mathbf{B}'$ are mutually independent),
- an isomorphism $\pi : (\mathbf{A}/\theta) \cong \mathbf{B}'$

in such a way that $h = i_{\mathbf{B}'} \circ (\pi \circ s_\theta)$, where $s_\theta : \mathbf{A} \to (\mathbf{A}/\theta)$ is uniquely determined by the choice of θ (hence its subscript) and $i_{\mathbf{B}'} : \mathbf{B}' \to \mathbf{B}$ is uniquely determined by the choice of $\mathbf{B}'$ (hence its subscript). If h is injective, then θ is the trivial partition and hence $h = i_{\mathbf{B}'} \circ \pi$ with $\pi : \mathbf{A} \cong \mathbf{B}'$ (recall the line below Definition 2.14 that we set $\mathbf{A}/\theta := \mathbf{A}$ in such case). If h is surjective, then $\mathbf{B}' = \mathbf{B}$ and hence $h = \pi \circ s_\theta$.

Therefore, we have

$$
\begin{aligned}
& \mathrm{Hom}(\mathbf{A}, \mathbf{B}) \\
= \; & \{h \mid h : \mathbf{A} \to \mathbf{B}\} \\
= \; & \biguplus_{\mathbf{C} \in (\mathcal{A}/\cong)} \; \biguplus_{\substack{\theta \text{ a partition of } \mathrm{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{C}]_{\cong}}} \; \biguplus_{\mathbf{B}' \subseteq \mathbf{B} : \mathbf{B}' \in [\mathbf{C}]_{\cong}} \{i_{\mathbf{B}'} \circ (\pi \circ s_\theta) \mid \pi : (\mathbf{A}/\theta) \cong \mathbf{B}'\}
\end{aligned}
$$

(in fact, the second $\biguplus$ and the third $\biguplus$ can switch their positions as they are mutually independent) and

$$
\begin{aligned}
\mathrm{Inj}(\mathbf{A}, \mathbf{B}) &= \biguplus_{\mathbf{B}' \subseteq \mathbf{B} : \mathbf{B}' \in [\mathbf{A}]_{\cong}} \{i_{\mathbf{B}'} \circ \pi \mid \pi : \mathbf{A} \cong \mathbf{B}'\}, \\
\mathrm{Sur}(\mathbf{A}, \mathbf{B}) &= \biguplus_{\substack{\theta \text{ a partition of } \mathrm{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}}} \{\pi \circ s_\theta \mid \pi : (\mathbf{A}/\theta) \cong \mathbf{B}\}.
\end{aligned}
$$

Since $\mathrm{dom}(\mathbf{A})$ and $\mathrm{dom}(\mathbf{B})$ are both finite, all the above $\biguplus$ are indeed disjoint unions of *finitely many* members that are finite and nonempty. It follows that

$$
\begin{aligned}
&\operatorname{hom}(\mathbf{A},\mathbf{B})\\
=\ & \sum_{\mathbf{C}\in(\mathcal{A}/\cong)} \ \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}):\\ (\mathbf{A}/\theta)\in[\mathbf{C}]_{\cong}}} \ \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{C}]_{\cong}} |\{i_{\mathbf{B}'}\circ(\pi\circ s_\theta) \mid \pi:(\mathbf{A}/\theta)\cong\mathbf{B}'\}|\\
=\ & \sum_{\mathbf{C}\in(\mathcal{A}/\cong)} \ \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}):\\ (\mathbf{A}/\theta)\in[\mathbf{C}]_{\cong}}} \ \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{C}]_{\cong}} \operatorname{iso}(\mathbf{A}/\theta,\mathbf{B}')\\
=\ & \sum_{\mathbf{C}\in(\mathcal{A}/\cong)} \ \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}):\\ (\mathbf{A}/\theta)\in[\mathbf{C}]_{\cong}}} \ \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{C}]_{\cong}} \operatorname{iso}(\mathbf{C},\mathbf{C})\\
=\ & \sum_{\mathbf{C}\in(\mathcal{A}/\cong)} \ \sum_{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}):(\mathbf{A}/\theta)\in[\mathbf{C}]_{\cong}} \ \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{C}]_{\cong}} \operatorname{aut}(\mathbf{C})\\
=\ & \sum_{\mathbf{C}\in(\mathcal{A}/\cong)} \operatorname{aut}(\mathbf{C})\times|\{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}) \mid (\mathbf{A}/\theta)\in[\mathbf{C}]_{\cong}\}|\times|\{\mathbf{B}'\subseteq\mathbf{B}\mid\mathbf{B}'\in[\mathbf{C}]_{\cong}\}|
\end{aligned}
$$

and

$$
\begin{aligned}
&\operatorname{inj}(\mathbf{A},\mathbf{B})\\
=\ & \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{A}]_{\cong}} |\{i_{\mathbf{B}'}\circ\pi \mid \pi:\mathbf{A}\cong\mathbf{B}'\}|\\
=\ & \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{A}]_{\cong}} \operatorname{iso}(\mathbf{A},\mathbf{B}')\\
=\ & \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{A}]_{\cong}} \operatorname{iso}(\mathbf{A},\mathbf{A})\\
=\ & \sum_{\mathbf{B}'\subseteq\mathbf{B}:\mathbf{B}'\in[\mathbf{A}]_{\cong}} \operatorname{aut}(\mathbf{A})\\
=\ & \operatorname{aut}(\mathbf{A})\times|\{\mathbf{B}'\subseteq\mathbf{B}\mid\mathbf{B}'\in[\mathbf{A}]_{\cong}\}|
\end{aligned}
$$

and

$$
\begin{aligned}
& \operatorname{sur}(\mathbf{A},\mathbf{B}) \\
= & \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}}} |\{\pi \circ s_\theta \mid \pi : (\mathbf{A}/\theta) \cong \mathbf{B}\}| \\
= & \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}}} \operatorname{iso}(\mathbf{A}/\theta, \mathbf{B}) \\
= & \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}}} \operatorname{iso}(\mathbf{B}, \mathbf{B}) \\
= & \sum_{\substack{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}): \\ (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}}} \operatorname{aut}(\mathbf{B}) \\
= & \operatorname{aut}(\mathbf{B}) \times |\{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}) \mid (\mathbf{A}/\theta) \in [\mathbf{B}]_{\cong}\}|,
\end{aligned}
$$

where, in each respective group of equalities for $\operatorname{hom}(\mathbf{A},\mathbf{B}), \operatorname{inj}(\mathbf{A},\mathbf{B}), \operatorname{sur}(\mathbf{A},\mathbf{B})$, the second equality follows because every homomorphism is uniquely determined by $[\mathbf{C}]_{\cong}, \theta, \mathbf{B}', \pi$ (in which π is allowed to vary given a combination of fixed $[\mathbf{C}]_{\cong}, \theta, \mathbf{B}'$), the third follows by part (c) and the fourth by part (b) of Remark 3.16.

Since $\mathbf{A}$ and $\mathbf{B}$ are arbitrary, we have, for arbitrary structures $\mathbf{C}$, that

$$
\begin{aligned}
\operatorname{inj}(\mathbf{C},\mathbf{B}) &= \operatorname{aut}(\mathbf{C}) \times |\{\mathbf{B}' \subseteq \mathbf{B} \mid \mathbf{B}' \in [\mathbf{C}]_{\cong}\}|, \\
\operatorname{sur}(\mathbf{A},\mathbf{C}) &= \operatorname{aut}(\mathbf{C}) \times |\{\theta \text{ a partition of } \operatorname{dom}(\mathbf{A}) \mid (\mathbf{A}/\theta) \in [\mathbf{C}]_{\cong}\}|.
\end{aligned}
$$

Substituting the last two equations into the last identity concerning $\operatorname{hom}(\mathbf{A},\mathbf{B})$, we conclude

$$
\operatorname{hom}(\mathbf{A},\mathbf{B}) = \sum_{\mathbf{C} \in (\mathcal{A}/\cong)} \operatorname{sur}(\mathbf{A},\mathbf{C}) \times \operatorname{inj}(\mathbf{C},\mathbf{B})/\operatorname{aut}(\mathbf{C}),
$$

as $\operatorname{aut}(\mathbf{C}) > 0$ (by Remark 3.16(b)).

As for the case of $\mathcal{U} = \mathcal{G}$, the above can be repeated with graphs in place of structures (V in place of dom, accordingly) and $\mathcal{G}$ in place of $\mathcal{A}$. Alternatively, consider arbitrary graphs $\mathbf{A}$ and $\mathbf{B}$, and note that $\operatorname{hom}(\mathbf{A},\mathbf{B}) = \operatorname{hom}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{B}^{\sigma(\mathrm{G})})$, $\operatorname{sur}(\mathbf{A},\mathbf{B}) =$

$\mathrm{sur}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{B}^{\sigma(\mathrm{G})})$, $\mathrm{inj}(\mathbf{A}, \mathbf{B}) = \mathrm{inj}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{B}^{\sigma(\mathrm{G})})$ and $\mathrm{aut}(\mathbf{A}) = \mathrm{aut}(\mathbf{A}^{\sigma(\mathrm{G})})$, by Remarks 2.4 and 3.11. Thus,

$$\begin{aligned}
& \mathrm{hom}(\mathbf{A}, \mathbf{B}) \\
= \; & \mathrm{hom}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{B}^{\sigma(\mathrm{G})}) \\
= \; & \sum_{\mathbf{C}' \in (\mathcal{A}[\sigma(\mathrm{G})]/\cong)} \mathrm{sur}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{C}') \times \mathrm{inj}(\mathbf{C}', \mathbf{B}^{\sigma(\mathrm{G})})/\mathrm{aut}(\mathbf{C}') \\
= \; & \sum_{\substack{\mathbf{C}' \in (\mathcal{A}[\sigma(\mathrm{G})]/\cong): \\ \mathbf{C}' \text{ is irreflexive and symmetric}}} \mathrm{sur}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{C}') \times \mathrm{inj}(\mathbf{C}', \mathbf{B}^{\sigma(\mathrm{G})})/\mathrm{aut}(\mathbf{C}') \\
= \; & \sum_{\mathbf{C} \in (\mathcal{G}/\cong)} \mathrm{sur}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{C}^{\sigma(\mathrm{G})}) \times \mathrm{inj}(\mathbf{C}^{\sigma(\mathrm{G})}, \mathbf{B}^{\sigma(\mathrm{G})})/\mathrm{aut}(\mathbf{C}^{\sigma(\mathrm{G})}) \\
= \; & \sum_{\mathbf{C} \in (\mathcal{G}/\cong)} \mathrm{sur}(\mathbf{A}, \mathbf{C}) \times \mathrm{inj}(\mathbf{C}, \mathbf{B})/\mathrm{aut}(\mathbf{C})
\end{aligned}$$

in which the third identity follows because $\mathrm{sur}(\mathbf{A}^{\sigma(\mathrm{G})}, \mathbf{C}') = 0$ if $\mathbf{C}'$ is not symmetric (i.e., $(a, b) \in E^{\mathbf{C}'}$ and $(b, a) \notin E^{\mathbf{C}'}$ for some distinct $a, b \in \mathrm{dom}(\mathbf{C}')$) and $\mathrm{inj}(\mathbf{C}', \mathbf{B}^{\sigma(\mathrm{G})}) = 0$ if $\mathbf{C}'$ is not irreflexive (i.e., $(a, a) \in E^{\mathbf{C}'}$ for some $a \in \mathrm{dom}(\mathbf{C}')$) and the fourth because of the one-to-one correspondence given in Remark 2.3 (assuming $\mathbf{C} \in (\mathcal{G}/\cong)$ if and only if $\mathbf{G}^{\sigma(\mathrm{G})} \in (\mathcal{A}[\sigma(\mathrm{G})]/\cong)$ for all $\mathbf{C} \in \mathcal{G}$). □

Homomorphism and Graph-Theoretic Properties

We first look at some properties concerning homomorphisms and graphs. The proposition below is immediate by definition and so we omit a proof.

Proposition 3.9. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be graphs. If* $\mathbf{G}$ *contains a subgraph* $\mathbf{G}' \cong \mathbf{K}_n$ *for some* $n \in \mathbb{Z}^+$*, then for every homomorphism* $h : \mathbf{G} \to \mathbf{H}$*, we have* $h(\mathbf{G}') \cong \mathbf{K}_n$*.* □

Recall Definition 2.7 that the chromatic number $\chi(\mathbf{G})$ of a graph $\mathbf{G}$ is the smallest integer $n \in \mathbb{Z}^+$ for which $\mathbf{G}$ is n-colorable or, equivalently, for which $\mathrm{hom}(\mathbf{G}, \mathbf{K}_n) > 0$, by

Example 3.17.

Proposition 3.10. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be graphs. If* $\mathbf{G} \to \mathbf{H}$*, then* $\chi(\mathbf{G}) \leq \chi(\mathbf{H})$.

Proof. Let $\mathbf{G}$ and $\mathbf{H}$ be two graphs such that $\mathbf{G} \to \mathbf{H}$. Since $\to$ is transitive (see Remark 3.10(b)), we have that for all $n \in \mathbb{Z}^+$, if $\mathbf{H} \to \mathbf{K}_n$, then $\mathbf{G} \to \mathbf{K}_n$; in other words, if $\hom(\mathbf{H}, \mathbf{K}_n) > 0$, then $\hom(\mathbf{G}, \mathbf{K}_n) > 0$. $\square$

Next, we look at properties concerning homomorphisms and graphs or structures in general. Recall that a walk in a graph or a structure is a sequence $w = (a_0, \ldots, a_n)$ in which each pair of consecutive entries are adjacent.

Definition 3.22. Let $\mathbf{A}, \mathbf{B} \in \mathcal{U}$, let $h : \mathbf{A} \to \mathbf{B}$ be a homomorphism, and let $w = (a_0, \ldots, a_n)$ be a walk in $\mathbf{A}$. The *image of* w *under* h is $h(w) := (h(a_0), \ldots, h(a_n))$.

Proposition 3.11. [33] *Let* $\mathbf{A}, \mathbf{B} \in \mathcal{U}$ *and* $h : \mathbf{A} \to \mathbf{B}$ *be a homomorphism.*

(a) *For every* $n \in \mathbb{Z}^+$*, if* w *is a walk of length* n *in* $\mathbf{A}$*, then* $h(w)$ *is a walk of length* n *in* $h(\mathbf{A})$*; in particular, if* w *is a closed walk of length* n *in* $\mathbf{A}$*, then* $h(w)$ *is a closed walk of length* n *in* $h(\mathbf{A})$.

(b) *If* $\mathbf{A}$ *is connected, then* $h(\mathbf{A})$ *is also connected.*

Proof. We will only prove for the case $\mathcal{U} = \mathcal{A}$, since the case $\mathcal{U} = \mathcal{G}$ is entirely analogous.

For part (a), let $w = (a_0, \ldots, a_n)$ be a walk of length n in $\mathbf{A}$, in which every pair of consecutive entries are adjacent. Then every pair of consecutive entries in the image $h(w) = (h(a_0), \ldots, h(a_n))$ are also adjacent, by the premise that $h : \mathbf{A} \to \mathbf{B}$ is a homomorphism. It follows that $h(w) = (h(a_0), \ldots, h(a_n))$ is a walk of length n in $h(\mathbf{A})$.

Note that if w is a closed walk, i.e., if $a_0 = a_n$, then $h(a_0) = h(a_n)$ and hence $h(w)$ is a closed walk.

For part (b), let $\mathbf{A}$ be connected. For distinct elements $c, d \in \text{dom}(h(\mathbf{A}))$, choose $a \in h^{-1}(c)$ and $b \in h^{-1}(d)$ (here h^{-1} means preimage, see Section 2.1), which must be distinct. Since $\mathbf{A}$ is connected, b is reachable from a, i.e., there is a walk $w = (a_0, \ldots, a_n)$ in $\mathbf{A}$ such that $a_0 = a$ and $a_n = b$. The image $h(w) = (h(a_0), \ldots, h(a_n))$ is a walk in $h(\mathbf{A})$, by part (a). Since $h(a_0) = h(a) = c$ and $h(a_n) = h(b) = d$, we have that d is reachable from c in $h(\mathbf{A})$. Therefore, $h(\mathbf{A})$ is also connected. □

Remark 3.17. (a) For graphs, the image of a path under a homomorphism is a walk but may not be a path, and the image of a cycle under a homomorphism is a closed walk but may not be a cycle. The same holds for structures.

(b) There is a subtle difference between the image of a cycle under a homomorphism for graphs and for structures, however. The image of a cycle in a graph may be acyclic, e.g., consider the homomorphisms from $\mathbf{C}_4$ to $\mathbf{K}_2$. In contrast, the image of a cycle in a structure must contain a cycle (as a subsequence) in it, due to the definition of homomorphism; in fact, if $\mathbf{A}$ and $\mathbf{B}$ are two structures for which there is a homomorphism $h : \mathbf{A} \to \mathbf{B}$, then h gives rise to a natural mapping $h' : V(\text{Inc}(\mathbf{A})) \to V(\text{Inc}(\mathbf{B}))$ (a *homomorphism* indeed) that preserves the labels in the edges in $E(\text{Inc}(\mathbf{A}))$ and it is easy to see that the image of a cycle in $\text{Inc}(\mathbf{A})$ under h' contains a cycle.

Corollary 3.12. (a) *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be graphs such that* $\mathbf{G} \to \mathbf{H}$*. If* $\mathbf{G}$ *contains a cycle of odd length* $n \geq 3$*, then* $\mathbf{H}$ *contains a cycle of odd length* m *such that* $n \geq m \geq 3$*.*

(b) *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be structures such that* $\mathbf{A} \to \mathbf{B}$*. If* $\mathbf{A}$ *contains a cycle of odd length* $n \in \mathbb{Z}^+$*, then* $\mathbf{B}$ *contains a cycle of odd length* m *such that* $n \geq m$*.*

Proof. For part (a), let $h : \mathbf{G} \to \mathbf{H}$ and let w be a cycle in $\mathbf{G}$ of odd length $n \geq 3$. Then, the image $h(w)$ is a closed walk in $h(\mathbf{G})$ of the same length n, by Proposition 3.11(a). It follows from Proposition 2.1 that $h(\mathbf{G})$ and hence $\mathbf{H}$ contain a cycle of odd length m such that $n \geq m \geq 3$.

For part (b), note that an analogue of Proposition 2.1 holds for structures (with suitable adaptations, of course, since cycles in structures have lengths ≥ 1). Then, argue as in part (a). $\square$

Corollary 3.13. (a) *Let* $\mathbf{A}, \mathbf{B} \in \mathcal{U}$ *such that* $\mathbf{A} \to \mathbf{B}$*. If* $\gamma_{\text{odd}}(\mathbf{A}) < \infty$*, then* $\gamma_{\text{odd}}(\mathbf{A}) \geq \gamma_{\text{odd}}(\mathbf{B})$*.*

(b) *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be structures such that* $\mathbf{A} \to \mathbf{B}$*. If* $\gamma(\mathbf{A}) < \infty$*, then* $\gamma(\mathbf{A}) \geq \gamma(\mathbf{B})$*.*

Proof. Part (a) immediately follows from Corollary 3.12. Part (b) follows from Proposition 3.11(a) and Remark 3.17(b). $\square$

3.2.2 Two Types of Homomorphism Counts

As the title of this dissertation suggests, homomorphism counts will play the central role. However, there are actually two different notions of homomorphism counts: The *actual number* of homomorphisms and the *indicator for the existence* of homomorphisms (0 or 1). The first notion, $\hom(\mathbf{A}, \mathbf{B})$, was introduced in Subsection 3.2.1.

Homomorphism Counts over Two Different Semirings

Stated in a more general (and hence more abstract) sense, $\hom(\mathbf{A}, \mathbf{B})$ is the *homomorphism count from* $\mathbf{A}$ *to* $\mathbf{B}$ *over* N, where $\mathrm{N} = (\mathbb{N}, +, \cdot, 0, 1)$ is called the *bag-set semiring*. Note that a nonempty set S with two elements $0, 1 \in S$ and two binary operations $+, \cdot$ on S is a *semiring* if

- for all $a, b, c \in S$: $(a + b) + c = a + (b + c)$,
- for all $a \in S$: $0 + a = a = a + 0$,
- for all $a, b \in S$: $a + b = b + a$,
- for all $a, b, c \in S$: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$,
- for all $a \in S$: $a \cdot 1 = a = 1 \cdot a$,
- for all $a, b, c \in S$: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$,
- for all $a, b, c \in S$: $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$,
- for all $a \in S$: $0 \cdot a = 0 = a \cdot 0$.

It is easy to see that $\mathrm{B} = (\mathbb{B}, +, \cdot, 0, 1)$ is a semiring with $\mathbb{B} := \{0, 1\}$ and the two binary operations $+, \cdot$ on $\mathbb{B}$ defined by

$$\begin{aligned} a + b &= \max(a, b), \\ a \cdot b &= \min(a, b), \end{aligned}$$

for all $a, b \in \mathbb{B}$, and B is called the *Boolean semiring*. This prompts the second notion of homomorphism counts.

Notation. For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, let $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B})$ be the notation to indicate the existence of a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, i.e., $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = 1$ if $\mathbf{A} \to \mathbf{B}$, otherwise $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = 0$.

This second notion of homomorphism counts is then called, in contrast, the *homomorphism count from* $\mathbf{A}$ *to* $\mathbf{B}$ *over* B. In view of this, we sometimes write $\hom_{\mathrm{N}}(\mathbf{A}, \mathbf{B})$ for $\hom(\mathbf{A}, \mathbf{B})$ to emphasize that the former notion of homomorphism counts is over the semiring N.

Remark 3.18. We saw in Remark 3.13 that $\hom_{\mathrm{N}}(*, *) = \hom(*, *)$ is invariant under isomorphism. Since $\hom_{\mathrm{B}}(*, *) = \operatorname{sgn}(\hom_{\mathrm{N}}(*, *))$ (see Proposition 3.14(a)), we have that $\hom_{\mathrm{B}}(*, *)$ is invariant under isomorphism as well. These imply that further notions developed based on homomorphism counts over B or over N, such as the (restricted) left and right profiles introduced in Chapter 4, are also invariant under isomorphism.

The next two propositions have a straightforward proof (omitted) using the *rule of sum* and the *rule of product*. The operations $\oplus$ (direct sum) and $\otimes$ (direct product) for both graphs and structures were given in Definitions 2.3 and 2.13, while $\uparrow$ (exponentiation) is specifically for structures and was given in Definition 2.13.

Proposition 3.14. *Let* $\mathbf{A}, \mathbf{B} \in \mathcal{U}$, *and let* K *be the semiring* B *or* N.

(a) $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = \operatorname{sgn}(\hom_{\mathrm{N}}(\mathbf{A}, \mathbf{B}))$, *where* $\operatorname{sgn} : \mathbb{N} \to \mathbb{B}, \operatorname{sgn}(n) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{otherwise.} \end{cases}$

(b) (*Additivity for* $\oplus$) *If* $\mathbf{B} = \bigoplus_{j=1}^{n} \mathbf{B}_j$ ($\mathbf{B}_1, \ldots, \mathbf{B}_n$ *are not necessarily the connected components of* $\mathbf{B}$), *then* $\hom_K(\mathbf{A}, \mathbf{B}) \geq \sum_{j=1}^{n} \hom_K(\mathbf{A}, \mathbf{B}_j)$; *the equality holds if* $\mathbf{A}$ *is connected.*

(c) (*Multiplicativity for* $\oplus$) *If* $\mathbf{A} = \bigoplus_{i=1}^{m} \mathbf{A}_i$ ($\mathbf{A}_1, \ldots, \mathbf{A}_m$ *are not necessarily the connected components of* $\mathbf{A}$), *then* $\hom_K(\mathbf{A}, \mathbf{B}) = \prod_{i=1}^{m} \hom_K(\mathbf{A}_i, \mathbf{B})$.

(d) *If* $\mathbf{A} = \bigotimes_{i=1}^{m} \mathbf{A}_i$*, then* $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{A}_i) = 1$ *for all* $i \in [m]$.

(e) (*Multiplicativity for* $\otimes$) *If* $\mathbf{B} = \bigotimes_{j=1}^{n} \mathbf{B}_j$*, then* $\hom_K(\mathbf{A}, \mathbf{B}) = \prod_{j=1}^{n} \hom_K(\mathbf{A}, \mathbf{B}_j)$. □

Proposition 3.15. [33] *Let* K *be the semiring* B *or* N. *For all structures* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{C}$, *we have* $\hom_K(\mathbf{A} \otimes \mathbf{C}, \mathbf{B}) = \hom_K(\mathbf{C}, \mathbf{B} \uparrow \mathbf{A})$. □

Corollary 3.16. *If* $\mathbf{G}$ *is a* 2*-colorable graph with* n *connected components where* $n \in \mathbb{Z}^+$, *then the number of* 2*-colorings of* $\mathbf{G}$ *is* 2^n.

Proof. If $\mathbf{G}$ is 2-colorable with connected components $\mathbf{G}_1, \ldots, \mathbf{G}_n$, then the number of 2-colorings of each $\mathbf{G}_i$ is 2 by Proposition 2.3, and hence $\hom(\mathbf{G}_i, \mathbf{K}_2) = 2$ by Example 3.17. By Proposition 3.14(c), it follows that the number of 2-colorings of $\mathbf{G}$ is $\hom(\mathbf{G}, \mathbf{K}_2) = \hom(\mathbf{G}_1, \mathbf{K}_2) \times \cdots \times \hom(\mathbf{G}_n, \mathbf{K}_2) = 2^n$. □

Remark 3.19. By Example 3.17 and Proposition 3.14(a), it follows that *for every* $n \in \mathbb{Z}^+$ *and every graph* $\mathbf{G}$, *we have* $\mathbf{G}$ *is* n*-colorable if and only if* $\hom_{\mathrm{B}}(\mathbf{G}, \mathbf{K}_n) = 1$. Recall Example 2.1(b) that for all $m, n \in \mathbb{Z}^+$ with $m > n$ and for all graphs $\mathbf{G}$, if $\mathbf{G}$ is n-colorable, then $\mathbf{G}$ is also m-colorable. This also follows from the previous statement, in view of Remark 3.12.

From the preceding remark it turns out that the decision problem n-`COLORABILITY` coincides with the problem: Given a graph $\mathbf{G}$, is $\hom_{\mathrm{B}}(\mathbf{G}, \mathbf{K}_n) = 1$? The time is ripe for us to look at a more general category of decision problems.

The (*uniform*) *constraint satisfaction problem* $\mathtt{CSP}(\mathbf{A}, \mathbf{B})$ is a decision problem that is parameterized by the two structures $\mathbf{A}$ and $\mathbf{B}$, in which the *scope* of the constraints is delineated by the relations in $\mathbf{A}$ and other part of the constraints by the relations in

$\mathbf{B}$, and that asks whether there exists an assignment of the elements of $\mathbf{A}$ to the elements of $\mathbf{B}$ such that the assignment satisfies all the constraints (see Chapter 6 of [27] for more details). This definition applies to graphs as well, with "vertices" and "edges" in place of "elements" and "relations", respectively. As the reader may have noticed, $\texttt{CSP}(\mathbf{A}, \mathbf{B})$ simply asks: Given $\mathbf{A}$ and $\mathbf{B}$, is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ or, in symbols, is $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = 1$?

Constraint satisfaction problems are a big family of decision problems in computer science and provide a general framework in which to formulate and study various decision problems. There are two variants of *nonuniform constraint satisfaction problem*: fixing $\mathbf{A}$ or fixing $\mathbf{B}$ in $\texttt{CSP}(\mathbf{A}, \mathbf{B})$.

We shall consider the corresponding class of the yes-instances of either variant of nonuniform constraint satisfaction problem so that, conversely, the nonuniform constraint satisfaction problem becomes the *membership problem* for the corresponding class.

Notations. Let $\mathbf{A} \in \mathcal{U}$.

(a) $\mathrm{CQ}(\mathbf{A}) := \{\mathbf{B} \in \mathcal{U} \mid \hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = 1\}$.

(b) $\mathrm{CSP}(\mathbf{A}) := \{\mathbf{B} \in \mathcal{U} \mid \hom_{\mathrm{B}}(\mathbf{B}, \mathbf{A}) = 1\}$.

Remark 3.20. The notation CQ stems from (*Boolean*) *conjunctive query* in the terminology of database theory, which is also known as a *primitive positive sentence*, a first-order sentence that has a special syntax (see Definition 5.2 and also Remark 5.2). For every $\mathbf{A}$ that is a nonempty structure (i.e., contains at least one tuple in a relation) or a nonempty graph (i.e., is not an independent set), the class $\{\mathbf{B} \in \mathcal{U} \mid \hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) = 1\}$ coincides with $\mathrm{Mod}(\varphi)$ for some primitive positive sentence φ related to $\mathbf{A}$ (see Proposition 5.9(b) and

Remark 5.3), and this explains our choice of CQ($\mathbf{A}$) for the class.

As can be inferred from the previous discussion, in general the computational complexity of the membership problem for CSP($\mathbf{A}$), also known as *expression complexity*, is NP-complete, since the membership problem for CSP($\mathbf{K}_3$) is indeed the well-known NP-complete problem 3-COLORABILITY (recall Section 2.2). In contrast, the computational complexity of the membership problem for CQ($\mathbf{A}$), also known as *data complexity*, is in P: Given an input $\mathbf{B}$, an intuitive algorithm tests, for every mapping $h : \mathrm{dom}(\mathbf{A}) \to \mathrm{dom}(\mathbf{B})$, whether h is a homomorphism; the total number of such mappings is $O(|\mathrm{dom}(\mathbf{B})|^{|\mathrm{dom}(\mathbf{A})|})$.

Homomorphic Equivalence

We saw in Remark 3.10 that the binary relation $\to$ over $\mathcal{U}$ is a preorder, namely, $\to$ is reflexive and transitive. In general, however, it is not symmetric. For example, in the case $\mathcal{U} = \mathcal{G}$, we have $\mathbf{I}_1 \to \mathbf{K}_2$ but $\mathbf{K}_2 \not\to \mathbf{I}_1$, by Proposition 3.9.

Notation. For all $\mathbf{A}$ and $\mathbf{B}$, we let $\mathbf{A} \leftrightarrow \mathbf{B}$ denote the statement "$\mathbf{A} \to \mathbf{B}$ and $\mathbf{B} \to \mathbf{A}$."

It turns out that $\leftrightarrow$ is also binary relation over $\mathcal{U}$ and, on top of that, it is reflexive and transitive (by the reflexivity and transitivity of $\to$), and moreover symmetric (due to the symmetry imposed by the statement for $\leftrightarrow$). That is to say, $\leftrightarrow$ *is an equivalence relation over* $\mathcal{U}$, called *homomorphic equivalence.*

Definition 3.23. Let $\mathbf{A}, \mathbf{B} \in \mathcal{U}$. We say that $\mathbf{A}$ *is homomorphically equivalent to* $\mathbf{B}$ if $\mathbf{A} \leftrightarrow \mathbf{B}$.

Notation. For $\mathbf{A} \in \mathcal{U}$, the *homomorphic-equivalence class of* $\mathbf{A}$ is denoted $[\mathbf{A}]_{\leftrightarrow} := \{\mathbf{B} \in \mathcal{U} \mid \mathbf{A} \leftrightarrow \mathbf{B}\}$.

Note that for all $\mathbf{A} \in \mathcal{U}$, we have $[\mathbf{A}]_{\leftrightarrow} = \mathrm{CQ}(\mathbf{A}) \cap \mathrm{CSP}(\mathbf{A})$.

Example 3.19. The two graphs $\mathbf{C}_3$ and $\mathbf{C}_3 \oplus \mathbf{C}_3$ are homomorphically equivalent.

This example reveals a general property about homomorphic equivalence and direct sums of graphs or of structures. We summarize some trivial properties below without proofs (as they are straightforward).

Proposition 3.17. *Let $\mathbf{A} \in \mathcal{U}$. For all $m, n \in \mathbb{Z}^+$, we have that $\bigoplus_m \mathbf{A}$ and $\bigoplus_n \mathbf{A}$ are homomorphically equivalent.* $\square$

Proposition 3.18. *For all $\mathbf{A}, \mathbf{B} \in \mathcal{U}$, the following are equivalent:*

(i) $\mathbf{A} \leftrightarrow \mathbf{B}$.

(ii) *For all $\mathbf{C} \in \mathcal{U}$, we have $\mathbf{C} \to \mathbf{A}$ if and only if $\mathbf{C} \to \mathbf{B}$.*

(iii) *For all $\mathbf{C} \in \mathcal{U}$, we have $\mathbf{A} \to \mathbf{C}$ if and only if $\mathbf{B} \to \mathbf{C}$.*

(iv) $\mathrm{CQ}(\mathbf{A}) = \mathrm{CQ}(\mathbf{B})$.

(v) $\mathrm{CSP}(\mathbf{A}) = \mathrm{CSP}(\mathbf{B})$. $\square$

Chapter 4

Graph Isomorphism and Its Relaxations

We will delve deeper into homomorphism counts (over B or over N) and investigate their expressive power in this chapter and the next. Certain homomorphism counts from or to a graph or a structure present an ensemble that reveals some interesting structural information about that graph or structure, and they will be the main objects of study next: the left profiles and the right profiles.

Recall the symbol $\mathcal{U}$ in Proviso 2 for $\mathcal{G}$ or $\mathcal{A}[\sigma]$ for some vocabulary σ and the set $\mathcal{F}/\cong$ corresponding to a subclass $\mathcal{F} \subseteq \mathcal{U}$, both of which were introduced in Chapter 2.

Definition 4.1. Let $\mathbf{A} \in \mathcal{U}$, and let K be the semiring B or N.

(a) The *left profile of* $\mathbf{A}$ *over* K is $\mathrm{lpf}_K(\mathbf{A}) := (\hom_K(\mathbf{F}, \mathbf{A}) \mid \mathbf{F} \in \mathcal{U}/\cong)$, and the *right profile of* $\mathbf{A}$ *over* K is $\mathrm{rpf}_K(\mathbf{A}) := (\hom_K(\mathbf{A}, \mathbf{F}) \mid \mathbf{F} \in \mathcal{U}/\cong)$.

(b) For every nonempty subclass $\mathcal{F} \subseteq \mathcal{U}$, the *left profile of* $\mathbf{A}$ *restricted to* $\mathcal{F}$ *over* K is

$\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) := (\hom_K(\mathbf{F}, \mathbf{A}) \mid \mathbf{F} \in \mathcal{F}/{\cong})$, and the *right profile of* $\mathbf{A}$ *restricted to* $\mathcal{F}$ *over* K is $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}) := (\hom_K(\mathbf{A}, \mathbf{F}) \mid \mathbf{F} \in \mathcal{F}/{\cong})$.

Note that, by notations, we have $\mathrm{lpf}_K(\mathbf{A}) = \mathrm{lpf}_K^{\mathcal{U}}(\mathbf{A})$ and $\mathrm{rpf}_K(\mathbf{A}) = \mathrm{rpf}_K^{\mathcal{U}}(\mathbf{A})$, for every $\mathbf{A} \in \mathcal{U}$. Since sequences are an alternative view of functions (cf. Section 2.1), the left and right profiles of an $\mathbf{A} \in \mathcal{U}$ restricted to a class $\mathcal{F}$ over a semiring $K \in \{\mathrm{B}, \mathrm{N}\}$ are indeed the mappings $\mathbf{B} \mapsto \hom_K(\mathbf{B}, \mathbf{A})$ and $\mathbf{B} \mapsto \hom_K(\mathbf{A}, \mathbf{B})$, respectively, for every $\mathbf{B} \in \mathcal{F}$.

As already mentioned in Remark 3.18, homomorphism counts, over either semiring B or N, are invariant under isomorphism. Hence, so are the left profiles and the right profiles: For all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, if $\mathbf{A} \cong \mathbf{B}$ then $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{B})$ and $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}) = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{B})$.

Notations. When the underlying semiring is the bag-set semiring N, for brevity we often omit its occurrence as the subscript in the notations $\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}, \mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}, \mathrm{lpf}_{\mathrm{N}}, \mathrm{rpf}_{\mathrm{N}}$ and write $\mathrm{lpf}^{\mathcal{F}}, \mathrm{rpf}^{\mathcal{F}}, \mathrm{lpf}, \mathrm{rpf}$ instead.

Example 4.1. By Examples 3.15 and 3.17, for every graph $\mathbf{G}$, the left profile $\mathrm{lpf}^{\mathcal{I}}(\mathbf{G})$ of $\mathbf{G}$ restricted to the class $\mathcal{I}$ of independent sets is the sequence $|V(\mathbf{G})|, |V(\mathbf{G})|^2, |V(\mathbf{G})|^3, \ldots$ and the right profile $\mathrm{rpf}^{\mathcal{K}}(\mathbf{G})$ of $\mathbf{G}$ restricted to the class $\mathcal{K}$ of cliques is the sequence of numbers of n-colorings of $\mathbf{G}$ for $n \in \mathbb{Z}^+$ (assuming that the graphs in $\mathcal{I}$ or in $\mathcal{K}$ are enumerated in increasing size).

By Remark 3.19, the right profile $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{K}}(\mathbf{G})$ of $\mathbf{G}$ restricted to $\mathcal{K}$ over the Boolean semiring B is a bit-sequence in which the n-th entry is $\mathrm{I}(\mathbf{G}$ is n-colorable$)$ for $n \in \mathbb{Z}^+$ (see Section 2.1 for the definition of indicator function).

In this chapter, we will focus on graphs and investigate the expressive power of homomorphism counts in characterizing various equivalence relations among graphs that are coarser than isomorphism, in the form of restricted left or right profiles. Much of the material is from [3].

4.1 Characterizing Graph Isomorphism in Left Profile and in Right Profile

As noted in Remark 3.10(a), an isomorphism is also a homomorphism (but not vice versa). In fact, there is a deeper connection between the two. The following theorem, due to L. Lovász, is a seminal result that has encouraged the study on characterizing various equivalence relations that are coarser than isomorphism in left profiles restricted to certain classes.

Theorem 4.1 (Lovász Theorem). [40] *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$, *we have* $\mathbf{A} \cong \mathbf{B}$ *if and only if* $\operatorname{lpf}(\mathbf{A}) = \operatorname{lpf}(\mathbf{B})$. □

The above theorem gives a characterization of isomorphism in left profile: *Isomorphism coincides with the equality of left profile.* Symmetrically, the next theorem, due to S. Chaudhuri and M. Vardi, gives a characterization of isomorphism in right profile: *Isomorphism coincides with the equality of right profile.*

Theorem 4.2 (Chaudhuri-Vardi Theorem). [9] *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$, *we have* $\mathbf{A} \cong \mathbf{B}$ *if and only if* $\operatorname{rpf}(\mathbf{A}) = \operatorname{rpf}(\mathbf{B})$. □

Remark 4.1. (a) Note the resemblance of the restatement of Proposition 3.4 given in Remark 3.2 to Theorems 4.1 and 4.2. Thus, if we enumerate the isomorphism types of graphs or of structures $[\mathbf{A}_1]_{\cong}, [\mathbf{A}_2]_{\cong}, \ldots$ and visualize the mapping $\hom(*,*)$ as a 2-dimensional infinite matrix

$$
\begin{array}{c|cccccc}
\hom(*,*) & [\mathbf{A}_1]_{\cong} & [\mathbf{A}_2]_{\cong} & \cdots & [\mathbf{A}_j]_{\cong} & \cdots \\
\hline
[\mathbf{A}_1]_{\cong} & \hom(\mathbf{A}_1,\mathbf{A}_1) & \hom(\mathbf{A}_1,\mathbf{A}_2) & \cdots & \hom(\mathbf{A}_1,\mathbf{A}_j) & \cdots \\
[\mathbf{A}_2]_{\cong} & \hom(\mathbf{A}_2,\mathbf{A}_1) & \hom(\mathbf{A}_2,\mathbf{A}_2) & \cdots & \hom(\mathbf{A}_2,\mathbf{A}_j) & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\
[\mathbf{A}_i]_{\cong} & \hom(\mathbf{A}_i,\mathbf{A}_1) & \hom(\mathbf{A}_i,\mathbf{A}_2) & \cdots & \hom(\mathbf{A}_i,\mathbf{A}_j) & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots
\end{array}
$$

then Theorems 4.1 and 4.2 say that *every column in the matrix is unique* and *every row in the matrix is unique*, respectively.

(b) Theorems 4.1 and 4.2 are no longer valid if lpf and rpf are replaced by $\mathrm{lpf}_{\mathrm{B}}$ and $\mathrm{rpf}_{\mathrm{B}}$: For every $\mathbf{A} \in \mathcal{U}$, we have $\mathrm{lpf}_{\mathrm{B}}(\mathbf{A}) = \mathrm{lpf}_{\mathrm{B}}(\mathbf{A} \oplus \mathbf{A})$ and $\mathrm{rpf}_{\mathrm{B}}(\mathbf{A}) = \mathrm{rpf}_{\mathrm{B}}(\mathbf{A} \oplus \mathbf{A})$, a consequence of $\mathbf{A}$ and $\mathbf{A} \oplus \mathbf{A}$ being *homomorphic equivalent* (by Proposition 3.17) but obviously $\mathbf{A}$ and $\mathbf{A} \oplus \mathbf{A}$ are not isomorphic.

In view of Definition 3.12, C-equivalence ($\equiv_{\mathrm{C}}$) is the equivalence relation over $\mathcal{U}$ that arises from satisfying the same C-sentences. By Remark 3.6, however, C-equivalence coincides with FO-equivalence, which in turn coincides with isomorphism, by Proposition 3.4.

Corollary 4.3. *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$*, the following are equivalent:*

(i) $\mathbf{A} \cong \mathbf{B}$.

(ii) $\mathbf{A} \equiv_{\mathrm{FO}} \mathbf{B}$.

(iii) $\mathbf{A} \equiv_{\mathrm{C}} \mathbf{B}$.

(iv) $\mathrm{lpf}^{\mathcal{U}}(\mathbf{A}) = \mathrm{lpf}^{\mathcal{U}}(\mathbf{B})$.

(v) $\mathrm{rpf}^{\mathcal{U}}(\mathbf{A}) = \mathrm{rpf}^{\mathcal{U}}(\mathbf{B})$. □

We will state and prove a result that is simultaneously more general than Theorems 4.1 and 4.2 for $\mathcal{U} = \mathcal{G}$, adopting, for arbitrary classes $\mathcal{F}$ of graphs, the notations

$$\begin{aligned}
\mathrm{Inj}(\mathcal{F}) &:= \{\mathbf{G} \in \mathcal{G} \mid \mathrm{inj}(\mathbf{G}, \mathbf{F}) > 0 \text{ for some } \mathbf{F} \in \mathcal{F}\}, \\
\mathrm{Sur}(\mathcal{F}) &:= \{\mathbf{G} \in \mathcal{G} \mid \mathrm{sur}(\mathbf{F}, \mathbf{G}) > 0 \text{ for some } \mathbf{F} \in \mathcal{F}\}, \\
\mathrm{Ext}(\mathcal{F}) &:= \mathrm{Inj}(\mathcal{F}) \cap \mathrm{Sur}(\mathcal{F}).
\end{aligned}$$

In other words, $\mathrm{Inj}(\mathcal{F})$ consists of graphs isomorphic to the subgraphs of all graphs in $\mathcal{F}$ (see Remark 3.14(a)), whereas $\mathrm{Sur}(\mathcal{F})$ consists of graphs isomorphic to the homomorphic images of all graphs in $\mathcal{F}$. Thus, it is obvious that $\mathcal{F} \subseteq \mathrm{Inj}(\mathcal{F})$ and $\mathcal{F} \subseteq \mathrm{Sur}(\mathcal{F})$ and, as a result, that $\mathcal{F} \subseteq \mathrm{Ext}(\mathcal{F})$. Furthermore, it is immediate by definition and Remark 3.15(a) that $\mathrm{Sur}(\mathrm{Sur}(\mathcal{F})) = \mathrm{Sur}(\mathcal{F})$ and $\mathrm{Inj}(\mathrm{Inj}(\mathcal{F})) = \mathrm{Inj}(\mathcal{F})$.

Theorem 4.4. *Let $\mathcal{F} \subseteq \mathcal{G}$ be nonempty. For all $\mathbf{G}$ and $\mathbf{H}$ in $\mathcal{F}$, the following are equivalent:*

(i) $\mathbf{G} \cong \mathbf{H}$.

(ii) $\mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$.

(iii) $\mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$.

Proof. We assume a linear order $<$ over the set $\mathcal{G}/\cong$ of graphs that is in increasing number of vertices and, among graphs having the same number of vertices, is in increasing number of edges, for which $\mathbf{G} < \mathbf{H}$ means that $\mathbf{G}$ precedes $\mathbf{H}$ in this linear order.

Let $\mathcal{F}$ be a nonempty class of graphs, and let $\mathbf{G}, \mathbf{H} \in \mathcal{F}$. Note that the directions from (i) to (ii) and from (i) to (iii) are both trivial because lpf and rpf are invariant under isomorphism (see Remark 3.18).

To prove (ii) implies (i), assume that $\mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$. It suffices to argue that $\mathrm{inj}(\mathbf{G},\mathbf{H}) > 0$ and $\mathrm{inj}(\mathbf{H},\mathbf{G}) > 0$: Then, by Remark 3.15(b), $|V(\mathbf{G})| = |V(\mathbf{H})|$ and $|E(\mathbf{G})| = |E(\mathbf{H})|$, hence an injective homomorphism $i : \mathbf{G} \to \mathbf{H}$ (there exists one by $\mathrm{inj}(\mathbf{G},\mathbf{H}) > 0$) is also an isomorphism $i : \mathbf{G} \cong \mathbf{H}$. For this goal, we will prove

$$\text{for every } \mathbf{F} \in \mathrm{Ext}(\mathcal{F}), \text{ it holds that } \mathrm{inj}(\mathbf{F},\mathbf{G}) = \mathrm{inj}(\mathbf{F},\mathbf{H}). \tag{+}$$

Then, since $\mathbf{G}$ and $\mathbf{H}$ are both in $\mathcal{F} \subseteq \mathrm{Ext}(\mathcal{F})$, by setting $\mathbf{F} = \mathbf{G}$ and setting $\mathbf{F} = \mathbf{H}$, respectively, we have $\mathrm{inj}(\mathbf{G},\mathbf{H}) = \mathrm{inj}(\mathbf{G},\mathbf{G}) > 0$ and $\mathrm{inj}(\mathbf{H},\mathbf{G}) = \mathrm{inj}(\mathbf{H},\mathbf{H}) > 0$ by Remark 3.16(b). Let $\mathbf{F} \in \mathrm{Ext}(\mathcal{F})$, by Proposition 3.8 we have

$$\begin{aligned}
\mathrm{hom}(\mathbf{F},\mathbf{G}) &= \sum_{\mathbf{E} \in \mathcal{G}/\cong} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{G})/\mathrm{aut}(\mathbf{E}) \\
&= \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{G})/\mathrm{aut}(\mathbf{E}) \\
&= \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong :\, \mathbf{E} < \mathbf{F} \text{ or } \mathbf{E} \cong \mathbf{F}} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{G})/\mathrm{aut}(\mathbf{E}) \\
&= \mathrm{inj}(\mathbf{F},\mathbf{G}) + \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong :\, \mathbf{E}<\mathbf{F}} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{G})/\mathrm{aut}(\mathbf{E}),
\end{aligned}$$

in which the second identity follows by considering those $\mathbf{E}$ in $\mathcal{G}/\cong$ for which $\mathrm{inj}(\mathbf{E},\mathbf{G}) > 0$ and $\mathrm{sur}(\mathbf{F},\mathbf{E}) > 0$ ($\mathrm{inj}(\mathbf{E},\mathbf{G}) > 0$ and $\mathbf{G} \in \mathcal{F}$ imply $\mathbf{E} \in \mathrm{Inj}(\mathcal{F})$, and $\mathrm{sur}(\mathbf{F},\mathbf{E}) > 0$ and $\mathbf{F} \in \mathrm{Ext}(\mathcal{F}) \subseteq \mathrm{Sur}(\mathcal{F})$ imply $\mathbf{E} \in \mathrm{Sur}(\mathrm{Sur}(\mathcal{F})) = \mathrm{Sur}(\mathcal{F})$, hence altogether we have $\mathbf{E} \in \mathrm{Ext}(\mathcal{F})$), the third by noting that if $\mathrm{sur}(\mathbf{F},\mathbf{E}) > 0$ then it must be either $\mathbf{E} < \mathbf{F}$ or $\mathbf{E} \cong \mathbf{F}$ because $|V(\mathbf{E})| \leq |V(\mathbf{F})|$ and $|E(\mathbf{E})| \leq |E(\mathbf{F})|$ by Remark 3.15(b), and the fourth by Remark 3.16(b) and (c). Thus,

$$\mathrm{inj}(\mathbf{F},\mathbf{G}) = \mathrm{hom}(\mathbf{F},\mathbf{G}) - \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong :\, \mathbf{E}<\mathbf{F}} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{G})/\mathrm{aut}(\mathbf{E}),$$

and likewise,

$$\mathrm{inj}(\mathbf{F},\mathbf{H}) = \hom(\mathbf{F},\mathbf{H}) - \sum_{\mathbf{E}\,\in\,\mathrm{Ext}(\mathcal{F})/\cong:\,\mathbf{E}<\mathbf{F}} \mathrm{sur}(\mathbf{F},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{H})/\mathrm{aut}(\mathbf{E}),$$

by which (+) can be easily proved by (strong) induction on the position of the representative of $[\mathbf{F}]_{\cong}$ in the linear order $<$ restricted to $\mathrm{Ext}(\mathcal{F})/\cong$ (and by Remark 3.16(c)), given that $\mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{lpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$.

Finally, the argument for the direction from (iii) to (i) will be dual to the previous one, with inj and sur (as well as Inj and Sur) switching their roles. Assume that $\mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$. It suffices to show that $\mathrm{sur}(\mathbf{G},\mathbf{H}) > 0$ and $\mathrm{sur}(\mathbf{H},\mathbf{G}) > 0$ because then, by Remark 3.15(b), $|V(\mathbf{G})| = |V(\mathbf{H})|$ and $|E(\mathbf{G})| = |E(\mathbf{H})|$, which implies that a surjective homomorphism $s : \mathbf{G} \to \mathbf{H}$ (whose existence is guaranteed by $\mathrm{sur}(\mathbf{G},\mathbf{H}) > 0$) is also an isomorphism $s : \mathbf{G} \cong \mathbf{H}$. Similarly, for this goal, we will prove

$$\text{for every } \mathbf{F} \in \mathrm{Ext}(\mathcal{F}) \text{, it holds that } \mathrm{sur}(\mathbf{G},\mathbf{F}) = \mathrm{sur}(\mathbf{H},\mathbf{F}). \tag{$*$}$$

It will follow, as $\mathbf{G}$ and $\mathbf{H}$ are both in $\mathcal{F} \subseteq \mathrm{Ext}(\mathcal{F})$, by respectively taking $\mathbf{F} = \mathbf{H}$ and $\mathbf{F} = \mathbf{G}$, that $\mathrm{sur}(\mathbf{G},\mathbf{H}) = \mathrm{sur}(\mathbf{H},\mathbf{H}) > 0$ and $\mathrm{sur}(\mathbf{H},\mathbf{G}) = \mathrm{sur}(\mathbf{G},\mathbf{G}) > 0$ by Remark 3.16(b). For $\mathbf{F} \in \mathrm{Ext}(\mathcal{F})$, it is true by Proposition 3.8 that

$$\begin{aligned}
\hom(\mathbf{G},\mathbf{F}) &= \sum_{\mathbf{E}\,\in\,\mathcal{G}/\cong} \mathrm{sur}(\mathbf{G},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{F})/\mathrm{aut}(\mathbf{E}) \\
&= \sum_{\mathbf{E}\,\in\,\mathrm{Ext}(\mathcal{F})/\cong} \mathrm{sur}(\mathbf{G},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{F})/\mathrm{aut}(\mathbf{E}) \\
&= \sum_{\mathbf{E}\,\in\,\mathrm{Ext}(\mathcal{F})/\cong:\,\mathbf{E}\,<\,\mathbf{F} \text{ or } \mathbf{E}\,\cong\,\mathbf{F}} \mathrm{sur}(\mathbf{G},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{F})/\mathrm{aut}(\mathbf{E}) \\
&= \mathrm{sur}(\mathbf{G},\mathbf{F}) + \sum_{\mathbf{E}\,\in\,\mathrm{Ext}(\mathcal{F})/\cong:\,\mathbf{E}<\mathbf{F}} \mathrm{sur}(\mathbf{G},\mathbf{E}) \times \mathrm{inj}(\mathbf{E},\mathbf{F})/\mathrm{aut}(\mathbf{E}),
\end{aligned}$$

where the second equality is obtained by similar considerations that $\mathrm{inj}(\mathbf{E},\mathbf{F}) > 0$ and $\mathrm{sur}(\mathbf{G},\mathbf{E}) > 0$ ($\mathrm{inj}(\mathbf{E},\mathbf{F}) > 0$ and $\mathbf{F} \in \mathrm{Ext}(\mathcal{F}) \subseteq \mathrm{Inj}(\mathcal{F})$ imply $\mathbf{E} \in \mathrm{Inj}(\mathrm{Inj}(\mathcal{F})) = \mathrm{Inj}(\mathcal{F})$, and $\mathrm{sur}(\mathbf{G}, \mathbf{E}) > 0$ and $\mathbf{G} \in \mathcal{F}$ imply $\mathbf{E} \in \mathrm{Sur}(\mathcal{F})$, which together imply $\mathbf{E} \in \mathrm{Ext}(\mathcal{F})$), the third follows by observing $\mathrm{inj}(\mathbf{E}, \mathbf{F}) > 0$ implies that either $\mathbf{E} < \mathbf{F}$ or $\mathbf{E} \cong \mathbf{F}$ because $|V(\mathbf{E})| \leq |V(\mathbf{F})|$ and $|E(\mathbf{E})| \leq |E(\mathbf{F})|$ by Remark 3.15(b), and the fourth likewise by Remark 3.16(b) and (c). Therefore,

$$\mathrm{sur}(\mathbf{G}, \mathbf{F}) = \mathrm{hom}(\mathbf{G}, \mathbf{F}) - \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong :\, \mathbf{E} < \mathbf{F}} \mathrm{sur}(\mathbf{G}, \mathbf{E}) \times \mathrm{inj}(\mathbf{E}, \mathbf{F})/\mathrm{aut}(\mathbf{E}),$$

and similarly,

$$\mathrm{sur}(\mathbf{H}, \mathbf{F}) = \mathrm{hom}(\mathbf{H}, \mathbf{F}) - \sum_{\mathbf{E} \in \mathrm{Ext}(\mathcal{F})/\cong :\, \mathbf{E} < \mathbf{F}} \mathrm{sur}(\mathbf{H}, \mathbf{E}) \times \mathrm{inj}(\mathbf{E}, \mathbf{F})/\mathrm{aut}(\mathbf{E}),$$

and with these equations we can easily prove $(*)$ by (strong) induction on the position of the representative of $[\mathbf{F}]_{\cong}$ in the linear order $<$ restricted to $\mathrm{Ext}(\mathcal{F})/\cong$ (in view of Remark 3.16(c)), given the condition that $\mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{G}) = \mathrm{rpf}^{\mathrm{Ext}(\mathcal{F})}(\mathbf{H})$. $\square$

This theorem is indeed a simultaneous generalization of Theorems 4.1 and 4.2 because it is clear that $\mathrm{Ext}(\mathcal{F}) = \mathcal{F}$ for the case $\mathcal{F} = \mathcal{G}$ (here $\mathrm{lpf} = \mathrm{lpf}^{\mathcal{G}}$ and $\mathrm{rpf} = \mathrm{rpf}^{\mathcal{G}}$ are the full left profile and the full right profile, respectively). As an immediate consequence of Theorem 4.4, in general, we have *for every nonempty class $\mathcal{F}$ of graphs, if* $\mathrm{Ext}(\mathcal{F}) = \mathcal{F}$, *then isomorphism over $\mathcal{F}$ is characterized by the equality of* $\mathrm{lpf}^{\mathcal{F}}$ *and by the equality of* $\mathrm{rpf}^{\mathcal{F}}$.

Remark 4.2. We present some common classes $\mathcal{F}$ for which $\mathrm{Ext}(\mathcal{F}) = \mathcal{F}$ holds.

(a) If $\mathrm{Inj}(\mathcal{F}) \subseteq \mathcal{F}$ or $\mathrm{Sur}(\mathcal{F}) \subseteq \mathcal{F}$ holds, then $\mathrm{Ext}(\mathcal{F}) = \mathcal{F}$ holds as well because $\mathrm{Ext}(\mathcal{F}) = \mathrm{Inj}(\mathcal{F}) \cap \mathrm{Sur}(\mathcal{F})$. In fact, $\mathrm{Inj}(\mathcal{F}) = \mathcal{F}$ holds for the following two cases:

- $\mathcal{F}$ is the class of all n-colorable graphs, for a fixed $n \in \mathbb{Z}^+$ (by Proposition 2.2(a))
- $\mathcal{F}$ is the class of all graphs of degree $\leq n$, for a fixed $n \in \mathbb{Z}^+$,

and $\mathrm{Sur}(\mathcal{F}) = \mathcal{F}$ holds for the following two cases:

- $\mathcal{F}$ is the class of all connected graphs (by Proposition 3.11(b))
- $\mathcal{F}$ is the class $\mathcal{K}$ of all cliques (by Proposition 3.9).

(b) Two other cases for classes $\mathcal{F}$ of *connected* graphs for which $\mathrm{Ext}(\mathcal{F}) = \mathcal{F}$ are:

- $\mathcal{F}$ is the class $\mathcal{P}$ of all path graphs,
- $\mathcal{F}$ is the class $\mathcal{T}$ of all free trees.

We shall only argue that $\mathrm{Ext}(\mathcal{P}) = \mathcal{P}$, the case for $\mathcal{T}$ being analogous. Since $\mathcal{P} \subseteq \mathrm{Ext}(\mathcal{P})$, it suffices to show $\mathrm{Ext}(\mathcal{P}) \subseteq \mathcal{P}$. First, observe that graphs in $\mathcal{P}$ are all connected and hence so are their homomorphic images, by Proposition 3.11(b). In other words, the graphs in $\mathrm{Sur}(\mathcal{P})$ are connected. Next, $\mathrm{Inj}(\mathcal{P})$ contains all graphs isomorphic to the subgraphs of path graphs. Therefore, every graph in $\mathrm{Ext}(\mathcal{P}) = \mathrm{Inj}(\mathcal{P}) \cap \mathrm{Sur}(\mathcal{P})$ is isomorphic to a connected subgraph of a graph in $\mathcal{P}$, and must be a path graph itself. It follows that $\mathrm{Ext}(\mathcal{P}) \subseteq \mathcal{P}$.

Although we only proved Theorem 4.4 for graphs (i.e., for $\mathcal{U} = \mathcal{G}$), it can be stated and proved analogously for structures (i.e., for $\mathcal{U} = \mathcal{A}$), e.g., with $\mathrm{Inj}(\mathcal{F})$, $\mathrm{Sur}(\mathcal{F})$ and $\mathrm{Ext}(\mathcal{F})$ defined analogously for classes of structures $\mathcal{F} \subseteq \mathcal{A}$, etc.

Note that the left profile in Theorem 4.1 and the right profile in 4.2 both involve “infinitely many” homomorphism counts. The next two corollaries, however, imply that “finitely many” homomorphism counts in the left profile or the right profile suffice to determine the isomorphism type of a graph or of a structure, respectively.

Corollary 4.5. [11] *For all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, let* $n := \min\{|V(\mathbf{G})|, |V(\mathbf{H})|\}$*. The following are equivalent:*

(i) $\mathbf{G} \cong \mathbf{H}$.

(ii) $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$*, where* $\mathcal{F} := \{\mathbf{F} \in \mathcal{G} \mid |V(\mathbf{F})| \leq n\}$.

(iii) $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$*, where* $\mathcal{F} := \{\mathbf{F} \in \mathcal{G} \mid |V(\mathbf{F})| \leq m\}$ *and* $m := \max\{2, n^3\}$.

Proof. Let $\mathbf{G}$ and $\mathbf{H}$ be given, and let $n := \min\{|V(\mathbf{G})|, |V(\mathbf{H})|\}$. It suffices to show that (ii) implies (i) and (iii) implies (i).

For the direction from (ii) to (i), assume that (ii) holds. Since $|V(\mathbf{I}_1)| = 1 \leq n$, (ii) implies that $|V(\mathbf{G})| = |V(\mathbf{H})| = n$, by Example 3.15. Then, it follows from the proof of Theorem 4.4 for the direction from (ii) to (i) that $\mathbf{G} \cong \mathbf{H}$, that is, (i) also holds.

For the direction from (iii) to (i), we restate Lemma 9.7 from [11] in an equivalent form for our purpose here: *For all* $k \geq 2$*, there are no graphs* $\mathbf{G}$ *and* $\mathbf{H}$ *such that* $|V(\mathbf{G})| \leq k$*,* $|V(\mathbf{H})| > k$ *and* $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$ *for all graphs* $\mathbf{F}$ *with* $|V(\mathbf{F})| \leq k^3$. Then, we prove not (i) implies not (iii) by assuming that $\mathbf{G} \not\cong \mathbf{H}$. We distinguish two cases.

Case 1. $|V(\mathbf{G})| = 1$ *or* $|V(\mathbf{H})| = 1$*.* Obviously, $\hom(\mathbf{G}, \mathbf{I}_2) \neq \hom(\mathbf{H}, \mathbf{I}_2)$ and $|V(\mathbf{I}_2)| = 2 = \max\{2, n^3\}$.

Case 2. $|V(\mathbf{G})| \geq 2$ *and* $|V(\mathbf{H})| \geq 2$*.* The restatement of Lemma 9.7 in [11] given previously implies the existence of a graph $\mathbf{F}$ with $|V(\mathbf{F})| \leq n^3 = \max\{2, n^3\}$ such that $\hom(\mathbf{G}, \mathbf{F}) \neq \hom(\mathbf{H}, \mathbf{F})$. $\square$

Corollary 4.6. *For all structures* $\mathbf{A}$ *and* $\mathbf{B}$*, let* $n := \min\{|\mathrm{dom}(\mathbf{A})|, |\mathrm{dom}(\mathbf{B})|\}$*. The following are equivalent:*

(i) $\mathbf{A} \cong \mathbf{B}$

(ii) $\mathrm{lpf}^{\mathcal{F}}(\mathbf{A}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{B})$, *where* $\mathcal{F} := \{\mathbf{C} \in \mathcal{A} \mid |\mathrm{dom}(\mathbf{C})| \leq n\}$

(iii) $\mathrm{rpf}^{\mathcal{F}}(\mathbf{A}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{B})$, *where* $\mathcal{F} := \{\mathbf{C} \in \mathcal{A} \mid |\mathrm{dom}(\mathbf{C})| \leq m\}$ *and* $m := \max\{2, n\}$.

Proof. It suffices to note that the empty structure $\mathbf{J}_1$ considered in Example 3.18(a) and the complete structure $\mathbf{S}_2$ considered in Example 3.18(b) both reveal the size of a structure via $\hom(\mathbf{J}_1, *)$ and $\hom(*, \mathbf{S}_2)$, respectively. Thus, for all arbitrarily given structures $\mathbf{A}$ and $\mathbf{B}$, with $n := \min\{|\mathrm{dom}(\mathbf{A})|, |\mathrm{dom}(\mathbf{B})|\}$, if (ii) or (iii) holds, then $|\mathrm{dom}(\mathbf{A})| = |\mathrm{dom}(\mathbf{B})| = n$, with which it immediately follows from a proof of the analogous version of Theorem 4.4 for structures that $\mathbf{A} \cong \mathbf{B}$, that is, (i) also holds. $\square$

To conclude this section, let us take a closer look at Theorems 4.1 and 4.2 from two different perspectives that have inspired me for relevant study.

First, as Theorem 4.1 tells, isomorphism over $\mathcal{G}$ is characterized by the equality of $\mathrm{lpf}^{\mathcal{G}}$. Thus, for a nonempty subclass $\mathcal{F} \subseteq \mathcal{G}$, the equality of $\mathrm{lpf}^{\mathcal{F}}$ induces a *relaxation of isomorphism* over $\mathcal{G}$, i.e., an equivalence relation $\equiv$ over $\mathcal{G}$ coarser than isomorphism. A natural question then arises: What interesting equivalence relations can be characterized by $\equiv$ induced this way? An extensive body of research has been motivated to address this question for various equivalence relations (see, for example, [15] and [16]). Symmetrically, Theorem 4.2 says that isomorphism over $\mathcal{G}$ is characterized by the equality of $\mathrm{rpf}^{\mathcal{G}}$, hence the same can be said and asked of equivalence relations $\equiv$ induced by the equality of $\mathrm{rpf}^{\mathcal{F}}$ for nonempty subclasses $\mathcal{F} \subseteq \mathcal{G}$. The paper [3] has a focus in this regard. These topics and their variants will be the theme of the remainder of this chapter.

Second, as Theorem 4.1 implies, given a fixed graph $\mathbf{G}$, the (equivalence) class

$[\mathbf{G}]_{\cong}$ can be defined in terms of $\mathrm{lpf}^{\mathcal{G}}$: For every graph $\mathbf{H}$, we have $\mathbf{H} \in [\mathbf{G}]_{\cong}$ if and only if $\mathrm{lpf}^{\mathcal{G}}(\mathbf{H}) \in \{\mathrm{lpf}^{\mathcal{G}}(\mathbf{G})\}$. Note the issue with $\mathrm{lpf}^{\mathcal{G}}$, that is, the full left profile involves "infinitely many" entries. By Corollary 4.5, however, it suffices to check "finitely many" entries to determine whether $\mathbf{H} \in [\mathbf{G}]_{\cong}$: For every graph $\mathbf{H}$, we have $\mathbf{H} \in [\mathbf{G}]_{\cong}$ if and only if $\mathrm{lpf}^{\mathcal{F}}(\mathbf{H}) \in \{\mathrm{lpf}^{\mathcal{F}}(\mathbf{G})\}$, where $\mathcal{F} := \{\mathbf{F} \in \mathcal{G} \mid |V(\mathbf{F})| \leq |V(\mathbf{G})|\}$. In view of Theorem 4.2, we have the same issue with $\mathrm{rpf}^{\mathcal{G}}$ when determining whether $\mathbf{H} \in [\mathbf{G}]_{\cong}$. By Corollary 4.5, likewise, we can check "finitely many" entries of $\mathrm{rpf}^{\mathcal{G}}$ for this: For every graph $\mathbf{H}$, we have $\mathbf{H} \in [\mathbf{G}]_{\cong}$ if and only if $\mathrm{rpf}^{\mathcal{F}'}(\mathbf{H}) \in \{\mathrm{rpf}^{\mathcal{F}'}(\mathbf{G})\}$, where $\mathcal{F}' := \{\mathbf{F} \in \mathcal{G} \mid |V(\mathbf{F})| \leq \max\{2, |V(\mathbf{G})|^3\}\}$. That is to say, the class $[\mathbf{G}]_{\cong}$, viewed as a decision problem for membership, admits an algorithm based on evaluating the finitely many certain homomorphism counts in the left profile or the right profile of the input graph $\mathbf{H}$. In an abstract form: The membership problem for a class $\mathcal{D} \subseteq \mathcal{G}$ admits an algorithm that decides, for a graph $\mathbf{H}$, whether $\mathbf{H} \in \mathcal{D}$ by checking whether $\mathrm{lpf}^{\mathcal{F}}(\mathbf{H}) \in X$ (or dually whether $\mathrm{rpf}^{\mathcal{F}}(\mathbf{H}) \in X$), for some fixed $\mathcal{F} \subseteq \mathcal{G}$ of a finite size $k \in \mathbb{Z}^+$ and $X \subseteq \mathbb{N}^k$. As for structures, recall the corresponding statements in Theorems 4.1 and 4.2 and Corollary 4.6, and the above discussion also applies. This raises a second question: What interesting classes of graphs or of structures admit an algorithm for the corresponding membership problem based on examining finitely many certain entries in a left profile or a right profile? We will explore these topics in the next chapter.

4.2 Relaxations of Graph Isomorphism vs. Restricted Profiles over the Bag-Set Semiring

As noted at the end of the previous section, we will study what equivalence relations coarser than isomorphism over graphs are characterized in restricted left or right profiles. Throughout this section, the homomorphism counts in the profiles are over the bag-set semiring N, and we omit the reference to it in the notations hom, lpf and rpf.

4.2.1 Two Introductory Examples

We have seen in Example 3.15 that for every $n \in \mathbb{Z}^+$ and every graph $\mathbf{G}$, it holds that $\hom(\mathbf{I}_n, \mathbf{G}) = |V(\mathbf{G})|^n$. This indicates that *the equivalence among graphs arising from having the same number of vertices coincides with* C^1*-equivalence, and both are characterized by the equality of* $\mathrm{lpf}^{\mathcal{I}}$ (see Remark 3.7(a) and Example 4.1). In fact, this statement remains true even when $\mathcal{I}$ is replaced by any nonempty subset of $\mathcal{I}$.

In contrast, however, this equivalence is not characterized in any $\mathrm{rpf}^{\mathcal{F}}$. To see this, consider $\mathbf{I}_3$ and $\mathbf{K}_3$, and observe that for every arbitrary graph $\mathbf{F}$, we have $\hom(\mathbf{I}_3, \mathbf{F}) > \hom(\mathbf{K}_3, \mathbf{F})$: If $|V(\mathbf{F})| < 3$, then $\hom(\mathbf{I}_3, \mathbf{F}) = |V(\mathbf{F})|^3 > 0 = \hom(\mathbf{K}_3, \mathbf{F})$ (the last equality by Proposition 3.9); if $|V(\mathbf{F})| \geq 3$, then $\hom(\mathbf{I}_3, \mathbf{F}) = |V(\mathbf{F})|^3 > \binom{|V(\mathbf{F})|}{3} \times 3! = \hom(\mathbf{K}_3, \mathbf{F})$.

Furthermore, Example 3.16 shows that left profile restricted to the singleton class $\{\mathbf{K}_2\}$ reveals the number of edges in a graph (indeed, due to Proposition 3.14(c), every nonempty class whose members are of the form $\bigoplus_n \mathbf{K}_2$ will do). Thus, *the equivalence that arises from having the same number of edges is characterized by the equality of* $\mathrm{lpf}^{\{\mathbf{K}_2\}}$.

Nevertheless, it is does not coincide with the equality of $\mathrm{rpf}^{\mathcal{F}}$ for any nonempty class $\mathcal{F}$ of graphs. If $\mathcal{F}$ contains a graph $\mathbf{F}$ of size ≥ 2, then consider the two independent sets $\mathbf{I}_1$ and $\mathbf{I}_2$, both of which have no edges, but $\mathrm{rpf}^{\mathcal{F}}(\mathbf{I}_1) \neq \mathrm{rpf}^{\mathcal{F}}(\mathbf{I}_2)$, since $\hom(\mathbf{I}_1, \mathbf{F}) = |V(\mathbf{F})| < 2 \times |V(\mathbf{F})| \leq |V(\mathbf{F})|^2 = \hom(\mathbf{I}_2, \mathbf{F})$. If $\mathcal{F}$ only contains graphs of size < 2, namely, $\mathcal{F} = \{\mathbf{I}_1\}$, then consider the two cliques $\mathbf{K}_2$ and $\mathbf{K}_3$, which have different numbers of edges, however, $\mathrm{rpf}^{\{\mathbf{I}_1\}}(\mathbf{K}_2) = (0) = \mathrm{rpf}^{\{\mathbf{I}_1\}}(\mathbf{K}_3)$.

4.2.2 Indistinguishability by Color Refinement

There are numerous problems in theoretical computer science whose computational complexity is yet to be determined, and `GRAPH-ISOMORPHISM` is a famous one among them. This problem asks (cf. Section 2.2): Given two graphs $\mathbf{G}$ and $\mathbf{H}$, are they isomorphic? Despite the currently unknown status of its computational complexity, a bunch of polynomial-time algorithms have been devised based on heuristics.

We will discuss two polynomial-time heuristic algorithms in this subsection and the next that are invariant under isomorphism: If two graphs are isomorphic, then the algorithm makes this conclusion; contrapositively, this means that the two graphs are really nonisomorphic if the algorithm says so. This is known as the important aspect of *soundness*. However, they do not possess another important aspect, known as *completeness*. That is to say, if the algorithm does not distinguish them, then they might still be nonisomorphic, it is just that the algorithm cannot detect this. The presentation of these algorithms here are adapted from [15].

This induces an equivalence relation that is a relaxation of isomorphism. A connection of this equivalence with the indistinguishability from different aspects (logic, homomorphism counts) for either algorithm has been established by other research. We will mention these connections and study its relation to right profiles.

We start in this subsection with a simple polynomial-time heuristic algorithm known as *color refinement*. The reader should be aware that color refinement and its enhanced variant introduced in the next subsection are both sound but incomplete.

We explain color refinement in words. Given a graph $\mathbf{G}$, we will compute a coloring of the vertices in $\mathbf{G}$, which is a function on the vertices, based on local information. Note that this is similar to the n-coloring introduced in Section 2.2 but it differs in two ways:

(1) The colors are not drawn from the set $\{0, \ldots, n-1\}$ but rather are more complicated objects.

(2) Two adjacent vertices may have the same color.

The procedure runs in iterations. Initially, every vertex has the same default color, and the coloring is $\mathrm{color}^0_{\mathbf{G}}$; the total number of distinct colors is 1. Iteratively, we refine the colors by computing the new color, $\mathrm{color}^{i+1}_{\mathbf{G}}(v)$, of each vertex v based on its old color $\mathrm{color}^i_{\mathbf{G}}(v)$ and on the number of occurrences of each combination of

- the adjacency of v to w and
- the old color, $\mathrm{color}^i_{\mathbf{G}}(w)$, of w,

with w ranging over $V(\mathbf{G})$; if the total number of distinct colors increases, then proceed to the next iteration, otherwise stop (for which we say the refinement *stabilizes*) and return the multiset of the colors of the vertices computed from the *previous* iteration.

To be more specific about the colors, for every tuple $\overline{v} = (v_1, \ldots, v_n)$ of n vertices in $\mathbf{G}$, we write $\mathrm{atp}(\overline{v})$ for the *atomic type* of the induced subgraph $\mathbf{G}[\{v_1, \ldots, v_n\}]$ (similar to the notion of adjacency matrix, see Remark 2.2), defined as the $(n \times n)$-matrix M whose entries are

$$M_{ij} := \begin{cases} 2 & \text{if } i = j, \\ 1 & \text{if } \{v_i, v_j\} \in E(\mathbf{G}), \\ 0 & \text{otherwise.} \end{cases}$$

Now, for every $v \in V(\mathbf{G})$, we let its initial color, $\mathrm{color}^0_{\mathbf{G}}(v)$, be 2 (inspired by $M_{ii} = 2$ for $i \in [n]$), and recursively let its new color, $\mathrm{color}^{i+1}_{\mathbf{G}}(v)$, be defined as the pair of its old color, $\mathrm{color}^i_{\mathbf{G}}(v)$, and the multiset,

$$\{\!\!\{(\mathrm{atp}(v, w), \mathrm{color}^i_{\mathbf{G}}(w)) \mid w \in V(\mathbf{G})\}\!\!\}.$$

Note that for each $w \in V(\mathbf{G})$, the pair $(\mathrm{atp}(v, w), \mathrm{color}^i_{\mathbf{G}}(w))$ embodies the combination of the adjacency of v to w (in $\mathrm{atp}(v, w)$) and the old color, $\mathrm{color}^i_{\mathbf{G}}(w)$, of w. The multiset itself counts the number of occurrences of each possible combination. The algorithm is presented as pseudocode in Figure 4.1.

As regards implementing this algorithm as a computer program, e.g., in Python, $\mathrm{color}^i_{\mathbf{G}}$ can be realized as a `dict` (dictionary, hash map) for all $i \in \mathbb{N}$, whereas $\mathrm{atp}(v)$ as a `list` and $\mathrm{color}^{i+1}_{\mathbf{G}}(v)$ as a `tuple` (whose second element is a `Multiset` in which the elements are all `tuple`s).

During the execution, vertices of the same old color may be assigned different new colors due to an update of information. However, vertices of different old colors will surely be assigned different new colors. In other words, all vertices are initially in the same group (with the same default color), and they iteratively split into smaller groups according to their new colors, until no more split happens.

Figure 4.1: Pseudocode for COLOR-REFINEMENT.

COLOR-REFINEMENT($\mathbf{G}$)

for $v \in V(\mathbf{G})$
    $\text{color}^0_{\mathbf{G}}(v) = 2$
$i = 0$ // i is the number of iterations in the subsequent **repeat-until** loop
repeat
    number-of-colors = (number of distinct colors in $\text{color}^i_{\mathbf{G}}$)
    for $v \in V(\mathbf{G})$
        $\text{color}^{i+1}_{\mathbf{G}}(v) = (\text{color}^i_{\mathbf{G}}(v), \boldsymbol{\{}(\text{atp}(v, w), \text{color}^i_{\mathbf{G}}(w)) \mid w \in V(\mathbf{G})\boldsymbol{\}})$
    new-number-of-colors = (number of distinct colors in $\text{color}^{i+1}_{\mathbf{G}}$)
    $i = i + 1$
until *new-number-of-colors* == *number-of-colors*
return $\boldsymbol{\{}\text{color}^{i-1}_{\mathbf{G}}(v) \mid v \in V(\mathbf{G})\boldsymbol{\}}$

A graph is *regular* if every vertex has the same degree (cf. Definition 2.2). Let $\mathbf{G}$ and $\mathbf{H}$ be two graphs. Obviously, if $\mathbf{G}$ is isomorphic to $\mathbf{H}$, then color refinement does not distinguish them, since it computes the coloring of a graph based on its structural information. By an analysis of the algorithm, however, one will find that *color refinement cannot even distinguish* $\mathbf{G}$ *and* $\mathbf{H}$ *when both are regular and have the same size and same degree*; in this situation, the algorithm stops right after the first iteration. That is to say, $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable by this algorithm.

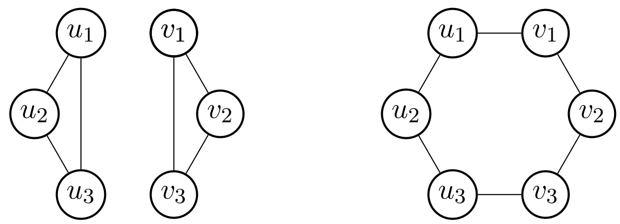


Figure 4.2: Two graphs $\mathbf{G}$ (left) and $\mathbf{H}$ (right) that are indistinguishable by color refinement.

Example 4.2. The graphs $\mathbf{G}$ and $\mathbf{H}$ depicted in Figure 4.2 are regular and, moreover, $|V(\mathbf{G})| = |V(\mathbf{H})|$ and $\deg(\mathbf{G}) = \deg(\mathbf{H})$. Therefore, they cannot be distinguished by color refinement.

The indistinguishability by color refinement yields an equivalence relation among graphs that is, by the previous discussion, a relaxation of isomorphism.

There is a characterization of this equivalence in logic. Recall that C^2 is first-order logic augmented with counting where the variables used in a formula are among z_0, z_1 (see Subsection 3.1.2).

Theorem 4.7. [7] *For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable by color refinement if and only if $\mathbf{G} \equiv_{\mathrm{C}^2} \mathbf{H}$.* $\square$

A different characterization of the indistinguishability by color refinement involves restricted left profiles. Recall that $\mathcal{T}$ denotes the class of all free trees (see Section 2.2).

Theorem 4.8. [16] *For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv_{\mathrm{C}^2} \mathbf{H}$ if and only if* $\mathrm{lpf}^{\mathcal{T}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}}(\mathbf{H})$. $\square$

From the preceding two theorems we immediately obtain the following characterization (a direct proof is given in [15]).

Corollary 4.9. *For all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G}$ *and* $\mathbf{H}$ *are indistinguishable by color refinement if and only if* $\mathrm{lpf}^{\mathcal{T}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}}(\mathbf{H})$*.* $\square$

We point out a key feature shared by all three mutually equivalent conditions in the three previous results: counting. Now we give an intuitive account for them based on that provided in [15]. Let $\mathbf{G}$ be a graph of n vertices. During the execution of color refinement on $\mathbf{G}$, an exploration starts out from every vertex and propagates to all vertices (including the one from which the exploration starts out), from each of which the propagation continues spreading in at most $n-1$ iterations until no new information (when the algorithm reaches stabilization), then these explorations are reported as a result. Indeed, an exploration starting out from the vertex v computes a rooted tree with v as the root node, whose child nodes are all vertices (including v) in $\mathbf{G}$ and are unordered among the siblings and, recursively, a rooted tree (as a subtree) is computed at each child node. The overall rooted tree of a vertex has height at most $n-1$, and the overall rooted trees computed at all the vertices of $\mathbf{G}$ comprise the returned colors. With the ability to count, and by reusing two variables (for v and u) in quantifications, these rooted trees can be collectively described in C^2 as "there are a total of n vertices v, and there are exactly n_1 vertices v of which the rooted tree has exactly n_1' child nodes u adjacent to v in $\mathbf{G}$ of which the rooted tree…, and has exactly n_2' child nodes u not adjacent to v in $\mathbf{G}$ of which the rooted tree…, and there are exactly n_2 vertices v of which the rooted trees…," which indeed is a C^2-sentence logically equivalent to $\chi_{\mathrm{C}^2}^{\mathbf{G}^{\sigma(\mathrm{G})}}$ that characterizes $\mathbf{G}^{\sigma(\mathrm{G})}$ (and hence $\mathbf{G}$) up to C^2-sentence (see Remark 3.7(b)). Moreover, the information of the number of vertices in $\mathbf{G}$ whose rooted tree is of each different type is captured by the counts $\hom(\mathbf{T}, \mathbf{G})$ for all free trees $\mathbf{T}$, and is collected in $\mathrm{lpf}^{\mathcal{T}}(\mathbf{G})$.

It turns out that, however, the indistinguishability by color refinement is not characterized by the equality of right profile restricted to any class.

Theorem 4.10. *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable by color refinement if and only if $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$.*

Proof. Let $\mathcal{F}$ be a nonempty class of graphs. We distinguish two cases as follows.

Case 1. $\mathcal{F}$ is a subclass of $\mathcal{I}$. Observe that $\mathbf{K}_2 \not\equiv_{\mathrm{C}^1} \mathbf{K}_3$ since they differ in size (by Remark 3.7(a)), and hence $\mathbf{K}_2 \not\equiv_{\mathrm{C}^2} \mathbf{K}_3$ since C^1 is a fragment of C^2. By Theorem 4.7, they are distinguished by color refinement. However, for every $n \in \mathbb{Z}^+$, we have $\hom(\mathbf{K}_2, \mathbf{I}_n) = \hom(\mathbf{K}_3, \mathbf{I}_n) = 0$ by Proposition 3.9, which implies that $\mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_2) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_3)$, whose entries are all 0.

Case 2. $\mathcal{F}$ is not a subclass of $\mathcal{I}$. Then some graph $\mathbf{F} \in \mathcal{F}$ contains an edge. We first consider the infinitely many pairs of graphs $\mathbf{G}_n$ and $\mathbf{H}_n$ for $n \geq 3$, where

$$V(\mathbf{G}_n) = V(\mathbf{H}_n) := \{u_1, \ldots, u_n\} \cup \{v_1, \ldots, v_n\}$$

and

$$\begin{aligned} E(\mathbf{G}_n) &:= \{\{u_i, u_j\} \mid 1 \leq i < j \leq n\} \cup \{\{v_i, v_j\} \mid 1 \leq i < j \leq n\}, \\ E(\mathbf{H}_n) &:= (E(\mathbf{G}) \setminus \{\{u_1, u_n\}, \{v_1, v_n\}\}) \cup \{\{u_1, v_1\}, \{u_n, v_n\}\}. \end{aligned}$$

The graphs $\mathbf{G}_3$ and $\mathbf{H}_3$ are drawn as $\mathbf{G}$ and $\mathbf{H}$, respectively, in Figure 4.2. It is clear for $n \geq 3$ that $\mathbf{G}_n$ and $\mathbf{H}_n$ are both regular and have the same size and same degree; by an earlier discussion, $\mathbf{G}_n$ is indistinguishable from $\mathbf{H}_n$ by color refinement.

Next, we argue that the following hold for every $n \geq 3$ and for every graph $\mathbf{E}$:

(1) $\hom(\mathbf{G}_n, \mathbf{E}) > 0$ if and only if $\mathbf{E}$ contains a subgraph isomorphic to $\mathbf{K}_n$.

(2) $\hom(\mathbf{H}_n, \mathbf{E}) > 0$ if and only if $\mathbf{E}$ contains a subgraph isomorphic to $\mathbf{K}_{n-1}$.

For (1), the "if" direction follows by taking the homomorphism $h : \mathbf{G} \to \mathbf{D}$ in which $h(u_i) = h(v_i) = w_i$ for all $i \in [n]$, where $\mathbf{D}$ is a subgraph of $\mathbf{E}$ isomorphic to $\mathbf{K}_n$ with vertex set $V(\mathbf{D}) = \{w_1, \dots, w_n\}$; the "only if" direction follows by Proposition 3.9, noting that the subgraph $\mathbf{G}[\{u_1, \dots, u_n\}]$ is isomorphic to $\mathbf{K}_n$. Similarly, for (2), let $\mathbf{D}$ be a subgraph of $\mathbf{E}$ isomorphic to $\mathbf{K}_{n-1}$ with vertex set $V(\mathbf{D}) = \{w_1, \dots, w_{n-1}\}$, then the "if" direction follows by taking the homomorphism $h : \mathbf{H} \to \mathbf{D}$ in which $h(u_n) = w_1$, $h(v_1) = w_{n-1}$ and $h(u_i) = h(v_{i+1}) = w_i$ for all $i \in [n-1]$; conversely, the "only if" direction follows by Proposition 3.9, noting that the subgraph $\mathbf{H}[\{u_1, \dots, u_{n-1}\}]$ is isomorphic to $\mathbf{K}_{n-1}$.

Finally, take the graph $\mathbf{F} \in \mathcal{F}$ which contains an edge. We will argue that

(3) $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}_n) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H}_n)$ for all $n \geq 3$

cannot be true for otherwise it leads to a contradiction: $\mathbf{F}$ *contains a subgraph isomorphic to* $\mathbf{K}_{n-1}$ *for all* $n \geq 3$ (note that $V(\mathbf{F})$ is finite). Assuming (3) to be true, we prove this statement by induction on n. The base case $n = 3$ holds because $\mathbf{F}$ contains an edge. By the induction hypothesis, $\mathbf{F}$ contains a subgraph isomorphic to $\mathbf{K}_{n-1}$, where $n \geq 3$. By (2), we have $\hom(\mathbf{H}_n, \mathbf{F}) > 0$. By the assumption that (3) is true and the fact that $\mathbf{F} \in \mathcal{F}$, we then have $\hom(\mathbf{G}_n, \mathbf{F}) = \hom(\mathbf{H}_n, \mathbf{F})$ and hence $\hom(\mathbf{G}_n, \mathbf{F}) > 0$. By (1), it follows that $\mathbf{F}$ contains a subgraph isomorphic to $\mathbf{K}_n$. $\square$

The next corollary follows as an immediate consequence of Theorems 4.7 and 4.10.

Corollary 4.11. *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$, *we have* $\mathbf{G} \equiv_{\mathrm{C}^2} \mathbf{H}$ *if and only if* $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$. $\square$

4.2.3 Indistinguishability by Weisfeiler-Leman Algorithm

Recall the last subsection that during the execution of color refinement on a graph, the vertices are initially assigned the same default color and are iteratively assigned a refined color until the stabilization of the refinement. The coloring is done on the *single* vertices.

This algorithm can be enhanced by coloring not the single vertices but rather the *tuples* of k vertices, which results in the *k-dimensional Weisfeiler-Leman algorithm* [50] (or *k-WL algorithm* for short) as an isomorphism test, which is the theme of this subsection. It turns out that color refinement is the special case $k = 1$, i.e., 1-WL algorithm. We will concern ourselves with $k \geq 2$ here. The presentation of this algorithm (parameterized by k) here is adapted from [15].

Fix an integer $k \geq 2$, we introduce this algorithm in words. Let $\mathbf{G}$ be an input graph. Initially, the default color that every k-tuple $\overline{v}$ of vertices is assigned, $\mathrm{color}^{k,0}_{\mathbf{G}}(\overline{v})$, is $\mathrm{atp}(\overline{v})$, where $\mathrm{atp}(\overline{v})$ is a $(k \times k)$-matrix introduced in the previous subsection. Note that this is different from color refinement in that now the k-tuples may not share the same initial color (in fact, $\mathrm{atp}(v) = 2$ for every *single vertex* $v \in V(\mathbf{G})$, that is why in color refinement we assign 2 to every vertex as the uniform initial color). Iteratively, we refine the colors by computing a new color $\mathrm{color}^{k,i+1}_{\mathbf{G}}(\overline{v})$ of each k-tuple $\overline{v}$ based on its old color $\mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v})$ and on the number of occurrences of each combination of

- the collective atomic type $\mathrm{atp}(\overline{v}w)$ of $\overline{v}$ with w and
- the old colors, $\mathrm{color}^{k,i}_{\mathbf{G}}(\overline{u})$, of all the k-tuples $\overline{u}$ obtained by substituting w for an entry in $\overline{v}$,

with w ranging over $V(\mathbf{G})$. As in color refinement, if the total number of colors increases, then proceed to the next iteration, otherwise stop (because the refinement stabilizes) and return the multiset of colors of the k-tuples computed from the previous iteration.

To be more precise about the new color of the k-tuple $\overline{v}$, let $\mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v})$ be the old color computed previously. We set $\mathrm{color}^{k,i+1}_{\mathbf{G}}(\overline{v})$, the new color of $\overline{v}$, to be the pair of its old color, $\mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v})$, and the multiset

$$\{\!\{(\mathrm{atp}(\overline{v}w), \mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v}(1/w)), \ldots, \mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v}(k/w))) \mid w \in V(\mathbf{G})\}\!\}.$$

Likewise, for every $w \in V(\mathbf{G})$, the sequence $(\mathrm{atp}(\overline{v}w), \mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v}(1/w)), \ldots, \mathrm{color}^{k,i}_{\mathbf{G}}(\overline{v}(k/w)))$ embodies the combination stated above. The multiset itself counts the number of occurrences of each possible combination. The algorithm is given as pseudocode in Figure 4.3.

As in color refinement, during the execution of k-WL algorithm, k-tuples of vertices having the same old color may be assigned different new colors because of an update of information, and those of different old colors will always be assigned different new colors. One aspect in which it differs from color refinement is that the k-tuples may have different initial colors (which are their respective atomic types). The k-tuples of vertices then iteratively split into smaller groups depending on the new colors they get, until no more splits.

Note that the algorithm is also invariant under isomorphism since the computation is again based on structural information. That is to say, if $\mathbf{G}$ and $\mathbf{H}$ are isomorphic graphs, then they are not distinguished by the algorithm (for any $k \geq 1$). As noted at the beginning of this subsection, however, this algorithm is never a complete isomorphism test. We have seen examples of nonisomorphic $\mathbf{G}$ and $\mathbf{H}$ that are indistinguishable when $k = 1$ previously. For $k \geq 2$, examples of such $\mathbf{G}$ and $\mathbf{H}$ can be found in [7]. Naturally, the indistinguishability

Figure 4.3: Pseudocode for k-Weisfeiler-Leman.

k-Weisfeiler-Leman($\mathbf{G}$)

for $\overline{v} \in V(\mathbf{G})^k$

  $\text{color}_{\mathbf{G}}^{k,0}(\overline{v}) = \text{atp}(\overline{v})$

$i = 0$ // i is the number of iterations in the subsequent **repeat-until** loop

repeat

  number-of-colors = (number of distinct colors in $\text{color}_{\mathbf{G}}^{k,i}$)

  for $\overline{v} \in V(\mathbf{G})^k$

    $\text{color}_{\mathbf{G}}^{k,i+1}(\overline{v}) =$

    $(\text{color}_{\mathbf{G}}^{k,i}(\overline{v}), \boldsymbol{\{}(\text{atp}(\overline{v}w), \text{color}_{\mathbf{G}}^{k,i}(\overline{v}(1/w)), \ldots, \text{color}_{\mathbf{G}}^{k,i}(\overline{v}(k/w))) \mid w \in V(\mathbf{G})\boldsymbol{\}})$

  new-number-of-colors = (number of distinct colors in $\text{color}_{\mathbf{G}}^{k,i+1}$)

  $i = i + 1$

until *new-number-of-colors* == *number-of-colors*

return $\boldsymbol{\{}\text{color}_{\mathbf{G}}^{k,i-1}(\overline{v}) \mid \overline{v} \in V(\mathbf{G})^k\boldsymbol{\}}$

by the k-dimensional Weisfeiler-Leman algorithm gives rise to an equivalence relation over graphs and, by the discussion above, is a relaxation of isomorphism.

Now we give an insight of k-dimensional Weisfeiler-Leman algorithm based on [15] similar to the one we gave of color refinement. Let $\mathbf{G}$ be a graph of n vertices. During the execution of k-WL on $\mathbf{G}$, an exploration starts out from every k-tuple $\overline{v}$ of vertices and propagates to all k-tuples $\overline{u}$ of vertices which differ from $\overline{v}$ in at most one entry, and the propagation continues spreading (there are at most $n^k - 1$ iterations of propagation since the total number of k-tuples is n^k) until no new information is obtained, i.e., when the algorithm stabilizes, and then the explorations are reported. Similar to color refinement, an exploration starting out from a k-tuple $\overline{v}$ computes a rooted tree with $\overline{v}$ as the root node, and its child nodes are those k-tuples $\overline{u}$ that differ from $\overline{v}$ in at most one entry and are also unordered among the siblings; recursively, a rooted tree as a subtree is computed at each child node. The overall rooted tree of $\overline{v}$ thus has height at most $n^k - 1$. The overall rooted trees computed at all k-tuples of vertices of $\mathbf{G}$ constitute the returned coloring. One thing to note is that now *the nodes of rooted trees themselves are no longer single vertices but rather k-tuples of vertices.* These overall rooted trees can be described, by the technique of reusing variables, as a C^{k+1}-sentence logically equivalent to $\chi^{\mathbf{G}^{\sigma(\mathrm{G})}}_{\mathrm{C}^{k+1}}$ that characterizes $\mathbf{G}^{\sigma(\mathrm{G})}$ (and hence $\mathbf{G}$) up to C^{k+1}-equivalence (cf. Remark 3.7(b)). As regards homomorphism counts, the information of the number of k-tuples of vertices in $\mathbf{G}$ whose rooted tree is of each different type is now captured by the number of homomorphisms from $\mathbf{F}$ to $\mathbf{G}$ that respect certain structural information of the *tree decompositions* $\mathbf{T}$ of $\mathbf{F}$, which in turn are captured by the counts $\hom(\mathbf{F}, \mathbf{G})$, for all graphs $\mathbf{F}$ of *treewidth* at most k, where we introduce these new notions as follows.

Definition 4.2. Let $\mathbf{G}$ be a graph.

(a) A *tree decomposition of* $\mathbf{G}$ is a free tree $\mathbf{T}$ in which each vertex t is associated with a *bag*, a nonempty subset $B_t \subseteq V(\mathbf{G})$, such that

- for every vertex $v \in V(\mathbf{G})$, we have $v \in B_t$ for some vertex t and the subgraph of $\mathbf{T}$ induced by the set $\{t \in V(\mathbf{T}) \mid v \in B_t\}$ of vertices is a subtree of $\mathbf{T}$, and
- for every edge $\{u, v\} \in E(\mathbf{G})$, there is a vertex t such that $u, v \in B_t$.

(b) The *width* of a tree decomposition $\mathbf{T}$ of $\mathbf{G}$ is defined to be $\max\{|B_t| \mid t \in V(\mathbf{T})\} - 1$.

The *treewidth of* $\mathbf{G}$ is defined to be $\min\{\text{width of } \mathbf{T} \mid \mathbf{T} \text{ is a tree decomposition of } \mathbf{G}\}$.

By definition, the independent sets $\mathbf{I}_1, \mathbf{I}_2, \ldots$ all have treewidth 0, the *bipartite* graphs (i.e., 2-colorable graphs) that contain an edge all have treewidth 1; the latter includes the forests and hence the free trees that contain an edge, as well as the path graphs $\mathbf{P}_2, \mathbf{P}_3, \ldots$. Furthermore, the cycle graphs $\mathbf{C}_3, \mathbf{C}_4, \ldots$ all have treewidth 2. Finally, for $n \in \mathbb{Z}^+$, the clique $\mathbf{K}_n$ has treewidth $n-1$.

Notation. For $k \in \mathbb{N}$, the class of all graphs of treewidth at most k is denoted $\mathcal{T}^k$.

Now we state a general version of Theorem 4.7 that connects indistinguishability by k-WL to C^{k+1}-equivalence and a general version of Theorem 4.8 that connects C^{k+1}-equivalence to the equality of left profile restricted to $\mathcal{T}^k$. As in Theorems 4.7 and 4.8, counting is a key feature shared by all three mutually equivalent conditions.

Theorem 4.12. [7] *Let $k \in \mathbb{Z}^+$. For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable by the k-dimensional Weisfeiler-Leman algorithm if and only if $\mathbf{G} \equiv_{\mathrm{C}^{k+1}} \mathbf{H}$.* $\square$

Theorem 4.13. [16] *Let $k \in \mathbb{Z}^+$. For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv_{\mathrm{C}^{k+1}} \mathbf{H}$ if and only if $\mathrm{lpf}^{\mathcal{T}^k}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}^k}(\mathbf{H})$.* $\square$

We immediately obtain the next characterization from the preceding two theorems (a direct proof is given in [15]).

Corollary 4.14. *Let $k \in \mathbb{Z}^+$. For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable by the k-dimensional Weisfeiler-Leman algorithm if and only if $\mathrm{lpf}^{\mathcal{T}^k}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}^k}(\mathbf{H})$.* $\square$

Note that Theorem 4.8 and Corollary 4.9 are indeed the special case with $k = 1$ of Theorem 4.13 and Corollary 4.14, respectively: The class $\mathcal{T}^1$ consists of all forests, including free trees, and, by Proposition 3.14(c), it follows that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathrm{lpf}^{\mathcal{T}^1}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}^1}(\mathbf{H})$ if and only if $\mathrm{lpf}^{\mathcal{T}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}}(\mathbf{H})$. Interestingly, Theorem 4.13 remains valid even when $k = 0$, because $\mathbf{G} \equiv_{\mathrm{C}^1} \mathbf{H}$ if and only if $|V(\mathbf{G})| = |V(\mathbf{H})|$ (see Remark 3.7(a)), and if and only if $\mathrm{lpf}^{\mathcal{I}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{I}}(\mathbf{H})$ (see the first example given in Subsection 4.2.1), since $\mathcal{T}^0 = \mathcal{I}$ by a previous discussion.

Remark 4.3. Let $k \in \mathbb{N}$. For $\mathcal{F} = \mathcal{T}^k$, it holds that $\mathrm{Inj}(\mathcal{F}) = \mathcal{F}$. Hence, by Remark 4.2(a), *isomorphism over* $\mathcal{T}^k$ *is characterized by the equality of* $\mathrm{lpf}^{\mathcal{T}^k}$. In other words, for all graphs $\mathbf{G}$ and $\mathbf{H}$ of treewidth at most k, they are isomorphic if and only if $\mathrm{lpf}^{\mathcal{T}^k}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}^k}(\mathbf{H})$. This, together with Theorem 4.13, implies that *isomorphism over* $\mathcal{T}^k$ *is characterized by* C^{k+1}*-equivalence*, which reproves and strengthens a result of M. Grohe and J. Mariño [30] (see Theorem 4 in that paper).

However, for all $k \in \mathbb{Z}^+$, the C^k-equivalence is not characterized by the equality of right profile restricted to any class $\mathcal{F}$ of graphs, seen in the next theorem (which generalizes Corollary 4.11).

Theorem 4.15. *For every* $k \in \mathbb{Z}^+$*, there is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G} \equiv_{\mathrm{C}^k} \mathbf{H}$ *if and only if* $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$. □

The next corollary immediately follows from Theorems 4.12 and 4.15.

Corollary 4.16. *For every* $k \in \mathbb{Z}^+$*, there is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G}$ *and* $\mathbf{H}$ *are indistinguishable by the* k*-dimensional Weisfeiler-Leman algorithm if and only if* $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$. □

Theorem 4.15 immediately follows from Theorem 4.17, which is presented and proved next. The logic formalism $\mathrm{C}^n_{\infty\omega}$ was briefly discussed in Remark 3.8. Put

$$\mathrm{C}^{\omega}_{\infty\omega} := \bigcup_{n\in\mathbb{Z}^+} \mathrm{C}^n_{\infty\omega}.$$

Therefore, if a class is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$, then it is not definable in $\mathrm{C}^n_{\infty\omega}$ for any $n \in \mathbb{Z}^+$.

Theorem 4.17. *If $\equiv$ is an equivalence relation over $\mathcal{G}$ that is finer than $\equiv_{\mathrm{C}^1}$ and coarser than $\equiv_{\mathrm{C}^k}$ for some $k \geq 2$, then there is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv \mathbf{H}$ if and only if $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$.*

Proof. Let $\equiv$ be an equivalence relation over graphs that is finer than C^1-equivalence and coarser than C^k-equivalence for some $k \geq 2$, and let $\mathcal{F}$ be a nonempty class of graphs. We distinguish two cases as follows.

Case 1. Every graph $\mathbf{F} \in \mathcal{F}$ is 2-colorable. Obviously, $\mathbf{K}_3 \not\equiv_{\mathrm{C}^1} \mathbf{K}_4$ since they differ in size (see Remark 3.7(a)), thus we have $\mathbf{K}_3 \not\equiv \mathbf{K}_4$ because $\equiv$ is finer than $\equiv_{\mathrm{C}^1}$. However, $\chi(\mathbf{K}_3) = 3$ and $\chi(\mathbf{K}_4) = 4$, both of which are greater than $\chi(\mathbf{F})$, hence $\hom(\mathbf{K}_3, \mathbf{F}) = 0 = \hom(\mathbf{K}_4, \mathbf{F})$, for every $\mathbf{F} \in \mathcal{F}$, which implies that $\mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_3) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_4)$.

Case 2. Some graph $\mathbf{F} \in \mathcal{F}$ is not 2-colorable. Then, by the Definable $\mathbf{H}$-Coloring Dichotomy Theorem A.1, we have $\mathrm{CSP}(\mathbf{F}) \neq \mathrm{Mod}(\psi)$ for any $\mathrm{C}^{\omega}_{\infty\omega}$-sentence ψ. In particular, $\mathrm{CSP}(\mathbf{F}) \neq \mathrm{Mod}(\varphi)$ for any $\mathrm{C}^k_{\infty\omega}$-sentence φ. By Remark 3.8(b), there are graphs $\mathbf{G} \in \mathrm{CSP}(\mathbf{F})$ and $\mathbf{H} \notin \mathrm{CSP}(\mathbf{F})$ such that $\mathbf{G} \equiv_{\mathrm{C}^k} \mathbf{H}$. It follows that $\mathbf{G} \equiv \mathbf{H}$ since $\equiv$ is coarser than $\equiv_{\mathrm{C}^k}$, and that $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$ since $\hom(\mathbf{G}, \mathbf{F}) > 0$ while $\hom(\mathbf{H}, \mathbf{F}) = 0$. $\square$

Remark 4.4. Let $k \in \mathbb{Z}^+$. The fragment of the counting logic C that consists of formulas φ for which the quantifier rank $\mathrm{qr}(\varphi)$ is at most k is denoted C_k, and C_k-*equivalence* over graphs is the equivalence arising from satisfying the same C_k-sentences. It was shown in [29] that *for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, they are* C_k*-equivalent if and only if* $\mathrm{lpf}^{\mathcal{T}_k}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}_k}(\mathbf{H})$, where $\mathcal{T}_k$ denotes the class of all graphs of *treedepth* at most k.

(a) It can shown that every C_k-sentence is logically equivalent to a C^k-sentence (indeed, this follows from a more general statement involving *formulas* that requires an inductive proof). Thus, C^k-equivalence implies C_k-equivalence.

(b) For all $s \in \mathbb{Z}^+$, the C^1-sentence $\exists^{=s} x\, x = x$ is also a C_1-sentence. Hence, C_1-equivalence implies C^1-equivalence (see Remark 3.7(a)), and vice versa (by the above part (a)).

Since C_1 is a fragment of C_k, we have that C_k-equivalence is finer than C_1-equivalence. Put all these together, it follows that *for all* $k \in \mathbb{Z}^+$*, the* C_k*-equivalence is finer than* C^1*-equivalence and coarser than* C^{k+1}*-equivalence and hence is not characterized in right profile restricted to any nonempty class* $\mathcal{F}$ *of graphs* (by Theorem 4.17).

4.2.4 Cospectrality

For every graph $\mathbf{G}$ of size n, the *characteristic polynomial of* $\mathbf{G}$ in the variable x is $p(\mathbf{G}, x) := \det(xI_n - M^{\mathbf{G}})$, where det denotes the determinant of a square matrix, $M^{\mathbf{G}}$ is the adjacency matrix of $\mathbf{G}$ (see Remark 2.2) and I_n is the n-dimensional identity matrix. The solutions, with multiplicities, to the equation $p(\mathbf{G}, x) = 0$ are the eigenvalues of the matrix $M^{\mathbf{G}}$.

We say two graphs $\mathbf{G}$ and $\mathbf{H}$ are *cospectral* if they have the same characteristic polynomial or, equivalently, if their adjacency matrices $M^{\mathbf{G}}$ and $M^{\mathbf{H}}$ have the same eigenvalues with multiplicities.

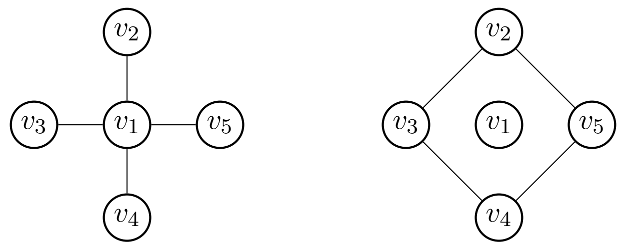


Figure 4.4: Two cospectral graphs $\mathbf{G}$ (left) and $\mathbf{H}$ (right), copied from Figure 2 of [15].

Example 4.3. The graphs $\mathbf{G}$ and $\mathbf{H}$ depicted in Figure 4.4 are cospectral.

In [15], H. Dell, M. Grohe and G. Rattan proved that cospectrality is characterized in left profile restricted to the class $\mathcal{C}$ of cycle graphs (see Proposition 9 in that paper, attributed to [49]).

Theorem 4.18. [15] *For all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are cospectral if and only if* $\mathrm{lpf}^{\mathcal{C}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{C}}(\mathbf{H})$. □

It turns out that cospectrality cannot be characterized in any restricted right profiles, seen next.

Theorem 4.19. *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G}$ and $\mathbf{H}$ are cospectral if and only if* $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$.

Proof. First, note that $\mathcal{C} \subseteq \mathcal{T}^2$ because $\mathbf{C}_n$ has treewidth at most 2 for all $n \in \mathbb{Z}^+$ (indeed, $\mathcal{C} \subsetneq \mathcal{T}^2$ since $\mathbf{I}_2 \in \mathcal{T}^2$ but $\mathbf{I}_2 \notin \mathcal{C}$). Hence, for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have

	$\mathbf{G}$ and $\mathbf{H}$ are C^3-equivalent	
iff	$\mathrm{lpf}^{\mathcal{T}^2}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{T}^2}(\mathbf{H})$	(by Theorem 4.13)
then	$\mathrm{lpf}^{\mathcal{C}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{C}}(\mathbf{H})$	(since $\mathcal{C} \subseteq \mathcal{T}^2$)
iff	$\mathbf{G}$ and $\mathbf{H}$ are cospectral	(by Theorem 4.18).

Therefore, cospectrality is coarser than C^3-equivalence (also see Theorem 2.1 in [14]).

Next, we have $\mathbf{I}_1 = \mathbf{C}_1 \in \mathcal{C}$. Hence, for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have

	$\mathbf{G}$ and $\mathbf{H}$ are cospectral	
iff	$\mathrm{lpf}^{\mathcal{C}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{C}}(\mathbf{H})$	(by Theorem 4.18).
then	$\hom(\mathbf{I}_1, \mathbf{G}) = \hom(\mathbf{I}_1, \mathbf{H})$	(since $\mathbf{I}_1 \in \mathcal{C}$)
iff	$\lvert V(\mathbf{G}) \rvert = \lvert V(\mathbf{H}) \rvert$	(see Example 3.15)
iff	$\mathbf{G}$ and $\mathbf{H}$ are C^1-equivalent	(see Remark 3.7(a)).

Thus, cospectrality is finer than C^1-equivalence.

We conclude by Theorem 4.17 with $k = 3$. □

4.2.5 Chromatic Equivalence

So far we have seen various relaxations of isomorphism among graphs that are characterized by the equality of some restricted left profile but not by the equality of any restricted right profile, and the reader may have the (false) impression that restricted left profiles always have more expressive power than their right counterparts in characterizing a relaxation of isomorphism. We will see a reverse situation in this subsection.

Let $\mathbf{G}$ be a graph. The *chromatic polynomial of* $\mathbf{G}$ in the variable x, denoted $\chi(\mathbf{G}, x)$, is the polynomial that gives the information of the number of n-colorings of $\mathbf{G}$ for all $n \in \mathbb{Z}^+$ (see Definition 2.7). More precisely, $\chi(\mathbf{G}, n)$ is equal to the number of n-colorings for all $n \in \mathbb{Z}^+$.

Example 4.4. We have the following chromatic polynomials of some common graphs:

$$\begin{aligned}
\chi(\mathbf{I}_n, x) &= x^n, && \text{for } n \in \mathbb{Z}^+; \\
\chi(\mathbf{C}_n, x) &= (x-1)^n + (-1)^n(x-1), && \text{for } n \geq 3; \\
\chi(\mathbf{K}_n, x) &= x(x-1)\cdots(x-n+1), && \text{for } n \in \mathbb{Z}^+.
\end{aligned}$$

Moreover, it is easy to see that

$$\chi(\mathbf{T}, x) \quad = \quad x(x-1)^{n-1}, \quad \text{for every free tree } \mathbf{T} \text{ of size } n \in \mathbb{Z}^+.$$

In particular,

$$\chi(\mathbf{P}_n, x) \quad = \quad x(x-1)^{n-1}, \quad \text{for } n \in \mathbb{Z}^+,$$

since $\mathbf{P}_n$ is also a free tree of size n.

We provide two useful techniques to derive the chromatic polynomial $\chi(\mathbf{G}, x)$ for an arbitrary graph $\mathbf{G}$ as follows.

Technique 1. Multiplicativity. If $\mathbf{G} = \mathbf{G}_1 \oplus \mathbf{G}_2$, i.e., if $\mathbf{G}$ is the direct sum of two graphs $\mathbf{G}_1$ and $\mathbf{G}_2$, then

$$\chi(\mathbf{G}, x) \quad = \quad \chi(\mathbf{G}_1, x) \cdot \chi(\mathbf{G}_2, x).$$

See Proposition 3.14(c).

Technique 2. Addition-Contraction Recursion. The base cases are for the cliques:

$$\chi(\mathbf{K}_n, x) \quad = \quad x(x-1)\cdots(x-n+1), \quad \text{for all } n \in \mathbb{Z}^+,$$

seen in Example 4.4. The recursive cases are for graphs $\mathbf{G}$ such that $\{u, v\} \notin E(\mathbf{G})$ for some distinct $u, v \in V(\mathbf{G})$:

$$\chi(\mathbf{G}, x) \quad = \quad \chi(\mathbf{G}_1, x) + \chi(\mathbf{G}_2, x),$$

where $\mathbf{G}_1$ is the graph with $V(\mathbf{G}_1) = V(\mathbf{G})$ and $E(\mathbf{G}_1) = E(\mathbf{G}) \cup \{\{u, v\}\}$, and $\mathbf{G}_2$ is the graph with

$$V(\mathbf{G}_2) = (V(\mathbf{G}) \setminus \{u, v\}) \cup \{w\},$$

Figure 4.5: Two chromatically equivalent graphs $\mathbf{G}$ (left) and $\mathbf{H}$ (right).

in which $w \notin V(\mathbf{G})$ is a fresh new vertex, and

$$\begin{aligned} E(\mathbf{G}_2) &= \{\{u', v'\} \in E(\mathbf{G}) \mid u' \neq u \text{ and } v' \neq v\} \cup \\ &\qquad \{\{w, w'\} \mid \{u, w'\} \in E(\mathbf{G}) \text{ or } \{v, w'\} \in E(\mathbf{G})\}. \end{aligned}$$

To see this, note that for $n \in \mathbb{Z}^+$, the n-colorings of $\mathbf{G}$ can be split into two groups: the first group in which u and v are colored differently, which are exactly the n-colorings of $\mathbf{G}_1$, and the second group in which u and v are colored the same, which are exactly the n-colorings of $\mathbf{G}_2$ (with w in place of u and v).

Note that $\chi(\mathbf{G}, x)$ has degree n and the leading coefficient (namely, the coefficient of the term x^n) is 1, for every graph $\mathbf{G}$ of size n.

We say two graphs $\mathbf{G}$ and $\mathbf{H}$ are *chromatically equivalent* if $\chi(\mathbf{G}, x) = \chi(\mathbf{H}, x)$ or, equivalently, if they have the same number of n-colorings for every $n \in \mathbb{Z}^+$.

Example 4.5. Consider the graphs $\mathbf{G}$ and $\mathbf{H}$ depicted in Figure 4.5, which are isomorphic to $\mathbf{I}_1 \oplus \mathbf{P}_3$ and $\mathbf{K}_2 \oplus \mathbf{K}_2$, respectively. Obviously, they are both 2-colorable. Moreover, they are chromatically equivalent since $\chi(\mathbf{G}, x) = x^2(x-1)^2 = \chi(\mathbf{H}, x)$.

In Example 3.17 we concluded that for $n \in \mathbb{Z}^+$ and every graph $\mathbf{G}$, the number of n-colorings of $\mathbf{G}$ is equal to $\hom(\mathbf{G}, \mathbf{K}_n)$. This immediately implies the characterization of chromatic equivalence in restricted right profile as follows.

Proposition 4.20. *For all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G}$ *and* $\mathbf{H}$ *are chromatically equivalent if and only if* $\mathrm{rpf}^{\mathcal{K}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{K}}(\mathbf{H})$*.* $\square$

However, as opposed to the results in previous subsections, here we have a reverse situation, which shows that it is not always the case that restricted left profile has more expressive power than restricted right profile in terms of characterizing an equivalence relation.

Theorem 4.21. *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G}$ *and* $\mathbf{H}$ *are chromatically equivalent if and only if* $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$*.*

We will prove this theorem in the remainder of this subsection. We first state and prove the two subsequent lemmas.

Lemma 4.22. *For every graph* $\mathbf{F}$*, if* $\mathbf{F}$ *is* 2*-colorable and connected and has size* ≥ 3*, then* $\mathrm{sur}(\mathbf{F}, \mathbf{P}_3) > 0$*.*

Proof. Recall Definition 2.5 that $V(\mathbf{K}_2) = \{v_1, v_2\}$, $E(\mathbf{K}_2) = \{\{v_1, v_2\}\}$, $V(\mathbf{P}_3) = \{v_1, v_2, v_3\}$ and $E(\mathbf{P}_3) = \{\{v_1, v_2\}, \{v_2, v_3\}\}$. Hence, $\mathbf{K}_2$ is a subgraph of $\mathbf{P}_3$.

In the following, we assume that $\mathbf{F}$ is an arbitrary graph that is 2-colorable and connected and has size ≥ 3.

Since $\mathbf{F}$ is 2-colorable, we have $\mathrm{hom}(\mathbf{F}, \mathbf{K}_2) > 0$ by Example 3.17. Moreover, since $\mathbf{F}$ is connected, we have $E(\mathbf{F}) \neq \emptyset$, which implies that $\mathbf{F}$ contains a subgraph isomorphic to $\mathbf{K}_2$. By Proposition 3.9, it follows that every homomorphism from $\mathbf{F}$ to $\mathbf{K}_2$ is surjective. Thus, $\mathrm{sur}(\mathbf{F}, \mathbf{K}_2) = \mathrm{hom}(\mathbf{F}, \mathbf{K}_2) > 0$.

Now, take an arbitrary surjective homomorphism $h : \mathbf{F} \to \mathbf{K}_2$. Since $|V(\mathbf{F})| \geq 3$, at least one of the two sets $h^{-1}(v_1)$ and $h^{-1}(v_2)$ has size at least 2 (here h^{-1} means preimage, see Section 2.1). Without loss of generality, we can assume that $|h^{-1}(v_1)| \geq 2$; indeed, if $|h^{-1}(v_1)| < 2$, then take the automorphism $\pi : \mathbf{K}_2 \cong \mathbf{K}_2$ such that $\pi(v_1) = v_2$ and $\pi(v_2) = v_1$, and consider the surjective homomorphism $(\pi \circ h) : \mathbf{F} \to \mathbf{K}_2$ in place of h in what follows. Thus, there are distinct vertices $u_1, u_2 \in V(\mathbf{F})$ such that $h(u_1) = h(u_2) = v_1$. As $\mathbf{F}$ is connected, there are (not necessarily distinct) vertices $u_3, u_4 \in V(\mathbf{F})$ such that $\{u_1, u_3\}, \{u_2, u_4\} \in E(\mathbf{F})$; note that $h(u_3) = h(u_4) = v_2$. It follows that the function $h' : V(\mathbf{F}) \to V(\mathbf{P}_3)$ with

$$h'(u) := \begin{cases} v_3 & \text{if } u = u_2, \\ \\ h(u) & \text{otherwise} \end{cases}$$

is a surjective homomorphism from $\mathbf{F}$ to $\mathbf{P}_3$. $\square$

Observe that $\oplus$ is a binary operation over $\mathcal{G}$ that is associative and commutative but that lacks an identity element. To facilitate a succinct form involving direct sums of graphs in the next lemma and later on in the proof of Theorem 4.21, we introduce an ad hoc graph (solely for this purpose), the *null graph*, denoted $\emptyset$, with vertex set $V(\emptyset) := \emptyset$ and edge set $E(\emptyset) := \emptyset$. Note that $\oplus$ in Definition 2.3 can be consistently extended to include the null graph $\emptyset$. Moreover, we have

$$\mathbf{F} \oplus \emptyset = \emptyset \oplus \mathbf{F} = \mathbf{F}$$

for all graphs $\mathbf{F} \in (\mathcal{G} \cup \{\emptyset\})$, which suggests that the null graph $\emptyset$ be the identity for the binary operation $\oplus$ on $\mathcal{G} \cup \{\emptyset\}$. We set $\bigoplus_0 \mathbf{F} := \emptyset$ for all graphs $\mathbf{F} \in \mathcal{G}$, and set $\mathbf{I}_0 := \emptyset$, the *independent set of size* 0.

Next, recall the two chromatically equivalent graphs $\mathbf{I}_1 \oplus \mathbf{P}_3$ and $\mathbf{K}_2 \oplus \mathbf{K}_2$ in Example 4.5. We delineate in the subsequent lemma exactly which graphs $\mathbf{F}$ do and do not distinguish them in terms of $\hom(\mathbf{F}, *)$.

Lemma 4.23. *For every graph* $\mathbf{F}$*, we have* $\hom(\mathbf{F}, \mathbf{I}_1 \oplus \mathbf{P}_3) \geq \hom(\mathbf{F}, \mathbf{K}_2 \oplus \mathbf{K}_2)$*, and the equality holds if and only if* $\mathbf{F}$ *is not* 2*-colorable or* $\mathbf{F}$ *is isomorphic to* $(\mathbf{I}_m \oplus \bigoplus_n \mathbf{K}_2)$ *for some* $m, n \in \mathbb{N}$ *such that* $m + n \geq 1$*.*

Proof. For brevity, we let $\mathbf{G} := (\mathbf{I}_1 \oplus \mathbf{P}_3)$ and $\mathbf{H} := (\mathbf{K}_2 \oplus \mathbf{K}_2)$ throughout this proof. We distinguish two cases depending on whether $\mathbf{F}$ is connected as follows.

Case 1. $\mathbf{F}$ *is connected.* We further divide into three cases as follows.

Case 1-1. $\mathbf{F}$ *is not* 2*-colorable.* Then, by Proposition 3.10, we have $\hom(\mathbf{F}, \mathbf{G}) = 0 = \hom(\mathbf{F}, \mathbf{H})$, because both $\mathbf{G}$ and $\mathbf{H}$ are 2-colorable.

Case 1-2. $\mathbf{F}$ *is isomorphic to* $\mathbf{I}_1$ *or* $\mathbf{K}_2$*.* Then, by Examples 3.15 and 3.16, respectively, we have $\hom(\mathbf{F}, \mathbf{G}) = 4 = \hom(\mathbf{F}, \mathbf{H})$.

Case 1-3. $\mathbf{F}$ *is* 2*-colorable but is not isomorphic to* $\mathbf{I}_1$ *or* $\mathbf{K}_2$*.* Since $\mathbf{F}$ is connected, it has size ≥ 3 and hence contains an edge (i.e., contains a subgraph isomorphic to $\mathbf{K}_2$). Hence, every homomorphic image of $\mathbf{F}$ is a connected graph (by Proposition 3.11(b)) and contains a subgraph isomorphic to $\mathbf{K}_2$ (by Proposition 3.9). Thus, for every homomorphism $h : \mathbf{F} \to \mathbf{G}$ or $h : \mathbf{F} \to \mathbf{H}$, the image $h(\mathbf{F})$ is isomorphic to $\mathbf{P}_2$ or $\mathbf{P}_3$. Therefore,

$$
\begin{aligned}
& \hom(\mathbf{F}, \mathbf{G}) & \\
= \; & 2 \times \operatorname{sur}(\mathbf{F}, \mathbf{P}_2) + \operatorname{sur}(\mathbf{F}, \mathbf{P}_3) & \text{(by Proposition 3.8)} \\
> \; & 2 \times \operatorname{sur}(\mathbf{F}, \mathbf{P}_2) & \text{(by Lemma 4.22)} \\
= \; & \hom(\mathbf{F}, \mathbf{H}) & \text{(by Proposition 3.8).}
\end{aligned}
$$

To summarize, if $\mathbf{F}$ is connected, then $\hom(\mathbf{F}, \mathbf{G}) \geq \hom(\mathbf{F}, \mathbf{H})$, and the equality holds if and only if $\mathbf{F}$ is not 2-colorable or $\mathbf{F}$ is isomorphic to $\mathbf{I}_1$ or $\mathbf{K}_2$.

Case 2. $\mathbf{F}$ *is not connected.* We can assume $\mathbf{F}$ has k connected components $\mathbf{F}_1, \ldots, \mathbf{F}_k$ (i.e., $\mathbf{F} = \bigoplus_{i=1}^{k} \mathbf{F}_i$) for some $k \in \mathbb{Z}^+$. It follows that

$$\begin{aligned}
& \hom(\mathbf{F}, \mathbf{G}) \\
= & \textstyle\prod_{i=1}^{k} \hom(\mathbf{F}_i, \mathbf{G}) \quad \text{(by Proposition 3.14(c))} \\
\geq & \textstyle\prod_{i=1}^{k} \hom(\mathbf{F}_i, \mathbf{H}) \quad \text{(by Case 1.)} \\
= & \hom(\mathbf{F}, \mathbf{H}) \qquad\qquad \text{(by Proposition 3.14(c))}.
\end{aligned}$$

Moreover, as $\mathbf{G}$ and $\mathbf{H}$ both contain a subgraph isomorphic to $\mathbf{K}_2$ and are 2-colorable, by Proposition 3.10 we have, for every $i \in [k]$, that $\hom(\mathbf{F}_i, \mathbf{G}) = 0$ if and only if $\hom(\mathbf{F}_i, \mathbf{H}) = 0$, and if and only if $\mathbf{F}_i$ is not 2-colorable. Therefore,

$$\begin{aligned}
& \text{the equality holds in the above inequality} \\
\text{iff} \quad & \hom(\mathbf{F}_i, \mathbf{G}) = 0 \ (= \hom(\mathbf{F}_i, \mathbf{H})) \text{ for some } i \in [k] \\
& \text{or } \hom(\mathbf{F}_i, \mathbf{G}) = \hom(\mathbf{F}_i, \mathbf{H}) > 0 \text{ for all } i \in [k] \\
\text{iff} \quad & \mathbf{F}_i \text{ is not 2-colorable for some } i \in [k] \qquad \text{(by the above discussion)} \\
& \text{or } \mathbf{F}_i \text{ is isomorphic to } \mathbf{I}_1 \text{ or } \mathbf{K}_2 \text{ for all } i \in [k] \text{ (by Case 1. and the above discussion)} \\
\text{iff} \quad & \mathbf{F} \text{ is not 2-colorable} \qquad \text{(by Proposition 2.2(b))} \\
& \text{or } \mathbf{F} \text{ is isomorphic to } (\mathbf{I}_m \oplus \textstyle\bigoplus_n \mathbf{K}_2) \text{ for some } m, n \in \mathbb{N} \text{ with } m + n = k. \qquad \square
\end{aligned}$$

Proof of Theorem 4.21. Let $\mathcal{F}$ be an arbitrary nonempty class of graphs, we argue that there are graphs $\mathbf{G}$ and $\mathbf{H}$ that are either chromatically equivalent while $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$ or not chromatically equivalent while $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$. We distinguish two cases as follows.

Case 1. Some graph $\mathbf{F} \in \mathcal{F}$ *is* 2*-colorable but is not isomorphic to* $(\mathbf{I}_m \oplus \bigoplus_n \mathbf{K}_2)$ *for any* $m, n \in \mathbb{N}$ *such that* $m + n \geq 1$. Take $\mathbf{G} := (\mathbf{I}_1 \oplus \mathbf{P}_3)$ and $\mathbf{H} := (\mathbf{K}_2 \oplus \mathbf{K}_2)$. Then, $\hom(\mathbf{F}, \mathbf{G}) > \hom(\mathbf{F}, \mathbf{H})$ by Lemma 4.23, hence $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$. However, $\mathbf{G}$ and $\mathbf{H}$ are chromatically equivalent, by Example 4.5.

Case 2. Every graph $\mathbf{F} \in \mathcal{F}$ *either is not* 2*-colorable or is isomorphic to* $(\mathbf{I}_m \oplus \bigoplus_n \mathbf{K}_2)$ *for some* $m, n \in \mathbb{N}$ *such that* $m+n \geq 1$. Take $\mathbf{G} := \mathbf{C}_8$ and $\mathbf{H} := (\mathbf{C}_4 \oplus \mathbf{C}_4)$. Then, $\mathbf{G}$ and $\mathbf{H}$ are not chromatically equivalent since they have different chromatic polynomials: $\chi(\mathbf{G}, x) = (x-1)^8 + (x-1)$ and $\chi(\mathbf{H}, x) = ((x-1)^4 + (x-1))^2$. However, no graph $\mathbf{F} \in \mathcal{F}$ distinguishes $\mathbf{G}$ and $\mathbf{H}$ via $\hom(\mathbf{F}, \mathbf{G})$ and $\hom(\mathbf{F}, \mathbf{H})$: If $\mathbf{F}$ is not 2-colorable, then $\hom(\mathbf{F}, \mathbf{G}) = 0 = \hom(\mathbf{F}, \mathbf{H})$ as both $\mathbf{G}$ and $\mathbf{H}$ are 2-colorable (by Proposition 3.10). Besides, if $\mathbf{F}$ is isomorphic to $(\mathbf{I}_m \oplus \bigoplus_n \mathbf{K}_2)$ for some $m, n \in \mathbb{N}$ such that $m + n \geq 1$, then $\hom(\mathbf{F}, \mathbf{G}) = 8^m \times 16^n = \hom(\mathbf{F}, \mathbf{H})$. □

4.2.6 Equivalence in Some Fragments of First-Order Logic

We have seen in Subsections 4.2.2 and 4.2.3 that for $k \in \mathbb{Z}^+$, the C^{k+1}-equivalence coincides with indistinguishability by k-dimensional Weisfeiler-Leman algorithm, and is characterized in $\mathrm{lpf}^{\mathcal{T}^k}$, the left profile restricted to the class $\mathcal{T}^k$ of graphs of treewidth at most k. It is natural to ask: Are the FO^k-equivalence and FO_k-equivalence able to be characterized in restricted left or right profiles, for any $k \in \mathbb{Z}^+$?

Fix a $k \in \mathbb{Z}^+$. Recall that FO^k is the fragment of first-order logic that consists of formulas whose variables (free or bound) are among $z_0, \ldots, z_{k-1}$, and that FO_k is the fragment of first-order logic that consists of formulas whose quantifier rank is at most k. Furthermore, for graphs $\mathbf{G}$ and $\mathbf{H}$, they are FO^k-equivalent (or FO_k-equivalent) precisely when they satisfy the same FO^k*-sentences* (or FO_k*-sentences*, respectively).

As the next proposition shows, *for every* $k \in \mathbb{Z}^+$*, neither* FO^k*-equivalence nor* FO_k*-equivalence is characterized in the left profile or in the right profile restricted to any nonempty class* $\mathcal{F}$ *of graphs.*

Proposition 4.24. *Let $k \in \mathbb{Z}^+$, and let $\equiv$ be* FO^k*-equivalence or* FO_k*-equivalence. The following hold:*

(a) *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv \mathbf{H}$ if and only if* $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$.

(b) *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv \mathbf{H}$ if and only if* $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$.

Proof. Let $k \in \mathbb{Z}^+$ and let $\equiv$ be FO^k-equivalence or FO_k-equivalence, We distinguish two cases depending on k.

Case 1. $k = 1$. We saw in Example 3.8 that all graphs are equivalent in the sense of $\equiv$.[1] That is to say, $\equiv$ is the coarsest equivalence relation over graphs. However, let $\mathbf{F}$ be an arbitrary graph. Then,

$$\hom(\mathbf{F}, \mathbf{F}) < 2 \times \hom(\mathbf{F}, \mathbf{F}) \leq \hom(\mathbf{F}, \mathbf{F} \oplus \mathbf{F})$$

by the fact that $\hom(\mathbf{F}, \mathbf{F}) > 0$ and by Proposition 3.14(b). This implies part (a). Next, if $\mathbf{F}$ has size $n \in \mathbb{Z}^+$, then

$$\hom(\mathbf{I}_1, \mathbf{F}) = n \neq 0 = \hom(\mathbf{K}_{n+1}, \mathbf{F})$$

by Example 3.15 and Proposition 3.9. This implies part (b).

Case 2. $k \geq 2$. Assume that $\mathcal{F}$ is an arbitrary nonempty class of graphs. We will give a construction of graphs $\mathbf{G}$ and $\mathbf{H}$ such that

- $\mathbf{G} \equiv \mathbf{H}$ and $\mathrm{lpf}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{lpf}^{\mathcal{F}}(\mathbf{H})$ in part (a), and

[1] Obviously, both FO^1-equivalence and FO_1-equivalence would be characterized in $\mathrm{lpf}^{\mathcal{F}}$ and in $\mathrm{rpf}^{\mathcal{F}}$ if $\mathcal{F} = \emptyset$ were allowed in the definition of restricted left and right profiles.

- either $\mathbf{G} \not\equiv \mathbf{H}$ and $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$, or $\mathbf{G} \equiv \mathbf{H}$ and $\mathrm{rpf}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{rpf}^{\mathcal{F}}(\mathbf{H})$, in part (b).

For part (a), let $\mathbf{F}$ be an arbitrary graph in $\mathcal{F}$ of size $n \in \mathbb{Z}^+$. Take $m := \max\{k, n\}$ and consider the two graphs $\mathbf{K}_m$ and $\mathbf{K}_{m+1}$. We let $\equiv'$ denote $\equiv_{\mathrm{FO}^m}$ (or $\equiv_{\mathrm{FO}_m}$) if $\equiv$ stands for $\equiv_{\mathrm{FO}^k}$ (or $\equiv_{\mathrm{FO}_k}$, respectively). By Remark 3.5, we have $\mathbf{K}_m \equiv' \mathbf{K}_{m+1}$ and hence $\mathbf{K}_m \equiv \mathbf{K}_{m+1}$. We have $\mathbf{F}$ is m-colorable by Example 2.1, and $\hom(\mathbf{F}, \mathbf{K}_m) > 0$ by Example 3.17. Take an arbitrary homomorphism $h : \mathbf{F} \to \mathbf{K}_m$, by Example 2.1(b) and in view of Example 3.17, the function $h' : V(\mathbf{F}) \to V(\mathbf{K}_{m+1})$ with the same mapping rule as h is also a homomorphism $h' : \mathbf{F} \to \mathbf{K}_{m+1}$. This implies a bijection between homomorphisms $h : \mathbf{F} \to \mathbf{K}_m$ and $h' : \mathbf{F} \to \mathbf{K}_{m+1}$ such that $h'(V(\mathbf{F})) \subseteq V(\mathbf{K}_m)$. In addition, let r be the smallest index $i \in [m]$ such that v_i is in the range $h(V(\mathbf{F}))$, i.e., $r := \min\{i \in [m] \mid v_i \in h(V(\mathbf{F}))\}$. It is easy to see that the mapping $h'' : V(\mathbf{F}) \to V(\mathbf{K}_{m+1})$ with

$$h''(v) := \begin{cases} v_{m+1} & \text{if } h(v) = v_r, \\ h(v) & \text{otherwise} \end{cases}$$

is a homomorphism $h'' : \mathbf{F} \to \mathbf{K}_{m+1}$ but is not one from $\mathbf{F}$ to $\mathbf{K}_m$. Thus, $\hom(\mathbf{F}, \mathbf{K}_m) < \hom(\mathbf{F}, \mathbf{K}_{m+1})$ and so $\mathrm{lpf}^{\mathcal{F}}(\mathbf{K}_m) \neq \mathrm{lpf}^{\mathcal{F}}(\mathbf{K}_{m+1})$. Take $\mathbf{G} := \mathbf{K}_m$ and $\mathbf{H} := \mathbf{K}_{m+1}$, and we are done with this part.

Finally, for part (b), we further distinguish two cases below.

Case 2(b)-1. $\mathcal{F} = \{\mathbf{I}_1\}$. Consider the two graphs $\mathbf{I}_1$ and $\mathbf{I}_2$. Since for $\varphi := \forall x \forall y\, x = y$ (note that φ is a FO^k-sentence and also a FO_k-sentence, because $k \geq 2$), $\mathbf{I}_1 \models \varphi$ while not $\mathbf{I}_2 \models \varphi$, we have that $\mathbf{I}_1 \not\equiv \mathbf{I}_2$. However, $\hom(\mathbf{I}_1, \mathbf{I}_1) = 1 = \hom(\mathbf{I}_2, \mathbf{I}_1)$, which implies that $\mathrm{rpf}^{\mathcal{F}}(\mathbf{I}_1) = \mathrm{rpf}^{\mathcal{F}}(\mathbf{I}_2)$. It suffices to take $\mathbf{G} := \mathbf{I}_1$ and $\mathbf{H} := \mathbf{I}_2$.

Case 2(b)-2. $\mathcal{F} \neq \{\mathbf{I}_1\}$. Since $\mathcal{F}$ is nonempty, there is a graph $\mathbf{F} \in \mathcal{F}$ of size $n \geq 2$. Take $m := \max\{k, n\} \geq 2$ and consider the graphs $\mathbf{I}_m$ and $\mathbf{I}_{m+1}$. As in part (a), we let $\equiv'$ denote $\equiv_{\mathrm{FO}^m}$ (or $\equiv_{\mathrm{FO}_m}$) if $\equiv$ stands for $\equiv_{\mathrm{FO}^k}$ (or $\equiv_{\mathrm{FO}_k}$, respectively). It follows that $\mathbf{I}_m \equiv' \mathbf{I}_{m+1}$, by Remark 3.5 if $\equiv'$ stands for $\equiv_{\mathrm{FO}^m}$, and by Remark 3.3 if $\equiv'$ stands for $\equiv_{\mathrm{FO}_m}$. Hence, $\mathbf{I}_m \equiv \mathbf{I}_{m+1}$. However, $\hom(\mathbf{I}_m, \mathbf{F}) = n^m < 2 \times n^m \leq n^{m+1} = \hom(\mathbf{I}_{m+1}, \mathbf{F})$ and it follows that $\mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_m) \neq \mathrm{rpf}^{\mathcal{F}}(\mathbf{K}_{m+1})$. We are done by taking $\mathbf{G} := \mathbf{K}_m$ and $\mathbf{H} := \mathbf{K}_{m+1}$. □

4.3 Relaxations of Graph Isomorphism vs. Restricted Profiles over the Boolean Semiring

In the previous section, we investigated several relaxations of isomorphism over graphs in regard to the existence of a characterization by the equality of restricted left profile or of restricted right profile over N, the bag-set semiring. As we will see shortly, most of them do not admit such a characterization over B, the Boolean semiring. In contrast, we explore next three relaxations of isomorphism that admit a characterization in restricted left or right profile over B.

4.3.1 Homomorphic Equivalence

Recall Definition 3.23 that for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, we say $\mathbf{A}$ and $\mathbf{B}$ are homomorphically equivalent (denoted $\mathbf{A} \leftrightarrow \mathbf{B}$) when $\mathbf{A} \to \mathbf{B}$ and $\mathbf{B} \to \mathbf{A}$ or, equivalently, when $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{B}) > 0$ and $\hom_{\mathrm{B}}(\mathbf{B}, \mathbf{A}) > 0$.

The proposition below is a simultaneous dual to Theorems 4.1 and 4.2 in terms of the underlying semiring of the left and right profiles, and it immediately follows from Proposition 3.18 (specifically the equivalence between (i), (ii) and (iii)).

Proposition 4.25. *For all* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathcal{U}$*, the following are equivalent:*

(i) $\mathbf{A} \leftrightarrow \mathbf{B}$.

(ii) $\mathrm{lpf}^{\mathcal{U}}_{\mathrm{B}}(\mathbf{A}) = \mathrm{lpf}^{\mathcal{U}}_{\mathrm{B}}(\mathbf{B})$.

(iii) $\mathrm{rpf}^{\mathcal{U}}_{\mathrm{B}}(\mathbf{A}) = \mathrm{rpf}^{\mathcal{U}}_{\mathrm{B}}(\mathbf{B})$. □

However, homomorphic equivalence over $\mathcal{G}$ is not characterized in any restricted left or right profile over N, as shown by the next result.

Proposition 4.26. (a) *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G} \leftrightarrow \mathbf{H}$ *if and only if* $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{H})$.

(b) *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G} \leftrightarrow \mathbf{H}$ *if and only if* $\mathrm{rpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{H})$.

Proof. Let $\mathbf{F}$ be an arbitrary graph. As seen in the proof of Proposition 4.24,

$$\hom(\mathbf{F}, \mathbf{F}) < \hom(\mathbf{F}, \mathbf{F} \oplus \mathbf{F}),$$

although $\mathbf{F}$ and $\mathbf{F} \oplus \mathbf{F}$ are homomorphically equivalent (by Proposition 3.17). This implies part (a).

Next, if $\mathbf{F}$ is isomorphic to $\mathbf{I}_1$, then, by Proposition 3.9,

$$\hom(\mathbf{K}_2, \mathbf{F}) = 0 = \hom(\mathbf{K}_3, \mathbf{F})$$

while $\mathbf{K}_2$ and $\mathbf{K}_3$ are not homomorphically equivalent; otherwise, $\mathbf{F}$ has $n \geq 2$ vertices, and

$$\hom(\mathbf{I}_1, \mathbf{F}) = n < 2n \leq n^2 = \hom(\mathbf{I}_2, \mathbf{F}),$$

although $\mathbf{I}_1$ and $\mathbf{I}_2$ are homomorphically equivalent (by Proposition 3.17). This implies part (b). $\square$

The two graphs $\mathbf{C}_3$ and $\mathbf{C}_3 \oplus \mathbf{C}_3$ in Example 3.19 are homomorphically equivalent. From Proposition 4.25, it follows that for every nonempty class $\mathcal{F}$ of graphs, $\mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_3) = \mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_3 \oplus \mathbf{C}_3)$ and $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_3) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_3 \oplus \mathbf{C}_3)$; in other words, the two graphs are not distinguished by means of restricted left or right profile over B. However, they are not equivalent in the sense of any equivalence relations listed below, which implies that none of these equivalence relations is characterized in restricted left or right profile over B:

(1) isomorphism, FO-equivalence and C-equivalence,

(2) equivalence arising from having the same number of vertices, C^1-equivalence and C_1-equivalence,

(3) equivalence arising from having the same number of edges,

(4) C^k-equivalence and indistinguishability by $(k-1)$-dimensional Weisfeiler-Leman algorithm, where $k \geq 2$,

(5) C_k-equivalence, where $k \in \mathbb{Z}^+$,

(6) cospectrality,

(7) chromatic equivalence,

(8) FO^k-equivalence, where $k \geq 2$,

(9) FO_k-equivalence, where $k \geq 2$.

For (5), note that C_k-equivalence implies C_1-equivalence because C_1 is a fragment of C_k. For (8) and (9), the spoiler has a simple winning strategy for the game $\mathrm{G}^k(\mathbf{C}_3, \mathbf{C}_3 \oplus \mathbf{C}_3)$ and the game $\mathrm{G}_k(\mathbf{C}_3, \mathbf{C}_3 \oplus \mathbf{C}_3)$: Place a pebble on a vertex in one copy of $\mathbf{C}_3$ in $\mathbf{C}_3 \oplus \mathbf{C}_3$ in the first move, and place a different pebble on a vertex in the other copy of $\mathbf{C}_3$ in $\mathbf{C}_3 \oplus \mathbf{C}_3$ in the second move.

Furthermore, the independent sets are the $\to$-minimals for the preorder $\to$ (see Remark 3.10(b)), and this means that $\hom_{\mathrm{B}}(\mathbf{I}_n, \mathbf{G}) = 1$ for all $n \in \mathbb{Z}^+$ and all graphs $\mathbf{G}$. Besides this, we saw in Example 3.8 that all graphs are FO^1-equivalent and FO_1-equivalent. Recall that $\mathcal{I}$ denotes the class of all independent sets.

Proposition 4.27. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be graphs. The following hold:*

(a) $\mathbf{G} \equiv_{\mathrm{FO}^1} \mathbf{H}$.

(b) $\mathbf{G} \equiv_{\mathrm{FO}_1} \mathbf{H}$.

(c) *For all nonempty classes* $\mathcal{F} \subseteq \mathcal{I}$*, we have* $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{H})$. □

For every graph $\mathbf{F}$, it is obvious that $\mathbf{F} \to \mathbf{F}$ and, if it has size $n \in \mathbb{Z}^+$, then $\mathbf{K}_{n+1} \not\to \mathbf{F}$ (by Proposition 3.9); this shows that there are graphs $\mathbf{G}$ and $\mathbf{H}$ such that $\hom_{\mathrm{B}}(\mathbf{G}, \mathbf{F}) = 1 \neq 0 = \hom(\mathbf{H}, \mathbf{F})$ (by setting $\mathbf{G} := \mathbf{F}$ and $\mathbf{H} := \mathbf{K}_{n+1}$).

Proposition 4.28. (a) *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G} \equiv_{\mathrm{FO}^1} \mathbf{H}$ *if and only if* $\mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{H})$*,*

(b) *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\mathbf{G} \equiv_{\mathrm{FO}_1} \mathbf{H}$ *if and only if* $\mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{H})$. □

4.3.2 Same Chromatic Number and Same Clique Number

We will investigate two more relaxations of graph isomorphism, which are coarser than homomorphic equivalence: the equivalence arising from having the same chromatic number (see Definition 2.7(b)) and the equivalence arising from having the same clique number. The relevant definition is as follows.

Definition 4.3. Let $\mathbf{G}$ be a graph. The *clique number of* $\mathbf{G}$, denoted $\omega(\mathbf{G})$, is the maximum $n \in \mathbb{Z}^+$ such that $\mathbf{K}_n$ is a subgraph of $\mathbf{G}$.

Interestingly, the chromatic number and the clique number of a graph $\mathbf{G}$ are dual in form when expressed as homomorphism counts: $\chi(\mathbf{G}) = \min\{n \in \mathbb{Z}^+ \mid \mathbf{G} \to \mathbf{K}_n\}$ or, equivalently, $\chi(\mathbf{G}) = \min\{n \in \mathbb{Z}^+ \mid \hom_{\mathrm{B}}(\mathbf{G}, \mathbf{K}_n) = 1\}$ and $\omega(\mathbf{G}) = \max\{n \in \mathbb{Z}^+ \mid \mathbf{K}_n \to \mathbf{G}\}$ or, equivalently, $\omega(\mathbf{G}) = \max\{n \in \mathbb{Z}^+ \mid \hom_{\mathrm{B}}(\mathbf{K}_n, \mathbf{G}) = 1\}$. Some examples are below.

- $\chi(\mathbf{I}_n) = \omega(\mathbf{I}_n) = 1$ for $n \in \mathbb{Z}^+$.
- $\chi(\mathbf{K}_n) = \omega(\mathbf{K}_n) = n$ for $n \in \mathbb{Z}^+$.
- $\chi(\mathbf{P}_1) = \omega(\mathbf{P}_1) = 1$ and $\chi(\mathbf{P}_n) = \omega(\mathbf{P}_n) = 2$ for $n \geq 2$.
- $\chi(\mathbf{C}_n) = \omega(\mathbf{C}_n) = n$ for $n \leq 3$, $\chi(\mathbf{C}_n) = \begin{cases} 2 & \text{if } n \text{ is even} \\ 3 & \text{otherwise} \end{cases}$ and $\omega(\mathbf{C}_n) = 2$ for $n \geq 4$.
- $\chi(\mathbf{T}) = \omega(\mathbf{T})$ for free trees $\mathbf{T}$ of size ≥ 2.

In particular, $\chi(\mathbf{C}_3) = \chi(\mathbf{C}_5)$ but $\mathbf{C}_3$ and $\mathbf{C}_5$ are not homomorphically equivalent (by Proposition 3.9 and the fact that $\mathbf{C}_3 = \mathbf{K}_3$); $\omega(\mathbf{C}_2) = \omega(\mathbf{C}_5)$ but $\mathbf{C}_2$ and $\mathbf{C}_5$ are not homomorphically equivalent (since $\chi(\mathbf{C}_2) = 2 < 3 = \chi(\mathbf{C}_5)$ and by Proposition 3.10). Therefore, the indistinguishability by chromatic number and by clique number are (strictly) coarser than homomorphic equivalence,[2] as shown by the following characterizations that are immediate from definitions (together with Proposition 4.25). Recall that $\mathcal{K}$ denotes the class of all cliques.

Proposition 4.29. *For all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, the following hold:*

(a) $\chi(\mathbf{G}) = \chi(\mathbf{H})$ *if and only if* $\mathrm{rpf}^{\mathcal{K}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{rpf}^{\mathcal{K}}_{\mathrm{B}}(\mathbf{H})$.

(b) $\omega(\mathbf{G}) = \omega(\mathbf{H})$ *if and only if* $\mathrm{lpf}^{\mathcal{K}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{K}}_{\mathrm{B}}(\mathbf{H})$. □

The next two propositions are dual to the preceding one, asserting that the indistinguishability by chromatic number and by clique number are not characterized by the equality of any restricted left profile and any restricted right profile, respectively.

Proposition 4.30. *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\chi(\mathbf{G}) = \chi(\mathbf{H})$ *if and only if* $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{G}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{H})$.

Proof. Assume, for the sake of contradiction, that there exists such a nonempty class $\mathcal{F}$. From the previous discussion, for all $n \in \mathbb{Z}^+$, we have $\chi(\mathbf{C}_{2n}) = 2 < 3 = \chi(\mathbf{C}_{2n+1})$, hence $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2n}) \neq \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2n+1})$ by the assumption. In particular, for all $m, n \in \mathbb{Z}^+$, $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2m+1}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2n+1})$. In the sequel, we show, instead, that for all $n \in \mathbb{Z}^+$, $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2n}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{C}_{2n+1})$, which contradicts a previous statement. For this purpose, fix an $n \in \mathbb{Z}^+$ and we show, for all $\mathbf{F} \in \mathcal{F}$, that $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n}) = \hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n+1})$. We distinguish two cases below depending on whether $\mathbf{F}$ is 2-colorable.

Case 1. $\mathbf{F}$ *is* 2*-colorable.* Then, $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{K}_2) = 1$ by Remark 3.19, and hence $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n}) = 1 = \hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n+1})$ by Remark 3.12.

[2] In fact, the indistinguishability by chromatic number is also strictly coarser than chromatic equivalence, which was considered in Subsection 4.2.5.

Case 2. $\mathbf{F}$ *is not* 2*-colorable.* Then, $\chi(\mathbf{F}) > 2$ and, by Proposition 2.4, $\mathbf{F}$ contains $\mathbf{C}_{2k+1}$ as a subgraph for some $k \in \mathbb{Z}^+$. Therefore, $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n}) = 0$ by Proposition 3.10. Let $m \in \mathbb{Z}^+$ such that $m > k$. Then, $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2m+1}) = 0$ by Corollary 3.12(a). Since $\mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_{2m+1}) = \mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{C}_{2n+1})$, we have $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{C}_{2n+1}) = 0$ as well. $\square$

Proposition 4.31. *There is no nonempty class* $\mathcal{F} \subseteq \mathcal{G}$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, we have* $\omega(\mathbf{G}) = \omega(\mathbf{H})$ *if and only if* $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{H})$.

Proof. Let $\mathcal{F}$ be an arbitrary nonempty class of graphs, we will show that there are graphs $\mathbf{G}$ and $\mathbf{H}$ such that either $\omega(\mathbf{G}) \neq \omega(\mathbf{H})$ and $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{H})$, or $\omega(\mathbf{G}) = \omega(\mathbf{H})$ and $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{H})$. We distinguish two cases below, depending on whether $\mathcal{F}$ is a subclass of $\mathcal{I}$, the class of all independent sets.

Case 1. $\mathcal{F}$ *is a subclass of* $\mathcal{I}$. We have $\omega(\mathbf{K}_2) = 2 < 3 = \omega(\mathbf{K}_3)$ by a previous discussion. However, $\hom_{\mathrm{B}}(\mathbf{K}_2, \mathbf{I}_n) = 0 = \hom_{\mathrm{B}}(\mathbf{K}_3, \mathbf{I}_n)$ for all $n \in \mathbb{Z}^+$ by Proposition 3.9, which implies $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{K}_2) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{K}_3)$. We are done by taking $\mathbf{G} := \mathbf{K}_2$ and $\mathbf{H} := \mathbf{K}_3$.

Case 2. $\mathcal{F}$ *is not a subclass of* $\mathcal{I}$. Then, there is a graph $\mathbf{F} \in \mathcal{F}$ that contains $\mathbf{K}_2$ as a subgraph. By Theorem 2.5, there is a graph $\mathbf{G}$ with $\chi(\mathbf{G}) = \chi(\mathbf{F}) + 1$ and $\gamma(\mathbf{G}) \geq 4$; note that $\mathbf{G}$ contains $\mathbf{K}_2$ as a subgraph and so $\omega(\mathbf{G}) = 2$, and clearly $\hom_{\mathrm{B}}(\mathbf{G}, \mathbf{F}) = 0$ by Proposition 3.10. By a previous discussion, however, we have $\omega(\mathbf{K}_2) = 2$. Moreover, $\hom_{\mathrm{B}}(\mathbf{K}_2, \mathbf{F}) = 1$, by Remark 3.12. Setting $\mathbf{H} := \mathbf{K}_2$, we get $\omega(\mathbf{G}) = \omega(\mathbf{H})$ and $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{G}) \neq \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{H})$. $\square$

The indistinguishability by chromatic number and by clique number are not even characterized in restricted left or right profile over N. A *proof* for the result below can be obtained by adapting the one for Proposition 4.26 (changing $\leftrightarrow$ to the corresponding $\equiv$), noting that

$$\chi(\bigoplus_m \mathbf{F}) = \chi(\bigoplus_n \mathbf{F}) \qquad \text{and} \qquad \omega(\bigoplus_m \mathbf{F}) = \omega(\bigoplus_n \mathbf{F})$$

for all $m, n \in \mathbb{Z}^+$ and all graphs $\mathbf{F}$.

Proposition 4.32. *Let $\equiv$ be the equivalence relation over $\mathcal{G}$ such that*

(1) $\mathbf{G} \equiv \mathbf{H}$ *if and only if* $\chi(\mathbf{G}) = \chi(\mathbf{H})$*, for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*, or*

(2) $\mathbf{G} \equiv \mathbf{H}$ *if and only if* $\omega(\mathbf{G}) = \omega(\mathbf{H})$*, for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$*.*

The following hold:

(a) *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv \mathbf{H}$ if and only if $\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{G}) = \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{H})$.*

(b) *There is no nonempty class $\mathcal{F} \subseteq \mathcal{G}$ such that for all graphs $\mathbf{G}$ and $\mathbf{H}$, we have $\mathbf{G} \equiv \mathbf{H}$ if and only if $\mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{G}) = \mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{H})$.* $\square$

4.4 Summary

The various relaxations of graph isomorphism studied in this chapter in terms of the existence of a characterization by the equality of left or right profile, over N or over B, restricted to a nonempty class $\mathcal{F}$ of graphs are summarized in Table 4.1.

Table 4.1: Equivalence relations over graphs vs. indistinguishability by the profiles.

EQUIVALENCE	$\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}$	$\mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}$	$\mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}$	$\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}$
isomorphism, FO-equivalence, C-equivalence	$\mathcal{F} = \mathcal{G}$	$\mathcal{F} = \mathcal{G}$	none	none
same number of vertices, C^1-equivalence, C_1-equivalence	$\mathcal{F} \subseteq \mathcal{I}$	none	none	none
same number of edges	$\mathcal{F} \subseteq \{\bigoplus_n \mathbf{K}_2 \mid n \in \mathbb{Z}^+\}$	none	none	none
C^2-equivalence, indistinguishability by color refinement	$\mathcal{F} = \mathcal{T}$	none	none	none
C^k-equivalence, indistinguishability by $(k-1)$-dimensional Weisfeiler-Leman algorithm, where $k \geq 2$	$\mathcal{F} = \mathcal{T}^{k-1}$	none	none	none
C_k-equivalence, where $k \in \mathbb{Z}^+$	$\mathcal{F} = \mathcal{T}_k$	none	none	none
cospectrality	$\mathcal{F} = \mathcal{C}$	none	none	none
chromatic equivalence	none	$\mathcal{F} = \mathcal{K}$	none	none
FO^1-equivalence, FO_1-equivalence	none (unless $\mathcal{F} = \emptyset$ allowed)	none (unless $\mathcal{F} = \emptyset$ allowed)	$\mathcal{F} \subseteq \mathcal{I}$	none
FO^k-equivalence, where $k \geq 2$	none	none	none	none
FO_k-equivalence, where $k \geq 2$	none	none	none	none
homomorphic equivalence	none	none	$\mathcal{F} = \mathcal{G}$	$\mathcal{F} = \mathcal{G}$
same chromatic number	none	none	none	$\mathcal{F} = \mathcal{K}$
same clique number	none	none	$\mathcal{F} = \mathcal{K}$	none

Chapter 5

Query Algorithms

The characterizations of isomorphism among graphs or structures given by Theorems 4.1 and 4.2 not only bring up questions about characterizing equivalence relations (over graphs) that are relaxations of isomorphism in restricted left or right profile, but also raise the issues of determining the membership of a graph or structure in a given class via queries of the form of homomorphism counts. Many results in this chapter are applicable for both graphs and structures (and are from [8, 51]), as opposed to the previous chapter where the focus was mainly on graphs. That said, most examples presented in this chapter are about graphs due to their simplicity. As before, we use the symbol $\mathcal{U}$ for the class $\mathcal{G}$ or the class $\mathcal{A}$ of structures (see Proviso 2).

5.1 Basic Definitions and Examples

Consider the scenario: Given a graph $\mathbf{G}$, is $|V(\mathbf{G})| \leq 5$? By Example 4.1, $\hom(\mathbf{I}_1, \mathbf{G}) = |V(\mathbf{G})|$ and this homomorphism count reveals the information about the number of vertices in a graph $\mathbf{G}$. We can turn this observation into a decision algorithm for the above problem that involves queries made in the framework of homomorphism counts: Given an input graph $\mathbf{G}$, answer "yes" as output if $\hom(\mathbf{I}_1, \mathbf{G}) \leq 5$ and "no" otherwise.

This scenario motivates the theme of this chapter, namely, the notion of *query algorithms*, in which the queries are in the form of homomorphism counts. It was introduced in [10], which was followed up by [8]. The formal definition of various query algorithms is as follows.

Definition 5.1. Let $\mathcal{D} \subseteq \mathcal{U}$ be a class, and let K be the semiring B or N.

(a) For every $n \in \mathbb{Z}^+$, every class $\mathcal{F} \subseteq \mathcal{U}$ of size n, and every set X of n-tuples over the underlying set of K, we say that $(\mathcal{F}, X)$ *is a left n-query algorithm for $\mathcal{D}$ over K* if for every $\mathbf{A} \in \mathcal{U}$, we have that $\mathbf{A} \in \mathcal{D}$ if and only if $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) \in X$, and similarly, we say that $(\mathcal{F}, X)$ *is a right n-query algorithm for $\mathcal{D}$ over K* if for every $\mathbf{A} \in \mathcal{U}$, we have that $\mathbf{A} \in \mathcal{D}$ if and only if $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}) \in X$.

(b) For every $n \in \mathbb{Z}^+$, we say that $\mathcal{D}$ *admits a left n-query algorithm over K* if there is a left n-query algorithm for $\mathcal{D}$ over K, and similarly, we say that $\mathcal{D}$ *admits a right n-query algorithm over K* if there is a right n-query algorithm for $\mathcal{D}$ over K.

(c) We say that $\mathcal{D}$ *admits a left query algorithm over K* if there is a left n-query algorithm for $\mathcal{D}$ over K for some $n \in \mathbb{Z}^+$, and similarly, we say that $\mathcal{D}$ *admits a right query algorithm over K* if there is a right n-query algorithm for $\mathcal{D}$ over K for some $n \in \mathbb{Z}^+$.

In case of $K = \mathrm{N}$, the above Definition 5.1(a) becomes more of an abstract notion of "algorithm" because the set X of n-tuples over the underlying set $\mathbb{N}$ of N can be undecidable.

Through the end of this section, we present a number of examples to illustrate the above definition, and we state and prove some simple results that will be useful in later sections.

Example 5.1. Fix an arbitrary graph $\mathbf{G}$. We have that $(\{\mathbf{G}\}, \mathbb{B})$ is both a left 1-query algorithm and a right 1-query algorithm over B for $\mathcal{G}$; changing $\mathbb{B}$ to $\mathbb{N}$, we obtain a left 1-query algorithm and a right 1-query algorithm over N for this class. ($\mathbb{B} = \{0, 1\}$ and B is the Boolean semiring, while N is the bag-set semiring, see Subsection 3.2.2.) Moreover, $(\{\mathbf{G}\}, \emptyset)$ is both a left 1-query algorithm and a right 1-query algorithm over B for the empty class $\emptyset$; it is also a left 1-query algorithm and a right 1-query algorithm over N for this class.

The independent sets are the graphs that contain no edges and are precisely the minimals for the preorder $\rightarrow$ over graphs (see Remark 3.10(b)). They are exactly those that have a homomorphism to $\mathbf{I}_1$.

Example 5.2. The class $\mathcal{I}$ of independent sets admits both a left 1-query algorithm $(\{\mathbf{K}_2\}, \{0\})$ and a right 1-query algorithm $(\{\mathbf{I}_1\}, \{1\})$ over B. Note that $(\{\mathbf{K}_2\}, \{0\})$ is also a left 1-query algorithm over N for $\mathcal{I}$. Moreover, this class admits a right 1-query algorithm over N as well, namely $(\{\mathbf{I}_1\}, \mathbb{Z}^+)$.

In the preceding examples, we see that if a class admits a left (or right) k-query algorithm over B, then it admits one over N. Indeed, this is true in general, and an intuitive explanation is that $\hom_{\mathrm{N}}$ reveal more information than $\hom_{\mathrm{B}}$, as evidenced by the upcoming proposition.

Proposition 5.1. *Let $\mathcal{D} \subseteq \mathcal{U}$ be a class, and let $n \in \mathbb{Z}^+$. In the following, for every set*

$X \subseteq \mathbb{B}^n$, we let $X' := \bigcup_{t \in X} X_t$, where

$$X_t := \{(t'_1, \dots, t'_n) \in \mathbb{N}^n \mid t'_i = 0 \text{ if and only if } t_i = 0 \text{ for all } i \in [n]\}$$

for every n-tuple $t = (t_1, \dots, t_n) \in \mathbb{B}^n$.

(a) *If $\mathcal{D}$ admits a left n-query algorithm $(\mathcal{F}, X)$ over B for some set $X \subseteq \mathbb{B}^n$, then it admits a left n-query algorithm $(\mathcal{F}, X')$ over N for the set X'.*

(b) *If $\mathcal{D}$ admits a right n-query algorithm $(\mathcal{F}, X)$ over B for some set $X \subseteq \mathbb{B}^n$, then it admits a right n-query algorithm $(\mathcal{F}, X')$ over N for the set X'.*

Proof. We only prove part (a), as part (b) is entirely analogous. With the conditions and definitions stated in the proposition, it is straightforward that for every $\mathbf{A} \in \mathcal{U}$,

$$\begin{array}{ll} & \mathbf{A} \in \mathcal{D} \\ \text{iff} & \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{A}) \in X \\ \text{iff} & \text{there is a } t \in X \text{ such that } \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{A}) = t \\ \text{iff} & \text{there is a } t \in X \text{ such that } \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{A}) \in X_t \\ \text{iff} & \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{A}) \in X'. \end{array}$$

□

Remark 5.1. Let $\mathcal{D} \subseteq \mathcal{U}$ be a class. The following are immediate by Definition 5.1 (also see [10]).

(a) *For every nonempty class $\mathcal{F} \subseteq \mathcal{U}$, we have that $\mathcal{D}$ admits a left query algorithm over K of the form $(\mathcal{F}, X)$ if and only if for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, we have that $\mathrm{lpf}^{\mathcal{F}}_K(\mathbf{A}) = \mathrm{lpf}^{\mathcal{F}}_K(\mathbf{B})$ and $\mathbf{A} \in \mathcal{D}$ together imply $\mathbf{B} \in \mathcal{D}$.*

(b) *For every nonempty class $\mathcal{F} \subseteq \mathcal{U}$, we have that $\mathcal{D}$ admits a right query algorithm over K of the form $(\mathcal{F}, X)$ if and only if for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, we have that $\mathrm{rpf}^{\mathcal{F}}_K(\mathbf{A}) = \mathrm{rpf}^{\mathcal{F}}_K(\mathbf{B})$*

and $\mathbf{A} \in \mathcal{D}$ *together imply* $\mathbf{B} \in \mathcal{D}$.

In fact, for the "if" direction, it suffices to take $X := \{\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{D}\}$ and $X := \{\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{D}\}$, respectively. We restate the above in an equivalent form that will be useful.

(c) *A class* $\mathcal{D} \subseteq \mathcal{U}$ *does not admit any left query algorithm over* K *if and only if for every finite nonempty class* $\mathcal{F} \subseteq \mathcal{U}$*, there are* $\mathbf{A} \in \mathcal{D}$ *and* $\mathbf{B} \notin \mathcal{D}$ *such that* $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{B})$.

(d) *A class* $\mathcal{D} \subseteq \mathcal{U}$ *does not admit any right query algorithm over* K *if and only if for every finite nonempty class* $\mathcal{F} \subseteq \mathcal{U}$*, there are* $\mathbf{A} \in \mathcal{D}$ *and* $\mathbf{B} \notin \mathcal{D}$ *such that* $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}) = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{B})$.

A clique is a graph in which every two distinct vertices are connected by an edge: For all $n \in \mathbb{Z}^+$, the clique $\mathbf{K}_n$ of size n has $\binom{n}{2} = n(n-1)/2$ edges in it, and it has chromatic number $\chi(\mathbf{K}_n) = n$. Propositions 3.9 and 3.10 together imply that for every graph $\mathbf{G}$, if $\mathbf{K}_n \to \mathbf{G}$ then $\mathbf{G}$ has chromatic number $\chi(\mathbf{G}) \geq n$.

Example 5.3. The class $\mathcal{K}$ of cliques admits a left 2-query algorithm

$$(\{\mathbf{I}_1, \mathbf{K}_2\}, \{(k, k(k-1)) \mid k \in \mathbb{Z}^+\})$$

over N. However, it does not admit any right query algorithm over N (hence not over B either, by Proposition 5.1(b)). To see this, for an arbitrary class $\mathcal{F}$ of n graphs ($n \in \mathbb{Z}^+$), let s be greater than the maximum size of the graphs in $\mathcal{F}$. Then, by Remark 5.1, the two graphs $\mathbf{K}_s \in \mathcal{K}$ and $(\mathbf{K}_s \oplus \mathbf{I}_1) \notin \mathcal{K}$ witness the assertion, because the graphs in $\mathcal{F}$ each have chromatic number $< s$ and hence $\mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{K}_s) = (\underbrace{0, \ldots, 0}_{n\text{-times}}) = \mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{K}_s \oplus \mathbf{I}_1)$ by

Proposition 3.10. (Note that $\mathbf{K}_s$ is connected while $\mathbf{K}_s \oplus \mathbf{I}_1$ is not, and this will be useful later in Example 5.8.) The same technique is employed in [10] to show that *the class of graphs containing an isolated vertex does not admit any right query algorithm over* N.

By Remark 5.1, the class $\mathcal{K}$ does not admit any left query algorithm over B, either: Consider the two graphs $\mathbf{K}_2 \in \mathcal{K}$ and $(\mathbf{K}_2 \oplus \mathbf{K}_2) \notin \mathcal{K}$. Observe that they are homomorphically equivalent, i.e., $\mathbf{K}_2 \leftrightarrow (\mathbf{K}_2 \oplus \mathbf{K}_2)$. By Proposition 3.18, we have $\mathbf{G} \to \mathbf{K}_2$ if and only if $\mathbf{G} \to (\mathbf{K}_2 \oplus \mathbf{K}_2)$, for every graph $\mathbf{G}$. Hence, $\mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{K}_2) = \mathrm{lpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{K}_2 \oplus \mathbf{K}_2)$.

The *bipartite graphs* are precisely the 2-colorable graphs or, equivalently, the acyclic graphs, which are exactly the forests (see Section 2.2). By Corollary 3.16, the number of 2-colorings of every bipartite graph of n connected components is exactly 2^n. In contrast, if a graph is not bipartite then it has no 2-coloring.

Example 5.4. The class $\mathcal{D}^{\mathrm{bip}}$ of bipartite graphs admits a right 1-query algorithm $(\{\mathbf{K}_2\}, \{1\})$ over B. By Proposition 5.1, it admits a right 1-query algorithm $(\{\mathbf{K}_2\}, \mathbb{Z}^+)$ over N. In fact, for every set $X \subseteq \mathbb{Z}^+$, we have that $(\{\mathbf{K}_n\}, X)$ is a right 1-query algorithm over N for $\mathcal{D}^{\mathrm{bip}}$ if and only if X contains $\{2^n \mid n \in \mathbb{Z}^+\}$ as a subset.

However, this class does not admit any left query algorithm over N (hence nor over B by Proposition 5.1). To see this, consider an arbitrary finite nonempty class $\mathcal{F}$ of graphs as a candidate. Let $n \geq 3$ be an odd integer greater than the maximum size of the graphs in $\mathcal{F}$. Then, take the two graphs $\mathbf{C}_{2n} \in \mathcal{D}^{\mathrm{bip}}$ and $(\mathbf{C}_n \oplus \mathbf{C}_n) \notin \mathcal{D}^{\mathrm{bip}}$. (In fact, we also have that $\mathbf{C}_{2n}$ is connected while $\mathbf{C}_n \oplus \mathbf{C}_n$ is not, which will turn out to be useful in Example 5.5). Therefore, $\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{C}_{2n}) = \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{C}_n \oplus \mathbf{C}_n)$ by Proposition 5.2 (presented next), and we conclude by Remark 5.1.

Proposition 5.2. *Let $n \geq 3$ be an integer. For every graph $\mathbf{G}$ of size $< n$, we have* $\hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_{2n}) = \hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_n \oplus \mathbf{C}_n)$.

Proof. Fix an integer $n \geq 3$, and let $\mathbf{G}$ be a graph of size $< n$. We distinguish two cases below.

Case 1. $\mathbf{G}$ is connected. We first argue that $\mathbf{C}_{2n}$ and $\mathbf{C}_n \oplus \mathbf{C}_n$ are indistinguishable by the number of *injective* homomorphisms from connected graphs of size $< n$:

$$\text{For all connected graphs } \mathbf{H} \text{ of size} < n \text{, we have } \mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) = \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n). \qquad (*)$$

The argument of $(*)$ is split into three cases.

Case $(*)$-1. $\mathbf{H}$ contains a cycle of length ≥ 3. Then $\mathbf{H}$ has girth $\gamma(\mathbf{H}) < n$. Since $\mathbf{C}_{2n}$ and $\mathbf{C}_n \oplus \mathbf{C}_n$ both have girth $\geq n$, we have $\mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) = 0 = \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n)$.

Case $(*)$-2. $\mathbf{H}$ contains no cycles of length ≥ 3 and has degree > 2. Then we immediately have $\mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) = 0 = \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n)$ because $\mathbf{C}_{2n}$ and $\mathbf{C}_n \oplus \mathbf{C}_n$ both have degree 2.

Case $(*)$-3. $\mathbf{H}$ contains no cycles of length ≥ 3 and has degree ≤ 2. Then $\mathbf{H}$ is a path graph $\mathbf{P}_k$ for some $k < n$. If $k = 1$, then we have $\mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) = 2n = \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n)$. If $k > 1$, then we have $\mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) = 4n = \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n)$.

Next, since $\mathbf{G}$ is a connected graph of size $< n$, its homomorphic image must also be a connected graph (by Proposition 3.11(b)) of size $< n$. By Proposition 3.8, we have

$$\hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_{2n}) = \sum_{H \text{ is connected and has size} < n} \mathrm{sur}(\mathbf{G}, \mathbf{H}) \cdot \mathrm{inj}(\mathbf{H}, \mathbf{C}_{2n}) / \mathrm{aut}(\mathbf{H})$$

and

$$\hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_n \oplus \mathbf{C}_n) = \sum_{H \text{ is connected and has size} < n} \mathrm{sur}(\mathbf{G}, \mathbf{H}) \cdot \mathrm{inj}(\mathbf{H}, \mathbf{C}_n \oplus \mathbf{C}_n) / \mathrm{aut}(\mathbf{H}),$$

and it follows by $(*)$ that $\hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_{2n}) = \hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_n \oplus \mathbf{C}_n)$.

Case 2. $\mathbf{G}$ *is not connected.* We can assume for some $k \geq 2$ that $\mathbf{G}$ consists of the connected components $\mathbf{G}_1, \ldots, \mathbf{G}_k$ (i.e., $\mathbf{G} = \bigoplus_{i=1}^{k} \mathbf{G}_i$). Then the above Case 1 yields $\hom_{\mathrm{N}}(\mathbf{G}_1, \mathbf{C}_{2n}) = \hom_{\mathrm{N}}(\mathbf{G}_1, \mathbf{C}_n \oplus \mathbf{C}_n), \ldots, \hom_{\mathrm{N}}(\mathbf{G}_k, \mathbf{C}_{2n}) = \hom_{\mathrm{N}}(\mathbf{G}_k, \mathbf{C}_n \oplus \mathbf{C}_n)$ and hence $\hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_{2n}) = \hom_{\mathrm{N}}(\mathbf{G}, \mathbf{C}_n \oplus \mathbf{C}_n)$ by Proposition 3.14(c). $\square$

Example 5.5. The class $\mathcal{D}^{\mathrm{bip-conn}}$ of bipartite connected graphs admits a right 1-query algorithm $(\{\mathbf{K}_2\}, \{2\})$ over N but no left query algorithm over N (hence nor over B by Proposition 5.1). One can argue as in Example 5.4 to see that $\mathcal{D}^{\mathrm{bip-conn}}$ does not admit any left query algorithm over N (note that n can be chosen to be an even integer ≥ 2 in this case).

Moreover, $\mathcal{D}^{\mathrm{bip-conn}}$ does not admit any right query algorithm over B: Take the two graphs $\mathbf{K}_2$ and $\mathbf{K}_2 \oplus \mathbf{K}_2$ considered in Example 5.3. Note that $\mathbf{K}_2 \in \mathcal{D}^{\mathrm{bip-conn}}$ while $(\mathbf{K}_2 \oplus \mathbf{K}_2) \notin \mathcal{D}^{\mathrm{bip-conn}}$ but the fact that they are homomorphically equivalent (that is, $\mathbf{K}_2 \leftrightarrow (\mathbf{K}_2 \oplus \mathbf{K}_2)$) implies that $\mathbf{K}_2 \to \mathbf{G}$ if and only if $(\mathbf{K}_2 \oplus \mathbf{K}_2) \to \mathbf{G}$, for every graph $\mathbf{G}$ (by Proposition 3.18). Therefore, $\mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{K}_2) = \mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{K}_2 \oplus \mathbf{K}_2)$.

We say that a class $\mathcal{D} \subseteq \mathcal{U}$ is *closed under homomorphic equivalence* if for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$ such that $\mathbf{A} \leftrightarrow \mathbf{B}$, we have $\mathbf{A} \in \mathcal{D}$ implies $\mathbf{B} \in \mathcal{D}$. Examples of classes closed under homomorphic equivalence include, for every $\mathbf{A} \in \mathcal{U}$, the two classes $\mathrm{CQ}(\mathbf{A}) = \{\mathbf{B} \in \mathcal{U} \mid \mathbf{A} \to \mathbf{B}\}$ and $\mathrm{CSP}(\mathbf{A}) = \{\mathbf{B} \in \mathcal{U} \mid \mathbf{B} \to \mathbf{A}\}$.

Example 5.6. The class $\mathrm{CQ}(\mathbf{K}_3)$ consists of all graphs that contain $\mathbf{K}_3$ as a subgraph, and admits a left 1-query algorithm $(\{\mathbf{K}_3\}, \{1\})$ over B and, by Proposition 5.1, a left 1-query algorithm $(\{\mathbf{K}_3\}, \mathbb{Z}^+)$ over N.

However, it does not admit any right query algorithm over N (and not over B, either, by Proposition 5.1): Suppose that $(\mathcal{F}, X)$ is a candidate right n-query algorithm over N for CQ($\mathbf{K}_3$). Let $s \geq 3$ be greater than the maximum chromatic number of the graphs in $\mathcal{F}$ (note that $\mathcal{F} \neq \emptyset$). Then $\mathbf{K}_s \in \mathrm{CQ}(\mathbf{K}_3)$ and $\mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{K}_s) = (\underbrace{0, \ldots, 0}_{n\text{-times}})$. By Lemma 2.5, there is a graph $\mathbf{G}$ with chromatic number $\chi(\mathbf{G}) = s$ and girth $\gamma(\mathbf{G}) \geq 4$, for which $\mathbf{G} \notin \mathrm{CQ}(\mathbf{K}_3)$ and $\mathrm{rpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{G}) = (\underbrace{0, \ldots, 0}_{n\text{-times}})$. It follows from Remark 5.1 that $(\mathcal{F}, X)$ is not a right n-query algorithm over N for CQ($\mathbf{K}_3$). (A stronger result is proven in [10].)

Example 5.7. The class CSP($\mathbf{K}_3$) consists of all graphs that are 3-colorable, namely those that have chromatic numbers ≤ 3. Note that it is symmetric to CQ($\mathbf{K}_3$) in the previous example in terms of the side of $\rightarrow$ (seen as a binary relation). Thus, it is not surprising that CSP($\mathbf{K}_3$) admits a right 1-query algorithm $(\{\mathbf{K}_3\}, \{1\})$ over B and, by Proposition 5.1, a right 1-query $(\{\mathbf{K}_3\}, \mathbb{Z}^+)$ over N.

Analogously to CQ($\mathbf{K}_3$), the class CSP($\mathbf{K}_3$) admits no left query algorithm over B or over N (also see [10]). It is, however, technically more involved to prove this latter assertion, which will be done in the next section (see Lemma 5.12).

The classes of graphs considered in Examples 5.3 and 5.5 are not closed under homomorphic equivalence, and we argued that they do not admit a left nor a right query algorithm over B by presenting a pair of graphs $\mathbf{G}$ and $\mathbf{H}$ which are homomorphically equivalent and of which exactly one is in the class. The argument follows from a more general result.

Proposition 5.3. *Let $\mathcal{D} \subseteq \mathcal{U}$ be a class. If $\mathcal{D}$ admits a left query algorithm or a right query algorithm over* B*, then it is closed under homomorphic equivalence.*

Proof. The case for right query algorithms over B is omitted since it can be proved analogously. Assume that $\mathcal{D}$ admits a left query algorithm $(\mathcal{F}, X)$ over B. For all $\mathbf{A}$ and $\mathbf{B}$, if $\mathbf{A} \leftrightarrow \mathbf{B}$, then,

$$\begin{array}{ll} & \mathbf{A} \in \mathcal{D} \\ \text{iff} & \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{A}) \in X \\ \text{iff} & \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{B}) \in X \quad \text{(by Proposition 4.25)} \\ \text{iff} & \mathbf{B} \in \mathcal{D}. \end{array}$$ □

Assume that $\mathcal{D} \subseteq \mathcal{U}$ is a finite class. If $\mathcal{D} = \emptyset$, then it admits both a left and a right query algorithm over B, by Example 5.1. However, if $\mathcal{D} \neq \emptyset$, then for some $\mathbf{A} \in \mathcal{D}$ and some $n \in \mathbb{Z}^+$, we have $\bigoplus_n \mathbf{A} \notin \mathcal{D}$ because $\mathcal{D}$ is finite, although it is homomorphically equivalent to $\mathbf{A}$. The next corollary follows from Proposition 5.3.

Corollary 5.4. *Let $\mathcal{D} \subseteq \mathcal{U}$ be a finite class. If $\mathcal{D}$ is nonempty, then it admits no left query algorithm and no right query algorithm over* B. □

Recall the scenario at the beginning of this section: Given a graph $\mathbf{G}$, is $|V(\mathbf{G})| \leq 5$? There we saw an (informal version of the) left 1-query algorithm $(\{\mathbf{I}_1\}, \{1, 2, 3, 4, 5\})$ over N for the class of graphs that are yes-instances. What may seem less trivial is that this class also admits a right query algorithm over N.

Proposition 5.5. [10] *Every finite class $\mathcal{D} \subseteq \mathcal{U}$ admits both a left query algorithm and a right query algorithm over* N.

Proof. Let $\mathcal{D} \subseteq \mathcal{U}$ be a finite class. We distinguish two cases below.

Case 1. $\mathcal{D}$ is empty. Then $(\{\mathbf{C}\}, \emptyset)$, where $\mathbf{C} \in \mathcal{U}$ is arbitrarily chosen, is simultaneously a left 1-query algorithm and a right 1-query algorithm over N (cf. Example 5.1).

Table 5.1: Finite classes vs. left query algorithms and right query algorithms.

FINITE CLASSES $\mathcal{D} \subseteq \mathcal{U}$	LEFT QUERY ALGORITHM	RIGHT QUERY ALGORITHM
$\mathcal{D} = \emptyset$	over both B and N	over both B and N
$\mathcal{D} \neq \emptyset$	over N only	over N only

Case 2. $\mathcal{D}$ is nonempty. Then let n be the maximum size of the members in $\mathcal{D}$. Take $\mathcal{F}$ for this specific n, as in Corollary 4.5(ii) or 4.6(ii), depending on whether $\mathcal{U} = \mathcal{G}$ or $\mathcal{U} = \mathcal{A}$, and take the set $X := \{\mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{C}) \mid \mathbf{C} \in \mathcal{D}\}$. It follows that $(\mathcal{F}, X)$ is a left query algorithm over N for $\mathcal{D}$ because for all $\mathbf{A} \in \mathcal{U}$, we have

$$
\begin{aligned}
& \mathbf{A} \in \mathcal{D} \\
\text{iff} \quad & \text{there is a } \mathbf{C} \in \mathcal{D} \text{ such that } \mathbf{A} \cong \mathbf{C} && \text{(classes are assumed to be closed under } \cong\text{)} \\
\text{iff} \quad & \text{there is a } \mathbf{C} \in \mathcal{D} \text{ such that } \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{A}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{C}) && \text{(by Corollary 4.5 or 4.6)} \\
\text{iff} \quad & \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{A}) \in X.
\end{aligned}
$$

As for a right query algorithm over N for $\mathcal{D}$, take $\mathcal{F}'$ for the value m that is a function of this specific n, as in Corollary 4.5(iii) or 4.6(iii), depending on whether $\mathcal{U} = \mathcal{G}$ or $\mathcal{U} = \mathcal{A}$, and take $X' := \{\mathrm{rpf}^{\mathcal{F}'}_{\mathrm{N}}(\mathbf{C}) \mid \mathbf{C} \in \mathcal{D}\}$. Then, we can argue that $(\mathcal{F}', X')$ is a right query algorithm over N for $\mathcal{D}$ analogously. □

A quick summary of when a finite class admits a left or a right query algorithm over B or over N is provided in Table 5.1.

As opposed to finite classes, the situation is more complicated for infinite classes. By Proposition 5.1, there are three cases in terms of the semirings B or N over which an infinite class admits a left query algorithm, and likewise for right query algorithm: (1) both B and N, (2) only N, and (3) neither B nor N.

For left query algorithms for infinite classes of graphs, we have seen case (1) in

Table 5.2: Finite classes vs. left query algorithms and right query algorithms.

INFINITE CLASSES $\subseteq \mathcal{G}$	LEFT QUERY ALGORITHM	RIGHT QUERY ALGORITHM
$\mathcal{G}$ (Example 5.1)	over both B and N	over both B and N
$\mathcal{I}$ (Example 5.2)	over both B and N	over both B and N
$\mathcal{K}$ (Example 5.3)	over N only	not over B nor over N
$\mathcal{D}^{\text{bip}}$ (Example 5.4)	not over B nor over N	over both B and N
$\mathcal{D}^{\text{bip-conn}}$ (Example 5.5)	not over B nor over N	over N only
CQ($\mathbf{K}_3$) (Example 5.6)	over both B and N	not over B nor over N
CSP($\mathbf{K}_3$) (Example 5.7)	not over B nor over N	over both B and N
$\mathcal{D}^{\text{conn}}$ (Example 5.8)	not over B nor over N	not over B nor over N

Examples 5.1, 5.2 and 5.6, case (2) in Example 5.3 and case (3) in Examples 5.4, 5.5 and 5.7. As to right query algorithms for infinite classes, case (1) is illustrated in Examples 5.1, 5.2, 5.4 and 5.7, case (2) in Example 5.5, and case (3) in Examples 5.3 and 5.6.

There is indeed a common infinite class of graphs that admits neither left nor right algorithm over B or N, as seen in the next example.

Example 5.8. The class $\mathcal{D}^{\text{conn}}$ of connected graphs admits no left query algorithm over N, by the argument in the second paragraph of Example 5.4 (also cf. Example 5.5), nor does it admit any right query algorithm over N, by the argument in the first paragraph of Example 5.3. By Proposition 5.1, these exclude the possibility that $\mathcal{D}^{\text{conn}}$ might admit a left or a right query algorithm over B.

Therefore, case (3) holds for both left query algorithms and right query algorithms for the class $\mathcal{D}^{\text{conn}}$. All these are summarized in Table 5.2.

Let us reconsider the scenario introduced at the beginning: Given a graph $\mathbf{G}$, is $|V(\mathbf{G})| \leq 5$? By Corollary 5.4 and Proposition 5.5, the class of yes-instances admits both a left and a right query algorithm over N but neither a left nor a right query algorithm over B. In fact, the same holds for the class of no-instances, i.e., its complement. In general, the collection of classes admitting a left (or right) query algorithm over a fixed semiring B or

N is closed under the Boolean operations on classes.

Proposition 5.6. [8, 10] *Let K be the semiring* B *or* N *in the following:*

(a) *If a class $\mathcal{D} \subseteq \mathcal{U}$ admits a left (or right) query algorithm over K, then so does its complement $\overline{\mathcal{D}} = \mathcal{U} \setminus \mathcal{D}$.*

(b) *If both the classes $\mathcal{D}_1, \mathcal{D}_2 \subseteq \mathcal{U}$ admit a left (or right) query algorithm over K, then so do their union $\mathcal{D}_1 \cup \mathcal{D}_2$ and their intersection $\mathcal{D}_1 \cap \mathcal{D}_2$.*

Proof. For simplicity, we will only consider $K = \mathrm{N}$, since the case $K = \mathrm{B}$ is analogous.

For part (a), suppose that $(\mathcal{F}, X)$ is a left n-query algorithm for the class $\mathcal{D}$ for some $n \in \mathbb{Z}^+$, some class $\mathcal{F} \subseteq \mathcal{U}$ of size n and some set $X \subseteq \mathbb{N}^n$. Then $(\mathcal{F}, \mathbb{N}^n \setminus X)$ is a left n-query algorithm for $\overline{\mathcal{D}}$ because for all $\mathbf{A} \in \mathcal{U}$, we have

$$
\begin{aligned}
& \mathbf{A} \in \overline{\mathcal{D}} \\
\text{iff} \quad & \mathbf{A} \notin \mathcal{D} \\
\text{iff} \quad & \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) \notin X \\
\text{iff} \quad & \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) \in (\mathbb{N}^n \setminus X).
\end{aligned}
$$

Right query algorithms are similar and hence are omitted.

For part (b), we recall the viewpoint in Section 2.1 that a sequence is a function. More precisely, for finite nonempty $\mathcal{F} \subseteq \mathcal{U}$ and for $\mathbf{A} \in \mathcal{U}$, the left profile $\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A})$ is a function $\mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) : \mathcal{F} \to \mathbb{N}$ that maps every $\mathbf{C}$ to $\hom_{\mathrm{N}}(\mathbf{C}, \mathbf{A})$ and, for every left query algorithm $(\mathcal{F}, X)$ over N, a tuple $t \in X$ is a function $t : \mathcal{F} \to \mathbb{N}$ that maps $\mathbf{C}$ to a natural number. (Likewise for right profiles and right query algorithms.)

For every left query algorithm $(\mathcal{F}, X)$ and every class $\mathcal{F}' \supseteq \mathcal{F}$, we write

$$X^{\mathcal{F}'} := \{t' : \mathcal{F}' \to \mathbb{N} \mid t'|_{\mathcal{F}} = t \text{ for some } t \in X\},$$

where $t'|_{\mathcal{F}}$ denotes the restriction of the function t' to the domain $\mathcal{F}$. It should be clear that the two left query algorithms $(\mathcal{F}, X)$ and $(\mathcal{F}', X^{\mathcal{F}'})$ determine the membership in the same class.

Now, given left query algorithms $(\mathcal{F}_1, X_1)$ and $(\mathcal{F}_2, X_2)$ for the classes $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, take $\mathcal{F} := \mathcal{F}_1 \cup \mathcal{F}_2$. It follows that $(\mathcal{F}, X_1^{\mathcal{F}} \cup X_2^{\mathcal{F}})$ is a left query algorithm for their union $\mathcal{D}_1 \cup \mathcal{D}_2$ because for all $\mathbf{A} \in \mathcal{U}$,

$$\begin{array}{ll} & \mathbf{A} \in (\mathcal{D}_1 \cup \mathcal{D}_2) \\ \text{iff} & \mathbf{A} \in \mathcal{D}_1 \text{ or } \mathbf{A} \in \mathcal{D}_2 \\ \text{iff} & \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}_1}(\mathbf{A}) \in X_1 \text{ or } \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}_2}(\mathbf{A}) \in X_2 \\ \text{iff} & \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) \in X_1^{\mathcal{F}} \text{ or } \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) \in X_2^{\mathcal{F}} \\ \text{iff} & \mathrm{lpf}_{\mathrm{N}}^{\mathcal{F}}(\mathbf{A}) \in (X_1^{\mathcal{F}} \cup X_2^{\mathcal{F}}). \end{array}$$

Likewise, $(\mathcal{F}, X_1^{\mathcal{F}} \cap X_2^{\mathcal{F}})$ is a left query algorithm for $\mathcal{D}_1 \cap \mathcal{D}_2$. The case for right query algorithms is similar and hence omitted. □

Note that while $\mathcal{D}^{\mathrm{bip}}$ admits a right query algorithm over N (cf. Example 5.4) and $\mathcal{D}^{\mathrm{conn}}$ of connected graphs does not (cf. Example 5.8), their intersection $(\mathcal{D}^{\mathrm{bip}} \cap \mathcal{D}^{\mathrm{conn}}) = \mathcal{D}^{\mathrm{bip-conn}}$ does admit one (cf. Example 5.5).

We close this section with a simple characterization of when a class admits a left query algorithm over B and of when a class admits a right query algorithm over B.

Proposition 5.7. *Let $\mathcal{D} \subseteq \mathcal{U}$ be a class. The following hold:*

(a) *$\mathcal{D}$ admits a left query algorithm over* B *if and only if it is a Boolean combination of* CQ *classes.*

(b) *$\mathcal{D}$ admits a right query algorithm over* B *if and only if it is a Boolean combination of*

CSP *classes.*

Proof. We only prove part (a), as part (b) is entirely analogous. We distinguish two cases as follows.

Case 1. $\mathcal{D}$ is empty. The "if" part is trivial (cf. Example 5.1), as is the "only if" part: Take $\mathrm{CQ}(\mathbf{A}) \cap \overline{\mathrm{CQ}(\mathbf{A})}$ for an arbitrary $\mathbf{A} \in \mathcal{U}$.

Case 2. $\mathcal{D}$ is nonempty. For the "if" direction, assume that $\mathcal{D}$ is a Boolean combination of the classes $\mathrm{CQ}(\mathbf{A}_1), \ldots, \mathrm{CQ}(\mathbf{A}_n)$ which is, without loss of generality, in the particular disjunctive normal form:

$$\mathcal{D} = \bigcup_{i=1}^{m} \bigcap_{j=1}^{n} \mathcal{D}_{ij},$$

where $m \in [2^n]$ and $\mathcal{D}_{ij} = \mathrm{CQ}(\mathbf{A}_j)$ or $\mathcal{D}_{ij} = \overline{\mathrm{CQ}(\mathbf{A}_j)}$ (the complement of $\mathrm{CQ}(\mathbf{A}_j)$) for all $j \in [n]$. Take $\mathcal{F} := \{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$. For all $i \in [m]$, let $t_i = (t_{i1}, \ldots, t_{in})$ be the n-tuple over $\mathbb{B}$ such that for all $j \in [n]$,

$$t_{ij} = \begin{cases} 1 & \text{if } \mathcal{D}_{ij} = \mathrm{CQ}(\mathbf{A}_j), \\ 0 & \text{otherwise.} \end{cases}$$

Take $X := \{t_1, \ldots, t_m\}$. It is straightforward to verify that $(\mathcal{F}, X)$ is a left query algorithm for $\mathcal{D}$ over B.

Conversely, for the "only if" direction, assume that $(\mathcal{F}, X)$ is a left query algorithm for $\mathcal{D}$ over B, where $\mathcal{F} = \{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$ and $X \subseteq \mathbb{B}^n$ is nonempty (since $\mathcal{D} \neq \emptyset$). For each n-tuple $t = (t_1, \ldots, t_n) \in X$, define the class

$$\mathcal{D}^t := \bigcap_{k=1}^{n} \mathcal{D}_k^t$$

in which

$$\mathcal{D}_k^t := \begin{cases} \mathrm{CQ}(\mathbf{A}_k) & \text{if } t_k = 1, \\ \overline{\mathrm{CQ}(\mathbf{A}_k)} & \text{otherwise,} \end{cases}$$

for all $k \in [n]$. It is easy to verify that $\mathcal{D} = \bigcup_{t \in X} \mathcal{D}^t$. $\square$

5.2 Left Query Algorithms

It is obvious that CQ classes admit a left query algorithm over B and over N (cf. Example 5.6). In this section, we will give characterizations for certain classes to have a left query algorithm over B or over N. The proposition below is straightforward yet useful in simplifying proofs.

Proposition 5.8. *Let K be the semiring* B *or* N. *For every class $\mathcal{D} \subseteq \mathcal{U}$, we have $\mathcal{D}$ admits a left query algorithm over K if and only if $\mathcal{D}$ admits a left query algorithm over K of the form $(\mathcal{F}, X)$ where every $\mathbf{A} \in \mathcal{F}$ is connected.*

Proof. The "if" direction is trivial. For the "only if" direction, assume that $(\mathcal{F}', X')$ is a left query algorithm for $\mathcal{D}$ over K. Let $\mathbf{A}_1, \ldots, \mathbf{A}_n$ be the connected components of the members in $\mathcal{F}'$. Take $\mathcal{F} := \{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$ and $X := \{\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{D}\}$. Note that for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, if $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{A}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{B})$, then $\mathrm{lpf}_K^{\mathcal{F}'}(\mathbf{A}) = \mathrm{lpf}_K^{\mathcal{F}'}(\mathbf{B})$ by Proposition 3.14(c), hence

$$\begin{aligned} & \mathbf{A} \in \mathcal{D} \\ \text{iff} \quad & \mathrm{lpf}_K^{\mathcal{F}'}(\mathbf{A}) \in X' \\ \text{iff} \quad & \mathrm{lpf}_K^{\mathcal{F}'}(\mathbf{B}) \in X' \\ \text{iff} \quad & \mathbf{B} \in \mathcal{D}. \end{aligned}$$

By Remark 5.1(a), it follows that $(\mathcal{F}, X)$ is a left query algorithm for $\mathcal{D}$ over K. $\square$

Definition 5.2. Let σ be a vocabulary.

(a) A *primitive positive σ-formula* is a first-order σ-formula of the form $\exists x_1 \cdots \exists x_m \bigwedge_{i=1}^{n} \varphi_i$ in which $m, n \in \mathbb{Z}^+$ and φ_i is an (atomic) relational formula for $i \in [n]$.

(b) Let $\varphi = \exists x_1 \cdots \exists x_m \bigwedge_{i=1}^{n} \varphi_i$ be a primitive positive σ-sentence. The *canonical structure of* φ is the σ-structure $\mathbf{A}^\varphi$ with domain $\mathrm{dom}(\mathbf{A}^\varphi) := \{x_1, \ldots, x_m\}$ and relations

$$R^{\mathbf{A}^\varphi} := \{(x_{j_1}, \ldots, x_{j_r}) \mid \varphi_i = R(x_{j_1}, \ldots, x_{j_r}) \text{ for some } i \in [n]\}$$

for all r-ary relation symbols $R \in \sigma$.

(c) Let $\mathbf{A}$ be a σ-structure with domain $\mathrm{dom}(\mathbf{A}) = \{a_1, \ldots, a_m\}$ such that $R^{\mathbf{A}} \neq \emptyset$ for some relation symbol $R \in \sigma$. The *canonical sentence of* $\mathbf{A}$ is the primitive positive σ-sentence

$$\varphi^{\mathbf{A}} := \exists x_1 \cdots \exists x_m \bigwedge \{R(x_{j_1}, \ldots, x_{j_r}) \mid R \in \sigma \text{ is } r\text{-ary and } (a_{j_1}, \ldots, a_{j_r}) \in R^{\mathbf{A}}\}$$

Notation. The fragment of $\mathrm{FO}[\sigma]$ consisting of all primitive positive σ-formulas is denoted $\mathrm{PP}[\sigma]$.

The next proposition is immediate from definition.

Proposition 5.9. *Let σ be a vocabulary.*

(a) *For all σ-structures $\mathbf{B}$ and all $\mathrm{PP}[\sigma]$-sentences φ, we have that $\mathbf{B} \models \varphi$ if and only if $\mathbf{A}^\varphi \to \mathbf{B}$ (i.e., $\mathrm{Mod}(\varphi) = \mathrm{CQ}(\mathbf{A}^\varphi)$).*

(b) *For all σ-structures $\mathbf{A}$ and $\mathbf{B}$ such that $R^{\mathbf{A}} \neq \emptyset$ for some $R \in \sigma$, we have that $\mathbf{A} \to \mathbf{B}$ if and only if $\mathbf{B} \models \varphi^{\mathbf{A}}$ (i.e., $\mathrm{CQ}(\mathbf{A}) = \mathrm{Mod}(\varphi^{\mathbf{A}})$).* $\square$

Remark 5.2. In the terminology of database theory, a primitive positive sentence is known as a (*Boolean*) *conjunctive query*, a canonical structure as a *canonical database* (*instance*) and a canonical sentence as a *canonical query*.

It turns out that for every σ-structure $\mathbf{A}$, the class $\mathrm{CQ}(\mathbf{A})$ is definable not only in first-order logic but indeed also by a Boolean combination of PP-sentences: If $R^{\mathbf{A}} \neq \emptyset$ for some $R \in \sigma$, then this holds by Proposition 5.9(b); otherwise, the class $\mathrm{CQ}(\mathbf{A}) = \mathcal{A}$ is equal to $\mathrm{Mod}(\varphi^{\mathbf{B}} \vee \neg\varphi^{\mathbf{B}})$ for any σ-structure $\mathbf{B}$ such that $R^{\mathbf{B}} \neq \emptyset$ for some $R \in \sigma$.

Remark 5.3. Via the one-to-one correspondence $\mathbf{G} \mapsto \mathbf{G}^{\sigma(\mathrm{G})}$ from graphs to digraphs (i.e., $\sigma(\mathrm{G})$-structures for which E is irreflexive and symmetric), Definition 5.2 can be adapted to graphs. There are two modifications to be made, however. In part (a), we disallow relational formulas φ_i to take the form $E(x, x)$. In part (b), we let $E^{\mathbf{A}^{\varphi}} := \{(x_1, x_2) \mid \varphi_i = E(x_1, x_2) \text{ or } \varphi_i = E(x_2, x_1) \text{ for some } i \in [n]\}$ in the canonical structure $\mathbf{A}^{\varphi}$. Part (c) remains unchanged. It follows that Proposition 5.9 extends to graphs, with $\sigma = \sigma(\mathrm{G})$, σ-structures (containing a tuple or not) changed to graphs (containing an edge or not, respectively) and canonical structures changed to their corresponding graphs, throughout the statements.

Proposition 5.10. *For every class $\mathcal{D} \subseteq \mathcal{U}$, we have that $\mathcal{D}$ is a Boolean combination of* CQ *classes if and only if $\mathcal{D}$ is definable by a Boolean combination of* PP*-sentences.*

Proof. In view of Remark 5.3, we only prove for $\mathcal{U} = \mathcal{A}$. Let σ be the underlying vocabulary. By Remark 3.9 and Proposition 5.6, it suffices to argue that

(1) for every structure $\mathbf{A}$, the class $\mathrm{CQ}(\mathbf{A})$ is definable by a Boolean combination φ of PP-sentences, and

(2) for every PP-sentence φ, there is a structure $\mathbf{A}$ such that $\mathrm{Mod}(\varphi) = \mathrm{CQ}(\mathbf{A})$.

Indeed, (2) immediately follows from Proposition 5.9(a). For (1), we distinguish two cases.

Case 1. $R^{\mathbf{A}} = \emptyset$ *for every* $R \in \sigma$*.* Then $\mathrm{CQ}(\mathbf{A}) = \mathcal{A}$ and by Proposition 5.9(b), we can take $\varphi := (\varphi^{\mathbf{B}} \vee \neg\varphi^{\mathbf{B}})$.

Case 2. $R^{\mathbf{A}} \neq \emptyset$ *for some* $R \in \sigma$*.* It immediately follows from Proposition 5.9(b) by taking $\varphi := \varphi^{\mathbf{A}}$. □

Proposition 5.11. *For every class* $\mathcal{D} \subseteq \mathcal{A}$*, if* $\mathcal{D}$ *admits a left query algorithm over* B*, then* $\mathcal{D}$ *is* FO*-definable and closed under homomorphic equivalence.*

Proof. It immediately follows from Propositions 5.3, 5.7(a) and 5.10. □

5.2.1 CSP **Classes and Their Unions**

Straight from definition we have for every graph $\mathbf{G}$ that $\mathrm{CSP}(\mathbf{G})$ admits the right 1-query algorithm $(\{\mathbf{G}\}, \{1\})$ over B and the right 1-query algorithm $(\{\mathbf{G}\}, \mathbb{Z}^+)$ over N. However, it is not yet clear when $\mathrm{CSP}(\mathbf{G})$ admits a left query algorithm over B or over N.

In this subsection, we will first present a characterization of CSP classes of *graphs* that admit a left query algorithm over B or over N, then we move on to show that the characterization also holds for CSP classes of *structures*, finally we conclude with further characterizations of *unions* of CSP classes of structures that admit a left query algorithm over these semirings.

In Example 5.7 we asserted that $\mathrm{CSP}(\mathbf{K}_3)$ admits no left query algorithm over B or over N. This assertion is a consequence of the next lemma.

Lemma 5.12. *Let* $\mathbf{G}$ *be a graph. The following are equivalent:*

(i) $\mathbf{G}$ *is* 1*-colorable*

(ii) $\mathrm{CSP}(\mathbf{G})$ *admits the left* 1*-query algorithm* $(\{\mathbf{K}_2\}, \{0\})$ *over* B

(iii) $\mathrm{CSP}(\mathbf{G})$ *admits the left* 1*-query algorithm* $(\{\mathbf{K}_2\}, \{0\})$ *over* N

(iv) $\mathrm{CSP}(\mathbf{G})$ *admits a left query algorithm over* B

(v) $\mathrm{CSP}(\mathbf{G})$ *admits a left query algorithm over* N*.*

Proof. For the direction from (i) to (ii), note that if $\mathbf{G}$ is 1-colorable, i.e., if $\mathbf{G}$ is an independent set, then $\mathrm{CSP}(\mathbf{G}) = \mathcal{I}$, the class of independent sets, which was shown to admit the left 1-query algorithm $(\{\mathbf{K}_2\}, \{0\})$ over B in Example 5.2. The implications from (ii) to (iv) and from (iii) to (v) are obvious, and the implications from (ii) to (iii) and (iv) to (v) are by Proposition 5.1. So we only prove not (i) implies not (v). If $\mathbf{G}$ is not 1-colorable, then it has chromatic number $\chi(\mathbf{G}) \geq 2$. We distinguish two cases as follows.

Case 1. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) = 2$*.* Then $\mathrm{CSP}(\mathbf{G})$ consists of exactly the 2-colorable graphs, that is, $\mathrm{CSP}(\mathbf{G}) = \mathcal{D}^{\mathrm{bip}}$, the class introduced in Example 5.4 and shown to not admit any left query algorithm over N there.

Case 2. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) \geq 3$*.* Let $\mathcal{F}$ be an arbitrary finite nonempty class of graphs, and let $n \geq 2$ be an integer greater than the maximum treewidth of the graphs in $\mathcal{F}$. Since $\mathbf{G}$ is not 2-colorable, the class $\mathrm{CSP}(\mathbf{G})$ is not definable in the logic $\mathrm{C}^{\omega}_{\infty\omega}$ by Theorem A.1. It follows that there are two graphs $\mathbf{H}_1 \in \mathrm{CSP}(\mathbf{G})$ and $\mathbf{H}_2 \notin \mathrm{CSP}(\mathbf{G})$ such that $\mathbf{H}_1 \equiv_{\mathrm{C}^n} \mathbf{H}_2$ (by Remark 3.8(b)) and hence $\mathrm{lpf}^{\mathcal{T}^{n-1}}(\mathbf{H}_1) = \mathrm{lpf}^{\mathcal{T}^{n-1}}(\mathbf{H}_2)$ (by Theorem 4.13), which implies that $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{H}_1) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{H}_2)$ since $\mathcal{F} \subseteq \mathcal{T}^{n-1}$. By Remark 5.1(a), $\mathrm{CSP}(\mathbf{G})$ does not admit any left query algorithm over N. □

As an immediate consequence of Lemma 5.12, we have that *deciding whether* CSP($\mathbf{G}$) *admits a left query algorithm over* B *is in* P because deciding whether $\mathbf{G}$ is 1-colorable, which amounts to deciding whether $\mathbf{G} \in \mathrm{CQ}(\mathbf{K}_2)$, can obviously be carried out in polynomial time.

An analysis of the condition that $\mathbf{G}$ is 1-colorable in Lemma 5.12 leads to our main characterization of CSP classes of graphs that admit a left query algorithm over B or N.

Theorem 5.13. *Let* $\mathbf{G}$ *be a graph. The following are equivalent:*

(i) CSP($\mathbf{G}$) *is* FO-*definable.*

(ii) CSP($\mathbf{G}$) *admits a left query algorithm over* B*.*

(iii) CSP($\mathbf{G}$) *admits a left query algorithm over* N*.*

Proof. By Lemma 5.12, it suffices to argue for the FO-definability of CSP($\mathbf{G}$) in the three cases below.

Case 1. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) = 1$. Then, $\mathrm{CSP}(\mathbf{G}) = \mathcal{I} = \mathrm{Mod}(\varphi)$, where $\varphi = \forall x \forall y \, \neg E(x, y)$.

Case 2. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) = 2$. Then, $\mathrm{CSP}(\mathbf{G}) = \mathcal{D}^{\mathrm{bip}}$. The construction of the two graphs witnessing that CSP($\mathbf{G}$) admits no left query algorithm over N in Example 5.4 also witnesses that CSP($\mathbf{G}$) is not FO-definable (see Example 3.10).

Case 3. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) \geq 3$. Then, CSP($\mathbf{G}$) is not $\mathrm{C}^{\omega}_{\infty\omega}$-definable by Theorem A.1 and hence not FO-definable, since FO is a fragment of $\mathrm{C}^{\omega}_{\infty\omega}$. □

The next dichotomy directly follows from the proof of Theorem 5.13.

Corollary 5.14. *Let* $\mathbf{G}$ *be a graph. If* $\mathbf{G}$ *is an independent set, then* $\mathrm{CSP}(\mathbf{G}) = \mathrm{Mod}(\varphi)$ *for the* FO*-sentence* $\varphi = \forall x \forall y \neg E(x, y)$*. Otherwise,* $\mathrm{CSP}(\mathbf{G})$ *is not* FO*-definable.* □

Next, we turn our attention to CSP classes of structures. It turns out that the characterization in Theorem 5.13 holds for CSP classes of structures as well.

Theorem 5.15. *Let* $\mathbf{A}$ *be a structure. The following are equivalent:*

(i) $\mathrm{CSP}(\mathbf{A})$ *is* FO*-definable.*

(ii) $\mathrm{CSP}(\mathbf{A})$ *admits a left query algorithm over* B.

(iii) $\mathrm{CSP}(\mathbf{A})$ *admits a left query algorithm over* N.

Before proving this theorem, we mention the following definition that will be needed.

Definition 5.3. Let $\mathcal{F}$ and $\mathcal{E}$ be finite classes of structures. The pair $(\mathcal{F}, \mathcal{E})$ is a *finite homomorphism duality* if for all structures $\mathbf{A}$, it holds that $\hom_{\mathrm{B}}(\mathbf{F}, \mathbf{A}) = 0$ for all $\mathbf{F} \in \mathcal{F}$ if and only if $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{D}) \neq 0$ for some $\mathbf{D} \in \mathcal{E}$.

Proof of Theorem 5.15. First, note that the direction from (ii) to (iii) immediately follows from Proposition 5.1, so we will prove the directions from (i) to (ii) and from (iii) to (i).

Next, for the direction from (i) to (ii), we note a result from [1] that *for all structures* $\mathbf{A}$*, it holds that* $\mathrm{CSP}(\mathbf{A})$ *is* FO*-definable if and only if* $(\mathcal{F}, \{\mathbf{A}\})$ *is a finite homomorphism duality for some finite class* $\mathcal{F}$ *of structures.* Now, assume that $\mathrm{CSP}(\mathbf{A})$ is FO-definable. If $\mathrm{CSP}(\mathbf{A}) = \mathcal{A}$ (cf. Example 3.18(b) for an example of such $\mathbf{A}$), which is equal to $\mathrm{Mod}(\exists x\, x = x)$, then obviously it admits a left query algorithm over B, say $(\{\mathbf{A}\}, \mathbb{B})$ (cf. Example 5.1). Otherwise, $\mathrm{CSP}(\mathbf{A}) \neq \mathcal{A}$, then there is a finite $\mathcal{F}$ such that

$(\mathcal{F}, \{A\})$ is a finite homomorphism duality by the aforementioned characterization given in [1]. Clearly, $\mathcal{F}$ is nonempty, and by taking the set $X := \{(\underbrace{0, \ldots, 0}_{n\text{-times}})\}$ where $n := |\mathcal{F}|$, we have that $(\mathcal{F}, X)$ is a left query algorithm over B for $\mathrm{CSP}(\mathbf{A})$.

Finally, for the direction from (iii) to (i), we will prove contrapositively that not (i) implies not (iii). For this purpose, let $\mathbf{A}$ be a structure for which $\mathrm{CSP}(\mathbf{A})$ is not FO-definable, and let σ be the underlying vocabulary. Our goal is to shown that *for every finite nonempty $\mathcal{F}$ of structures, there are two structures $\mathbf{P} \in \mathrm{CSP}(\mathbf{A})$ and $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A})$ such that* $\mathrm{lpf}^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{Q})$, for which we assume that all $\mathbf{F} \in \mathcal{F}$ are connected due to Proposition 5.8.

Let $\mathcal{F}$ be such a class, and let $n \geq 2$ be greater than the maximum size of the structures in $\mathcal{F}$. Moreover, let $\mathcal{D}^{\mathrm{conn}}_{\delta<n}$ denote the class of all connected structures $\mathbf{C}$ of diameter $\delta(\mathbf{C}) < n$. Note that $\mathcal{F} \subseteq \mathcal{D}^{\mathrm{conn}}_{\delta<n}$.

We introduce relevant definitions and constructions from [38] as follows.

(1) For every equivalence relation $\equiv$ over $\mathrm{dom}(\mathbf{A})$, the *quotient of $\mathbf{A}$ by $\equiv$*, denoted $\mathbf{A}/{\equiv}$ is the quotient of $\mathbf{A}$ by the partition $\{[a]_{\equiv} \mid a \in \mathrm{dom}(\mathbf{A})\}$ (cf. Definition 2.14 for the quotient of $\mathbf{A}$ by a partition θ of $\mathrm{dom}(\mathbf{A})$).

Let $n \in \mathbb{Z}^+$.

(2) The structure $\mathbf{L}_n$ has domain $\mathrm{dom}(\mathbf{L}_n) := \{0, \ldots, n\}$ and relation $R^{\mathbf{L}_n} := \bigcup_{i=0}^{n-1} \{i, i+1\}^r$ for every r-ary relation symbol $R \in \sigma$.

(3) Let $\equiv_n$ be the equivalence relation over the domain of the product structure $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$ such that for all $(a_1, a_2, b), (a_1', a_2', b') \in \mathrm{dom}(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)$, we have $(a_1, a_2, b) \equiv_n (a_1', a_2', b')$ if

- $a_1 = a_1'$ and $b = b' = 0$, or
- $a_2 = a_2'$ and $b = b' = n$, or
- $a_1 = a_1'$, $a_2 = a_2'$ and $b = b'$.

Let $\mathbf{B}_0(\mathbf{A}, n)$ and $\mathbf{B}_1(\mathbf{A}, n)$ be the substructure of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$ induced by the sets

$$\{[(a_1, a_2, b)]_{\equiv_n} \mid a_1, a_2 \in \mathrm{dom}(\mathbf{A}) \text{ and } b \in [0, n-1]\}$$

and

$$\{[(a_1, a_2, b)]_{\equiv_n} \mid a_1, a_2 \in \mathrm{dom}(\mathbf{A}) \text{ and } b \in [n]\},$$

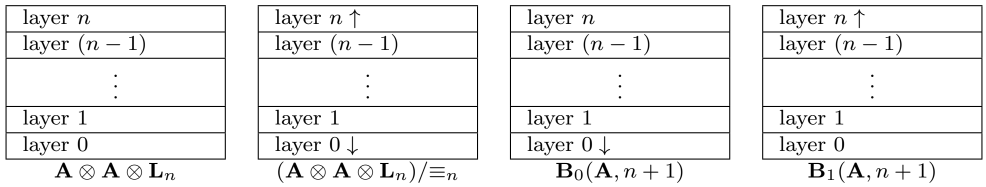
respectively. That is, $\mathbf{B}_0(\mathbf{A}, n)$ is induced by removing the elements whose third coordinate is n and $\mathbf{B}_1(\mathbf{A}, n)$ by removing the elements whose third coordinate is 0, in $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$.

For convenience, let us call the substructure of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$ induced by the elements whose third coordinate is in $[i, j]$ the *layer(s) i-j of* $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$, and call *layer i* for layer(s) i-i; likewise for $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$. Thus, $\mathbf{B}_0(\mathbf{A}, n)$ is the layer(s) 0-$(n-1)$ of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$ and $\mathbf{B}_1(\mathbf{A}, n)$ is the layer(s) 1-n of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$.

Remark 5.4. Note that the elements in a tuple of a relation only span across two consecutive layers of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$, and likewise for $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$.

A visualization for the four structures $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$, $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$, $\mathbf{B}_0(\mathbf{A}, n+1)$ and $\mathbf{B}_1(\mathbf{A}, n+1)$ is given in Figure 5.1. (Layer 0 and layer n, i.e., the bottom layer and the top layer, from $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$, are marked with a down arrow $\downarrow$ and an up arrow $\uparrow$, respectively.)

Figure 5.1: Visualization of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$, $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$, $\mathbf{B}_0(\mathbf{A}, n+1)$ and $\mathbf{B}_1(\mathbf{A}, n+1)$ in the proof of Theorem 5.15.



Remark 5.5. For all $n \in \mathbb{Z}^+$, $i \in [0, n-1]$, $j \in [0, n-i-1]$, $k \in [0, n-i-j]$ and $l \in [0, n-i-j+1]$, the following are straight from definition.

(a) Layer(s) 0-i of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$ is isomorphic to layer(s) 0-i of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_{n+1})/{\equiv_{n+1}}$, and layer(s) $(n-i)$-n of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}$ is isomorphic to layer(s) $(n-i+1)$-$(n+1)$ of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_{n+1})/{\equiv_{n+1}}$.

(b) Layer(s) $(i+1)$-$(i+j+1)$ of $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_{n+1})/{\equiv_{n+1}}$, layer(s) $(i+k)$-$(i+j+k)$ of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n$ and layer(s) $(i+l)$-$(i+j+l)$ of $\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_{n+1}$ are mutually isomorphic.

Now, let $\mathbf{P} := (\mathbf{B}_0(\mathbf{A}, n+1) \oplus \mathbf{B}_1(\mathbf{A}, n+1))$ and $\mathbf{Q} := (((\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}) \oplus (\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n))$. Then, we refer to the two results from [38].

(a) *For every* $n \in \mathbb{Z}^+$, *we have* $\mathbf{B}_0(\mathbf{A}, n) \to \mathbf{A}$ *and* $\mathbf{B}_1(\mathbf{A}, n) \to \mathbf{A}$.

(b) CSP($\mathbf{A}$) *is* FO*-definable if and only if* $((\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n}) \to \mathbf{A}$ *for some* $n \in \mathbb{Z}^+$.

By Proposition 3.14(c), it follows that $\mathbf{P} \in \mathrm{CSP}(\mathbf{A})$ while $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A})$.

Next, we will show that $\hom(\mathbf{C}, \mathbf{P}) = \hom(\mathbf{C}, \mathbf{Q})$ *for all structures* $\mathbf{C} \in \mathcal{D}^{\mathrm{conn}}_{\delta<n}$, i.e., $\mathrm{lpf}^{\mathcal{D}^{\mathrm{conn}}_{\delta<n}}(\mathbf{P}) = \mathrm{lpf}^{\mathcal{D}^{\mathrm{conn}}_{\delta<n}}(\mathbf{Q})$. Then, it will immediately follow that $\mathrm{lpf}^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{Q})$ since $\mathcal{F} \subseteq \mathcal{D}^{\mathrm{conn}}_{\delta<n}$.

Without loss of generality, assume that in terms of the third coordinate, the

Figure 5.2: Visualization of $\mathbf{P}$ and $\mathbf{Q}$ in the proof of Theorem 5.15.

$\mathbf{P}$	$\mathbf{Q}$
layer $(2n+1)\uparrow$	layer $(2n+1)$
layer $2n$	layer $2n$
$\vdots$	$\vdots$
layer $(n+2)$	layer $(n+2)$
layer $(n+1)$	layer $(n+1)$
layer n	layer $n\uparrow$
layer $(n-1)$	layer $(n-1)$
$\vdots$	$\vdots$
layer 1	layer 1
layer $0\downarrow$	layer $0\downarrow$

$\mathbf{B}_0(\mathbf{A}, n+1)$-portion precedes the $\mathbf{B}_1(\mathbf{A}, n+1)$-portion in $\mathbf{P}$, and the $((\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n})$-portion precedes the $(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)$-portion in $\mathbf{Q}$. More precisely, let

$$
\begin{aligned}
\mathrm{dom}(\mathbf{P}) &:= \{(a_1, a_2, b) \mid (a_1, a_2, b) \in \mathrm{dom}(\mathbf{B}_0(\mathbf{A}, n+1))\} \uplus \\
&\quad\ \{(a_1, a_2, b+n+1) \mid (a_1, a_2, b) \in \mathrm{dom}(\mathbf{B}_1(\mathbf{A}, n+1))\}, \\
\mathrm{dom}(\mathbf{Q}) &:= \{(a_1, a_2, b) \mid (a_1, a_2, b) \in \mathrm{dom}((\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)/{\equiv_n})\} \uplus \\
&\quad\ \{(a_1, a_2, b+n+1) \mid (a_1, a_2, b+n+1) \in \mathrm{dom}(\mathbf{A} \otimes \mathbf{A} \otimes \mathbf{L}_n)\}.
\end{aligned}
$$

We adopt the same naming scheme concerning layer(s) i-j as earlier. A visualization for the two structures $\mathbf{P}$ and $\mathbf{Q}$ are given in Figure 5.2.

Let

- $\mathbf{P}_1$ and $\mathbf{Q}_1$ be the direction sum of layers 0-$(n-1)$ and layers $(n+1)$-$2n$ of $\mathbf{P}$ and $\mathbf{Q}$, respectively,
- $\mathbf{P}_2$ and $\mathbf{Q}_2$ be the layers 1-n of $\mathbf{P}$ and $\mathbf{Q}$, respectively, and
- $\mathbf{P}_3$ and $\mathbf{Q}_3$ be the layers $(n+2)$-$(2n+1)$, respectively.

By Remark 5.5, there are three isomorphisms $\pi_1 : \mathbf{P}_1 \cong \mathbf{Q}_1$, $\pi_2 : \mathbf{P}_2 \cong \mathbf{Q}_3$ and $\pi_3 : \mathbf{P}_3 \cong \mathbf{Q}_2$.

Let $\mathbf{C}' \in \mathcal{D}_{\delta<n}^{\mathrm{conn}}$ be an arbitrary structure, it is easy to establish a bijection f from the set $\mathrm{Inj}(\mathbf{C}', \mathbf{P})$ to the set $\mathrm{Inj}(\mathbf{C}', \mathbf{Q})$ of *injective* homomorphisms: For every injective homomorphism $i : \mathbf{C}' \to \mathbf{P}$, the image $i(\mathbf{C}')$ is contained as a substructure in either $\mathbf{P}_1$, $\mathbf{P}_2$ or $\mathbf{P}_3$ because $\mathbf{C}'$ is connected and has diameter $\delta(\mathbf{C}') < n$ (and by Remark 5.4), and likewise for every injective homomorphism $i' : \mathbf{C}' \to \mathbf{Q}$. Thus, let

$$f(h) := \begin{cases} \pi_1 \circ i & \text{if } i(\mathbf{C}') \subseteq \mathbf{P}_1, \\ \pi_2 \circ i & \text{if } i(\mathbf{C}') \subseteq \mathbf{P}_2, \\ \pi_3 \circ i & \text{if } i(\mathbf{C}') \subseteq \mathbf{P}_3, \end{cases}$$

i.e., $i(\mathbf{C}') \subseteq \mathbf{P}_1$ if and only if $(\pi_1 \circ i)(\mathbf{C}') \subseteq \mathbf{Q}_1$, $i(\mathbf{C}') \subseteq \mathbf{P}_2$ if and only if $(\pi_1 \circ i)(\mathbf{C}') \subseteq \mathbf{Q}_3$, and $i(\mathbf{C}') \subseteq \mathbf{P}_3$ if and only if $(\pi_1 \circ i)(\mathbf{C}') \subseteq \mathbf{Q}_2$.

Hence,

$$\mathrm{inj}(\mathbf{C}', \mathbf{P}) = \mathrm{inj}(\mathbf{C}', \mathbf{Q}) \text{ for all } \mathbf{C}' \in \mathcal{D}_{\delta<n}^{\mathrm{conn}}. \tag{$*$}$$

Let $\mathbf{C} \in \mathcal{D}_{\delta<n}^{\mathrm{conn}}$ be an arbitrary structure. Then, for homomorphisms (not necessarily injective) $h : \mathbf{C} \to \mathbf{P}$ and $h' : \mathbf{C} \to \mathbf{Q}$, the images $h(\mathbf{C})$ and $h'(\mathbf{C})$ are both in $\mathcal{D}_{\delta<n}^{\mathrm{conn}}$,

by Proposition 3.11. We conclude

$$
\begin{aligned}
& \hom(\mathbf{C},\mathbf{P}) && \\
= & \sum_{\mathbf{C}'\in(\mathcal{A}/\cong)} \mathrm{sur}(\mathbf{C},\mathbf{C}') \times \mathrm{inj}(\mathbf{C}',\mathbf{P})/\mathrm{aut}(\mathbf{C}') && \text{(by Proposition 3.8)} \\
= & \sum_{\mathbf{C}'\in((\mathcal{A}/\cong)\cap\mathcal{D}^{\mathrm{conn}}_{\delta<n})} \mathrm{sur}(\mathbf{C},\mathbf{C}') \times \mathrm{inj}(\mathbf{C}',\mathbf{P})/\mathrm{aut}(\mathbf{C}') && \text{(by the above discussion)} \\
= & \sum_{\mathbf{C}'\in((\mathcal{A}/\cong)\cap\mathcal{D}^{\mathrm{conn}}_{\delta<n})} \mathrm{sur}(\mathbf{C},\mathbf{C}') \times \mathrm{inj}(\mathbf{C}',\mathbf{Q})/\mathrm{aut}(\mathbf{C}') && \text{(by } (*)) \\
= & \sum_{\mathbf{C}'\in(\mathcal{A}/\cong)} \mathrm{sur}(\mathbf{C},\mathbf{C}') \times \mathrm{inj}(\mathbf{C}',\mathbf{Q})/\mathrm{aut}(\mathbf{C}') && \text{(by the above discussion)} \\
= & \hom(\mathbf{C},\mathbf{Q}) && \text{(by Proposition 3.8),}
\end{aligned}
$$

as desired. □

It follows that *deciding whether* CSP($\mathbf{A}$) *admits a left query algorithm over* B *is* NP*-complete* because deciding whether CSP($\mathbf{A}$) is FO-definable is NP-complete [38]. (Compare this to CSP($\mathbf{G}$) for a graph $\mathbf{G}$, for which the corresponding decision problem is in P, as seen earlier.[1]) In fact, this statement as well as the characterization given in Theorem 5.15 generalizes to *unions* of CSP classes of structures, as can be seen next.

Theorem 5.16. *Let* $\mathbf{A}_1, \ldots, \mathbf{A}_n$ *be structures. The following are equivalent:*

(i) $\bigcup_{i=1}^n \mathrm{CSP}(\mathbf{A}_i)$ *is* FO*-definable.*

(ii) $\bigcup_{i=1}^n \mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* B*.*

(iii) $\bigcup_{i=1}^n \mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* N*.*

[1] Interestingly, it is known that CSP($\mathbf{C}_3$) is NP-complete because the membership problem is the same as `3-COLORABILITY`. In contrast, however, it is also known that deciding whether a digraph (i.e., a σ(G)-structure) admits a homomorphism to a directed 3-cycle is in polynomial time (see [33]).

Proof. It is shown in [5] that *for all structures* $\mathbf{A}_1, \ldots, \mathbf{A}_n$*, the class* $\bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *is* FO-*definable if and only if* $\mathrm{CSP}(\mathbf{A}_i)$ *is* FO-*definable for all* $i \in [n]$. This theorem then follows from Theorem 5.15 and Theorem 5.17 presented next. □

Let $\mathbf{A}_1, \ldots, \mathbf{A}_n$ be structures. We say that $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are *pairwise incomparable* (in terms of the preorder $\rightarrow$, see Remark 3.10(b)) if for all distinct $i, j \in [n]$, it holds that $\mathbf{A}_i \not\rightarrow \mathbf{A}_j$. Note that if $\mathbf{A}_i \rightarrow \mathbf{A}_j$, then $\mathrm{CSP}(\mathbf{A}_i) \subseteq \mathrm{CSP}(\mathbf{A}_j)$ and hence removing $\mathrm{CSP}(\mathbf{A}_i)$ from the union $\bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ results in the same class. The next theorem implies that *deciding whether* $\bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* $\mathbb{B}$ *is* NP*-complete.*

Theorem 5.17. *Let* K *be the semiring* $\mathbb{B}$ *or* $\mathbb{N}$*. For all pairwise incomparable structures* $\mathbf{A}_1, \ldots, \mathbf{A}_n$*, the union* $\bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* K *if and only if* $\mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* K *for all* $i \in [n]$*.*

Proof. Fix a semiring K. The statement becomes trivial for $n = 1$, so we assume $n \geq 2$ in what follows. The "if" direction follows from Proposition 5.6(b). For the "only if" direction, we will prove its contraposition as follows. Assume that there is some $i \in [n]$ such that $\mathrm{CSP}(\mathbf{A}_i)$ does not admit any left query algorithm over K and, without loss of generality, that $i = 1$.

Let $\mathcal{F}$ be an arbitrary finite nonempty class of *connected* structures. Then, it follows from the preceding assumption and Remark 5.1(a) that there are two structures $\mathbf{P}'$ and $\mathbf{Q}'$ such that

(1) $\mathbf{P}' \in \mathrm{CSP}(\mathbf{A}_1)$,

(2) $\mathbf{Q}' \notin \mathrm{CSP}(\mathbf{A}_1)$,

(3) $\mathrm{lpf}^{\mathcal{F}}(\mathbf{P}') = \mathrm{lpf}^{\mathcal{F}}(\mathbf{Q}')$.

In view of Proposition 5.8, it suffices to present *two structures* $\mathbf{P} \in \bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *and* $\mathbf{Q} \notin \bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *such that* $\mathrm{lpf}^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{Q})$.

For this purpose, observe

(4) $\mathbf{A}_1 \in (\mathrm{CSP}(\mathbf{A}_1) \setminus \bigcup_{i=2}^{n} \mathrm{CSP}(\mathbf{A}_i))$,

by the premise that $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are pairwise incomparable.

Take $\mathbf{P} := \mathbf{P}' \oplus \mathbf{A}_1$ and $\mathbf{Q} := \mathbf{Q}' \oplus \mathbf{A}_1$. By Proposition 3.14(b), it follows that for every $\mathbf{F} \in \mathcal{F}$, $\hom(\mathbf{F}, \mathbf{P}) = \hom(\mathbf{F}, \mathbf{P}') + \hom(\mathbf{F}, \mathbf{A}_1) = \hom(\mathbf{F}, \mathbf{Q}') + \hom(\mathbf{F}, \mathbf{A}_1) = \hom(\mathbf{F}, \mathbf{Q})$, where the first and the third equalities follow from the fact that $\mathbf{F}$ is connected. Thus, $\mathrm{lpf}^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}^{\mathcal{F}}(\mathbf{Q})$.

Moreover, by (1) and (4) we have that $\mathbf{P} \in \mathrm{CSP}(\mathbf{A}_1) \subseteq \bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$. In addition, by (2) we have that $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A}_1)$ and, by (4), that $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A}_i)$ for any $i \in [2, n]$ by (4). Therefore, $\mathbf{Q} \notin \bigcup_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$. □

By Proposition 5.6(a), the complexity result and the characterization carries over to the *complements* of CSP classes, $\overline{\mathrm{CSP}(\mathbf{A})}$, i.e., we have that *deciding whether* $\overline{\mathrm{CSP}(\mathbf{A})}$ *admits a left query algorithm over* B *is* NP*-complete* and, by Remark 3.9, that the characterization in Theorem 5.15 holds for $\overline{\mathrm{CSP}(\mathbf{A})}$.

By Proposition 3.14(e), the complexity result and the characterization hold for *intersections* of CSP classes, $\bigcap_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$, i.e., we have that *deciding whether* $\bigcap_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$ *admits a left query algorithm over* B *is* NP*-complete* and that the characterization in Theorem 5.15 extends to $\bigcap_{i=1}^{n} \mathrm{CSP}(\mathbf{A}_i)$.

5.2.2 Homomorphic-Equivalence Classes and Their Unions

In this subsection, we turn our attention from CSP classes to homomorphic-equivalence classes of structures, in view of the characterization in Theorem 5.15. Recall that for all structures $\mathbf{A}$,

$$[\mathbf{A}]_{\leftrightarrow} := \{\mathbf{B} \in \mathcal{A} \mid \mathbf{A} \leftrightarrow \mathbf{B}\}.$$

It is immediate that $[\mathbf{A}]_{\leftrightarrow} = \mathrm{CQ}(\mathbf{A}) \cap \mathrm{CSP}(\mathbf{A})$ (see Subsection 3.2.2).

Lemma 5.18. *Let K be the semiring* B *or* N*. For every structure $\mathbf{A}$, we have $[\mathbf{A}]_{\leftrightarrow}$ admits a left query algorithm over K if and only if* $\mathrm{CSP}(\mathbf{A})$ *admits a left query algorithm over K.*

Proof. Fix a semiring K. For the "if" part, observe that

$$[\mathbf{A}]_{\leftrightarrow} = \mathrm{CQ}(\mathbf{A}) \cap \mathrm{CSP}(\mathbf{A}).$$

It is obvious that $\mathrm{CQ}(\mathbf{A})$ admits a left query algorithm over K. The claim follows by Proposition 5.6(b).

For the "only if" part. Assume that $\mathrm{CSP}(\mathbf{A})$ does not admit any left query algorithm over K. Let $\mathcal{F}$ be an arbitrary finite nonempty set of structures, which without loss of generality we may assume to be all connected, by Proposition 5.8. Then, there are two structures $\mathbf{P} \in \mathrm{CSP}(\mathbf{A})$ and $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A})$ such that $\mathrm{lpf}^{\mathcal{F}}_K(\mathbf{P}) = \mathrm{lpf}^{\mathcal{F}}_K(\mathbf{Q})$.

Take $\mathbf{P}' := \mathbf{P} \oplus \mathbf{A}$ and $\mathbf{Q}' := \mathbf{Q} \oplus \mathbf{A}$. By Proposition 3.14(c), we have $\mathbf{P}' \to \mathbf{A}$ since $\mathbf{P} \to \mathbf{A}$, and we have $\mathbf{Q}' \not\to \mathbf{A}$ since $\mathbf{Q} \not\to \mathbf{A}$. Hence, $\mathbf{P}' \in [\mathbf{A}]_{\leftrightarrow}$ while $\mathbf{Q}' \notin [\mathbf{A}]_{\leftrightarrow}$. However, for every structure $\mathbf{F} \in \mathcal{F}$ (which is assumed to be connected),

$$\hom_K(\mathbf{F}, \mathbf{P}') = \hom_K(\mathbf{F}, \mathbf{P}) + \hom_K(\mathbf{F}, \mathbf{A}) = \hom_K(\mathbf{F}, \mathbf{Q}) + \hom_K(\mathbf{F}, \mathbf{A}) = \hom_K(\mathbf{F}, \mathbf{Q}'),$$

for which the first and the third equalities follow from Proposition 3.14(b), and the second one from the fact $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q})$. Thus, $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P}') = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q}')$, and as a result we have that $[\mathbf{A}]_{\leftrightarrow}$ does not admit any left query algorithm over K (by Proposition 5.8 in view of Remark 5.1). □

Theorem 5.19. *For every structure* $\mathbf{A}$*, we have* $[\mathbf{A}]_{\leftrightarrow}$ *admits a left query algorithm over* B *if and only if* $[\mathbf{A}]_{\leftrightarrow}$ *admits a left query algorithm over* N.

Proof. It immediate follows from Theorem 5.15 and Lemma 5.18. □

As a result, by Lemma 5.18 we have that *deciding whether* $[\mathbf{A}]_{\leftrightarrow}$ *admits a left query algorithm over* B *is* NP*-complete* because it is equivalent to deciding whether $\mathrm{CSP}(\mathbf{A})$ admits one, which was seen to be NP-complete in the previous subsection. This statement as well as the characterization in Theorem 5.19 extends to unions of homomorphic-equivalence classes.

Theorem 5.20. *For all structures* $\mathbf{A}_1, \ldots, \mathbf{A}_n$*, the union* $\bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *admits a left query algorithm over* B *if and only if* $\bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *admits a left query algorithm over* N.

Proof. It immediately follows from Theorem 5.19 and the next theorem. □

Let $\mathbf{A}_1, \ldots, \mathbf{A}_n$ be structures. As remarked previously, for each $i \in [n]$, deciding whether $[\mathbf{A}_i]_{\leftrightarrow}$ admits a left query algorithm over B is NP-complete. This, together with Theorem 5.21 below, implies that *deciding whether* $\bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *admits a left query algorithm over* B *is* NP*-complete.*

Theorem 5.21. *Let K be the semiring* $\mathbb{B}$ *or* $\mathbb{N}$*. For all structures $\mathbf{A}_1, \ldots, \mathbf{A}_n$, the union $\bigcup_{i=1}^{n} [\mathbf{A}_i]_\leftrightarrow$ admits a left query algorithm over K if and only if $[\mathbf{A}_i]_\leftrightarrow$ admits a left query algorithm over K for all $i \in [n]$.*

Proof. Fix a semiring K. The "if" part immediately follows from Proposition 5.6(b). For the "only if" part, we prove by induction on n. The base case $n = 1$ is trivial since the statement is the same on either side.

For the inductive case $n \geq 2$, note that, without loss of generality, we can assume that $[\mathbf{A}_1]_\leftrightarrow, \ldots, [\mathbf{A}_n]_\leftrightarrow$ are pairwise disjoint: If $[\mathbf{A}_i]_\leftrightarrow = [\mathbf{A}_j]_\leftrightarrow$ for some distinct $i, j \in [n]$, then we can remove, say, $[\mathbf{A}_j]_\leftrightarrow$ from the union and the resulting union class will remain unchanged.

We will prove the contraposition: *If $[\mathbf{A}_i]_\leftrightarrow$ does not admit any left query algorithm over K for some $i \in [n]$, then neither does $\bigcup_{i=1}^{n} [\mathbf{A}_i]_\leftrightarrow$.* Observe that $\rightarrow$ is a preorder among $\mathbf{A}_1, \ldots, \mathbf{A}_n$ (cf. Remark 3.10(b)) and hence there is a maximal which is, without loss of generality, assumed to be $\mathbf{A}_n$: For all $i \in [n-1]$, we have $\mathbf{A}_n \not\rightarrow \mathbf{A}_i$. Next, we distinguish two cases.

Case 1. $[\mathbf{A}_n]_\leftrightarrow$ admits a left query algorithm over K. Then, there is an $i \in [n-1]$ such that $[\mathbf{A}_i]_\leftrightarrow$ does not admit any left query algorithm over K. By the induction hypothesis, we have $\bigcup_{i=1}^{n-1} [\mathbf{A}_i]_\leftrightarrow$ does not admit any left query algorithm over K. By Proposition 5.6, it follows that $\bigcup_{i=1}^{n} [\mathbf{A}_i]_\leftrightarrow$ does not admit any left query algorithm over K either, because $\bigcup_{i=1}^{n-1} [\mathbf{A}_i]_\leftrightarrow = (\bigcup_{i=1}^{n} [\mathbf{A}_i]_\leftrightarrow) \cap \overline{[\mathbf{A}_n]_\leftrightarrow}$.

Case 2. $[\mathbf{A}_n]_\leftrightarrow$ does not admit any left query algorithm over K. By Lemma 5.18, we have that $\mathrm{CSP}(\mathbf{A}_n)$ does not admit any left query algorithm over K either. Now, let $\mathcal{F}$

be an arbitrary finite nonempty class of connected structures (in view of Proposition 5.8). Then, there are two structures $\mathbf{P} \in \mathrm{CSP}(\mathbf{A}_n)$ and $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A}_n)$ such that $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q})$, by Remark 5.1. Our goal then is to *present two structures* $\mathbf{P'} \in \bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *and* $\mathbf{Q'} \notin \bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *such that* $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P'}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q'})$.

Take $\mathbf{P'} := \mathbf{P} \oplus \mathbf{A}_n$ and $\mathbf{Q'} := \mathbf{Q} \oplus \mathbf{A}_n$. It follows that $\mathbf{P'} \in [\mathbf{A}_n]_{\leftrightarrow}$ since $\mathbf{P} \in \mathrm{CSP}(\mathbf{A}_n)$. Hence, $\mathbf{P'} \in \bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$. Moreover, we have $\mathbf{Q'} \notin [\mathbf{A}_n]_{\leftrightarrow}$ since $\mathbf{Q} \notin \mathrm{CSP}(\mathbf{A}_n)$, and for all $i \in [n-1]$, we have $\mathbf{Q'} \notin [\mathbf{A}_i]_{\leftrightarrow}$ since $\mathbf{A}_n \not\to \mathbf{A}_i$. Hence, $\mathbf{Q'} \notin \bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$. However, for every structure $\mathbf{F} \in \mathcal{F}$, which is connected, we have

$$\begin{aligned} & \hom_K(\mathbf{F}, \mathbf{P'}) \\ = \; & \hom_K(\mathbf{F}, \mathbf{P}) + \hom_K(\mathbf{F}, \mathbf{A}_n) && \text{(by Proposition 3.14(b))} \\ = \; & \hom_K(\mathbf{F}, \mathbf{Q}) + \hom_K(\mathbf{F}, \mathbf{A}_n) && \text{(since } \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q})\text{)} \\ = \; & \hom_K(\mathbf{F}, \mathbf{Q'}) && \text{(by Proposition 3.14(b)).} \end{aligned}$$

Thus, $\mathrm{lpf}_K^{\mathcal{F}}(\mathbf{P'}) = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q'})$, and we conclude that $\bigcup_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ does not admit any left query algorithm over K either (by Proposition 5.8 and Remark 5.1). □

By Proposition 5.6(a), we have that the characterization in Theorem 5.19 also holds for the *complements* of homomorphic-equivalence classes, $\overline{[\mathbf{A}]_{\leftrightarrow}}$, and that *deciding whether* $\overline{[\mathbf{A}]_{\leftrightarrow}}$ *admits a left query algorithm over* $\mathbb{B}$ *is* NP*-complete.*

Furthermore, the complexity result and the characterization extend to *intersections* of homomorphic-equivalence classes, $\bigcap_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$. In fact, for all distinct $i, j \in [n]$, either $[\mathbf{A}_i]_{\leftrightarrow} = [\mathbf{A}_j]_{\leftrightarrow}$ or $[\mathbf{A}_i]_{\leftrightarrow} \cap [\mathbf{A}_j]_{\leftrightarrow} = \emptyset$; in other words, either $\bigcap_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ collapses to some $[\mathbf{A}_i]_{\leftrightarrow}$ or $\bigcap_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow} = \emptyset$. Therefore, *deciding whether* $\bigcap_{i=1}^n [\mathbf{A}_i]_{\leftrightarrow}$ *admits a left query algorithm over* $\mathbb{B}$ *is* NP*-complete* and the characterization in Theorem 5.19 holds for

$\bigcap_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$.

It is not yet clear whether the full characterization involving FO-definability in Theorem 5.16 holds for homomorphic-equivalence classes of structures and their unions, complements and intersections. We will see that the answer is "yes" in the next subsection.

5.2.3 Two More Characterizations

In this subsection, we give two more characterizations about admitting a left query algorithm (over B or over N), one in Theorem 5.22 and the other in Corollary 5.25. These characterizations, as well as other results mentioned here, are from [8].

At the beginning of Section 5.2, we saw that if a class $\mathcal{D}$ admits a left query algorithm over B, then it is FO-definable and closed under homomorphic equivalence (cf. Proposition 5.11). It turns out that the converse is also true.

Theorem 5.22. *Let $\mathcal{D}$ be a class of structures. The following are equivalent:*

(i) *$\mathcal{D}$ admits a left query algorithm over* B*.*

(ii) *$\mathcal{D}$ is a Boolean combination of* CQ *classes.*

(iii) *$\mathcal{D}$ is definable by a Boolean combination of* PP*-sentences.*

(iv) *$\mathcal{D}$ is* FO*-definable and closed under homomorphic equivalence.* □

The preceding theorem immediately implies the corollary below, by which we have a full characterization as in Theorem 5.16 for homomorphic-equivalence classes and their complements, intersections and unions, as remarked at the end of the previous subsection.

Corollary 5.23. *For every class $\mathcal{D}$ of structures closed under homomorphic equivalence, we have $\mathcal{D}$ admits a left query algorithm over* B *if and only if $\mathcal{D}$ is* FO*-definable.* □

A more general characterization than Theorem 5.15 is given in Corollary 5.25, which is a direct consequence of Theorem 5.24 next.

Theorem 5.24. *Let $\mathcal{D}$ be a class of structures closed under homomorphic equivalence. For every finite nonempty class $\mathcal{F}$ of connected structures, we have $\mathcal{D}$ admits a left query algorithm over* B *of the form $(\mathcal{F}, X)$ if and only if $\mathcal{D}$ admits a left query algorithm over* N *of the form $(\mathcal{F}, X')$.* □

Corollary 5.25. *Let $\mathcal{D}$ be a class of structures closed under homomorphic equivalence. The following are equivalent:*

(i) *$\mathcal{D}$ is* FO*-definable.*

(ii) *$\mathcal{D}$ admits a left query algorithm over* B*.*

(iii) *$\mathcal{D}$ admits a left query algorithm over* N*.* □

In particular, for every class $\mathcal{D}$ of structures that is a Boolean combination of CSP classes or of homomorphic-equivalence classes, it is closed under homomorphic equivalence; hence, it is FO-definable if and only if it admits a left query algorithm over B, and if and only if it admits a left query algorithm over N.

5.3 Right Query Algorithms

A prominent example of a class admitting a right query algorithm over B is a CSP (cf. Example 5.7). In this section, we derive characterizations for certain classes to admit a right query algorithm over B (and over N in some limited cases).

5.3.1 CQ Classes and Their Boolean Combinations

For every graph $\mathbf{G}$, we have, by definition, that $\mathrm{CQ}(\mathbf{G})$ admits the left 1-query algorithm $(\{\mathbf{G}\}, \{1\})$ over B and the left 1-query algorithm $(\{\mathbf{G}\}, \mathbb{Z}^+)$ over N (by Proposition 5.1) and, by Remark 5.2, that $\mathrm{CQ}(\mathbf{G})$ is FO-definable. However, it is not yet clear when $\mathrm{CQ}(\mathbf{G})$ admits a right query algorithm over B or over N.

Analogous to Subsection 5.2.1, in this subsection we investigate CQ classes in terms of right query algorithms (mainly over B). We will first give a characterization of CQ classes of *graphs* that admit a right query algorithm over B (and also over N), then we proceed to show that a similar characterization can be obtained for CQ classes of *structures*, and finally we conclude with a more general characterization that will imply one for *Boolean combinations* of CQ classes of structures that admit a right query algorithm over B.

Lemma 5.26. *Let* $\mathbf{G}$ *be a graph. The following are equivalent:*

(i) $\mathbf{G}$ *is* 2*-colorable.*

(ii) $\mathrm{CQ}(\mathbf{G})$ *admits the right* 1*-query algorithm* $(\{\mathbf{I}_1\}, \mathbb{B})$ *or the right* 1*-query algorithm* $(\{\mathbf{I}_1\}, \{0\}))$ *over* B.

(iii) $\mathrm{CQ}(\mathbf{G})$ *admits the right* 1*-query algorithm* $(\{\mathbf{I}_1\}, \mathbb{N})$ *or the right* 1*-query algorithm* $(\{\mathbf{I}_1\}, \{0\}))$ *over* N.

(iv) $\mathrm{CQ}(\mathbf{G})$ *admits a right query algorithm over* B.

(v) $\mathrm{CQ}(\mathbf{G})$ *admits a right query algorithm over* N.

Proof. The directions from (ii) to (iv) and from (iii) to (v) are trivial, while those from (ii) to (iii) and from (iv) to (v) are by Proposition 5.1.

For the direction from (i) to (ii), we consider the two cases below.

Case 1. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) = 1$. Then, $\mathbf{G}$ is an independent set, which means that $\mathrm{CQ}(\mathbf{G}) = \mathcal{G}$. By Example 5.1, we have $(\{\mathbf{I}_1\}, \mathbb{B})$ as a right 1-query algorithm over B for $\mathrm{CQ}(\mathbf{G})$.

Case 2. $\mathbf{G}$ *has chromatic number* $\chi(\mathbf{G}) = 2$. Then, $\mathrm{CQ}(\mathbf{G}) = \mathrm{CQ}(\mathbf{K}_2)$ because $\mathbf{G}$ is homomorphically equivalent to $\mathbf{K}_2$, which implies for every graph $\mathbf{H}$ that $\mathbf{H} \in \mathrm{CQ}(\mathbf{G})$ if and only if $\mathbf{H} \notin \mathcal{I}$, the class of independent sets. By Example 5.2 and (the proof of) Proposition 5.6(a), $\mathrm{CQ}(\mathbf{G})$ admits the right 1-query algorithm $(\{\mathbf{I}_1\}, \{0\})$ over B.

Finally, we argue that not (i) implies not (v). If $\mathbf{G}$ is not 2-colorable, then it must contain $\mathbf{C}_n$ as a subgraph where $n \geq 3$ is an odd integer, by Proposition 2.4. Note that $\mathbf{G}$ has odd girth $\gamma_{\mathrm{odd}}(\mathbf{G}) \leq n$.

Assume that $\mathcal{F}$ is an arbitrary class of k graphs, where $k \in \mathbb{Z}^+$. Take $\mathbf{K}_p$ where p is greater than the size of $\mathbf{G}$ as well as the maximum size of the graphs in $\mathcal{F}$. By Theorem 2.5, there exists a graph $\mathbf{H}$ with girth $\gamma(\mathbf{H}) \geq n+1$ and chromatic number $\chi(\mathbf{H})$ greater than the maximum chromatic number of the graphs in $\mathcal{F}$. Hence, $\gamma_{\mathrm{odd}}(\mathbf{H}) \geq n+1$ because $\gamma_{\mathrm{odd}}(\mathbf{H}) \geq \gamma(\mathbf{H})$. It follows that $\mathbf{K}_p \in \mathrm{CQ}(\mathbf{G})$ while $\mathbf{H} \notin \mathrm{CQ}(\mathbf{G})$ (by Corollary 3.13(a)), but $\mathrm{rpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{K}_p) = (\underbrace{0, \ldots, 0}_{k\text{-times}}) = \mathrm{rpf}^{\mathcal{F}}_{\mathrm{N}}(\mathbf{H})$. The result follows from Remark 5.1. $\square$

In the context for graphs, that is, when $\mathcal{U} = \mathcal{G}$, we say that a $\mathrm{PP}[\sigma(\mathrm{G})]$-sentence φ is *acyclic* if the graph corresponding to its canonical structure $\mathbf{A}^{\varphi}$ is acyclic.

An analysis of the condition that $\mathbf{G}$ is 2-colorable in Lemma 5.26 leads to a further characterization of when $\mathrm{CQ}(\mathbf{G})$ admits a right query algorithm over B and over N, in analogy to Theorem 5.13.

Theorem 5.27. *Let* $\mathbf{G}$ *be a graph. The following are equivalent:*

(i) $\mathrm{CQ}(\mathbf{G})$ *is definable by a Boolean combination of acyclic* PP*-sentences.*

(ii) $\mathrm{CQ}(\mathbf{G})$ *admits a right query algorithm over* B.

(iii) $\mathrm{CQ}(\mathbf{G})$ *admits a right query algorithm over* N.

Proof. By Theorem 5.26, it suffices to show that *for every graph* $\mathbf{G}$*, it is* 2*-colorable if and only if* $\mathrm{CQ}(\mathbf{G})$ *is definable by a Boolean combination of acyclic* PP*-sentences.* We distinguish two cases as follows.

Case 1. $\mathbf{G}$ *is* 2*-colorable.* By the proof of Theorem 5.26, if $\mathbf{G}$ has chromatic number $\chi(\mathbf{G}) = 1$, then $\mathrm{CQ}(\mathbf{G}) = \mathcal{G}$ and hence we can choose $\varphi := (\exists x \exists y\, E(x, y) \vee \neg\exists x \exists y\, E(x, y))$. Otherwise, $\chi(\mathbf{G}) = 2$, hence $\mathrm{CQ}(\mathbf{G}) = \mathrm{CQ}(\mathbf{K}_2)$ and it suffices to take $\varphi := \exists x \exists y\, E(x, y)$. Obviously, in both situations, $\mathrm{CQ}(\mathbf{G}) = \mathrm{Mod}(\varphi)$ and φ is a Boolean combination of acyclic PP-sentences.

Case 2. $\mathbf{G}$ *is not* 2*-colorable.* Then, $\mathbf{G}$ contains a cycle of odd length ≥ 3, by Proposition 2.4. That is, $\gamma_{\mathrm{odd}}(\mathbf{G}) < \infty$. Suppose, for the contrary, that $\mathrm{CQ}(\mathbf{G}) = \mathrm{Mod}(\varphi)$, where φ is a Boolean combination of the acyclic $\mathrm{PP}[\sigma(\mathrm{G})]$-sentences $\varphi_1, \dots, \varphi_n$, where $n \in \mathbb{Z}^+$. We have $\mathbf{G} \models \varphi$ since $\mathbf{G} \in \mathrm{CQ}(\mathbf{G})$. Let $\mathbf{F}_1, \dots, \mathbf{F}_n$ be the graphs corresponding to the canonical structures $\mathbf{A}^{\varphi_1}, \dots, \mathbf{A}^{\varphi_n}$; note that $\mathbf{F}_i$ is acyclic and $E(\mathbf{F}_i) \neq \emptyset$ for all $i \in [n]$. Without loss of generality, we assume that φ is in disjunctive normal form such that for every $i \in [n]$, each disjunct of φ contains either φ_i or $\neg\varphi_i$ in its conjunction and, in addition, we assume for some $k \in [0, n]$ that $\varphi_1, \dots, \varphi_k$ are those among $\varphi_1, \dots, \varphi_n$ that hold in $\mathbf{G}$. We further distinguish two cases.

Case 2-1. $k = 0$. Then, $\mathbf{G} \models (\neg\varphi_1 \wedge \cdots \wedge \neg\varphi_n)$ and hence $\mathrm{Mod}(\neg\varphi_1 \wedge \cdots \wedge \neg\varphi_n) \subseteq$

$\mathrm{Mod}(\varphi)$. For all $i \in [n]$, as $E(\mathbf{F}_i) \neq \emptyset$, we have $\mathbf{F}_i \not\to \mathbf{I}_1$ and hence $\mathbf{I}_1 \models \neg\varphi_i$ (by Proposition 5.9(a) in view of Remark 5.3). Therefore, $\mathbf{I}_1 \models (\neg\varphi_1 \wedge \cdots \wedge \neg\varphi_n)$. It follows that $\mathbf{I}_1 \in \mathrm{Mod}(\varphi) = \mathrm{CQ}(\mathbf{G})$, a contradiction because $\mathbf{G}$ is not 2-colorable.

Case 2-2. $k > 0$. Then, $\mathbf{G} \models (\bigwedge_{i=1}^{k} \varphi_i \wedge \bigwedge_{i=k+1}^{n} \neg\varphi_i)$ and hence $\mathrm{Mod}(\bigwedge_{i=1}^{k} \varphi_i \wedge \bigwedge_{i=k+1}^{n} \neg\varphi_i) \subseteq \mathrm{Mod}(\varphi)$. For all $i \in [n]$, we have $\mathbf{F}_i \to \mathbf{G}$ if and only if $i \in [k]$, by Proposition 5.9(a) in view of Remark 5.3. Take $\mathbf{H} = \mathbf{F}_1 \oplus \cdots \oplus \mathbf{F}_k$. Then, $\mathbf{H}$ is an acyclic graph (thus, $\gamma_{\mathrm{odd}}(\mathbf{H}) = \infty$) and $\mathbf{H} \to \mathbf{G}$ (by Proposition 3.14(c)). In fact, for all $i \in [n]$, we have $\mathbf{F}_i \to \mathbf{H}$ if and only if $i \in [k]$:

- If $i \in [k]$, then obviously $\mathbf{F}_i \to \mathbf{H}$ as $\mathbf{F}_i \subseteq \mathbf{H}$.
- If $i \notin [k]$, then $\mathbf{F}_i \not\to \mathbf{H}$ since $\mathbf{H} \to \mathbf{G}$ and $\mathbf{F}_i \not\to \mathbf{G}$.

By Proposition 5.9(a) in view of Remark 5.3 again, we have $\mathbf{H} \models (\bigwedge_{i=1}^{k} \varphi_i \wedge \bigwedge_{i=k+1}^{n} \neg\varphi_i)$. It follows that $\mathbf{H} \in \mathrm{Mod}(\varphi) = \mathrm{CQ}(\mathbf{G})$, a contradiction to Corollary 3.13(a). □

From the proof of Lemma 5.26, we can extract a trichotomy for CQ classes that is analogous to Corollary 5.14 for CSP classes.

Corollary 5.28. *Let* $\mathbf{G}$ *be a graph, and consider the acyclic* PP*-sentence* $\varphi = \exists x \exists y\, E(x, y)$*. If* $\mathbf{G}$ *is an independent set, then* $\mathrm{CQ}(\mathbf{G}) = \mathrm{Mod}(\varphi \vee \neg\varphi)$*. If* $\mathbf{G}$ *is a bipartite graph containing an edge, then* $\mathrm{CQ}(\mathbf{G}) = \mathrm{Mod}(\varphi)$*. Otherwise,* $\mathrm{CQ}(\mathbf{G})$ *is not definable by any Boolean combination of acyclic* PP*-sentences.* □

In the context of structures, i.e., when $\mathcal{U} = \mathcal{A}$, we say that a PP-sentence φ is *acyclic* if the canonical structure $\mathbf{A}^{\varphi}$ is acyclic. For $\mathbf{A} \in \mathcal{U}$, we say that the class $\mathrm{CQ}(\mathbf{A})$ is *acyclic* if $\mathbf{A}$ is acyclic.

A proof for the next proposition can be done analogously to the one for Proposition 5.10 and hence is omitted.

Proposition 5.29. *For every class $\mathcal{D} \subseteq \mathcal{U}$, we have that $\mathcal{D}$ is a Boolean combination of acyclic* CQ *classes if and only if $\mathcal{D}$ is definable by a Boolean combination of acyclic* PP*-sentences.* $\square$

Next, we will turn attention to CQ classes of structures in view of the characterization in Theorem 5.27. Many of the results presented next as well as their proofs are from [8].

We have for CQ classes of structures a similar characterization to Theorem 5.27 for CQ classes of graphs, and it is an instance of Corollary 5.34 presented later on. Indeed, such a characterization holds for Boolean combinations of CQ classes of structures, and we will see it after Corollary 5.34.

Theorem 5.30. *Let $\mathbf{A}$ be a structure. The following are equivalent:*

(i) CQ($\mathbf{A}$) *admits a right query algorithm over* B.

(ii) CQ($\mathbf{A}$) *is definable by a Boolean combination of acyclic* PP*-sentences.*

(iii) CQ($\mathbf{A}$) *is a Boolean combination of* FO*-definable* CSP *classes.* $\square$

We saw in Example 5.6 that the class CQ($\mathbf{K}_3$) of *graphs* does not admit any right query algorithm over B (nor over N).

Similarly, if we consider a counterpart to $\mathbf{K}_3$ that is a structure, namely, the σ(G)-structure $\mathbf{A}$ for which $\mathrm{dom}(\mathbf{A}) := \{a_1, a_2, a_3\}$ and $E^{\mathbf{A}} := \{(a_1, a_2), (a_2, a_3), (a_3, a_1)\}$, then *the class* CQ($\mathbf{A}$) *of* σ(G)*-structures does not admit any right query algorithm over* B, either. In fact, $\mathbf{A}$ has girth $\gamma(\mathbf{A}) = 3$, and the statement that $\mathrm{CQ}(\mathbf{A})$ does not admit such an algorithm follows from the theorem presented next known as the *Sparse Incomparability Lemma* for structures (cf. Lemma 2.5). More precisely, for every finite nonempty class $\mathcal{F}$ of $\sigma(\mathrm{G})$-structures, we let n be the maximum size of the $\sigma(\mathrm{G})$-structures in $\mathcal{F}$. The theorem then guarantees the existence of a structure $\mathbf{B}$ of girth $\gamma(\mathbf{B}) \geq 4$ for which $\hom_{\mathrm{B}}(\mathbf{A}, \mathbf{C}) = \hom_{\mathrm{B}}(\mathbf{B}, \mathbf{C})$ for all $\sigma(\mathrm{G})$-structures $\mathbf{C}$ of size $\leq n$, hence $\mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{A}) = \mathrm{lpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{B})$. Obviously, we have $\mathbf{A} \in \mathrm{CQ}(\mathbf{A})$. However, $\mathbf{B} \notin \mathrm{CQ}(\mathbf{A})$ by Corollary 3.13(b).

Theorem 5.31. [37] *Let $m, n \in \mathbb{Z}^+$. For every structure $\mathbf{A}$, there is a structure $\mathbf{B}$ of girth $\gamma(\mathbf{B}) \geq m$ such that for all structures $\mathbf{C}$ of size $\leq n$, we have $\mathbf{A} \to \mathbf{C}$ if and only if $\mathbf{B} \to \mathbf{C}$.* □

We see that the above class $\mathrm{CQ}(\mathbf{A})$ is FO-definable and closed under homomorphic equivalence, and it admits a left query algorithm over B (by the discussions in Sections 5.1 and 5.2). Nevertheless, we also see that it does not admit any right query algorithm B, just earlier. It is natural to ask: What is a characterization of classes being FO-definable and closed under homomorphic equivalence and admitting a right query algorithm over B at the same time? Equivalently (by Theorem 5.22), what is a characterization of classes admitting both a left query algorithm and a right query algorithm over B? We have an answer as follows.

Theorem 5.32. *Let $\mathcal{D}$ be a class of structures. The following are equivalent:*

(i) *$\mathcal{D}$ admits both a left query algorithm and a right query algorithm over* B.

(ii) *$\mathcal{D}$ admits a left query algorithm over* N *and a right query algorithm over* B.

(iii) *$\mathcal{D}$ is definable by a Boolean combination of acyclic* PP*-sentences.*

(iv) *$\mathcal{D}$ is a Boolean combination of acyclic* CQ *classes.*

(v) *$\mathcal{D}$ is a Boolean combination of* FO*-definable* CSP *classes.*

To prove this theorem, we will need the following result. The reader is reminded of Definition 5.3 for a finite homomorphism duality.

Theorem 5.33. [24] *Let $\mathbf{A}$ be a structure. The following are equivalent:*

(i) *$\mathbf{A}$ is homomorphically equivalent to an acyclic structure.*

(ii) *There exists a finite homomorphism duality of the form $(\{\mathbf{A}\}, \mathcal{E})$.*

(iii) *There exists a finite homomorphism duality of the form $(\{\mathbf{A}\}, \mathcal{E})$ for which* $\mathrm{CSP}(\mathbf{E})$ *is* FO*-definable for every structure $\mathbf{E} \in \mathcal{E}$.*

Proof of Theorem 5.32. The equivalence between (i) and (ii) follows from Corollary 5.25 in view of Proposition 5.3, and the equivalence between (iii) and (iv) is given in Proposition 5.29. The direction from (v) to (i) follows by Proposition 5.3, Proposition 5.7(b) and Theorem 5.22.

For the direction from (iv) to (v), assume that $\mathcal{D}$ is a Boolean combination of the classes $\mathrm{CQ}(\mathbf{A}_1), \ldots, \mathrm{CQ}(\mathbf{A}_n)$, where $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are all acyclic structures. For all $i \in [n]$, since $\mathbf{A}_i \leftrightarrow \mathbf{A}_i$ and $\mathbf{A}_i$ is acyclic, by Theorem 5.33 we have that $\mathrm{CQ}(\mathbf{A}_i) = \bigcap_{\mathbf{E} \in \mathcal{E}} \overline{\mathrm{CSP}(\mathbf{E})}$, where $\mathrm{CSP}(\mathbf{E})$ is FO-definable for every structure $\mathbf{E} \in \mathcal{E}$. It follows that $\mathcal{D}$ is a Boolean combination of FO-definable CSP classes.

Finally, for the direction from (i) to (iii), assume that $\mathcal{D}$ admits a right query algorithm $(\mathcal{F}, X)$ over B and a left query algorithm over B, which, by Theorem 5.22, implies that $\mathcal{D}$ is definable by a Boolean combination φ of $\varphi_1, \ldots, \varphi_k$ that are PP-sentences, where $k \in \mathbb{Z}^+$. In other words, $\mathcal{D} = \mathrm{Mod}(\varphi)$. Let $m \in \mathbb{Z}^+$ be greater than the maximum size among $\mathbf{A}^{\varphi_1}, \ldots, \mathbf{A}^{\varphi_i}$.

For all $i \in [k]$, let $\mathbf{A}_{(i,1)}, \ldots, \mathbf{A}_{(i,j_i)}$ enumerate the acyclic homomorphic images of $\mathbf{A}^{\varphi_i}$, where $j_i \in \mathbb{N}$, and let

$$\varphi_i' := \begin{cases} \varphi^{\mathbf{C}_i} \wedge \neg\varphi^{\mathbf{C}_i} & \text{if } j_i = 0, \\ \bigvee_{r=1}^{j_i} \varphi^{\mathbf{A}_{(i,r)}} & \text{otherwise,} \end{cases}$$

where $\mathbf{C}_i$ is an arbitrarily chosen acyclic structure. Then, it follows that $\mathrm{Mod}(\varphi_i')$ admits a right query algorithm over B:

- If $j_i = 0$, then $\mathrm{Mod}(\varphi_i') = \emptyset$ and hence admits a trivial right query algorithm over B (cf. Example 5.1).
- Otherwise, for all $r \in [j_i]$, we have that $\mathbf{A}_{(i,r)} \leftrightarrow \mathbf{A}_{(i,r)}$ and, since $\mathbf{A}_{(i,r)}$ is acyclic, that $\mathrm{CQ}(\mathbf{A}_{(i,r)})$ is equal to a Boolean combination of (indeed, an intersection of the respective complements of) CSP classes by Theorem 5.33, and hence it admits a right query algorithm over B, by Proposition 5.7(b). Thus, $\mathrm{Mod}(\varphi^{\mathbf{A}_{(i,r)}})$ admits a right query algorithm over B because $\mathrm{CQ}(\mathbf{A}_{(i,r)}) = \mathrm{Mod}(\varphi^{\mathbf{A}_{(i,r)}})$, by Proposition 5.9(b). Consequently, $\mathrm{Mod}(\varphi_i')$ admits a right query algorithm over B, by Proposition 5.6(b) in view of Remark 3.9.

Let φ' be obtained from φ by replacing φ_i with φ_i' for all $i \in [k]$, then it follows, from the previous discussion, and by Proposition 5.6 in view of Remark 3.9, that $\mathrm{Mod}(\varphi')$ admits a right query algorithm over B, say $(\mathcal{F}', X')$. Let n be the maximum size of the structures occurring in $\mathcal{F} \cup \mathcal{F}'$.

By construction, φ' is a Boolean combination of acyclic PP-sentences. We are done after we show that $\mathrm{Mod}(\varphi') = \mathrm{Mod}(\varphi)$ (recall $\mathcal{D} = \mathrm{Mod}(\varphi)$).

For this purpose, let $\mathbf{A}$ be an arbitrary structure. With the integers m and n chosen above, Theorem 5.31 guarantees the existence of a structure $\mathbf{B}$ of girth $\gamma(\mathbf{B}) \geq m$ such that for all structures $\mathbf{C}$ of size $\leq n$, we have $\mathbf{A} \to \mathbf{C}$ if and only if $\mathbf{B} \to \mathbf{C}$, which implies that

(1) $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{A}) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{B})$,

(2) $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}'}(\mathbf{A}) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}'}(\mathbf{B})$.

Next, we argue that for all $i \in [k]$, it holds that $\mathbf{B} \in \mathrm{Mod}(\varphi_i)$ if and only if $\mathbf{B} \in \mathrm{Mod}(\varphi_i')$. We distinguish two cases.

Case 1. $\mathbf{A}^{\varphi_i}$ has no acyclic homomorphic images. In particular, $\mathbf{A}^{\varphi_i}$ contains a cycle (thus has a finite girth) and $\mathbf{A}^{\varphi_i} \not\to \mathbf{B}$ by Corollary 3.13(b) and the fact that $\mathbf{B}$ has girth $\gamma(\mathbf{B}) \geq m$. Thus, $\mathbf{B} \notin \mathrm{Mod}(\varphi_i)$ by Proposition 5.9(a), and $\mathbf{B} \notin \emptyset = \mathrm{Mod}(\varphi^{\mathbf{C}_i} \wedge \neg\varphi^{\mathbf{C}_i}) = \mathrm{Mod}(\varphi_i')$.

Case 2. $\mathbf{A}^{\varphi_i}$ has acyclic homomorphic images. Assume that these images are $\mathbf{A}_{(i,1)}, \ldots, \mathbf{A}_{(i,j_i)}$. Then, $\mathbf{A}^{\varphi_i} \to \mathbf{A}_{(i,1)}, \ldots, \mathbf{A}^{\varphi_i} \to \mathbf{A}_{(i,j_i)}$. If $\mathbf{A}_{(i,r)} \to \mathbf{B}$ for some $r \in [j_i]$, then $\mathbf{A}^{\varphi_i} \to \mathbf{B}$. Conversely, if $\mathbf{A}^{\varphi_i} \to \mathbf{B}$, then $\mathbf{A}_{(i,r)} \subseteq \mathbf{B}$ and hence $\mathbf{A}_{(i,r)} \to \mathbf{B}$ for some $r \in [j_i]$, because $\mathbf{B}$ has girth $\gamma(\mathbf{B}) \geq m$, which is greater than the size of $\mathbf{A}^{\varphi_i}$. That is, $\mathbf{A}^{\varphi_i} \to \mathbf{B}$ if and only if $\mathbf{A}_{(i,r)} \to \mathbf{B}$ for some $r \in [j_i]$. By Proposition 5.9(a) in view of Remark 3.9, we have $\mathbf{B} \in \mathrm{Mod}(\varphi_i)$ if and only if $\mathbf{B} \in \mathrm{Mod}(\bigvee_{r=1}^{j_i} \varphi^{\mathbf{A}_{(i,r)}}) = \mathrm{Mod}(\varphi_i')$.

Hence,

(3) $\mathbf{B} \in \mathrm{Mod}(\varphi)$ if and only if $\mathbf{B} \in \mathrm{Mod}(\varphi')$.

Putting all things together, we have

$$
\begin{array}{ll}
 & \mathbf{A} \in \mathrm{Mod}(\varphi) \\
\text{iff} & \mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{A}) \in X \\
\text{iff} & \mathrm{rpf}^{\mathcal{F}}_{\mathrm{B}}(\mathbf{B}) \in X \qquad \text{(by (1))} \\
\text{iff} & \mathbf{B} \in \mathrm{Mod}(\varphi) \\
\text{iff} & \mathbf{B} \in \mathrm{Mod}(\varphi') \qquad \text{(by (3))} \\
\text{iff} & \mathrm{rpf}^{\mathcal{F}'}_{\mathrm{B}}(\mathbf{B}) \in X' \\
\text{iff} & \mathrm{rpf}^{\mathcal{F}'}_{\mathrm{B}}(\mathbf{A}) \in X' \quad \text{(by (2))} \\
\text{iff} & \mathbf{A} \in \mathrm{Mod}(\varphi'),
\end{array}
$$

as desired. □

Since for every class $\mathcal{D}$, it is FO-definable and closed under homomorphic equivalence if and only if it admits a left query algorithm over B (by Theorem 5.22), as a consequence of Theorem 5.32, we have the following characterization similar to Corollary 5.25.

Corollary 5.34. *Let $\mathcal{D}$ be a* FO*-definable class of structures closed under homomorphic equivalence. The following are equivalent:*

(i) *$\mathcal{D}$ admits a right query algorithm over* B*.*

(ii) *$\mathcal{D}$ is definable by a Boolean combination of acyclic* PP*-sentences.*

(iii) *$\mathcal{D}$ is a Boolean combination of* FO*-definable* CSP *classes.* □

In particular, all Boolean combinations of CQ classes of structures admit a left query algorithm over B and hence are FO-definable and closed under homomorphic equivalence (by the discussions in Sections 5.1 and 5.2). As a consequence of Corollary 5.34, they admit a right query algorithm over B if and only if they are definable by a Boolean combination φ of acyclic PP-sentences, and if and only if they are a Boolean combination of FO-definable CSP classes.

5.3.2 Homomorphic-Equivalence Classes and Their Unions

We turn to homomorphic-equivalence classes of structures and a characterization of when a union of such classes admits a right query algorithm over B or over N. Many of the results presented in this subsection as well as their proofs are from [8]. We start with an analogue for CQ classes of structures to Lemma 5.18.

Lemma 5.35. *Let K be the semiring* B *or* N*. For every structure $\mathbf{A}$, we have that $[\mathbf{A}]_{\leftrightarrow}$ admits a right query algorithm over K if and only if* $\mathrm{CQ}(\mathbf{A})$ *admits a right query algorithm over K.*

Proof. Fix a semiring K. For the "if" part, observe that

$$[\mathbf{A}]_{\leftrightarrow} = \mathrm{CQ}(\mathbf{A}) \cap \mathrm{CSP}(\mathbf{A}).$$

It is obvious that $\mathrm{CSP}(\mathbf{A})$ admits a right query algorithm over K. The claim follows by Proposition 5.6(b).

For the "only if" part. Assume that $\mathrm{CQ}(\mathbf{A})$ does not admit any right query algorithm over K. Let $\mathcal{F}$ be an arbitrary finite nonempty set of structures. For the class $\mathcal{F}' := \{\mathbf{F} \uparrow \mathbf{A} \mid \mathbf{F} \in \mathcal{F}\}$, there are two structures $\mathbf{P} \in \mathrm{CQ}(\mathbf{A})$ and $\mathbf{Q} \notin \mathrm{CQ}(\mathbf{A})$ such that $\mathrm{rpf}_K^{\mathcal{F}'}(\mathbf{P}) = \mathrm{rpf}_K^{\mathcal{F}'}(\mathbf{Q})$, by Remark 5.1, which implies $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A} \otimes \mathbf{P}) = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A} \otimes \mathbf{Q})$, by Proposition 3.15. Our goal is to *present two structures $\mathbf{P'} \in [\mathbf{A}]_{\leftrightarrow}$ and $\mathbf{Q'} \notin [\mathbf{A}]_{\leftrightarrow}$ such that* $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{P'}) = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{Q'})$.

Take $\mathbf{P}' := \mathbf{A} \otimes \mathbf{P}$ and $\mathbf{Q}' := \mathbf{A} \otimes \mathbf{Q}$. Then, it is immediate that $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{P}') = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{Q}')$. By parts (d) and (e) of Proposition 3.14, we have $\mathbf{P}' \in [\mathbf{A}]_{\leftrightarrow}$ since $\mathbf{A} \to \mathbf{P}$, and we have $\mathbf{Q}' \notin [\mathbf{A}]_{\leftrightarrow}$ since $\mathbf{A} \not\to \mathbf{Q}$. Thus, by Remark 5.1, $[\mathbf{A}]_{\leftrightarrow}$ does not admit any right query algorithm over K. $\square$

Theorem 5.36. *Let* $\mathbf{A}$ *be a structure. The following are equivalent:*

(i) $[\mathbf{A}]_{\leftrightarrow}$ *admits a right query algorithm over* B.

(ii) $\mathrm{CQ}(\mathbf{A})$ *admits a right query algorithm over* B.

(iii) $\mathbf{A}$ *is homomorphically equivalent to an acyclic structure.*

Proof. The equivalence between (i) and (ii) immediately follows from Lemma 5.35.

For the direction from (iii) to (i), assume that $\mathbf{A}$ is homomorphically equivalent to an acyclic structure. Then, by Theorem 5.33, there is a finite homomorphism duality of the form $(\{\mathbf{A}\}, \mathcal{E})$. Therefore, $\mathrm{CQ}(\mathbf{A}) = \bigcap_{\mathbf{E}\in\mathcal{E}} \overline{\mathrm{CSP}(\mathbf{E})}$, which, by Proposition 5.7(b), admits a right query algorithm over B. Moreover, $\mathrm{CSP}(\mathbf{A})$ obviously admits a right query algorithm over B. Since $[\mathbf{A}]_{\leftrightarrow} = \mathrm{CQ}(\mathbf{A}) \cap \mathrm{CSP}(\mathbf{A})$, it follows by Proposition 5.6(b) that $[\mathbf{A}]_{\leftrightarrow}$ admits a right query algorithm over B.

Finally, for the direction from (ii) to (iii), we will argue as in the proof for the direction from (i) to (iii) of Theorem 5.32. Assume that $\mathrm{CQ}(\mathbf{A})$ has a right query algorithm $(\mathcal{F}, X)$ over B. Then, $\mathbf{A}$ must have acyclic homomorphic images: If all homomorphic images of $\mathbf{A}$ contain a cycle, then in particular, $\mathbf{A}$ contains a cycle and hence has a finite girth. Let $m > |\mathrm{dom}(\mathbf{A})| \geq \gamma(\mathbf{A})$ and let n be the maximum size of the structures in $\mathcal{F}$, then Theorem 5.31 implies the existence of a structure $\mathbf{B}$ with $\gamma(\mathbf{B}) \geq m$ such that for all structures $\mathbf{C}$ of size $\leq n$, it holds that $\mathbf{A} \to \mathbf{C}$ if and only if $\mathbf{B} \to \mathbf{C}$. Thus, $\mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{A}) = \mathrm{rpf}_{\mathrm{B}}^{\mathcal{F}}(\mathbf{B})$ and we have $\mathbf{A} \in \mathrm{CQ}(\mathbf{A})$ if and only if $\mathbf{B} \in \mathrm{CQ}(\mathbf{A})$, a contradiction because $\mathbf{A} \not\to \mathbf{B}$ (by Corollary 3.13).

Therefore, $\mathbf{A}$ has acyclic homomorphic images $\mathbf{A}_1, \ldots, \mathbf{A}_s$. Take $\varphi' := \bigvee_{r=1}^{s} \varphi^{\mathbf{A}_r}$, and argue as in the proof in Theorem 5.32 (for the direction from (i) to (iii)), we have that $\mathrm{Mod}(\varphi')$ has a right query algorithm $(\mathcal{F}', X')$ over B. Let n be the maximum size of the structures in $\mathcal{F} \cup \mathcal{F}'$, and let $m > |\mathrm{dom}(\mathbf{A})| \geq \gamma(\mathbf{A})$. We can likewise argue that $\mathrm{Mod}(\varphi^{\mathbf{A}}) = \mathrm{Mod}(\varphi')$, with $\varphi^{\mathbf{A}}$ in the role of φ in the proof there (note that $\mathrm{CQ}(\mathbf{A}) = \mathrm{Mod}(\varphi^{\mathbf{A}})$ by Proposition 5.9(b)). Since $\mathbf{A} \in \mathrm{CQ}(\mathbf{A}) = \mathrm{Mod}(\varphi^{\mathbf{A}}) = \mathrm{Mod}(\varphi')$, we have $\mathbf{A} \in \mathrm{Mod}(\varphi^{\mathbf{A}_r})$ and hence $\mathbf{A}_r \to \mathbf{A}$ (by Proposition 5.9(b)) for some $r \in [s]$. Besides, we have $\mathbf{A} \to \mathbf{A}_r$, since $\mathbf{A}_r$ is a homomorphic image of $\mathbf{A}$. Therefore, $\mathbf{A}$ is homomorphically equivalent to $\mathbf{A}_r$, an acyclic structure. $\square$

Thus, from Theorem 5.36 it holds that *deciding whether* $\mathbf{A}$ *is homomorphically equivalent to an acyclic structure is* NP*-complete*: Testing whether $\mathbf{A}$ is homomorphically equivalent to an acyclic structure amounts to testing whether the *core of* $\mathbf{A}$ is acyclic, a task that is in NP and whose NP-hardness follows from [12]. Thus, it follows that both *deciding whether* $\mathrm{CQ}(\mathbf{A})$ *admits a right query algorithm over* B *is* NP*-complete* and that *deciding whether* $[\mathbf{A}]_{\leftrightarrow}$ *admits a right query algorithm over* B *is* NP*-complete.*

Finally, we present an analogue of Theorem 5.21 for right query algorithms over B or over N, from which it will follow that *deciding whether* $\bigcup_{i=1}^{n} [\mathbf{A}_i]_{\leftrightarrow}$ *admits a right query algorithm over* B *is* NP*-complete.*

Theorem 5.37. *Let K be the semiring* B *or* N*. For all structures $\mathbf{A}_1, \ldots, \mathbf{A}_n$, the union*

$\bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ *admits a right query algorithm over* K *if and only if* $[\mathbf{A}_i]_{\leftrightarrow}$ *admits a right query algorithm over* K *for all* $i \in [n]$*.*

Proof. Fix a semiring K. The "if" part immediately follows from Proposition 5.6(b). For the "only if" part, we prove by induction on n. The base case $n = 1$ is trivial since the statement is the same on either side.

For the inductive case $n \geq 2$, note that, without loss of generality, we can assume that $[\mathbf{A}_1]_{\leftrightarrow}, \ldots, [\mathbf{A}_n]_{\leftrightarrow}$ are pairwise disjoint: If $[\mathbf{A}_i]_{\leftrightarrow} = [\mathbf{A}_j]_{\leftrightarrow}$ for some distinct $i, j \in [n]$, then we can remove, say, $[\mathbf{A}_j]_{\leftrightarrow}$ from the union and the resulting union class will remain unchanged.

We will prove the contraposition: *If* $[\mathbf{A}_i]_{\leftrightarrow}$ *does not admit any right query algorithm over* K *for some* $i \in [n]$*, then neither does* $\bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$. Observe that $\to$ is a preorder among $\mathbf{A}_1, \ldots, \mathbf{A}_n$ (cf. Remark 3.10(b)) and hence there is a minimal which is, without loss of generality, assumed to be $\mathbf{A}_n$: For all $i \in [n-1]$, we have $\mathbf{A}_i \not\to \mathbf{A}_n$. Next, we distinguish two cases.

Case 1. $[\mathbf{A}_n]_{\leftrightarrow}$ *admits a right query algorithm over* K*.* Then, there is an $i \in [n-1]$ such that $[\mathbf{A}_i]_{\leftrightarrow}$ does not admit any right query algorithm over K. By the induction hypothesis, we have $\bigcup_{i=1}^{n-1}[\mathbf{A}_i]_{\leftrightarrow}$ does not admit any right query algorithm over K. By Proposition 5.6, it follows that $\bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ does not admit any right query algorithm over K either, because $\bigcup_{i=1}^{n-1}[\mathbf{A}_i]_{\leftrightarrow} = (\bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}) \cap \overline{[\mathbf{A}_n]_{\leftrightarrow}}$.

Case 2. $[\mathbf{A}_n]_{\leftrightarrow}$ *does not admit any right query algorithm over* K*.* By Lemma 5.35, we have that $\mathrm{CQ}(\mathbf{A}_n)$ does not admit any right query algorithm over K either. Now, let $\mathcal{F}$ be an arbitrary finite nonempty class of structures. Then, for the class $\mathcal{F}' := \{\mathbf{F} \uparrow \mathbf{A}_n \mid \mathbf{F} \in \mathcal{F}\}$, there are two structures $\mathbf{P} \in \mathrm{CQ}(\mathbf{A}_n)$ and $\mathbf{Q} \notin \mathrm{CQ}(\mathbf{A}_n)$ such that $\mathrm{rpf}_K^{\mathcal{F}'}(\mathbf{P}) = \mathrm{rpf}_K^{\mathcal{F}'}(\mathbf{Q})$, by Remark 5.1, which implies $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}_n \otimes \mathbf{P}) = \mathrm{rpf}_K^{\mathcal{F}}(\mathbf{A}_n \otimes \mathbf{Q})$, by Proposition 3.15. Our goal then is to *present two structures* $\mathbf{P}' \in \bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ *and* $\mathbf{Q}' \notin \bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ *such that* $\mathrm{rpf}_K^{\mathcal{F}}(\mathbf{P}') = \mathrm{lpf}_K^{\mathcal{F}}(\mathbf{Q}')$.

Take $\mathbf{P}' := \mathbf{A}_n \otimes \mathbf{P}$ and $\mathbf{Q}' := \mathbf{A}_n \otimes \mathbf{Q}$. By parts (d) and (e) of Proposition 3.14, we have

- $\mathbf{P}' \in [\mathbf{A}_n]_{\leftrightarrow}$ since $\mathbf{P} \in \mathrm{CQ}(\mathbf{A}_n)$,
- $\mathbf{Q}' \notin [\mathbf{A}_n]_{\leftrightarrow}$ since $\mathbf{Q} \notin \mathrm{CQ}(\mathbf{A}_n)$,
- for all $i \in [n-1]$, $\mathbf{Q}' \notin [\mathbf{A}_i]_{\leftrightarrow}$ since $\mathbf{A}_i \not\to \mathbf{A}_n$.

Therefore, $\mathbf{P}' \in \bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ and $\mathbf{Q}' \notin \bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$. It follows from Remark 5.1 that $\bigcup_{i=1}^{n}[\mathbf{A}_i]_{\leftrightarrow}$ does not admit any right query algorithm over K either. $\square$

5.4 Adaptive Homomorphism-Count Queries and Graph Isomorphism

In [10], Y. Chen, J. Flum, M. Liu, and Z. Xun show that *there is a construction of two graphs* $\mathbf{E}_1 = \mathbf{E}_1(n)$ *and* $\mathbf{E}_2 = \mathbf{E}_2(n)$ *as functions of* $n \in \mathbb{N}^+$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$ *each of size* n*, they are isomorphic if and only if* $\hom(\mathbf{E}_1, \mathbf{G}) = \hom(\mathbf{E}_1, \mathbf{H})$ *and* $\hom(\mathbf{E}_2, \mathbf{G}) = \hom(\mathbf{E}_2, \mathbf{H})$. This results in a procedure shown in Figure 5.3 for the decision problem `GRAPH-ISOMORPHISM` (see Section 2.2 for more details), with access to an oracle that computes homomorphism counts over N. Recall that $\hom(\mathbf{I}_1, \mathbf{G}) = |V(\mathbf{G})|$ for all graphs $\mathbf{G}$.

Figure 5.3: Pseudocode for GRAPH-ISOMORPHISM.

```
GRAPH-ISOMORPHISM(G, H)
n = hom(I_1, G)
n' = hom(I_1, H)
if n ≠ n'
    return "no"
else
    construct E_1 = E_1(n)
    n_1 = hom(E_1, G)
    n'_1 = hom(E_1, H)
    if n_1 ≠ n'_1
        return "no"
    else
        construct E_2 = E_2(n)
        n_2 = hom(E_2, G)
        n'_2 = hom(E_2, H)
        if n_2 ≠ n'_2
            return "no"
        else
            return "yes"
```

They also show that it is, in general, optimal in terms of the number of queries made to the oracle – at most three for each input graph – for a procedure of the problem `GRAPH-ISOMORPHISM` in which all the queries take the form $\hom(\mathbf{D}, *)$, where $\mathbf{D}$ is a graph that depends on the values of the previous queries (i.e., the queries $\hom(\mathbf{D}, *)$ are *adaptive*) and has a predetermined construction. Furthermore, they show that no procedure that makes a bounded number of adaptive queries of the dual form $\hom(*, \mathbf{D})$ in which $\mathbf{D}$ has a predetermined construction is a match in regard to `GRAPH-ISOMORPHISM`.

However, if we allow a mix of queries of either form in a procedure, then we obtain one that makes at most two adaptive queries to the oracle for each input graph. In the following, we state and prove a result from [51].

Theorem 5.38. *There is a construction of a graph* $\mathbf{F} = \mathbf{F}(n)$ *as a function of* $n \in \mathbb{Z}^+$ *such that for all graphs* $\mathbf{G}$ *and* $\mathbf{H}$ *of size* n*, they are isomorphic if and only if* $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$.

This gives rise to a procedure shown in Figure 5.4 for `GRAPH-ISOMORPHISM` that is already optimal in terms of the number of queries made to the oracle for each input graph.

Remark 5.6. The devil is in the details: The constructions of $\mathbf{E}_1(n)$ and $\mathbf{E}_2(n)$ collectively and of $\mathbf{F}(n)$ in the two aforementioned procedures both have a super-exponential size in n on the one hand, and computing homomorphism counts over N is #P-complete in general [17] on the other. Therefore, they do not, by any means, settle the complexity status of `GRAPH-ISOMORPHISM`, which is only known to be in NP.

Proof of Theorem 5.38. Recall that, for $n \in \mathbb{Z}^+$, we write $\bigoplus_n \mathbf{G}$ for the direct sum of n isomorphic copies of the graph $\mathbf{G}$.

Figure 5.4: Pseudocode for Graph-Isomorphism-Revised.

```
Graph-Isomorphism-Revised(G, H)
n = hom(I_1, G)
n' = hom(I_1, H)
if n ≠ n'
    return "no"
else
    construct F = F(n)
    m = hom(G, F)
    m' = hom(H, F)
    if m ≠ m'
        return "no"
    else
        return "yes"
```

First, observe that given $D \in \mathbb{Z}^+$, every sequence $(a_0, \ldots, a_t)$ of fixed size $t+1$ in which $a_i \in \{0, \ldots, D-1\}$ can be encoded by a unique integer $a_0 \times D^0 + \cdots + a_t \times D^t$.

Now, let $n \in \mathbb{Z}^+$ be given, and let $\mathbf{F}_1, \ldots, \mathbf{F}_s$ enumerate the respective representatives of the isomorphism types of all graphs of size at most n. Our goal is to construct $\mathbf{F}$ such that, for a suitable $D \in \mathbb{Z}^+$, *certain digits in the D-ary representation of* $\hom(\mathbf{G}, \mathbf{F})$ *are* $\hom(\mathbf{G}, \mathbf{F}_1), \ldots, \hom(\mathbf{G}, \mathbf{F}_s)$*, for all graphs* $\mathbf{G}$ *of size* n. By the proof of Theorem 4.4 (for the direction from (iii) to (i)), *the counts* $\hom(\mathbf{G}, \mathbf{F}_1), \ldots, \hom(\mathbf{G}, \mathbf{F}_s)$ *are sufficient to* *determine the isomorphism type of* $\mathbf{G}$ *among all graphs of size* n.

For this purpose, put

$$\mathbf{F} := \bigoplus_{j=1}^{s} \left(\bigoplus\nolimits_{D^{e_j}} \mathbf{F}_j \right),$$

where $e_1, \dots, e_s, D \in \mathbb{Z}^+$ will be determined later. We write $\mathcal{E}^+_{\leq n}$ for the set of all integers that are the sum of at most n (not necessarily distinct) integers from $\{e_1, \dots, e_s\}$. By Proposition 3.14(b) and (c), it follows that if $\mathbf{G} = \mathbf{G}_1 \oplus \cdots \oplus \mathbf{G}_r$ is an arbitrary graph of size n with the r connected components $\mathbf{G}_1, \dots, \mathbf{G}_r$, where $r \in [n]$, then

$$\begin{aligned} & \hom(\mathbf{G}, \mathbf{F}) \\ = & \prod_{i=1}^{r} \hom(\mathbf{G}_i, \mathbf{F}) \\ = & \prod_{i=1}^{r} \sum_{j=1}^{s} \hom(\mathbf{G}_i, \mathbf{F}_j) \times D^{e_j} \qquad\qquad (+) \\ = & \sum_{e \in \mathcal{E}^+_{\leq n}} \left(\sum_{\substack{j_1, \dots, j_r \in [s] \text{ with} \\ e_{j_1} + \cdots + e_{j_r} = e}} (\hom(\mathbf{G}_1, \mathbf{F}_{j_1}) \times \cdots \times \hom(\mathbf{G}_r, \mathbf{F}_{j_r})) \right) \times D^e. \end{aligned}$$

The two properties will be desirable:

(1) The outer summation on the last line of (+) is the D-ary representation of $\hom(\mathbf{G}, \mathbf{F})$.

(2) For all $j \in [s]$, the digit for the D^{re_j}-place in this D-ary representation of $\hom(\mathbf{G}, \mathbf{F})$ is $\hom(\mathbf{G}_1, \mathbf{F}_j) \times \cdots \times \hom(\mathbf{G}_r, \mathbf{F}_j) = \hom(\mathbf{G}, \mathbf{F}_j)$.

In what follows, we determine the values of $e_1, \dots, e_s$ and D to achieve the two desirable properties, based on the (arbitrary) graph $\mathbf{G} = \mathbf{G}_1 \oplus \cdots \oplus \mathbf{G}_r$ of size n mentioned previously.

Choose

$$\begin{aligned} e_1 &:= 1, \\ e_{j+1} &:= n^j + n^{j-1} + \cdots + 1 \quad \text{for all } j \in [s-1]. \end{aligned}$$

It follows that $e_{j+1} > ne_j$ for all $j \in [s-1]$ and that $e_1, \ldots, e_s$ is a strictly increasing sequence. Hence, it is easy to show that every integer $e \in \mathcal{E}^+_{\leq n}$ can be expressed as a unique summation of at most n (not necessarily distinct) integers from $\{e_1, \ldots, e_s\}$ up to permutation of the summands.[2] In particular, for every $j \in [s]$, we have $re_j \in \mathcal{E}^+_{\leq n}$ and the only such summation for re_j is $\underbrace{e_j + \cdots + e_j}_{r\text{-times}}$. Thus, property (2) holds when property (1) also holds, that is, when D is sufficiently large.

To determine the value of D, we derive an upper bound on the inner summation on the last line of (+) for arbitrary $e \in \mathcal{E}^+_{\leq n}$:

$$\begin{aligned} & \sum_{\substack{j_1, \ldots, j_r \in [s] \text{ with} \\ e_{j_1} + \cdots + e_{j_r} = e}} (\hom(\mathbf{G}_1, \mathbf{F}_{j_1}) \times \cdots \times \hom(\mathbf{G}_r, \mathbf{F}_{j_r})) \\ \leq & \sum_{\substack{j_1, \ldots, j_r \in [s] \text{ with} \\ e_{j_1} + \cdots + e_{j_r} = e}} (\hom(\mathbf{G}_1, \mathbf{K}_n) \times \cdots \times \hom(\mathbf{G}_r, \mathbf{K}_n)) \\ = & \sum_{\substack{j_1, \ldots, j_r \in [s] \text{ with} \\ e_{j_1} + \cdots + e_{j_r} = e}} \hom(\mathbf{G}, \mathbf{K}_n) \\ \leq & \sum_{\substack{j_1, \ldots, j_r \in [s] \text{ with} \\ e_{j_1} + \cdots + e_{j_r} = e}} n^n \quad \leq \quad r! \times n^n \quad \leq \quad n!n^n, \end{aligned}$$

where the second last inequality follows from the aforementioned fact about unique summation up to permutation of summands. Set $D := n!n^n + 1$, then property (1) holds.

Finally, it remains to argue that, given two arbitrary graphs $\mathbf{G}$ and $\mathbf{H}$ of size n, they are isomorphic if and only if $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$. The "only if" direction is

[2] For example, the two summations $1+2+2+3+3+3$ and $2+1+3+3+2+3$ are the same up to permutation of the summands.

trivial. Before we deal with the "if" direction, let us notice that $\hom(\mathbf{G}, \mathbf{F}) > 0$ because $\hom(\mathbf{G}, \mathbf{G}) > 0$ and $\mathbf{G}$ is among $\mathbf{F}_1, \ldots, \mathbf{F}_s$, which are subgraphs of $\mathbf{F}$; moreover, the exponents of D for the nonzero digits in the D-ary representation of $\hom(\mathbf{G}, \mathbf{F})$ indicate the number of connected components there are in $\mathbf{G}$, by property (1) and the aforementioned fact about unique summation up to permutation of summands (see the last line of (+)). The same also holds for $\mathbf{H}$. Now, for the "if" direction, assume that $\hom(\mathbf{G}, \mathbf{F}) = \hom(\mathbf{H}, \mathbf{F})$, then they are identical when expressed in D-ary representation and hence agree on all digits. In particular, they have the same number of connected components. Furthermore, by property (2), we have $\hom(\mathbf{G}, \mathbf{F}_j) = \hom(\mathbf{H}, \mathbf{F}_j)$ for all $j \in [s]$. Thus, by an earlier discussion, $\mathbf{G}$ and $\mathbf{H}$ are isomorphic. $\square$

Chapter 6

Concluding Remarks

Logic and homomorphism counts each provide a framework to study various computational problems. The investigation of the expressive power of the framework provided by different sorts of logic formalisms began as early as in the early 20th century, and that in the scope of finite structures, as some researchers suggested (see, e.g., [39]), was initiated in 1950 due to the seminal theorem by B. Trakhtenbrot [48] that the set of first-order sentences satisfied by *some* finite structure is not decidable or, equivalently, that the set of first-order sentences satisfied by *all* finite structures is not recursively enumerable. These researchers maintain that Trakhtenbrot's Theorem marked the inception of the now full-fledged field known as finite model theory. It is noteworthy that Fagin's Theorem [22] is a landmark result, which states that the decision problems in NP correspond exactly to the classes $\mathrm{Mod}(\varphi)$ where φ is a sentence of *existential second-order logic* – a logic formalism that extends first-order logic with *relation variables* (for relations over the domain of a structure or over the vertex set of a graph) which, when used in a sentence, must be existentially quantified and precede all quantifications of the (ordinary) variables in the prenex normal form of that sentence – in the vocabulary of the input structures of the corresponding decision problem; in other words, "NP coincides with existential second-order logic." In contrast, the investigation of the expressive power of the framework provided by homomorphism counts has only begun in recent decades.

In this dissertation, we consider both frameworks and focus on the latter, carrying out an investigation of its expressive power in (1) capturing various equivalence relations over graphs that are relaxations of isomorphism in Chapter 4, and (2) query algorithms over two common semirings for testing properties of graphs or structures in Chapter 5.

A different aspect in which to investigate the expressive power of homomorphism counts is graph parameter. Recall in Chapter 2 that a graph parameter is a function that maps a graph to a (real) number and that is invariant under isomorphism, and in Chapter 3 we saw that the number of n-colorings (obviously a graph parameter) of a graph $\mathbf{G}$ coincides with the homomorphism count $\hom(\mathbf{G}, \mathbf{K}_n)$ (see Example 3.17). In [26], a necessary and sufficient condition is given of when a graph parameter $f : \mathcal{G} \to \mathbb{R}$ can be expressed in the form $\hom(*, \mathbf{H})$ for a fixed *weighted graph* $\mathbf{H}$, i.e., for every graph $\mathbf{G}$, we have $f(\mathbf{G}) = \hom(\mathbf{G}, \mathbf{H})$. It is natural to ask a dual question.

Future Work 1. Find a necessary and sufficient condition for when a graph parameter $f : \mathcal{G} \to \mathbb{R}$ can be expressed in the form $\hom(\mathbf{H}, *)$ for some fixed (weighted) graph $\mathbf{H}$, i.e., for every graph $\mathbf{G}$, we have $f(\mathbf{G}) = \hom(\mathbf{H}, \mathbf{G})$.

Undoubtedly, it will require a great amount of knowledge and a combination of mathematical tools developed in various fields to achieve these goals, as exemplified in [41].

In Chapter 4, we saw that many results concerning the (non)existence of a characterization in restricted profile of various relaxations of graph isomorphism are obtained by a tailor-made argument, while only a few – C^k-equivalence, C_k-equivalence, cospectrality, to be precise – follow from a higher-level *metatheorem* (Theorem 4.17), which, amounts to giving a necessary condition for the existence of a characterization of a relaxation of graph isomorphism in restricted right profile over $\mathbb{N}$.

More recently, a necessary condition for L-equivalence over graphs able to be characterized in restricted left profile over $\mathbb{N}$ for certain logics L is given in [47]. Inspired by this and Theorem 4.17, we propose the following.

Future Work 2. Find a necessary and sufficient condition for when a relaxation of graph isomorphism can be characterized in restricted left profile, or in restricted right profile, over a semiring $\mathbb{B}$ or $\mathbb{N}$.

We saw in Chapter 5 (cf. Proposition 5.9) that for every structure (or graph) $\mathbf{A}$, the class $\mathrm{CQ}(\mathbf{A})$ of structures (or graphs, respectively) that admit a homomorphism to $\mathbf{A}$ coincides with the model class $\mathrm{Mod}(\varphi^{\mathbf{A}})$, where $\varphi^{\mathbf{A}}$ is the canonical sentence for $\mathbf{A}$, a PP-sentence that describes $\mathbf{A}$ in a "positive" way, i.e., with (existentially quantified) variables representing the elements in the domain of $\mathbf{A}$ the sentence $\varphi^{\mathbf{A}}$ asserts the presence of the tuples in the relations in $\mathbf{A}$ but not the absence of the tuples not in them; conversely, for every PP-sentence φ, the model class $\mathrm{Mod}(\varphi)$ coincides with the class $\mathrm{CQ}(\mathbf{A}^{\varphi})$ of structures (or graphs, respectively) that admit a homomorphism to $\mathbf{A}$.

It is natural to ask whether there exists a logic formalism L such that

(1) for every structure (or graph) $\mathbf{A}$, there exists an L-sentence $\psi^{\mathbf{A}}$ that depends on $\mathbf{A}$

such that CSP($\mathbf{A}$) coincides with Mod($\psi^{\mathbf{A}}$), and

(2) for every L-sentence ψ, there exists a structure (or graph, respectively) $\mathbf{A}^{\psi}$ that depends on ψ such that Mod(ψ) coincides with CSP($\mathbf{A}^{\psi}$).

For simplicity, we assume $\mathbf{A}$ is a structure in the rest of this paragraph. Intuitively, to derive such an L we would use *set variables* (for relations over the domain of a structure) and let $\psi^{\mathbf{A}}$ describe $\mathbf{A}$ in a "negative" way, i.e., with (existentially quantified) set variables representing the elements in the domain of $\mathbf{A}$ the L-sentence $\psi^{\mathbf{A}}$ asserts the absence of the tuples in the relations in $\mathbf{A}$ but not the presence of the tuples in them: The idea is that, for all structures $\mathbf{B}$, a function $h : \mathrm{dom}(\mathbf{B}) \to \mathrm{dom}(\mathbf{A})$ is a homomorphism if and only if for all r-ary relation symbols R and all elements $a_1, \ldots, a_r \in \mathrm{dom}(\mathbf{A})$ with $(a_1, \ldots, a_r) \notin R^{\mathbf{A}}$, there are no $b_1 \in h^{-1}(a_1), \ldots, b_r \in h^{-1}(a_r)$ such that $(b_1, \ldots, b_r) \in R^{\mathbf{B}}$. By extending first-order logic with set variables which, when used in a sentence, must be existentially quantified and precede all quantifications of the (ordinary) variables in the prenex normal form of that sentence, we obtain *monadic existential second-order logic*, where *monadic* refers to the fact that in this logic all relation variables have arity 1 (thus are set variables since sets are unary relations). Recall Fagin's Theorem earlier that NP coincides with existential second-order logic, T. Feder and M. Vardi introduced in [23] the logic *monotone monadic strict* NP *without inequalities* (abbreviated as MMSNP), a fragment of existential second-order logic (NP), in the quest of a large syntactically defined subclass of NP that exhibits a dichotomy – every decision problem in this subclass either is in P or is NP-complete, in view of Schaefer's Theorem [46]. It turns out that for $L =$ MMSNP, the above (1) holds. However, it is known that (2) does not necessarily hold [23, 43]. In [42], F. Madelaine and

I. Stewart give a characterization of when (2) holds.

We studied in Chapter 5 classes of graphs or of structures as to whether they admit a query algorithm, left or right, over the two common semirings, B and N. In [11], Y. Chen, J. Flum, M. Liu, and Z. Xun examine this for classes of graphs and over N, and they give several sufficient conditions of when a class (of graphs) of a particular type admit a left or a right query algorithm (over N) and present some concrete examples of classes that do not. In contrast, we have a characterization (i.e., a sufficient and necessary condition) of when a class of structures admits a left query algorithm over B in Theorem 5.22; in particular, a class admits such a query algorithm if and only if it is definable by a Boolean combination of PP-sentences (equivalence between (i) and (iii)), if and only if it is FO-definable and closed under homomorphic equivalence (equivalence between (i) and (iv)). It is yet unknown, however, what an analogous characterization would be for the remaining cases, and we propose the following future direction of research.

Future Work 3. Find a characterization of when a class of structures admits a left query algorithm over N and a characterization of when a class of graphs admits a left query algorithm over B or over N.

Since a (left) query algorithm over N differs from one over B in such a way that the homomorphism-count queries are evaluated for the actual number, rather than the existence, of homomorphisms, it is conceivable that a characterization of classes of structures admitting a left query algorithm over N in a form analogous to Theorem 5.22 might be obtained by conducting similar arguments, considering

- a generalization CQ_S of CQ classes parameterized by subsets $S \subseteq \mathbb{N}$ such that

$$\mathrm{CQ}_S(\mathbf{A}) := \{\mathbf{B} \mid \hom(\mathbf{A}, \mathbf{B}) \in S\}$$

  to establish the corresponding equivalence between (i) and (ii), and

- a suitable fragment of the counting logic C that extends PP with the counting mechanism to establish the corresponding equivalence between (ii) and (iii),

where the corresponding statement (iv) becomes irrelevant in this regard. As for classes of graphs, a key step in deriving an analogous characterization of classes of graphs admitting a left query algorithm over B would be to examine whether the relevant results in [45] hold in the context of graphs as well, whereas an analogous characterization of classes of graphs admitting a left query algorithm over N may be obtained following similar considerations as the above for classes of graphs.

While we expanded the characterization in Proposition 5.7(a) to one in Theorem 5.22, we did not expand its counterpart in Proposition 5.7(b) in any way, nor did we consider characterizations of classes of graphs or of structures as to whether they admit a right query algorithm over N. Acknowledging this, we propose the next future direction of research.

Future Work 4. Expand the characterization of classes of graphs or of structures admitting a right query algorithm over B. Find a characterization of classes of graphs or of structures admitting a right query algorithm over N.

In particular, we saw in Proposition 5.9 that CQ classes of structures correspond to model classes of PP-sentences (except for structures whose relations are all empty) and in Theorem 5.22 that CQ classes are involved in Boolean combinations for those classes that admit a left query algorithm over B (equivalence between (i) and (ii)). Moreover, in a previous discussion, we saw, symmetrically, that CSP classes correspond to model classes of certain MMSNP-sentences. Therefore, an equivalence analogous to that between (i) and (iii) in Theorem 5.22 can be immediately obtained. Besides, it is reasonable to assume, for some logic L based on MMSNP, that a class of structures admits a right query algorithm over B if and only if it is L-definable and closed under homomorphic equivalence, in an analogous form to (iv) in Theorem 5.22. As for characterizations concerning right query algorithms over N, it is plausible by considering, analogously,

- a generalization CSP_S of CSP classes parameterized by subsets $S \subseteq \mathbb{N}$ such that

  $$\mathrm{CSP}_S(\mathbf{A}) := \{\mathbf{B} \mid \hom(\mathbf{B}, \mathbf{A}) \in S\}$$

  for equivalence analogous to that between (i) and (ii) in Theorem 5.22, and

- a suitable variant of the logic MMSNP that incorporates certain counting mechanism for equivalence analogous to that between (ii) and (iii) in Theorem 5.22.

Finally, we saw in Theorem 5.24 (and Corollary 5.25) that homomorphism counts over N, while carrying more information than their counterparts over B, do not have more advantages when it comes to left query algorithms for classes of structures that are closed under homomorphic equivalence. It is natural to ask a dual question for right query algorithms (for classes of structures).

Future Work 5. Determine whether for every class $\mathcal{D}$ of structures closed under homomorphic equivalence, it admits a right query algorithm over B of the form $(\mathcal{F}, X)$ if and only if it admits a right query algorithm over N of the form $(\mathcal{F}', X')$.

We saw an abundance of evidence in Chapters 4 and 5 that left profiles and right profiles behave very differently, although they both give a characterization of isomorphism (Theorems 4.1 and 4.2). It is for certain that a clearer picture can be obtained by resolving the above open problems.

It is my sincere hope that this dissertation will help foster the appreciation of finite model theory and homomorphism counts and invite more contributions to the development of these beautiful fields.

Bibliography

[1] Albert Atserias. On digraph coloring problems and treewidth duality. *European Journal of Combinatorics*, 29(4):796–820, 2008.

[2] Albert Atserias, Andrei A. Bulatov, and Anuj Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

[3] 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.

[4] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[5] Meghyn Bienvenu, Balder Ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, csp, and mmsnp. *ACM Transactions on Database Systems (TODS)*, 39(4):1–44, 2014.

[6] Andrei A. Bulatov. H-coloring dichotomy revisited. *Theoretical Computer Science*, 349(1):31–39, 2005.

[7] 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.

[8] Balder ten Cate, Víctor Dalmau, Phokion G. Kolaitis, and Wei-Lin Wu. When do homomorphism counts help in query algorithms? *arXiv preprint arXiv:2304.06294*, 2023.

[9] Surajit Chaudhuri and Moshe Y. Vardi. Optimization of *Real* conjunctive queries. In Catriel Beeri, editor, *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 25-28, 1993, Washington, DC, USA*, pages 59–70. ACM Press, 1993.

[10] Yijia Chen, Jörg Flum, Mingjun Liu, and Zhiyang Xun. On queries determined by a constant number of homomorphism counts. *arXiv preprint arXiv:2111.13269*, 2021.

[11] Yijia Chen, Jörg Flum, Mingjun Liu, and Zhiyang Xun. On algorithms based on finitely many homomorphism counts. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[12] Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *International Conference on Principles and Practice of Constraint Programming*, pages 310–326. Springer, 2002.

[13] Anuj Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143(2):154–174, 1998.

[14] Anuj Dawar, Simone Severini, and Octavio Zapata. Pebble games and cospectral graphs. *Electronic Notes in Discrete Mathematics*, 61:323–329, 2017.

[15] 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, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 40:1–40:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

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

[17] Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures & Algorithms*, 17(3-4):260–289, 2000.

[18] Heinz-Dieter Ebbinghaus and Jörg Flum. Finite model theory. *Springer Monographs in Mathematics. Springer-Verlag, Berlin*, 2006.

[19] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic*. Springer, 2nd edition, 1995.

[20] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[21] Paul Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[22] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of computation*, 7:43–73, 1974.

[23] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone nonadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[24] Jan Foniok, Jaroslav Nešetřil, and Claude Tardif. Generalised dualities and maximal finite antichains in the homomorphism order of relational structures. *European Journal of Combinatorics*, 29(4):881–899, 2008.

[25] Roland Fraïssé. Sur quelques classifications des systèmes de relations. *Publications Scientifiques de l'Université d'Alger*, Series A, 1:35–182, 1954.

[26] Michael Freedman, László Lovász, and Alexander Schrijver. Reflection positivity, rank connectivity, and homomorphism of graphs. *Journal of the American Mathematical Society*, 20(1):37–51, 2007.

[27] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007.

[28] Martin Grohe. *Descriptive complexity, canonisation, and definable graph structure theory*, volume 47. Cambridge University Press, 2017.

[29] Martin Grohe. Counting bounded tree depth homomorphisms. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual*

ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020, pages 507–520. ACM, 2020.

[30] Martin Grohe and Julian Mariño. Definability and descriptive complexity on databases of bounded tree-width. In Catriel Beeri and Peter Buneman, editors, *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer, 1999.

[31] Martin Grohe and Daniel Neuen. Recent advances on the graph isomorphism problem. *arXiv preprint arXiv:2011.01366*, 2020.

[32] Martin Grohe and Pascal Schweitzer. The graph isomorphism problem. *Communications of the ACM*, 63(11):128–134, 2020.

[33] Pavol Hell and Jaroslav Nešetřil. *Graphs and homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and its Applications.* Oxford University Press, 2004.

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

[35] Neil Immerman. *Descriptive complexity.* Springer Science & Business Media, 2012.

[36] Phokion G Kolaitis and Moshe Y Vardi. Conjunctive-query containment and constraint satisfaction. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 205–213, 1998.

[37] Gábor Kun. Constraints, mmsnp and expander relational structures. *Combinatorica*, 33(3):335–347, 2013.

[38] Benoit Larose, Cynthia Loten, and Claude Tardif. A characterisation of first-order constraint satisfaction problems. *Logical Methods in Computer Science*, 3, 2007.

[39] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

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

[41] László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012.

[42] Florent Madelaine and Iain A Stewart. Constraint satisfaction, logic and forbidden patterns. *SIAM Journal on Computing*, 37(1):132–163, 2007.

[43] Florent R. Madelaine and Iain A. Stewart. Some problems not definable using structure homomorphisms. *Ars Combinatoria*, 67:153–160, 2003.

[44] Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Cambridge University Press, 2017.

[45] Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM (JACM)*, 55(3):1–53, 2008.

[46] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978.

[47] Tim Seppelt. Logical equivalences, homomorphism indistinguishability, and forbidden minors. *arXiv preprint arXiv:2302.11290*, 2023.

[48] Boris A. Trakhtenbrot. The impossibility of an algorithm for the decision problem for finite domains (russian). *Doklady Akademii Nauk SSSR*, 70:569–572, 1950.

[49] Edwin R Van Dam and Willem H Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272, 2003.

[50] Boris Weisfeiler. *On construction and identification of graphs*, volume 558. Springer, 2006.

[51] Wei-Lin Wu. Query algorithms based on homomorphism counts. In *Structure Meets Power Workshop (Contributed Talks)*, page 24, 2022.

Appendix A

The Definable H-Coloring Dichotomy Theorem

The logic programming language *Datalog* allows us to write recursive queries and, when viewed as a logic formalism, is denoted DL. We will not go into the details of DL, and an interested reader can consult any reference on finite model theory, e.g., [18, 27, 35, 39].

Definition A.1. Let σ be a vocabulary and let $n \in \mathbb{Z}^+$. The fragment of Datalog σ-formulas φ such that the variables, free or bound, appearing in φ are among $z_0, \ldots, z_{n-1}$ is denoted $\mathrm{DL}^n[\sigma]$, or DL^n when the vocabulary σ is understood from the context.

Our goal in this appendix is to prove the following theorem (Theorem 11 from [3]). Recall the logic formalism $\mathrm{C}^n_{\infty\omega}$ (where $n \in \mathbb{Z}^+$) mentioned in Remark 3.8, and we let

$$\mathrm{C}^\omega_{\infty\omega} := \bigcup_{n \in \mathbb{Z}^+} \mathrm{C}^n_{\infty\omega}.$$

Theorem A.1 (Definable **H**-Coloring Dichotomy Theorem)**.** *Let* **H** *be a graph. If* **H** *is* 2*-colorable, then* CSP(**H**) = Mod($\neg\varphi$) *for some* DL^4*-sentence* φ*. Otherwise,* CSP(**H**) $\neq$ Mod(ψ) *for any* $\mathrm{C}^{\omega}_{\infty\omega}$*-sentence* ψ*.*

The discussion in the sequel through the end of this appendix will lead to a *proof* of Theorem A.1.

Let **H** be an arbitrary graph. We distinguish three cases below.

Case 1. **H** *has chromatic number* $\chi(\mathbf{H}) = 1$*.* Then, $\mathbf{H} \in \mathcal{I}$ is an independent set. Clearly, CSP(**H**) $= \mathcal{I} =$ Mod($\neg\varphi$), where $\varphi \in \mathrm{DL}^2$ is obtained from a Datalog program that contains exactly one clause in which the head is a 0-ary predicate and the body is $E(x, y)$.

Case 2. **H** *has chromatic number* $\chi(\mathbf{H}) = 2$*.* Then, **H** contains an edge and is 2-colorable. Thus, CSP(**H**) is the class of 2-colorable graphs, which are exactly those graphs that contain no cycle of odd length ≥ 3 (by Proposition 2.4). The existence of a cycle of odd length ≥ 3 can be expressed as a Datalog program (see, e.g., Example 2.6.20 in [27]) that uses 4 variables. Let φ be the corresponding DL^4-sentence, then it follows that CSP(**H**) = Mod($\neg\varphi$).

Case 3. **H** *is not* 2*-colorable.* We need to show that CSP(**H**) is not $\mathrm{C}^{\omega}_{\infty\omega}$-definable. This requires significantly more work, which we do in the subsequent steps.

Step 1: Transition from CSP Classes of Graphs to CSP Classes of the Corresponding σ(G)-Structures

First note that for every graph **G**, we have $\{\mathbf{F}^{\sigma(\mathrm{G})} \mid \mathbf{F} \in \mathrm{CSP}(\mathbf{G})\} \subseteq \mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$ and that the equality holds if and only if **G** is an independent set.

Example A.1. Consider $\mathbf{K}_2$. The $\sigma(\mathrm{G})$-structures in $\mathrm{CSP}(\mathbf{K}_2^{\sigma(\mathrm{G})})$ all satisfy the *axioms of graph theory* (see Example 3.1):

$$\begin{aligned}
\varphi_{\text{irreflx}} &:= \forall x \neg E(x,x), \\
\varphi_{\text{sym}} &:= \forall x \forall y (E(x,y) \to E(y,x)).
\end{aligned}$$

However, for the $\sigma(\mathrm{G})$-structure $\mathbf{K}$ with domain $\mathrm{dom}(\mathbf{K}) = \{v_1, v_2\}$ and relation $E^{\mathbf{K}} = \{(v_1, v_2)\}$, we have $\mathbf{K} \not\models \varphi_{\text{sym}}$ but $\mathbf{K} \in \mathrm{CSP}(\mathbf{K}_2^{\sigma(\mathrm{G})})$.

Nevertheless, as we will see shortly, for every graph $\mathbf{G}$, it holds that $\mathrm{CSP}(\mathbf{G})$ is $\mathrm{C}^{\omega}_{\infty\omega}$-definable if and only if $\mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$ is $\mathrm{C}^{\omega}_{\infty\omega}$-definable. That is to say, $\mathrm{C}^{\omega}_{\infty\omega}$-definability is preserved for transition from CSP classes of graphs to CSP classes of the corresponding $\sigma(\mathrm{G})$-structures (and back).

Definition A.2. Let $\mathbf{G}$ be a graph. For every $\sigma(\mathrm{G})$-structure $\mathbf{A}$, we say that $\mathbf{A}$ is an *orientation of* $\mathbf{G}$ if

- $\mathrm{dom}(\mathbf{A}) = \mathrm{dom}(\mathbf{G}^{\sigma(\mathrm{G})})$, and
- for all $a, b \in \mathrm{dom}(\mathbf{A})$, we have $(a,b) \in E^{\mathbf{A}}$ or $(b,a) \in E^{\mathbf{A}}$ if and only if $(a,b) \in E^{\mathbf{G}^{\sigma(\mathrm{G})}}$.

Note that in particular, for every graph $\mathbf{G}$, the corresponding $\sigma(\mathrm{G})$-structure $\mathbf{G}^{\sigma(\mathrm{G})}$ is an orientation of $\mathbf{G}$. Moreover, the $\sigma(\mathrm{G})$-structure $\mathbf{K}$ considered in Example A.1 is an orientation of $\mathbf{K}_2$. We restate Remark 3.11 as the next lemma.

Lemma A.2. *Let* $\mathbf{G}$ *be a graph. For every graph* $\mathbf{F}$*, we have that* $\mathbf{F} \in \mathrm{CSP}(\mathbf{G})$ *if and only if* $\mathbf{F}^{\sigma(\mathrm{G})} \in \mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$. □

Definition A.3. (a) For every $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$-formula φ, the $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$-formula φ^+ is defined inductively as follows:

$$
\begin{aligned}
(x = y)^+ &:= x = y,\\
(E(x,y))^+ &:= (E(x,y) \vee E(y,x)),\\
(\neg\varphi)^+ &:= \neg\varphi^+,\\
(\varphi \vee \psi)^+ &:= (\varphi^+ \vee \psi^+),\\
(\exists x\varphi)^+ &:= \exists x\varphi^+,\\
(\exists^{\geq n} x\varphi)^+ &:= \exists^{\geq n} x\varphi^+,\\
(\bigvee \Phi)^+ &:= \bigvee \{\varphi^+ \mid \varphi \in \Phi\}.
\end{aligned}
$$

(b) For every $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$-formula φ, let $\varphi^* := (\varphi_{\mathrm{irreflx}} \wedge \varphi^+)$.

Obviously, if φ is an $\mathrm{C}^{\omega}_{\infty\omega}[\sigma(\mathrm{G})]$-formula, then so are φ^+ and φ^*. The next lemma summarizes simple yet useful properties of the $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$-formulas φ^+.

Lemma A.3. *Let* $\mathbf{F}$ *be a graph, and let* φ *be an* $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$*-sentence. The following hold:*

(a) $\mathbf{F} \models \varphi$ *if and only if* $\mathbf{F} \models \varphi^+$.

(b) *For every orientation* $\mathbf{A}$ *of* $\mathbf{F}$*, we have* $\mathbf{F} \models \varphi^+$ *if and only if* $\mathbf{A} \models \varphi^+$.

Proof. For both parts, prove by induction on φ the more general statement for $\mathrm{C}_{\infty\omega}[\sigma(\mathrm{G})]$-*formulas* φ together with variable assignments for $\mathbf{F}$. □

Theorem A.4. *For every graph* $\mathbf{G}$*, we have that* $\mathrm{CSP}(\mathbf{G})$ *is* $\mathrm{C}^{\omega}_{\infty\omega}$*-definable if and only if* $\mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$ *is* $\mathrm{C}^{\omega}_{\infty\omega}$*-definable.*

Proof. For the "if" direction, let φ be an $\mathrm{C}^{\omega}_{\infty\omega}[\sigma(\mathrm{G})]$-sentence such that $\mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})}) = \mathrm{Mod}(\varphi)$. Then, for every graph $\mathbf{F}$, we have

$\mathbf{F} \in \mathrm{CSP}(\mathbf{G})$

iff $\mathbf{F}^{\sigma(\mathrm{G})} \in \mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$ (by Lemma A.2)

iff $\mathbf{F}^{\sigma(\mathrm{G})} \models \varphi$ (by assumption)

iff $\mathbf{F} \models \varphi$ (by Definition 3.11).

It follows that $\mathrm{CSP}(\mathbf{G}) = \mathrm{Mod}(\varphi)$.

For the "only if" direction, let ψ be an $\mathrm{C}^{\omega}_{\infty\omega}[\sigma(\mathrm{G})]$-sentence such that $\mathrm{CSP}(\mathbf{G}) = \mathrm{Mod}(\psi)$. Then, for every $\sigma(\mathrm{G})$-structure $\mathbf{A}$, we have

$\mathbf{A} \in \mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$

iff there is a graph $\mathbf{F}$ such that $\mathbf{A}$ is an orientation of $\mathbf{F}$ and $\mathbf{F}^{\sigma(\mathrm{G})} \in \mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})})$

iff there is a graph $\mathbf{F}$ such that $\mathbf{A}$ is an orientation of $\mathbf{F}$ and $\mathbf{F} \in \mathrm{CSP}(\mathbf{G})$

(by Lemma A.2)

iff there is a graph $\mathbf{F}$ such that $\mathbf{A}$ is an orientation of $\mathbf{F}$ and $\mathbf{F} \models \psi$

(by assumption)

iff there is a graph $\mathbf{F}$ such that $\mathbf{A}$ is an orientation of $\mathbf{F}$ and $\mathbf{F} \models \psi^+$

(by Lemma A.3(a))

iff there is a graph $\mathbf{F}$ such that $\mathbf{A}$ is an orientation of $\mathbf{F}$ and $\mathbf{A} \models \psi^+$

(by Lemma A.3(b))

iff $\mathbf{A} \models \varphi_{\mathrm{irreflx}}$ and $\mathbf{A} \models \psi^+$

iff $\mathbf{A} \models \psi^*$ (by definition)

It follows that $\mathrm{CSP}(\mathbf{G}^{\sigma(\mathrm{G})}) = \mathrm{Mod}(\psi^*)$ □

It is known that *the class* $\mathrm{CSP}(\mathbf{K}_3)$ *of* 3*-colorable graphs is not* $\mathrm{C}^{\omega}_{\infty\omega}$*-definable* (see Theorem 4.11 and Remark 4.12 in [13]). By Theorem A.4, it follows that *the class* $\mathrm{CSP}(\mathbf{K}_3^{\sigma(\mathrm{G})})$ *of* $\sigma(\mathrm{G})$*-structures is not* $\mathrm{C}^{\omega}_{\infty\omega}$*-definable.*

For the next step, we need to bring in a partial ordering $\leq_{\mathrm{DL}}$ among classes of structures called *Datalog-reducibility* from Definition 1 in [2] (denoted $\leq_{\mathrm{datalog}}$ there). By the paragraph below Definition 1 in [2], $\mathrm{C}^{\omega}_{\infty\omega}$-definability is preserved downwards by $\leq_{\mathrm{DL}}$, i.e., for all classes $\mathcal{D}$ and $\mathcal{F}$ of structures such that $\mathcal{D} \leq_{\mathrm{DL}} \mathcal{F}$, if $\mathcal{F}$ is $\mathrm{C}^{\omega}_{\infty\omega}$-definable, then so is $\mathcal{D}$ (indeed, for every DL^n-sentence, there is a logically equivalent $\mathrm{C}^{n}_{\infty\omega}$-sentence, see Theorem 4.1 of [36]).

Our goal in the next step is to show that $\mathrm{CSP}(\mathbf{K}_3^{\sigma(\mathrm{G})}) \leq_{\mathrm{DL}} \mathrm{CSP}(\mathbf{H}^{\sigma(\mathrm{G})})$, which, together with the earlier statement that the class $\mathrm{CSP}(\mathbf{K}_3^{\sigma(\mathrm{G})})$ of $\sigma(\mathrm{G})$-structures is not $\mathrm{C}^{\omega}_{\infty\omega}$-definable, implies that $\mathrm{CSP}(\mathbf{H}^{\sigma(\mathrm{G})})$ is not $\mathrm{C}^{\omega}_{\infty\omega}$-definable. Then, it will follow that $\mathrm{CSP}(\mathbf{H})$ is not $\mathrm{C}^{\omega}_{\infty\omega}$-definable, either, by Theorem A.4.

Step 2: Prove $\mathrm{CSP}(\mathbf{K}_3^{\sigma(\mathrm{G})}) \leq_{\mathrm{DL}} \mathrm{CSP}(\mathbf{H}^{\sigma(\mathrm{G})})$

The subsequent presentation will follow closely Subsection IV.B in [3], and the terms and results apply to arbitrary vocabularies σ.

Recall the definitions of an induced substructure $\mathbf{A}[S]$ of a structure $\mathbf{A}$ by a nonempty set $S \subseteq \mathrm{dom}(\mathbf{A})$ and the quotient $\mathbf{A}/\theta$ by a partition θ of $\mathrm{dom}(\mathbf{A})$ in Section 2.3. Note that there is a one-to-one correspondence between an equivalence relation $\equiv$ over $\mathrm{dom}(\mathbf{A})$ and a partition θ of $\mathrm{dom}(\mathbf{A})$. In the sequel, we will identify an equivalence with the corresponding partition when taking the quotient of a structure by that *equivalence relation.*

Definition A.4. Let σ be a vocabulary.

(a) Let $\mathbf{A}$ be a σ-structure $\mathbf{A}$. For $a \in \mathrm{dom}(\mathbf{A})$, let P_a be a new unary relation symbol.

The *singleton-expansion of* $\mathbf{A}$, denoted $\overline{\mathbf{A}}$, is the $(\sigma \cup \{P_a \mid a \in \mathrm{dom}(\mathbf{A})\})$-structure with domain $\mathrm{dom}(\overline{\mathbf{A}}) := \mathrm{dom}(\mathbf{A})$ and relations $R^{\overline{\mathbf{A}}} := R^{\mathbf{A}}$ for all $R \in \sigma$ and $P_a^{\overline{\mathbf{A}}} := \{a\}$ for all $a \in \mathrm{dom}(\mathbf{A})$.

(b) A *primitive positive* σ*-formula with equality allowed* (abbreviated: PPE[σ]-formula) is a first-order σ-formula of the form $\exists x_1 \cdots \exists x_m \bigwedge_{i=1}^{n} \varphi_i$ in which $m, n \in \mathbb{Z}^+$ and φ_i is an atomic (equational or relational) formula for $i \in [n]$.

Definition A.5. Let σ be a vocabulary and let $\mathbf{A}$ be a σ-structure. For every r-ary relation T over $\mathrm{dom}(\mathbf{A})$, we say that T *is* PPE-*definable in* $\mathbf{A}$ if there is a PPE-formula $\varphi(x_1, \ldots, x_r)$ such that $T = \{(a_1, \ldots, a_r) \in \mathrm{dom}(\mathbf{A})^r \mid \mathbf{A} \models \varphi(a_1, \ldots, a_r)\}$.

Recall from Chapter 2 that for all $\mathbf{A}$ and $\mathbf{B}$ in $\mathcal{U}$, we write $\mathbf{A} \subseteq \mathbf{B}$ to mean that $\mathbf{A}$ is a subgraph of $\mathbf{B}$ (if $\mathcal{U} = \mathcal{G}$) or $\mathbf{A}$ is a substructure of $\mathbf{B}$ (if $\mathcal{U} = \mathcal{A}$).

Definition A.6. Let $\mathbf{A} \in \mathcal{U}$.

(a) For every $\mathbf{B} \subseteq \mathbf{A}$, we say $\mathbf{B}$ is a *core of* $\mathbf{A}$ if

- there is a homomorphism $h : \mathbf{A} \to \mathbf{B}$ where $h|_{\mathrm{dom}(\mathbf{B})}$ is the identity mapping on $\mathrm{dom}(\mathbf{B})$, and
- for every $\mathbf{C} \subseteq \mathbf{B}$, if the statement (1) holds for $\mathbf{C}$ in place of $\mathbf{B}$, then $\mathbf{B} = \mathbf{C}$.

(b) We say $\mathbf{A}$ is a *core* if it is a core of itself.

In fact, it can be shown that a core of a graph or a structure is unique up to isomorphism (see [33]), henceforth we shall say *the* core of a graph or a structure.

Obviously, for every graph $\mathbf{G}$, if $\mathbf{G}'$ is the core of $\mathbf{G}$, then $\mathrm{CSP}(\mathbf{G}) = \mathrm{CSP}(\mathbf{G}')$.

Hence, *without loss of generality, we may assume that our non-2-colorable graph* $\mathbf{H}$ *is a core.*

Theorem A.5. [6] *If* $\mathbf{G}$ *is a non-2-colorable core graph, then there are a subset* $S \subseteq V(\mathbf{G})$ *and an equivalence relation* $\equiv$ *over* S *such that*

(1) S *is* PPE-*definable in* $\overline{\mathbf{G}^{\sigma(\mathrm{G})}}$,

(2) $\equiv$ *is* PPE-*definable in* $\overline{\mathbf{G}^{\sigma(\mathrm{G})}}[S]$,

(3) $(\mathbf{G}^{\sigma(\mathrm{G})}[S])/\equiv$ *is isomorphic to* $\mathbf{K}_3^{\sigma(\mathrm{G})}$. □

Remark A.1. This theorem follows from the implication from not (b) to not (c) in Theorem 1 of [6]. Moreover, condition (2) in Theorem A.5 is implied by the proof of Theorem 1 there and is stronger than the condition that $\equiv$ is PPE-definable in $\overline{\mathbf{G}^{\sigma(\mathrm{G})}}$ in that theorem.

Theorem A.6. [2] *Let* $\mathbf{A}$ *be a structure with* $|\mathrm{dom}(\mathbf{A})| \geq 2$*, let* S *be a subset of* $\mathrm{dom}(\mathbf{A})$ *and let* $\equiv$ *be an equivalence relation over* $\mathrm{dom}(\mathbf{A})$*. The following hold:*

(a) $\mathrm{CSP}(\mathbf{A}) \leq_{\mathrm{DL}} \mathrm{CSP}(\overline{\mathbf{A}})$.

(b) *If* $\equiv$ *is* PPE-*definable in* $\mathbf{A}$*, then* $\mathrm{CSP}(\mathbf{A}/\equiv) \leq_{\mathrm{DL}} \mathrm{CSP}(\mathbf{A})$.

(c) *If* S *is* PPE-*definable in* $\mathbf{A}$*, then* $\mathrm{CSP}(\mathbf{A}[S]) \leq_{\mathrm{DL}} \mathrm{CSP}(\mathbf{A})$.

(d) *If* $\mathbf{A}$ *is a core, then* $\mathrm{CSP}(\overline{\mathbf{A}}) \leq_{\mathrm{DL}} \mathrm{CSP}(\mathbf{A})$. □

Remark A.2. Statement (a) of this theorem follows from a special case of Lemma 11 in [2], statements (b) and (c) are special cases of Theorem 18 in that paper, and statement (d) follows from the second part of Lemma 19 there.

Thus, for our non-2-colorable core graph $\mathbf{H}$, we apply Theorem A.5 to get the subset S and the equivalence relation $\equiv$. Let σ_1 be the vocabulary of $\overline{\mathbf{H}^{\sigma(\mathrm{G})}}[S]$ and σ_2 be the vocabulary of $\overline{\mathbf{H}^{\sigma(\mathrm{G})}[S]}$. Note that $\sigma_1 = \sigma_2 \cup \{P_a \mid a \notin S\}$ and that for every σ_1-structure $\mathbf{A}$, we have $\mathbf{A} \in \mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}}[S])$ if and only if

- $P_a^{\mathbf{A}} = \emptyset$ for all $a \notin S$, and
- $\mathbf{A}' \in \mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}[S]})$ for the σ_2-structure $\mathbf{A}'$ obtained from $\mathbf{A}$ by removing $P_a^{\mathbf{A}}$ for all $a \notin S$.

Thus, if $\mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}}[S])$ is definable by an $\mathrm{C}^{\omega}_{\infty\omega}[\sigma_1]$-sentence φ, then $\mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}[S]})$ is definable by the $\mathrm{C}^{\omega}_{\infty\omega}[\sigma_2]$-sentence obtained from φ by replacing, for all $a \notin S$, the atomic formulas $P_a(x)$ with $\neg x = x$. Then, we consider $\mathbf{H}^{\sigma(\mathrm{G})}$ and apply Theorem A.6, so that

$$
\begin{array}{lll}
 & \mathrm{CSP}(\mathbf{H}^{\sigma(\mathrm{G})}) & \\
\geq_{\mathrm{DL}} & \mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}}) & \text{(by Theorem A.6(d))} \\
\geq_{\mathrm{DL}} & \mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}}[S]) & \text{(by Theorem A.6(c))} \\
\geq_{\mathrm{DL}} & \mathrm{CSP}(\overline{\mathbf{H}^{\sigma(\mathrm{G})}[S]}) & \text{(by the above discussion)} \\
\geq_{\mathrm{DL}} & \mathrm{CSP}(\mathbf{H}^{\sigma(\mathrm{G})}[S]) & \text{(by Theorem A.6(a))} \\
\geq_{\mathrm{DL}} & \mathrm{CSP}((\mathbf{H}^{\sigma(\mathrm{G})}[S])/{\equiv}) & \text{(by Theorem A.6(b))} \\
= & \mathrm{CSP}(\mathbf{K}_3^{\sigma(\mathrm{G})}) & \text{(by condition (3) in Theorem A.5),}
\end{array}
$$

as desired. $\square$