Game comonads and beyond:

compositional constructions
for logic and algorithms

Adam Peter Connolly

University of Cambridge
Computer Laboratory
St. Catharine's College

October 2022

This dissertation is submitted for
the degree of Doctor of Philosophy

For Ellie, a chuisle mo chroí.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text.

It does not exceed the prescribed word limit for the relevant Degree Committee.

Game comonads and beyond:

compositional constructions for logic and algorithms

Adam Peter Connolly

Summary

Game comonads represent a rare application of category theoretic methods to the fields of finite model theory and descriptive complexity. First introduced by Abramsky, Dawar and Wang in 2017, these new constructions exposed connections between Spoiler-Duplicator games used in logic, related algorithms for constraint satisfaction and structure isomorphism, and well-known parameters such as treewidth and treedepth. The compositional framework for logical resources emerging from these comonads has proved an important tool in generalising results from finite model theory and new game comonads have been invented for a range of different logics and algorithms. However, this framework has previously been limited by its inability to express logics which are strictly stronger than those captured by Abramsky, Dawar and Wang's pebbling comonad, $\mathbb{P}_k$.

In this thesis, we show for the first time how to overcome these limitations by extending the reach of compositional techniques for logic and algorithms in a number of directions. Firstly, we deepen our understanding of the comonad $\mathbb{P}_k$, which previously captured the strongest logic of any game comonad. Doing so, we reveal new connections between the Kleisli category of $\mathbb{P}_k$ and k-variable logics extended with different forms of quantification, including limited counting quantifiers and unary generalised quantifiers.
Secondly, we show how to construct a new family of game comonads $\mathbb{H}_{n,k}$ which capture logics extended by generalised quantifiers of all arities. This construction leads to new variants of Hella's k-pebble n-bijective game, new structural parameters generalising treewidth, and new techniques for constructing game comonads.
Finally, we expand the realm of compositional methods in finite model theory beyond comonads, introducing new constructions on relational structures based on other aspects of category theory. In the first instance, we show that lifting well-known linear-algebraic monads on **Set** to the category of relational structures gives a compositional semantics to linear programming approximations of homomorphism and an elegant framework for studying these techniques. Furthermore, we use presheaves to give a new semantics for pebble games and algorithms for constraint satisfaction and structure isomorphism. Building on analogous work in quantum contextuality, we use a common invariant based on cohomology to invent efficient algorithms for approximating homomorphism and isomorphism and prove that these are far more powerful than those currently captured by game comonads.

Acknowledgments

I thank my supervisor, Anuj Dawar, for the wonderful years I have spent as his student. I am grateful for his patience, support and enthusiastic collaboration and this thesis owes a huge debt to everything he has taught me.

Throughout my doctoral research, I have had the great privilege of working closely with outstanding scientists and mathematicians. Foremost among them, Samson Abramsky, has provided guidance and wisdom throughout and I have enjoyed our close collaboration on the cohomological algorithms behind Chapter 8. The game comonads community has been a constant source of discussion and inspiration, in particular, Tomáš Jakl, Nihil Shah, Tom Paine, Yoàv Montacute, Dan Marsden, Amin Karamlou and Luca Reggio. Standa Živný and Jakub Opršal generously hosted me in Oxford and Vienna, teaching me all I know about PCSPs, and helping to shape Chapters 7 and 8. I have also benefitted greatly from the advice of Martin Hyland, Andy Pitts and Glynn Wynskel, and I reserve special thanks for Thomas Forster and Julian Holstein who encouraged me to start this PhD in the first place.

I thank EPSRC, the University of Cambridge and St. Catharine's College without whose financial support this thesis could not have been produced. The Alan Turing Institute's *PhD Enrichment Programme* was also a fantastic asset to my research and fostered the work in Chapter 8.

Doing a PhD during a global pandemic presented many challenges and I'm grateful to my friends and family who tackled these with me. I thank especially Paul and Judith Howcroft for taking me in during the early days of March 2020.

In a gloomy era of working from home, I had the immense good fortune to share mine with the brightest of stars: Oliver Rhodes, Oisín Faust, Ryan Young, Todd Gillespie and Elizabeth Howcroft. Thank you for keeping the lights on as I finished writing up.

Finally, to my parents, Sandra and Michael, my sister, Meghan and my partner (and chief proofreader), Ellie. Without you, I would never have come so far — thank you.

Contents

Chapter 1

Introduction

Mathematical logic is sometimes regarded as arcane and obscure by mathematicians (not to mention members of the general public) but it has long been recognised as an essential tool in theoretical computer science. Reflecting on logic's "*unusual effectiveness*" in the field at the end of the last century, Phokion Kolaitis [56] claimed that

> *. . . logic has permeated through computer science during the past thirty years much more than it has through mathematics during the past one hundred years.*

As evidence, his co-authors in that paper lay out examples of foundational contributions of logical methods to topics as disparate as database theory, computational complexity, semantics of programming languages and automated verification. The "*effectiveness*" is undeniable. However, the field of logic in computer science is a divided one, dominated by two main traditions which have different tools and motivations and share little by way of interaction. The first, the *logic and algorithms* tradition, uses logic as a description language. It includes applying the mathematics of finite model theory to the relationship between expressing properties of finite structures and computing them. The second, the *logic and semantics* tradition, uses logic as a language for inference. It includes applying category theory and categorical logic to understand compositional structures in programming languages and computation.

The discovery of game comonads by Abramsky, Dawar and Wang [6] in 2017 began an exciting new dialogue between these different logical traditions in theoretical computer science. Their construction, the *pebbling comonad*, showed how pebble games used in the logic and algorithms tradition to study the expressive power of k-variable logics, and the local consistency and Weisfeiler-Leman algorithms, could be recreated using comonads on the category of relational structures. Not only did this construction provide a compositional semantics for pebble games and their associated algorithms, it also unearthed new connections between different variations of the games and the well-known structural parameter treewidth. Further connections have since flourished under the *Structure and*

Power programme launched by Abramsky and Shah [11] in 2018. New game comonads have been discovered which capture other model-comparison games and structural parameters, and have allowed a general theory to emerge. A full introduction to this new field of research is provided in Chapter 3. While this theory has created new bridges between the model-theoretic and categorical traditions, there are many unexplored directions where the applicability of these methods is untested.

This thesis expands this new field by increasing the strength of the underlying logics and algorithms which can be captured by compositional semantics. In [6], Abramsky, Dawar and Wang show that the morphisms and isomorphisms of the *Kleisli category* of the pebbling comonad capture important approximations to homomorphisms and isomorphisms of relational structures. These approximations have well-known characterisations in terms of both logic and algorithms. In terms of logics, the Kleisli morphisms are related to existential-positive infinitary first-order logic with k variables, $\exists^{+}\mathcal{L}^{k}_{\infty}$ and the Kleisli isomorphisms to infinitary first-order logic with k variables and counting quantifiers, $\mathcal{C}^{k}_{\infty}$. These logics are important in descriptive complexity theory for proving upper bounds on the expressiveness of extensions of first-order logic with powers of recursion and simple counting, namely, `DATALOG` [67] and inflationary fixpoint logic with counting, FPC. In terms of algorithms, the Kleisli morphisms give a semantics to local consistency algorithms in constraint satisfaction while Kleisli isomorphisms capture the Weisfeiler-Leman algorithm for structure isomorphism. These connections are fascinating and we spend some of this thesis developing them. However, the last 30 years has seen advances in both descriptive complexity and algorithms for constraint satisfaction and structure isomorphism which greatly expand on the power of the logics and algorithms captured by the pebbling comonad. Providing a compositional semantics to these advances motivates this thesis.

Descriptive complexity theory examines the relationship between the logical resources needed to express properties of finite structures and the computational resources needed to compute them. This field of research began, in essence, with Fagin's Theorem [43] which showed that the properties expressible in *existential second-order logic* ($\exists\mathbf{SO}$) are exactly those decidable in *non-deterministic polynomial time* (`NPTIME`). This result raised an interesting question: if $\exists\mathbf{SO} = \mathtt{NPTIME}$, is there a logic which captures *polynomial time* in the same way? This question, which is still the central open problem in descriptive complexity theory, began with Chandra and Harel [29] and was formulated precisely by Gurevich [54]. The logic FPC, which is bounded by the infinitary counting logic of the pebbling comonad, is an important part of this history. However, in 1992, Cai, Fürer and Immerman [27] demonstrated a `PTIME` property on graphs which cannot be defined in this logic. Since this result, some in the field have focused on the question of what power should be added to FPC to get closer to capturing `PTIME`. This search led to the study of fixpoint logics extended with generalised quantifiers, pioneered by Hella and Kolaitis and Väänänen. A result of Dawar [32] showed that if there is a logic for `PTIME`, it can be described as FPC extended by a vectorized family of generalised quantifiers. The question of which family would suffice led to the study of generalised quantifiers inspired by linear-algebra, as well as the invention of rank logic and linear-algebraic logic. These were top candidates for capturing `PTIME` until 2021 when they were shown by Lichter [72, 36] to be insufficient. This thesis investigates whether we can use game comonads or other compositional methods to reason about these more powerful logics.

Progress in descriptive complexity theory also has close links to the study of algorithms for constraint satisfaction and structure isomorphism. Developments in these fields serve as further motivation to push compositional methods beyond the limits of the pebbling comonad. For structure isomorphism, the relationship between FPC and the well-known Weisfeiler-Leman isomorphism test is established by Cai, Fürer and Immerman [27], who use the inexpressibility result mentioned above to show that Weisfeiler-Leman is not strong enough to decide isomorphism of structures. Babai's recent breakthrough [15] in finding a quasipolynomial-time algorithm for graph isomorphism builds on a tradition of refining such tests using algebraic methods. This tradition began with Weisfeiler and Leman's original paper [90] and incorporates methods from group theory developed in the 1980s in the work of Luks, Babai and others [75, 16]. For constraint satisfaction problems, recent progress has seen equally exciting breakthroughs. Bulatov, Jeavons and Krokhin [24] pioneered the algebraic approach to these problems, establishing deep connections between the computational complexity of CSPs over a certain template structure and the algebraic properties of the template's *polymorphisms*. This approach culminated with the resolution of the Feder-Vardi Dichotomy Conjecture [44] by Bulatov [25] and Zhuk [91], which classified all "islands of tractability" for constraint satisfaction problems in the process. The local consistency algorithms for CSP captured by the pebbling comonad represents just one of these "islands", in particular solving CSP on templates of *bounded width*. It is thus interesting to investigate whether the compositional methods suggested by the game comonads project can by applied beyond this and whether doing so reveals new connections between algorithms for constraint satisfaction and isomorphism.

1.1 Contributions of this thesis

This thesis represents a novel critique of game comonads as a tool for logic and algorithms. It explores for the first time the application of these methods to approximations to homomorphism and isomorphism which go beyond the realm of the logic FPC and the local algorithms of k-consistency and k-Weisfeiler-Leman. Chapter 2 sets out preliminary definitions and notation. Chapter 3 provides a new motivation for the study of this subject and a history of the field from the perspective of the descriptive power of the logics. The first major technical contributions come in Chapters 4 and 5 which deepen our understanding of the relationship between the pebbling comonad and logical quantification. This is done firstly by identifying logical fragments corresponding to maps which are intermediate between morphisms and isomorphisms of the Kleisli category of the pebbling comonad and, secondly, by expanding a result of Kolaitis and Väänänen to recast these logics in terms of unary generalised quantifiers. The second major contribution, presented in Chapter 6, is the construction of a new family of game comonads, $\mathbb{H}_{n,k}$, for logics expanded by generalised quantifiers of all arities, capturing a very powerful class of logics with historical significance in descriptive complexity theory. In the process of doing so, we invent new games for different generalised quantifier logics and discover a new natural form of structural decomposition which is related to these games by the $\mathbb{H}_{n,k}$ comonad.

This thesis does not, however, limit itself solely to comonadic constructions in the style of Abramsky, Dawar and Wang. The contribution of Chapters 7 and 8 is to go beyond comonads, breaking new and exciting ground by applying compositional methods to some other natural approximations of homomorphism and isomorphism which arise from logic and algorithms. In Chapter 7, we show that linear programming approximations to homomorphism, which are captured by no known game comonads, are naturally described by monads on the category of relational structures. We also develop some preliminary lines of investigation towards a general theory here which may prove to be dual to that of game comonads. In Chapter 8, we make an ambitious connection between the theory of presheaves and approximations to homomorphism and isomorphism. In doing so we draw interesting parallels with the use of these methods in quantum contextuality and, borrowing techniques from that field, we develop new cohomological algorithms for the homomorphism and isomorphism problems. These combine the power of local approximations captured by the comonads and linear-algebraic approximations captured by the monads of Chapter 7. We show that these methods are powerful enough to distinguish state-of-the-art counterexamples in descriptive complexity and we use this to lay the foundations for interesting future work.

1.2 Collaborations and previous work

We conclude this introduction by acknowledging the parts of this thesis which have appeared in previous publications or result from collaboration with others. Chapter 3 is an attempt at an original motivation and (selective) history of the game comonads project. As such, it is influenced by other reviews of the topic by Abramsky [2] and Dawar [33], and conversations with other devotees of game comonads. The language of *Structure* and *Power* used throughout the thesis is borrowed from Abramsky and Shah [11]. The relationship between game comonads and generalised quantifiers emerged from joint work with Anuj Dawar which was first presented at the *29th EACSL Annual Conference on Computer Science Logic* [80]. An extended version of this work [79] has been submitted to the journal *Logical Methods in Computer Science* and contains the principal results and constructions of Chapter 6, along with the definitions of the modified Hella games and the Generalised Hella's Theorem of Chapter 5. The approach to homomorphism and isomorphism of relational structures via presheaves and cohomology given in Chapter 8 was first presented at the *47th International Symposium on Mathematical Foundations of Computer Science* [78]. This work benefitted hugely from the guidance and encouragement of Anuj Dawar and Samson Abramsky. The relationship between flasque subpresheaves and k-consistency in Section 8.2.3 is entirely due to Samson Abramsky, appearing in his technical report on an earlier version of this work [1]. It is reproduced here with permission. Finally, it is noted that my previous publications use my surname in the *leagan Gaelach*; they are signed *Ó Conghaile* instead of *Connolly*.

Chapter 2

Preliminaries and definitions

In this chapter, we fix notation that is used throughout the thesis, introduce some common prerequisite definitions and collect other material that is not original to this thesis and would otherwise break the flow of the main research chapters. As such, it is intended to be referred back to as indicated in Chapters 3 to 8 rather than being read through as a standalone chapter.

2.1 Mathematical basics and notation

This section introduces some standard mathematical notation and definitions from set theory, combinatorics and algebra.

Sets and functions For every positive integer $n \in \mathbb{Z}$, we write $[n]$ for the set $\{1, \ldots, n\}$. Let A and B be sets. The set $A \times B$ is the set of all pairs (a, b) where $a \in A$ and $b \in B$. We write $f \colon A \to B$ for a function from A to B. A function is *injective* if for any $a, a' \in A$ $f(a) = f(a') \implies a = a'$, *surjective* if for any $b \in B$ there is some $a \in A$ with $f(a) = b$, and *bijective* if it is both injective and surjective. We sometimes represent a function f as the subset of $A \times B$ consisting of all pairs $(a, f(a))$. A *partial function*, $p \colon A \rightharpoonup B$ is a set of pairs $p \subset A \times B$ such that for each $a \in A$ there is at most 1 pair $(a, b) \in p$. The *domain* of p is the set $\mathbf{dom}(p) = \{a \mid \exists b \text{ s.t. } (a, b) \in p\}$.

Relations and quotients For any set A, a *binary relation* on A is subset $R \subset A^2$. For any such binary relation we abbreviate the statement $(a, b) \in R$ using the infix notation aRb. A binary relation R is said to be *reflexive* if, for all $a \in A$, aRa. R is said to be *symmetric* if for all $a, b \in A$,

$$aRb \implies bRa.$$

R is said to be *transitive* if for any $a, b, c \in A$,

$$aRb \text{ and } bRc \implies aRc.$$

For any binary relation $R \in A^2$, we define the *reflexive, symmetric or transitive closure of* R to be the smallest relation $\overline{R} \supset R$ that is (respectively) reflexive, symmetric or transitive. That such a set exists and is unique is an easy exercise. We call a binary relation R an *equivalence relation* if it is reflexive, symmetric and transitive. For any equivalence relation $\sim$ on a set A and any element $a \in A$, we define the *equivalence class of* a as the set $[a] = \{b \mid a \sim b\}$. By the definition of an equivalence relation, these classes have the property that $a \sim b \iff [a] = [b]$ and if $a \not\sim b$ then the sets $[a]$ and $[b]$ are disjoint. Given an equivalence relation $\sim$ on A, we define the *quotient of* A *by* $\sim$ as the set $A/\sim := \{[a] \mid a \in A\}$. The *quotient map* for $\sim$ is the function $q_\sim : A \to A/\sim$ is the function defined by $f(a) := [a]$ for each $a \in A$.

Orders and trees Let A be a set and $\leq$ be a binary relation on A. $\leq$ is a *partial order* on A if it is reflexive and transitive. $\leq$ is said to be a *linear order* on A if additionally it is total, meaning that for any $a, b \in A$ either $a \leq b$ or $b \leq a$. For any $a, b \in A$ if $a \leq b$ and $a \neq b$ we write $a < b$. A *tree* T is a set with a partial order $\leq$ such that for all $t \in T$, the set $\{x \mid x \leq t\}$ is linearly ordered by $\leq$ and such that there is an element $r \in T$ called the *root* such that $r \leq t$ for all $t \in T$. If $t < t'$ in T and there is no x with $t < x < t'$, we call t' a *child* of t and t the *parent* of t'.

Lists and tuples Let A be a set. We write (finite) *lists* of elements in A between square brackets as $[a_1, \ldots, a_n]$. We write ϵ for the empty list. A^* denotes the set of all lists of elements of A, including ϵ, while A^+ denotes the set of all non-empty lists in A^*. If $s \in A^*$ is a list then we write $|s|$ for the length of s. For any whole number $1 \leq m \leq |s|$, we write $s[m]$ for the mth element of the list s and $s[: m]$ for the prefix

$$[s[1], \ldots, s[m]]$$

of s. For two lists $s, t \in A^*$ of lengths m and l we write $s; t$ for the *concatenation* of s and t, i.e. the list

$$[s[1], \ldots s[m], t[1], \ldots, t[l]].$$

For any element $a \in A$ and list $s \in A^*$, we write $s; a$ as shorthand for the concatenation $s; [a]$.

Tuples are similar to lists but treated slightly differently in the text. For any set A and non-negative integer n, the set A^n is the set of all tuples $(a_1, \ldots, a_n)$ where $a_i \in A$ for each $1 \leq i \leq n$. We write tuples using boldface, for example $\mathbf{a} = (a_1, \ldots, a_n) \in A^n$. Similarly to lists, for $1 \leq i \leq n$ we write $\mathbf{a}[i]$ for the ith element of the tuple $\mathbf{a} \in A^n$. This is often abbreviated by dropping the boldface and using a subscript, i.e. a_i.

Bipartite graphs While we largely study graphs as an example of a class of relational structures, see Section 2.2, we recall here some notation and results about the classical theory of bipartite graphs. A bipartite graph $\mathcal{G}$ consists of two disjoint sets of vertices A and B and a subset $E \subset A \times B$ which we call the set of edges of $\mathcal{G}$, saying that there is an edge between vertices $a \in A$ and $b \in B$ if $(a, b) \in E$. We write for any $a \in A$ we define the neighbourhood of a as

$$\mathcal{N}(a) := \{b \mid (a, b) \in E\}$$

and dually, for any $b \in B$, we write $\mathcal{N}(b)$ for the neighbourhood of b. For any subset S of either A or B, we write $\mathcal{N}(S)$ for the union $\bigcup_{x \in S} \mathcal{N}(x)$. We call a subset of edges $M \subset E$ a *matching* if every vertex of $\mathcal{G}$ appears in at most one pair $(a, b) \in M$. A matching in $\mathcal{G}$ is said to be *total on* A, if there each vertex $a \in A$ appears in exactly one edge of M and define *total on* B symmetrically. We say that a matching in $\mathcal{G}$ is *perfect* if it is total on both A and B.

Hall's marriage theorem, stated originally for finding unique representatives of sets in some collection rather than matchings in graphs [55], gives important equivalent conditions for the existence of different matchings in a bipartite graph. We state it here for bipartite graphs.

Theorem 2.1 (Hall's Marriage Theorem [55]). *For any bipartite graph $\mathcal{G}$ on sets A and B we have the following equivalences:*

- *$\mathcal{G}$ contains a matching which is total on A if, and only if, for every subset $S \subset A$,*

$$|A| \leq |\mathcal{N}(A)|,$$

- *$\mathcal{G}$ contains a matching which is total on B if, and only if, for every subset $S \subset B$,*

$$|B| \leq |\mathcal{N}(B)|,$$

- *$\mathcal{G}$ contains a total matching if, and only if, both of the previous two conditions hold.*

Algebraic structures The algebraic structures used in this thesis are standard from any introductory undergraduate course on algebra. An algebraic structure $\mathbf{A}$ is given as a set A equipped with *operations*, i.e. functions of the type $A^n \to A$ for $n \geq 0$, which satisfy certain axioms. Nullary operations of the type $A^0 \to A$ are called *constants*. In this style we make the following definitions.

A *monoid* $\mathbf{M}$ is a set M with a constant $1 \in M$ (called the *identity*) and a binary operation $\cdot \colon M^2 \to M$ (called *multiplication*) such that $\cdot$ is *associative*, meaning that

$$\forall a, b, c \in M, \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

and 1 is the *identity of* $\cdot$ meaning that

$$\forall a \in M, \quad a \cdot 1 = 1 \cdot a = a.$$

A monoid $\mathbf{M}$ is *commutative* if

$$\forall a, b \in M, \quad a \cdot b = b \cdot a.$$

A *group* $\mathbf{G}$ is a monoid such that every element $g \in G$ has an *inverse*, meaning that

$$\forall g \in G \; \exists h \in G, \quad g \cdot h = h \cdot g = 1.$$

If a group is commutative we sometimes say that it is *abelian*. In this case we write the constant as 0 and the multiplication as $+$.

A *semiring* $\mathbf{S}$ is a set S with constants $0_{\mathbf{S}}$ and $1_{\mathbf{S}}$ and binary operations $+_{\mathbf{S}}$ and $\cdot_{\mathbf{S}}$ such that $(S, 0_{\mathbf{S}}, +_{\mathbf{S}})$ is a commutative monoid, $(S, 1_{\mathbf{S}}, \cdot_{\mathbf{S}})$ is a monoid and $\cdot_{\mathbf{S}}$ *distributes over* $+_{\mathbf{S}}$, meaning that

$$\forall a, b, c \in S, \quad a \cdot_{\mathbf{S}} (b +_{\mathbf{S}} c) = (a \cdot_{\mathbf{S}} b) +_{\mathbf{S}} (a \cdot_{\mathbf{S}} c).$$

A *ring* $\mathbf{R}$ is a semiring such that $(S, 0_{\mathbf{S}}, +_{\mathbf{S}})$ is an abelian group. A ring is *commutative* if $\cdot_{\mathbf{S}}$ is commutative. A *field* $\mathbb{F}$ is a ring such that $0_{\mathbf{S}} \neq 1_{\mathbf{S}}$, the set $S \setminus \{0_{\mathbf{S}}\}$ is closed under $\cdot_{\mathbf{S}}$, meaning that

$$\forall a, b \in S \setminus \{0_{\mathbf{S}}\}, \quad a \cdot_{\mathbf{S}} b \in S \setminus \{0_{\mathbf{S}}\}$$

and $(S \setminus \{0_{\mathbf{S}}\}, 1_{\mathbf{S}}, \cdot_{\mathbf{S}})$ is a commutative group.

For a semiring $\mathbf{S}$, a *left* $\mathbf{S}$*-semimodule* $\mathbf{M}$ is an abelian group where additionally, for each $s \in \mathbf{S}$, there is a unary operation, called *scalar multiplication by* s which sends any $a \in M$ to $s \cdot m$, such that the following hold for all $s, r \in S$ and $a, b \in M$

$$\begin{aligned} s \cdot (a + b) &= (s \cdot a) + (s \cdot b) \\ (s +_{\mathbf{S}} r) \cdot a &= (s \cdot a) + (r \cdot a) \\ r \cdot (s \cdot a) &= (r \cdot_{\mathbf{S}} s) \cdot a \\ 1_{\mathbf{S}} \cdot a &= a. \end{aligned}$$

If $\mathbf{S}$ is a ring we say $\mathbf{M}$ is a *left* $\mathbf{S}$*-module*. If $\mathbf{S}$ is a field we say $\mathbf{M}$ is a *vector space over* $\mathbf{S}$. Note that the use of the word *left* in these definitions refers to the third equation above which state that composing multiplication of scalars corresponds to multiplication *on the left* in $\mathbf{S}$. When $\mathbf{S}$ is commutative we no longer need the distinction.

2.2 Finite model theory

Finite model theory is the extensively studied mathematical theory of how finite objects relate to the logical sentences they satisfy. Descriptive complexity is one of the key twentieth century applications of this theory in computer science. It relates the expressive power of the logical languages needed to describe a class of finite objects with the computational power needed to recognise it. These fields provide both important influence and motivation to this thesis. This section recalls only the most essential definitions for the material to come but an interested reader is referred to the textbook treatments of Libkin [71] and Immerman [59] for much more comprehensive discussions of finite model theory and descriptive complexity respectively.

Finite relational structures A (finite) relational signature σ is a finite set of symbols $\sigma = \{R_1, \dots R_m\}$ where each symbol $R \in \sigma$ has an associated non-negative integer $ar(R)$ called the *arity of* R. A relational structure $\mathcal{A}$ over the signature σ is a tuple

$$\langle A, R_1^{\mathcal{A}}, \dots, R_m^{\mathcal{A}} \rangle$$

where A is the *underlying set* of $\mathcal{A}$ and for each $R \in \sigma$ $R^{\mathcal{A}} \subset A^{ar(R)}$ is a set of tuples of length $ar(R)$, which we call the *related tuples of* R *in* $\mathcal{A}$. Throughout this thesis, we use the cursive letter (e.g. $\mathcal{A}$) for the whole relational structure and the plain letter (e.g. A) for the underlying set. A relational structure $\mathcal{A}$ is said to be *finite* if A is a finite set.

Homomorphisms and isomorphisms Let $\mathcal{A}$ and $\mathcal{B}$ be two relational structures over a common signature σ. For any function $f\colon A \to B$ between the underlying sets of these and any symbol $R \in \sigma$ with $ar(R) = n$, we say that f *preserves the relation* R if

$$\forall \mathbf{a} = (a_1, \dots, a_n) \in A^n, \quad \mathbf{a} \in R^{\mathcal{A}} \implies (f(a_1), \dots, f(a_n)) \in R^{\mathcal{B}}.$$

We say that f *reflects the relation* R if the condition above holds with the implication reversed. We call f a *homomorphism* if it preserves every relation $R \in \sigma$. When such an f exists we write $\mathcal{A} \to \mathcal{B}$. If f is an injection, surjection or bijection we call it an *injective, surjective or bijective homomorphism* respectively. A bijective homomorphism which additionally reflects every relation $R \in \sigma$ is called an *isomorphism.* If there is an isomorphism between $\mathcal{A}$ and $\mathcal{B}$ we write $\mathcal{A} \cong \mathcal{B}$. A partial function $p \subset A \times B$ which preserves all relations $R \in \sigma$ is called a *partial homomorphism.* A partial homomorphism which is injective and also reflects all relations $R \in \sigma$ is called a *partial isomorphism.*

Congruences and quotient structures Let $\mathcal{A}$ be a relational structure over some signature σ and let $\sim$ be an equivalence relation on the underlying set A. For any pair of tuples $\mathbf{a} = (a_1, \dots a_n)$ and $\mathbf{a}' = (a'_1, \dots a'_n) \in A^n$ such that for each $1 \leq i \leq n$ $a_i \sim a'_i$, we write $\mathbf{a} \sim \mathbf{a}'$. We say that $\sim$ is a *congruence on* $\mathcal{A}$ if for every relation $R \in \sigma$ of arity n and any pair of tuples $\mathbf{a}, \mathbf{a}' \in A^n$ such that $\mathbf{a} \sim \mathbf{a}'$, we have that

$$(a_1, \dots, a_n) \in R^{\mathcal{A}} \iff (a'_1, \dots, a'_n) \in R^{\mathcal{A}}.$$

For any relational structure $\mathcal{A}$ over the signature σ and congruence $\sim$ on $\mathcal{A}$, we define the *quotient structure of $\mathcal{A}$ by $\sim$* as the structure $\mathcal{A}/\sim$ over the signature σ with underlying set $A/\sim$ and relations $R^{\mathcal{A}/\sim}$ defined as the set

$$\{([a_1], \dots, [a_n]) \mid (a_1, \dots a_n) \in R^{\mathcal{A}}\},$$

for each $R \in \sigma$. It can be proven that for any such congruence this is the unique relational structure on the set $A/\sim$ such that the quotient map $q_\sim \colon A \to A/\sim$ both preserves and reflects each relation $R \in \sigma$.

It is not hard to see that the set $R^{\mathcal{A}/\sim}$ given above is only well-defined when $\sim$ is a congruence. If $\sim$ is not a congruence, there will not in general be any relational structure on $A/\sim$ such that $q_\sim$ both preserves and reflects each $R \in \sigma$. In this case, in this thesis we define the quotient structure $\mathcal{A}/\sim$ to be the "minimal" structure on $A/\sim$ such that $q_\sim$ both preserves each $R \in \sigma$. This is done by defining $R^{\mathcal{A}/\sim}$ for each $R \in \sigma$ with arity n to be the set

$$\{([a_1], \dots [a_n]) \mid \exists \mathbf{a}' \in R^{\mathcal{A}} \text{ s.t. } \mathbf{a}' \sim (a_1, \dots a_n)\}.$$

Tree decompositions and treewidth The notions of treewidth and the related tree decompositions of graphs and more generally relational structures have been rediscovered several times in graph theory and finite model theory. Perhaps most notably they are used as an important part of Robertson and Seymour's famous programme on graph minors [86]. For the definitions in this section we rely on Grohe's book [50]. In particular, we define a tree decomposition, following Definition 4.1.1 of [50] as follows.

Definition 2.2. *A* tree decomposition *of a (finite) σ-structure $\mathcal{A}$ is a pair (T, B) with T a tree and $B : T \to 2^A$ such that:*

1. *For every $a \in A$ the set $\{t \mid a \in B(t)\}$ induces a subtree of T; and*
2. *for all relational symbols $R \in \sigma$ and related tuples $\mathbf{a} \in R^{\mathcal{A}}$, there exists a node $t \in T$ such that $\mathbf{a} \subset B(t)$.*

In any such tree decomposition, for every $t \in T$ we call the set $B(t)$ the *bag at t*. We say that a tree decomposition has *width n* if the size of the largest bag is n, i.e. $n = \max_{t \in T} |B(t)| - 1$. The *treewidth* of a structure $\mathcal{A}$, is the smallest m for which there exists a tree decomposition of $\mathcal{A}$ of width m. We write this $\mathbf{tw}(\mathcal{A})$ for this parameter. In a similar manner, we can define the notion of a *path decomposition* as a tree decomposition where the tree T is in fact a linearly ordered set P. The parameter *pathwidth* is then the smallest m for which there exists a path decomposition of width m.

An alternative game-theoretic characterisation of treewidth is given by the (rather entertaining) *cops-and-robbers game* on the structure $\mathcal{A}$, which was first defined by Aigner and Fromme [12]. The game involves two players, the robber and the cops, taking turns to move around the structure. The cops try to trap the robber while the robber seeks to evade capture. The difficulty of the game (for the cops) is measured by the resources (number of cops, improved movement capabilities etc.) required by the cops to succeed in their aim. Intuitively, the more *densely connected* the structure $\mathcal{A}$ is the more difficult the robber is to apprehend on $\mathcal{A}$. For example, it is known that a structure has treewidth less than k if, and only if k cops "with helicopters" can win the cops-and-robbers game on $\mathcal{A}$. A good survey of the many variantions of this game (for the case of graphs) is given in the excellent textbook by Bonato and Nowakowski [21].

Logic In the most abstract sense, we take a *logic* $\mathcal{L}$ to be function which assigns to each relational structure σ a collection $\mathcal{L}[\sigma]$ of *formulas* $\phi(\mathbf{x})$ with some list $\mathbf{x}$ of free variables and a *semantics relation* $\models_{\mathcal{L}[\sigma]}$ which defines for each σ-structure $\mathcal{A}$ and choice of parameters $\mathbf{a} \in A^n$ whether $\mathcal{A}, \mathbf{a}$ *satisfies* any given formula $\phi(\mathbf{x})$ with $|x| \leq n$. If some choice $\mathcal{A}, \mathbf{a}$ satisfies a formula $\phi(\mathbf{x})$ in $\mathcal{L}[\sigma]$ we write $\mathcal{A}, \mathbf{a} \models_{\mathcal{L}[\sigma]} \phi(\mathbf{x})$, we drop the subscript when the logic $\mathcal{L}$ is clear from context.

For logics $\mathcal{L}$ and $\mathcal{L}'$, we say that formulas $\phi(\mathbf{x}) \in \mathcal{L}[\sigma]$ and $\phi'(\mathbf{x}) \in \mathcal{L}'[\sigma]$ are *equivalent* if for every σ-structure $\mathcal{A}$ and tuple $\mathbf{a}$ we have

$$\mathcal{A}, \mathbf{a} \models_{\mathcal{L}[\sigma]} \phi(\mathbf{x}) \iff \mathcal{A}, \mathbf{a} \models_{\mathcal{L}'[\sigma]} \phi'(\mathbf{x}).$$

We say that $\mathcal{L}$ and $\mathcal{L}'$ are *equivalent* of for every signature σ and every $\phi \in \mathcal{L}$ there is an equivalent formula $\overline{\phi} \in \mathcal{L}'[\sigma]$ and, similarly, for every $\psi \in \mathcal{L}'[\sigma]$ there is an equivalent formula $\overline{\psi} \in \mathcal{L}[\sigma]$. If this is the case, we write $\mathcal{L} \equiv \mathcal{L}'$.

Fix a logic $\mathcal{L}$ and signature σ. For any two σ-structures $\mathcal{A}$ and $\mathcal{B}$ and tuples $\mathbf{a} \in A^m$ and $\mathbf{b} \in B^m$, we write $\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}} \mathcal{B}, \mathbf{b}$ if for every formula $\phi(\mathbf{x}) \in \mathcal{L}[\sigma]$ we have

$$\mathcal{A}, \mathbf{a} \models \phi(\mathbf{x}) \implies \mathcal{B}, \mathbf{b} \models \phi(\mathbf{x}).$$

We say that $\mathcal{A}$ and $\mathcal{B}$ are *equivalent over* $\mathcal{L}$ (or $\mathcal{L}$*-equivalent*) if we have both $\mathcal{A} \Rrightarrow_{\mathcal{L}} \mathcal{B}$ and $\mathcal{B} \Rrightarrow_{\mathcal{L}} \mathcal{A}$.

We assume a standard syntax and semantics for first-order logic (as in [71]), which we denote **FO**. We write $\mathcal{L}_\infty$ for the infinitary logic that is obtained from **FO** by allowing conjunctions and disjunctions over arbitrary sets of formulas. We write $\exists^+\mathcal{L}_\infty$ and $\exists^+\mathbf{FO}$ for the restriction of $\mathcal{L}_\infty$ and **FO** to existential positive formulas, i.e. those without negations or universal quantifiers. We use natural number superscripts to denote restrictions of the logic to a fixed number of variables. So, in particular $\mathbf{FO}^k, \mathcal{L}^k_\infty$ and $\exists^+\mathcal{L}^k_\infty$ denote the k-variable fragments of **FO**, $\mathcal{L}_\infty$ and $\exists^+\mathcal{L}_\infty$ respectively. Similarly, we use subscripts on the names of the logic to denote the fragments limited to a fixed nesting depth of quantifiers. Thus, $\mathbf{FO}_r, \mathcal{L}_{\infty,r}$ and $\exists^+\mathcal{L}_{\infty,r}$ denote the fragments of **FO**, $\mathcal{L}_\infty$ and $\exists^+\mathcal{L}_\infty$ with quantifier depth at most r. We write $\mathcal{C}$ to denote the extension of $\mathcal{L}_\infty$ where we are allowed quantifiers $\exists^{\geq i}$ for each natural number i. The quantifier is to be read as "there exists at least i elements…". We are mainly interested in the k-variable fragments of this logic $\mathcal{C}^k$. When we are interested in emphasising that $\mathcal{C}$ represents the addition of counting quantifiers to $\mathcal{L}_\infty$ we write it as $\mathcal{L}_\infty(\#)$.

Spoiler-Duplicator games Let $\mathcal{A}$ and $\mathcal{B}$ be relational structures over a common signature. We saw earlier in this section that the homomorphism and isomorphism relations $\mathcal{A} \to \mathcal{B}$ and $\mathcal{A} \cong \mathcal{B}$ are witnessed by functions which preserve or preserve and reflect the relations of σ. In finite model theory, these relations are often too strong and we often prefer to consider the relations $\Rrightarrow_{\mathcal{L}}$ and $\equiv_{\mathcal{L}}$ over some logic $\mathcal{L}$. This raises the question of how we witness these relations in a convenient way. Spoiler-Duplicator games are an important way to do this which were introduced by Ehrenfeucht [42] (after earlier work of Roland Fraïssé) to witness the relation $\equiv_{\mathbf{FO}}$. The original game is defined as follows.

Definition 2.3. *For relational structures $\mathcal{A}$ and $\mathcal{B}$ over the same signature, the* back-and-forth Ehrenfeucht-Fraïssé game between $\mathcal{A}$ and $\mathcal{B}$, *written* $\mathbf{EF}(\mathcal{A}, \mathcal{B})$ *is played by two players, Spoiler and Duplicator, as follows.*

At the start of each round the position *is given by a partial isomorphism $p \subset A \times B$. Spoiler begins by choosing one of the structures $\mathcal{A}$ or $\mathcal{B}$ and selecting an element from that structure. Duplicator responds by choosing an element from the other structure. Writing (a, b) for the pair of chosen elements, the new position is given by $p' = p \cup \{(a, b)\}$. If p' fails to be a partial isomorphism then the game ends and we say that Spoiler has won. Otherwise the game continues to another round. We say that Duplicator wins if she can prevent Spoiler from winning.*

We write $\mathbf{EF}_n(\mathcal{A}, \mathcal{B})$ for the version of this game which terminates after n rounds.

We say that Duplicator has a *winning strategy* for the game $\mathbf{EF}_n(\mathcal{A}, \mathcal{B})$ if she can play in such a way that she always wins, regardless of how Spoiler plays. Ehrenfeucht's original paper proves the following theorem which is a model for all other Spoiler-Duplicator games.

Theorem 2.4 (Ehrenfeucht-Fraïssé Theorem)**.** *For any two relational structures $\mathcal{A}$, $\mathcal{B}$, Duplicator has a winning strategy for the game $\mathbf{EF}_n(\mathcal{A}, \mathcal{B})$ with the starting position $p = \{(a_1, b_1), \ldots, (a_m, b_m)\}$ if and only if*

$$\mathcal{A}, a_1, \ldots a_m \equiv_{\mathbf{FO}_n} \mathcal{B}, b_1, \ldots b_m.$$

There are many variants of the game $\mathbf{EF}$ in use throughout finite model theory which capture other other logical relations in the same manner as Theorem 2.4. We briefly define the following three to give a flavour of the possible variations.

The *one-way Ehrenfeucht-Fraïssé game from $\mathcal{A}$ to $\mathcal{B}$*, written $\exists\mathbf{EF}(\mathcal{A},\mathcal{B})$ is played as before by Spoiler and Duplicator on the structures $\mathcal{A}$ and $\mathcal{B}$. In this game, the position at the start of each round is only required to be a partial homomorphism and Spoiler can only play on the structure $\mathcal{A}$. Spoiler wins if the position at the end of the round fails to be a partial homomorphism. There is an analogous result to Theorem 2.4 which states that Duplicator has a winning strategy for this game with starting position $\{(a_1, b_1), \dots, (a_m, b_m)\}$ if, and only if, $\mathcal{A}, a_1, \dots a_m \Rrightarrow_{\exists^+\mathbf{FO}_n} \mathcal{B}, b_1, \dots b_m$.

The *k-pebble bijection game between $\mathcal{A}$ and $\mathcal{B}$*, written $\mathbf{Bij}^k(\mathcal{A},\mathcal{B})$, is played by Spoiler and Duplicator on the structures $\mathcal{A}$ and $\mathcal{B}$ and was defined for example by Immerman [59] after a game of Hella. In this game, the position p at the start of each round is required to be a partial isomorphism of size at most k. At the start of each round, if the position has size k, Spoiler must choose some pair to remove from p to give a new position p'. Duplicator then provides a bijection $f\colon A \to B$ and Spoiler chooses an element $a \in A$. The position at the end of the round is then given by $p'' = p' \cup \{(a, f(a))\}$ and Spoiler wins if this fails to be a partial isomorphism. The analogous result to Theorem 2.4 says that Duplicator has a winning strategy for this game with starting position $\{(a_1, b_1), \dots, (a_m, b_m)\}$ if, and only if, $\mathcal{A}, a_1, \dots a_m \equiv_{\mathcal{C}^k} \mathcal{B}, b_1, \dots b_m$.

The *k-pebble n-bijective game between $\mathcal{A}$ and $\mathcal{B}$*, written $\mathbf{Bij}^n_k(\mathcal{A},\mathcal{B})$, is played by Spoiler and Duplicator on the structures $\mathcal{A}$ and $\mathcal{B}$ and was defined by Hella [58]. In this game, the position p at the start of each round is required to be a partial isomorphism of size at most k. At the start of each round, Duplicator provides a bijection $f\colon A \to B$ and Spoiler chooses subsets $p' \subset p$ and $S \subset A$ such that $|S| \leq n$ and $|S| + |p'| \leq k$. The position at the end of the round is then given by $p'' = p' \cup \{(a, f(a))\}_{a \in S}$ and Spoiler wins if this fails to be a partial isomorphism. The analogous result to Theorem 2.4 says that Duplicator has a winning strategy for this game with starting position $\{(a_1, b_1), \dots, (a_m, b_m)\}$ if, and only if, $\mathcal{A}, a_1, \dots a_m$ and $\mathcal{B}, b_1, \dots b_m$ are equivalent over k-variable infinitary first order logic extended by all *generalised quantifiers of arity n* which we define in Chapter 5.

2.3 Complexity theory and algorithms

Computational complexity theory studies the amount of *resources* (e.g. time, space, quantum entanglement) required to perform computations. Algorithms are step-by-step methods of solving computational problems which use a certain amount of resources. For a full background on complexity theory and its relationship to logic we refer the reader to Immerman's textbook treatment [59]

Complexity classes We reference several complexity classes throughout this thesis. `PTIME` is the class of all sets $S \subset \{0, 1\}^*$ whose membership is decidable using a polynomial-time deterministic Turning machine, i.e. a machine T for which there is some k for which

T terminates after $\mathcal{O}(n^k)$ deterministic steps on inputs of size n. *Quasi-polynomial time*, written `QuasiP`, is the class of sets decidable on deterministic Turing machines which terminate after $2^{\mathcal{O}((\log(n))^c)}$ steps on inputs of length n for some fixed positive real number c. `NPTIME` is the class of decidable using a polynomial-time *non-deterministic* Turning machine. A (polynomial-time) reduction from one set $S \subset \{0,1\}^*$ to another set $S' \subset \{0,1\}^*$ is a polynomial-time deterministic Turing machine for which, given any input x, the output $T(x)$ is in S if, and only if, $x \in S'$. We say that a decision problem S is `NP-hard` if for any problem S' in `NPTIME` there is a polynomial-time reduction to S. A problem S is `NP-complete` if it is both in `NPTIME` and `NP-hard`.

Descriptive complexity theory Descriptive complexity theory attempts to relate the classes of relational structures definable in certain logics to the computational complexity of these classes. To assign a computational complexity (in the sense of the last paragraph) to a class of relational structures, we need a way of encoding relational structures as strings from the set $\{0,1\}^*$. A standard encoding is given in Definition 2.1 of [59]. We say that a logic $\mathcal{L}$ *defines* a class of relational structures K over the signature σ if there is a formula $\phi_K \in \mathcal{L}[\sigma]$ such that for every σ-structure $\mathcal{A}$, $\mathcal{A} \in K \iff \mathcal{A} \models \phi_K$. We say that a logic, $\mathcal{L}$, *captures* a complexity class `C` if a class of relational structures K is defined by $\mathcal{L}$ if, and only if, K is decidable in `C`. The foundational result in descriptive complexity theory is due to Fagin [43] and states that existential second-order logic $\exists$**SO** captures `NPTIME`.

A logic for `PTIME`? This raises the question of whether there is a similar logic for `PTIME`. While Gurevich famously conjectured [54] that no such logic exists, there have been several candidate logics for `PTIME` throughout the history of descriptive complexity theory. Two important examples referenced in this thesis are fixed-point logic with counting and rank logic. *Fixed-point logic with counting* (written FPC) is first-order logic extended with operators for inflationary fixed-points and counting, for example see [45]. Rank logic is first-order logic extended with operators for inflationary fixed-points and computing ranks of matrices over finite fields, see [82]. Each has since been proven not to capture `PTIME`, for FPC see Cai, Fürer and Immerman [27], for rank logic see Lichter [72].

CSP and structure isomorphism Assuming a fixed relational signature σ, we write CSP for the set of all pairs of σ-structures $(\mathcal{A}, \mathcal{B})$ such that there is a homomorphism witnessing $\mathcal{A} \to \mathcal{B}$. We use $CSP(\mathcal{B})$ to denote the set of relational structures $\mathcal{A}$ such that $(\mathcal{A}, \mathcal{B}) \in CSP$. We also use CSP and $CSP(\mathcal{B})$ to denote the decision problem on these sets. For general $\mathcal{B}$, $CSP(\mathcal{B})$ is well-known to be `NP-complete`. However for certain structures $\mathcal{B}$ the problem is in `PTIME`. Indeed, the Bulatov-Zhuk Dichotomy Theorem (formerly the Feder-Vardi Dichotomy Conjecture) states that for any $\mathcal{B}$ $CSP(\mathcal{B})$ is either `NP`-complete or it is `PTIME`. Working out efficient algorithms which decide $CSP(\mathcal{B})$ for larger and larger classes of $\mathcal{B}$ was an active area of research which culminated in Bulatov and Zhuk's exhaustive classes of algorithms [25, 91].

Similarly, we write SI for the set of all pairs of σ-structures $(\mathcal{A}, \mathcal{B})$ such that there is an isomorphism witnessing $\mathcal{A} \cong \mathcal{B}$. The decision problem for this set is also thought not to be in `PTIME` however there are no general hardness results known for this. The best known algorithm (in the case where σ is the signature of graphs) is Babai's[15] which is in `QuasiP`.

2.4 Cai-Fürer-Immerman constructions

In Chapter 8, we make extensive use of the Cai-Fürer-Immerman (CFI) construction, which originally was used to prove that FPC does not capture `PTIME` (see Theorem 2.6) and has since appeared in many different guises in descriptive complexity theory. We define a version of this here and state the main known results about it.

Following Lichter[72], we define the general CFI construction $\mathbf{CFI}_q(G, g)$ for q a prime power, $G = (G, <)$ an ordered undirected graph and g a function from the edge set of G to $\mathbb{Z}_q$. The idea is that the construction encodes a system of linear equations over $\mathbb{Z}_q$ into G while the function g "twists" these equations in a certain way. For CFI structures, $\mathbf{CFI}_q(G, g)$ the property $\sum g = 0$ is sometimes called the *CFI property*. The following well-known fact (see [82], for example) shows that this property is closed under isomorphisms and is useful in our later arguments.

Fact 2.5. *For any prime power, q, ordered graph G, and functions g, h from the edges of G to $\mathbb{Z}_q$,*

$$\mathbf{CFI}_q(G, g) \cong \mathbf{CFI}_q(G, h) \iff \sum g = \sum h$$

$\mathbf{CFI}_q(G, g)$ is built in three steps. First, we define a gadget which replaces each vertex of G with elements that form a ring. Secondly, we define relations between gadgets which impose consistency equations between gadgets. Finally, the function g is used to insert the important twists into the consistency equations. We now describe this in detail below, following a presentation by Lichter[72].

Vertex gadgets For any vertex $x \in G$, let $\mathcal{N}(x)$ be the neighbourhood of x in G (i.e. those vertices which share edges with x) and let $\mathbb{Z}_q^{\mathcal{N}(x)}$ denote the ring of functions from $\mathcal{N}(x)$ to the ring $\mathbb{Z}_q$. We will replace each vertex x of the base graph with a gadget whose vertices are the following subset of $\mathbb{Z}_q^{\mathcal{N}(x)}$,

$$A_x = \{\mathbf{a} \in \mathbb{Z}_q^{\mathcal{N}(x)} \mid \sum_{y \in \mathcal{N}(x)} \mathbf{a}(y) = 0\}$$

The relations on the gadget are for each y in $\mathcal{N}(x)$ a symmetric relation

$$I_{x,y} = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a}(y) = \mathbf{b}(y)\}$$

and a directed cycle encoded by the relation

$$C_{x,y} = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a}(y) = \mathbf{b}(y) + 1\}$$

Together these impose the ring structure of $\mathbb{Z}_q^{\mathcal{N}(x)}$ onto the vertices of the gadget.

Edge equations Next define a relation between gadgets for each edge $\{x, y\}$ in G and each constant $c \in \mathbb{Z}_q$ of the form

$$E_{\{x,y\},c} = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \in A_x,\ \mathbf{b} \in A_y,\ \mathbf{a}(y) + \mathbf{b}(x) = c\}$$

Putting it together with a twist We finally define the structure $\mathbf{CFI}_q(G, g)$ as $\langle A, \prec, R_I, R_C, R_{E,0}, R_{E,1}, \ldots, R_{E,q-1}\rangle$ where the universe is $A = \cup_x A_x$ where $\prec$ is the linear pre-order

$$\prec = \bigcup_{x<y} A_x \times A_y$$

and the edge relations $R_{E,c}$ are interpreted according to the twists in g as

$$R_{E,c} = \bigcup_{e \in E} E_{e,c+g(e)}$$

where the sum in the subscript is over $\mathbb{Z}_q$ For the relations R_I and R_C we deviate slightly from Lichter's construction and interpret these as ternary relations of the following form

$$R_I = \bigcup_{\{x,y\} \in E} I_{x,y} \times A_y$$
$$R_C = \bigcup_{\{x,y\} \in E} C_{x,y} \times A_y$$

We now recall the two major separation results based on this construction. The first is a landmark result of descriptive complexity from the early 1990's.

Theorem 2.6 (Cai, Fürer, Immerman [27])**.** *There is a class of ordered (3-regular) graphs $\mathcal{G} = \{G_n\}_{n \in \mathbb{N}}$ such that in the respective class of CFI structures*

$$\mathcal{K} = \{\mathbf{CFI}_2(G, g) \mid G \in \mathcal{G}, g \colon V(G) \to \mathbb{Z}_2\}$$

the CFI property is decidable in polynomial-time but cannot be expressed in FPC.

The second is a recent breakthrough due to Moritz Lichter.

Theorem 2.7 (Lichter [72])**.** *There is a class of ordered graphs $\mathcal{G} = \{G_n\}_{n \in \mathbb{N}}$ such that in the respective class of CFI structures*

$$\mathcal{K} = \{\mathbf{CFI}_{2^k}(G, g) \mid G \in \mathcal{G}\}$$

the CFI property is decidable in polynomial-time (indeed, expressible in choiceless polynomial time) but cannot be expressed in rank logic.

2.5 Category theory

The comonadic construction of Abramsky, Dawar and Wang [6] which initiated research on game comonads, emerged from applying the mathematics of category theory to Spoiler-Duplicator games. This thesis continues in this tradition but remains firmly motivated by concerns in finite model theory and descriptive complexity. We thus do not assume any knowledge of category theory and introduce the preliminaries of the field here. For a more comprehensive introduction to category theory we refer to Chapter 1 of Leinster's textbook [70].

Basics A *category* $\mathbf{C}$ consists of

- a class $ob(\mathbf{C})$ called the *objects* of $\mathbf{C}$,
- for each pair of objects $A, B \in ob(\mathbf{C})$, a class $\mathbf{C}(A, B)$ of *morphisms from* A *to* B, which we write as arrows $A \xrightarrow{f} B$; and
- for any $A, B, C \in ob(\mathbf{C})$ a function $\circ\colon \mathbf{C}(B, C) \times \mathbf{C}(A, B) \to \mathbf{C}(A, C)$ called *composition in* $\mathbf{C}$

such that for every $A \in ob(\mathbf{C})$ there is an object $1_A \in \mathbf{C}(A, A)$ for which $(\mathbf{C}(A, A), \circ, 1_A)$ is a monoid and $\circ$ is associative in the sense that for any morphisms f, g, h of the appropriate types $f \circ (g \circ h) = (f \circ g) \circ h$. We say that two objects in $\mathbf{C}$ are *isomorphic* if there are two morphisms $A \xrightarrow{f} B \xrightarrow{g} A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$. We write $\mathbf{C}^{op}$ for the *dual of a category* $\mathbf{C}$ which is a category with the same objects as $\mathbf{C}$ and with morphisms $\mathbf{C}^{op}(A, B) := \mathbf{C}(B, A)$ and composition defined using composition in $\mathbf{C}$. The most common categories we will refer to are the category of sets **Set** whose objects are sets and morphisms are functions and the category $\mathcal{R}(\sigma)$ of relational structures over σ with morphisms being homomorphisms.

For any two categories $\mathbf{C}$ and $\mathbf{D}$ a *functor* $F\colon \mathbf{C} \to \mathbf{D}$ is a function $F\colon ob(\mathbf{C}) \to ob(\mathbf{D})$ and, for all $A, B \in ob(\mathbf{C})$ a function $F\colon \mathbf{C}(A, B) \to \mathbf{D}(F(A), F(B))$ such that $F(1_A) = 1_{F(A)}$ and for any $A \xrightarrow{f} B \xrightarrow{g} C$, $F(g \circ f) = F(g) \circ F(f)$. A functor $F\colon \mathbf{C} \to \mathbf{C}$ is called an *endofunctor on* $\mathbf{C}$. For any category the *identity endofunctor* $\mathbf{id_C}\colon \mathbf{C} \to \mathbf{C}$ acts as the identity function on both objects and morphisms. For any two functors $F, G\colon \mathbf{C} \to \mathbf{D}$, a *natural transformation* $\alpha\colon F \implies G$ is a choice of morphism $\alpha_A \in \mathbf{D}(F(A), G(A))$ for every $A \in \mathbf{C}$ such that, for any $f \in \mathbf{C}(A, B)$ we have that $G(f) \circ \alpha_A = \alpha_B \circ F(f)$.

Monads and comonads We now define the notion of a *comonad* which is essential to the compositional approach to Spoiler-Duplicator games pioneered by Abramsky, Dawar and Wang [6].

Definition 2.8. *A* comonad $\mathbb{T}$ *on a category* $\mathbf{C}$ *can be defined in two different equivalent ways as an* Eilenberg-Moore triple *or a* Kleisli triple. *We give both definitions here.*

Eilenberg-Moore Triple *A comonad* $\mathbb{T}$ *on* $\mathbf{C}$ *is a triple* $(\mathbb{T}, \epsilon, \delta)$ *where* $\mathbb{T}$ *is an endofunctor of* $\mathbf{C}$, *and* ϵ *and* δ *are natural transformations, called the* counit *and* comultiplication *of* $\mathbb{T}$ *and having types* $\mathbb{T} \implies \mathbf{id_C}$ *and* $\mathbb{T} \implies \mathbb{T}\mathbb{T}$ *such that the following diagrams commute.*

$$\begin{tikzcd}
\mathbb{T}A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & \mathbb{T}\mathbb{T}A \arrow[d, "\mathbb{T}\delta_A"] \\
\mathbb{T}\mathbb{T}A \arrow[r, "\delta_{\mathbb{T}A}"'] & \mathbb{T}\mathbb{T}\mathbb{T}A
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{T}A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] \arrow[dr, equal] & \mathbb{T}\mathbb{T}A \arrow[d, "\mathbb{T}\epsilon_A"] \\
\mathbb{T}\mathbb{T}A \arrow[r, "\epsilon_{TA}"'] & \mathbb{T}A
\end{tikzcd}$$

Kleisli Triple *A comonad* $\mathbb{T}$ *on* $\mathbf{C}$ *is a triple* $(\mathbb{T}, \epsilon, \overline{(\cdot)})$ *where* $\epsilon \colon \mathbb{T} \implies \mathbf{id_C}$ *is a natural transformation and for every pair of objects* A, B *in* $\mathbf{C}$ *the* Kleisli coextension $\overline{(\cdot)}$ *is a function from* $\mathbf{C}(\mathbb{T}A, B)$ *to* $\mathbf{C}(\mathbb{T}A, \mathbb{T}B)$ *such that*

- $\overline{\epsilon_A} = \mathbf{id}_{\mathbb{P}A}$ *for any* $A \in ob(\mathbf{C})$
- $\epsilon_B \circ \overline{f} = f$ *for any morphism* $f \in \mathbf{C}(\mathbb{P}A, B)$
- $\overline{g \circ \overline{f}} = \overline{g} \circ \overline{f}$ *for any morphisms* $f \in \mathbf{C}(\mathbb{P}A, B)$ *and* $g \in \mathbf{C}(\mathbb{P}A, B)$

An *monad* is the dual construction of a comonad and is defined in Eilenberg Moore form as follows.

Definition 2.9. *A monad* $\mathbb{M}$ *on* $\mathbf{C}$ *is a triple* $(\mathbb{M}, \eta, \mu)$ *where* $\mathbb{M}$ *is an endofunctor of* $\mathbf{C}$, *and* η *and* μ *are natural transformations, called the* unit *and* multiplication *of* $\mathbb{M}$ *and having types* $\mathbf{id_C} \implies \mathbb{M}$ *and* $\mathbb{M}\mathbb{M} \implies \mathbb{M}$ *such that the analogous diagrams to those in Definition 2.8 where all arrows are reversed commute.*

We define the notions of the Kleisli category of a comonad and coalgebras for a comonad in Chapter 3. The dual notions for monads which we call the Kleisli category of the monad and algebras of the monad are defined by reversing the direction of the arrows in the comonadic definitions.

Chapter 3

A review of game comonads

Spoiler-Duplicator games, as we have seen in the previous chapter, have proved an extremely useful tool to Finite Model Theory and Descriptive Complexity. Variations of these games have been created to capture many fragments of logic and finding strategies for the Duplicator in these games continues to be one of the main ways to prove expressiveness lower bounds for properties of finite relational structures. The use of these games has been very successful in Descriptive Complexity but until recently there has been no framework for studying these games collectively. Instead, games were introduced ad hoc for different logic fragments and Duplicator strategies are usually constructed concretely and studied separately for each game.

Game comonads provide exactly such a framework for Spoiler-Duplicator games. This category-theoretic approach was first introduced in the case of pebble games by Abramsky, Dawar and Wang [6] and the framework has been expanded to include Ehrenfeucht-Fraïssé games as well as games for modal logic [11], guarded logic [8] and restricted conjunction logic [77]. These are comonads on the category $\mathcal{R}(\sigma)$ where certain morphisms (called I-morphisms) in their Kleisli categories capture strategies for Duplicator in the respective one-way Spoiler-Duplicator games. While this gives a neat compositional semantics to Duplicator strategies for a wide-range of one-way games, the true power of these comonads comes from the connections they establish between different games. Indeed, for each comonad there is a corresponding one-way, back-and-forth and bijective game for which Duplicator strategies are reflected, each in a different way, in the Kleisli category of the comonad. For many of the known game comonads these games each correspond to a well-studied game from finite model theory. Furthermore, and perhaps more surprisingly, the coalgebras of these comonads also correspond to well-studied structural decompositions which, in different comonads, characterise parameters such as treedepth, treewidth, and pathwidth. The comonads which we review in this chapter give a rich sense of what it means to be a Spoiler-Duplicator game and connect many notions which appeared before as ad hoc and unrelated in finite model theory. They have also been used to find general category-theoretic proofs to theorems from finite model theory which have previously relied directly on Spoiler-Duplicator games, including new generalisations of Lovasz's classic homomorphism counting result [74].

In this chapter, we review the current body of research into comonadic semantics for Spoiler-Duplicator games used in finite model theory. The first section discusses the motivation for this approach to Spoiler-Duplicator games, the second introduces the main examples of game comonads discovered so far and the final section discusses the recent work towards a general theory of these phenomena and their applications in finite model theory.

3.1 Motivation for the pebbling comonad

In this section, we motivate the comonadic approach to Spoiler-Duplicator games by considering the compositional nature of Duplicator strategies in these games. In particular, we consider strategies for the one-way pebble game of Kolaitis and Vardi and show how Abramsky, Dawar and Wang's pebbling comonad construction arises directly from a natural way of composing these. The approach taken in this section differs in style to the that given in Abramsky, Dawar and Wang's original paper but the construction, which is discussed in more depth in the next section, is the same.

3.1.1 Categories in Finite Model Theory

In finite model theory, the category studied is usually taken to be $\mathcal{R}(\sigma)$ the category with objects being the relational structures over some finite signature σ and maps between structures being homomorphisms. While categories are rarely mentioned explicitly in most work on finite model theory, this category theoretic structure has sometimes appeared implicitly for example in Hell and Nešetřil's work on cores of graphs [57]. One notable exception to this is the treatment of adjoint functors on categories of graphs by Foniok and Tardif [46] and the extension of this to gadget constructions by Krokhin, Opršal, Wrochna and Zivný [69]. Note that the category $\mathcal{R}(\sigma)$ differs from the category of σ-structures regularly considered in classical model theory where maps are *elementary embeddings*, see for example the review of Baldwin [17]. Borrowing the notation of Libkin [71], we call this category $\mathtt{STRUCT}[\sigma]$ to distinguish it from $\mathcal{R}(\sigma)$.

For applications in descriptive complexity theory, however, the category $\mathcal{R}(\sigma)$ has, in general, too few morphisms to capture the relations of interest between structures. This is because the relations $\mathcal{A} \to \mathcal{B}$ or $\mathcal{A} \cong \mathcal{B}$ are too strong to be useful in proving complexity lower bounds. Indeed, for any structures $\mathcal{A}$ and $\mathcal{B}$ deciding if there is a homomorphism $\mathcal{A} \to \mathcal{B}$ is the same as deciding the constraint satisfaction problem $\mathbf{CSP}(\mathcal{A}, \mathcal{B})$ and over finite structures $\mathcal{A} \cong \mathcal{B}$ is the same as equivalence of $\mathcal{A}$ and $\mathcal{B}$ over full first-order logic. As a result, it is more common to study coarser relations such as $\Rrightarrow_{\mathcal{L}}$ or $\equiv_{\mathcal{L}'}$ over some logics

$\mathcal{L}, \mathcal{L}'$ which are usually fragments of first-order logic, possibly extended by quantifiers or arbitrary conjunctions and disjunctions. It is therefore an interesting question to ask what is the right category to consider when we are interested in studying these coarser relations. It is, in fact, this question for the relation $\Rrightarrow_{\exists^+\mathcal{L}^k}$ which originally motivated the work by Abramsky, Dawar & Wang on the pebbling comonad.

One naïve approach to defining this category is to directly use the definition of $\Rrightarrow_{\exists^+\mathcal{L}^k}$. Recall that for two relational structures $\mathcal{A}$ and $\mathcal{B}$ we have that $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$ if and only if for all sentences $\phi \in \exists^+\mathcal{L}^k$ if $A \models \phi$ then $B \models \phi$. Suppose we then defined a category, k-$\mathcal{R}(\sigma)$, which had structures with signature σ as objects and for any two such structures $\mathcal{A}$ and $\mathcal{B}$ there is a unique morphism in k-$\mathcal{R}(\sigma)(\mathcal{A}, \mathcal{B})$ if and only if $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$. By the definition above it is clear that identity and composition laws hold and that this is indeed a category. However, it is not a particularly interesting category. With at most one morphism between any two structures, k-$\mathcal{R}(\sigma)(\mathcal{A}, \mathcal{B})$ is, in fact, a thin category. Thin categories, of which any partially ordered set is an example, are particularly degenerate examples of categories. One consequence of this which makes k-$\mathcal{R}(\sigma)$ particularly uninteresting for descriptive complexity is that isomorphism between two structures $\mathcal{A}$ and $\mathcal{B}$ in a thin category is equivalent to the existence of a map from $\mathcal{A}$ to $\mathcal{B}$ and a map from $\mathcal{B}$ to $\mathcal{A}$. In the category, $\mathcal{R}(\sigma)$ this condition is known as homomorphic equivalence and is notably distinct from isomorphism. In the category k-$\mathcal{R}(\sigma)$, isomorphism would simply be equivalent to the relation $\equiv_{\exists^+\mathcal{L}^k}$. This would appear to be insufficient to capture the types of relations of interest in descriptive complexity. Indeed, while $\Rrightarrow_{\exists^+\mathcal{L}^k}$ is important particularly in the descriptive complexity of `DATALOG` [67], equivalence is usually considered over stronger logics such as $\mathcal{L}^k$ or $\mathcal{C}^k$. As a result, we expect that the "correct" categories for descriptive complexity would also exhibit a different homomorphic equivalence and isomorphism. So these categories must not be thin and we need the maps to be more extensive objects which witness relations of the form $\mathcal{A} \Rrightarrow_L \mathcal{B}$ and $\mathcal{A} \equiv_{L'} \mathcal{B}$.

Exactly such an object is given by the Spoiler-Duplicator games reviewed in the previous chapter. Recall, for example, the following theorem of Kolaitis and Vardi characterising the relation $\Rrightarrow_{\exists^+\mathcal{L}^k}$ in terms of the one-way k-pebble game.

Theorem 3.1. *For any positive integer k and two relational structures $\mathcal{A}$ and $\mathcal{B}$ over a common (finite) signature σ, the following are equivalent*

1. *$\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$; and*
2. *Duplicator has a winning strategy in the one-way k-pebble game from $\mathcal{A}$ to $\mathcal{B}$.*

This result suggests the set of winning strategies for Duplicator in this one-way k-pebble game between $\mathcal{A}$ and $\mathcal{B}$ as a good candidate for the set of morphisms in a richer version of the category k-$\mathcal{R}(\sigma)$. There are two obstacles to creating this category. Firstly, we need an explicit description of what is meant by a "strategy" for Duplicator. Secondly, we need to check that, for whatever notion of strategy we settle on, it satisfies the necessary composition and identity laws to form a category. In the next two sections, we address these two issues for a particular definition of strategy and, in doing so, reveal a comonadic structure.

3.1.2 Duplicator Strategies in the Pebble Game

Recall the one-way k-pebble game of Kolaitis and Vardi on relational structures $\mathcal{A}$ and $\mathcal{B}$ with underlying sets A and B which we reviewed in Chapter 2. Positions in this game are sets of pairs $(a_i, b_i) \in A \times B$ which are indexed by a subset of the numbers $i \in [k]$. This is often pictured as pairs of pebbles labelled from the set $[k]$ with one pebble in each pair placed on an element of A and the corresponding pebble on an element of B. A position is winning for Spoiler if the set $\{(a_i, b_i) \mid i \in [k]\}$ is *not* a partial homomorphism from $\mathcal{A}$ to $\mathcal{B}$. The game starts with the empty position where no pairs are defined. Then in each round, starting with position γ, Spoiler chooses an index $i \in [k]$ and an element $a \in A$. Duplicator responds by choosing an element $b \in B$. Then, letting γ_i be the position γ with the pair indexed by i removed (if it was defined in the first place), the new position at the end of the round is given by $\gamma_i \cup \{(a, b)\}$ where (a, b) is indexed by i. This is usually described by Spoiler picking up the pair of pebbles labelled by i (whether they are in play or not) and placing one of them on $a \in A$ and letting Duplicator place the other one where she chooses in B. Spoiler wins the game if after any round of the game the position is winning for Spoiler. Duplicator wins if she can play the game forever while preventing Spoiler from winning.

As we have just seen that Duplicator wins this game if and only if $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$ and so the winning strategies for Duplicator are a good candidate for the map in our category for $\Rrightarrow_{\exists^+\mathcal{L}^k}$. In this section, we show how to form this category, first by defining a precise notion of Duplicator strategy in this game and secondly by showing how these strategies compose.

Part I : Defining Duplicator strategies There are several possible ways of defining a strategy for Duplicator in the game described above. Indeed, strategies can be deterministic or not and the responses of Duplicator in each round may depend only on the position in that round or on the whole sequence of Spoiler moves up to it. In this presentation, we choose to develop deterministic strategies which take into account the entire history of Spoiler's moves. We call these *deterministic sequential* strategies or just sequential strategies. As we see this formalisation of strategies emphasises the tree-like structure of the logical formulas which underpin this game.

In the game above, each round consists of a Spoiler move followed by a Duplicator move. A Spoiler move is a pair in the set $A \times [k]$ and a Duplicator move is simply an element of B. The history of Spoiler moves in rounds up to and including the nth round is a list $[(a_1, p_1), \ldots, (a_n, p_n)] \in (A \times [k])^n$. Thus the set of all *Spoiler histories* is the set $\mathbf{Sp}_k(A) := (A \times [k])^+$ of non-empty lists of Spoiler moves and for any $s \in \mathbf{Sp}_k(A)$ the last element of the list $\omega(s) = (\omega_{pos}(s), \omega_{peb}(s))$ is the most recent move made by Spoiler in s. We say that a pebble with index $i \in [k]$ is *live* in s if there is some prefix s' of s such that $\omega_{peb}(s') = i$ and we write $\mathbf{lp}(s)$ for the set of live pebbles in s. For any pebble index $i \in \mathbf{lp}(s)$ we say that the i^{th} *live prefix of* s is the longest prefix s^i of s such that $\omega_{peb}(s^i) = i$. The live pebbles s tell us which pebbles are in play after s has been played by Spoiler and the live prefixes s identify the last round in which each pebble in play was moved. We can now define a *deterministic sequential strategy for Duplicator* in the existential k-pebble game as any function $f : \mathbf{Sp}_k(A) \to B$. We should interpret such a strategy as defining for any Spoiler history of $s \in \mathbf{Sp}_k(A)$ of length n, Duplicator's response in the n^{th} round after Spoiler has played the moves specified by s.

To decide whether such a strategy is winning for Duplicator we need to know where the pebbles are positioned after each possible sequence of moves. Suppose $s \in \mathbf{Sp}_k(A)$ is a history of Spoiler moves on A and $f : \mathbf{Sp}_k(A) \to B$ is a Duplicator strategy in the existential k-pebble game between $\mathcal{A}$ and $\mathcal{B}$. The *position after s under f* is defined as the partial function

$$\mathbf{Pos}(s, f) = \{(\omega_{pos}(s^i), f(s^i)) \mid i \in \mathbf{lp}(s)\}.$$

This should be understood as the map which, for each pebble i which is in play after s is played by Spoiler, maps the last position $\omega_{pos}(s^i) \in A$ on which Spoiler placed the pebble i to Duplicator's response $f(s^i) \in B$ in the round where Spoiler last moved pebble i. Now we can say that f is a *winning sequential strategy for Duplicator in the existential k-pebble game* if for all Spoiler histories $s \in \mathbf{Sp}_k(A)$, $\mathbf{Pos}(s, f)$ is a partial homomorphism.

Now that we have a concrete representation of winning Duplicator strategies of the k pebble game from $\mathcal{A}$ to $\mathcal{B}$, we now need to explain how to compose these with winning strategies for the game from $\mathcal{B}$ to $\mathcal{C}$ to get a winning strategy for the game from $\mathcal{A}$ to $\mathcal{C}$. That is what we do in the next section.

Part II: Composing sequential strategies At first glance, given sequential Duplicator winning strategies in the existential k-pebble games from $\mathcal{A}$ to $\mathcal{B}$ and from $\mathcal{B}$ to $\mathcal{C}$ which we would like to compose, there appears to be a mismatch of types. Indeed a sequential strategy from A to B is a map $f_1 : \mathbf{Sp}_k(A) \to B$ and a sequential strategy from $\mathcal{B}$ to $\mathcal{C}$ is a map $f_2 : \mathbf{Sp}_k(B) \to C$. As functions these strategies clearly do not compose. So we need to come up with a new means of composing these objects. We see in this section how to make this composition work.

Informally, it is not difficult to see what to do to create a Duplicator strategy in the existential k-pebble game from $\mathcal{A}$ to $\mathcal{C}$ using the strategies f_1 and f_2. In order to respond to any sequence of Spoiler moves on A, we first use f_1 to respond to Spoiler's moves on B. From an outsider perspective, observing just the structure B, this appears as someone choosing certain indexed pebbles and placing them on elements of B. The sequence of moves which emerges in this way can be seen as a sequence of Spoiler moves on B. Now we can use the strategy f_2 to play on C in response to this virtual Spoiler. The result is a strategy in the existential k-pebble game from $\mathcal{A}$ to $\mathcal{C}$.

Formally, this composition was obtained by transforming the first strategy of the form $f_1 \colon \mathbf{Sp}_k(A) \to B$ into a function of the form $\overline{f_1} \colon \mathbf{Sp}_k(A) \to \mathbf{Sp}_k(B)$ which translates sequences of Spoiler moves on A to sequences of Spoiler moves on B. The composed strategy is then the composition $f_2 \circ \overline{f_1}$. This type of composition, which is natural for strategies may seem strange for functions. However, as we see in the next section, this structure is actually a known construction in category theory. Namely, this is the Kleisli form of a comonad.

Seeing that the condition of being a winning strategy is preserved in this composition is simply a consequence of the composition of partial homomorphisms being a homomorphism. Indeed, after any series of moves $s \in \mathbf{Sp}_k(A)$ the position $\mathbf{Pos}(s, f_1)$ a partial homomorphism, and the position $\mathbf{Pos}(\overline{f_1}(s), f_2)$ is a partial homomorphism. Then $\mathbf{Pos}(s, f_2 \circ \overline{f_1})$ is a partial homomorphism as it is the composition of two partial homomorphisms. We now see how this structure is naturally interpreted as a comonad on $\mathcal{R}(\sigma)$.

3.1.3 Outlines of a comonad

In the last section, we saw how to make the winning Duplicator strategies in the existential k-pebble game into explicit maps between relational structures which compose and form a category which has more structure than the thin category for $\Rrightarrow_{\exists^+\mathcal{L}^k}$ mentioned in the section before. We now see how this category, which we built from natural considerations about how the underlying game is played, can actually be seen as arising from the original category $\mathcal{R}(\sigma)$ by means of a standard category-theoretic object known as a comonad.

Recall the Kleisli triple formulation of a comonad from Chapter 2. This describes a comonad on a category $\mathbf{C}$ by three pieces of data $(\mathbb{P}, \epsilon, \bar{\cdot})$ where $\mathbb{P}$ is a functor on the base category $\mathbf{C}$, $\epsilon \colon \mathbb{P} \implies \mathbf{id_C}$ is a natural transformation and $\bar{\cdot}$ is an operator which transforms maps of the form $\mathbb{P}A \to B$ into maps $\mathbb{P}A \to \mathbb{P}B$. To form a comonad these pieces of data need to satisfy certain comonad laws which we recall later. These laws are exactly those required to form a category $\mathcal{K}(\mathbb{P})$ where the objects are exactly those from $\mathbf{C}$, the maps $\mathcal{K}(\mathbb{P})(A, B)$ are exactly those maps in $\mathbf{C}(\mathbb{P}A, B)$, the identity on A is the map ϵ_A and composition of maps $f \in \mathcal{K}(\mathbb{P})(A, B)$ and $g \in \mathcal{K}(\mathbb{P})(B, C)$ is $g \circ \overline{f}$. The category is known as the *Kleisli category of the comonad* $\mathbb{P}$. This structure should be immediately reminiscent of the category we formed by composing strategies in the previous section. In the rest of this section, we show how to make this connection fully formal in the case of the existential k-pebble game. This is a reconstruction of the pebbling comonad $\mathbb{P}_k$ from Abramsky, Dawar, and Wang [6] from a slightly different starting point. Their original construction, which we see in the next section, uses a different but equivalent formulation of a comonad.

We give this construction in two parts. First, we define the data of the comonad by constructing a relational structure $\mathbb{P}_k\mathcal{A}$ on the set of Spoiler histories $\mathbf{Sp}_k(A)$ such that the homomorphisms $\mathcal{R}(\sigma)(\mathbb{P}_k\mathcal{A}, \mathcal{B})$ are precisely the winning strategies in the one-way k-pebble game and then formally defining the identity strategies ϵ_A and the operator $\bar{\cdot}$ which we sketched in the last section. After this, we prove that $\mathbb{P}_k$ is a comonad on $\mathcal{R}(\sigma)$ by showing that this data satisfies the comonad laws for a Kleisli triple.

Part I: Defining the data The exposition of sequential strategies for Duplicator in the one-way k-pebble game from the last section gives us a guide for how to construct the relational structure $\mathbb{P}_k\mathcal{A}$. Indeed, taking the elements of this structure to be $\mathbf{Sp}_k(A) = (A \times [k])^+$ as we had in the last section gives, for any $\mathcal{A}$ and $\mathcal{B}$, the set of functions $B^{\mathbb{P}_k A}$ is precisely the set of sequential Duplicator strategies from $\mathcal{A}$ to B. We now see that we can lift the relations from $\mathcal{A}$ to $\mathbb{P}_k\mathcal{A}$ in such a way that the homomorphisms in this set are exactly the winning sequential strategies for Duplicator in the existential k-pebble game.

We saw in the last section that the condition for Duplicator to win the existential k-pebble game from $\mathcal{A}$ to $\mathcal{B}$ is that for each history of Spoiler moves $s \in \mathbf{Sp}_k(A)$, the position of the pebbles after Duplicator has responded to the last play of s is a partial homomorphism. We also saw that for any Duplicator strategy f this position, $\mathbf{Pos}(s, f)$ was determined by the responses provided by f to the live prefixes of s, $\mathbf{LP}(s) = \{s^i \mid i \in \mathbf{lp}(s)\}$. Recall that each live prefix s^i represents the last time in the sequence of moves s where Spoiler has repositioned the pebble indexed by i and so the element $\omega_{pos}(s^i) \in A$ is the position of Spoiler's pebble i after the sequence of moves s. This lets us define $\mathbb{P}_k$ as follows.

Definition 3.2. *For any $\mathcal{A} \in \mathcal{R}(\sigma)$, the structure $\mathbb{P}_k\mathcal{A} \in \mathcal{R}(\sigma)$ has the underlying set $\mathbf{Sp}_k(A)$ and for any relational symbol R in σ with arity n, $(s_1, \ldots, s_n) \in R^{\mathbb{P}_k\mathcal{A}}$ if and only if there is some $s \in \mathbf{Sp}_k(A)$ such that each s_i is a live prefix of s and $(\omega_{pos}(s_1), \ldots, \omega_{pos}(s_n)) \in R^{\mathcal{A}}$. For any homomorphism $f\colon \mathcal{A} \to \mathcal{B}$, the map $\mathbb{P}_k f\colon \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{B}$ is defined on any $s = [(a_1, p_1), \ldots (a_m, p_m)]$ as*

$$\mathbb{P}_k f(s) = [(f(a_1), p_1), \ldots (f(a_m), p_m)]$$

We now check that this is indeed a functor.

Lemma 3.3. *$\mathbb{P}_k$ is a functor.*

Proof. We need to check that for any homomorphism $f\colon \mathcal{A} \to \mathcal{B}$, $\mathbb{P}_k f$ is a homomorphism and that for any homomorphisms $f\colon \mathcal{A} \to \mathcal{B}$ and $g\colon \mathcal{B} \to \mathcal{C}$, we have $\mathbb{P}_k(g \circ f) = (\mathbb{P}_k g) \circ (\mathbb{P}_k f)$.

To see that $\mathbb{P}_k f$ is a homomorphism, first note that $\mathbb{P}_k f$ preserves length and the prefix relation on lists in $\mathbb{P}_k\mathcal{A}$ and that it doesn't change the sequence of pebbles in the list, in the sense that projecting $s \in \mathbb{P}_k\mathcal{A}$ and $\mathbb{P}_k f(s) \in \mathbb{P}_k\mathcal{B}$ pointwise onto the second component, we get the same list in $[k]^+$. Together this means that if s' is a live prefix of s then $\mathbb{P}_k f(s')$ is a prefix of $\mathbb{P}_k f(s)$.

Now, by definition of $\mathbb{P}_k\mathcal{A}$, any tuple $(s_1, \ldots, s_n)$ is in $R^{\mathbb{P}_k\mathcal{A}}$ for some relational symbol R, if and only if there is some $s \in \mathbb{P}_k\mathcal{A}$ such that each of the s_i is a live prefix of s and we have that $(\omega_{pos}(s_1), \ldots, \omega_{pos}(s_n)) \in R^{\mathcal{A}}$. As $\mathbb{P}_k f$ preserves the relation of being a live prefix, we have that each of the images $\mathbb{P}_k f(s_1), \ldots, \mathbb{P}_k f(s_n)$ are live prefixes of $\mathbb{P}_k f(s)$. Furthermore, as $\omega_{pos}(\mathbb{P}_k f(s_i)) = f(\omega_{pos}(s_i))$ and f is a homomorphism we have that $(\omega_{pos}(\mathbb{P}_k f(s_1)), \ldots, \omega_{pos}(\mathbb{P}_k f(s_n)))$ is in $R^{\mathcal{B}}$ and so $(\mathbb{P}_k f(s_1), \ldots, \mathbb{P}_k f(s_n)) \in R^{\mathbb{P}_k\mathcal{B}}$, as required.

To see that $\mathbb{P}_k$ behaves well with respect to composition, we note that $\mathbb{P}_k f$ applies f pointwise on lists and so composition of $\mathbb{P}_k f$ and $\mathbb{P}_k g$ is just a pointwise composition of f and g. In particular on any $s \in \mathbb{P}_k\mathcal{A}$

$$\begin{aligned}
\mathbb{P}_k g(\mathbb{P}_k f(s)) &= \mathbb{P}_k g([(f(a_1), p_1), \ldots, (f(a_m), p_m)]) \\
&= [(g(f(a_1)), p_1), \ldots, (g(f(a_m)), p_m)] \\
&= [((g \circ f)(a_1), p_1), \ldots, ((g \circ f)(a_m), p_m)] \\
&= \mathbb{P}_k(g \circ f)(s)
\end{aligned}$$

as required.

□

We would like to say that any homomorphism $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ is a winning strategy for Duplicator. This is nearly true but with one small caveat. Suppose that there are two distinct live prefixes s^i and s^j of some $s \in \mathbb{P}_k\mathcal{A}$ but $\omega_{pos}(s^i) = \omega_{pos}(s^j)$. This can be thought of as s being a sequence of moves which results in two distinct pebbles being placed by Spoiler on the same element of A. However, there is no relation in $\mathbb{P}_k\mathcal{A}$ which forces the Duplicator responses $f(s^i)$ and $f(s^j)$ to be equal. This means that even if $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ is a homomorphism in general $\mathbf{Pos}(s, f)$ need not even be a partial function, let alone a partial homomorphism. This inspires the next definition.

Definition 3.4. *For any signature σ and any $\mathcal{A} \in \mathcal{R}(\sigma)$, the* I-structure *$\mathcal{A}$ is the structure $\mathcal{A} \in \mathcal{R}(\sigma^+)$ where $\sigma^+ = \sigma \cup \{I\}$, $I^{\mathcal{A}} = \{(a, a) \mid a \in A\}$ and for all $R \in \sigma$ the σ^+-structure agrees with the σ-structure $\mathcal{A}$.*

As defined, an I-structure is a relational structure with an extra relation I observing the equality of elements in A. We now see that the winning sequential strategies for Duplicator in the game over σ structures are precisely the homomorphisms $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ for I-structures $\mathcal{A}$ and $\mathcal{B}$.

Lemma 3.5. *For any σ-structures $\mathcal{A}, \mathcal{B}$, $f\colon \mathbf{Sp}_k(A) \to B$ is a winning sequential strategy for Duplicator in the existential k-pebble game from A to B if and only if $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ is a homomorphism for the I-structures $\mathcal{A}$ and $\mathcal{B}$.*

Proof. f is a winning strategy for Duplicator in the existential k-pebble game if and only if for every $s \in \mathbf{Sp}_k(A)$, $\mathbf{Pos}(s, f)$ is a partial homomorphism. So f is not winning for Duplicator precisely when there is some s such that $\mathbf{Pos}(s, f)$ is either (a) not a partial function or (b) is a partial function but not a partial homomorphism. The first case is equivalent to there being an s and two distinct live prefixes s^i and s^j of s such that $\omega_{pos}(s^i) = \omega_{pos}(s^j)$ but $f(s^i) \neq f(s^j)$. This is equivalent to the statement that $(s^i, s^j) \in \mathbb{P}_k\mathcal{A}$ but $(f(s^i), f(s^j)) \notin I^B$. So, case (a) is equivalent to $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ not preserving the I relation. Case (b) on the other hand is equivalent to the existence of some tuple $(s_1, \ldots, s_m)$ which are all live prefixes of s such that $(\omega_{pos}(s_1), \ldots, \omega_{pos}(s_m)) \in R^{\mathcal{A}}$ but $(f(s_1), \ldots f(s_m)) \notin R^{\mathcal{B}}$. This is equivalent to there existing a tuple $(s_1, \ldots, s_m) \in R^{\mathbb{P}_k\mathcal{A}}$ such that $(f(s_1), \ldots f(s_m)) \notin R^{\mathcal{B}}$ which is simply the statment that f does not preserve the relation R. Putting these two cases together we have that f is not a winning strategy for the existential k-pebble game if and only if there is some relational symbol S in σ^+ such that f does not preserve S.

□

Having defined the relational structure of $\mathbb{P}_k\mathcal{A}$ and shown that homomorphisms can indeed be used to classify winning strategies in the existential k-pebble game, it remains to define ϵ and $\bar{\cdot}$.

For any A in $\mathcal{R}(\sigma)$ $\epsilon_A\colon \mathbb{P}_k\mathcal{A} \to A$ should be the identity strategy for Duplicator on A. The intuitively obvious definition for this is the strategy where Duplicator responds to Spoiler pebbling an element of A by pebbling the exact same element. Formally defined, this is the function which takes any sequence of Spoiler moves $s \in \mathbf{Sp}_k(A)$ to the most recently pebbled element, $\omega_{peb}(s)$. It is clear from the definitions that this is a homomorphism and ϵ is natural, i.e. for any homomorphism $f\colon \mathcal{A} \to \mathcal{B}$, $f \circ \epsilon_A = \epsilon_B \circ \mathbb{P}_k f$.

The definition of $\bar{\cdot}$ was covered informally in the last section but we give the formal definition here. The aim is to turn any homomorphism $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ into a homomorphism $\overline{f}\colon \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{B}$. Following the procedure in the last section, we define the operator as follows.

Definition 3.6. *For any homomorphism $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ we define the map $\overline{f}\colon \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{B}$ by defining its action on an arbitrary sequence $s = [(a_1, p_1), \ldots, (a_n, p_n)] \in \mathbb{P}_k\mathcal{A}$ as*

$$\overline{f}(s) = [(f(s_1), p_1), \ldots, (f(s_n), p_n)]$$

where, for each $1 \leq i \leq n$, s_i is the prefix of the first i elements of s.

To see that the map $\overline{f}$ is indeed a homomorphism, we note that $\overline{f}$ acts pointwise on the lists in $\mathbb{P}_k\mathcal{A}$ and doesn't change any of the pebble indices. Thus, we can say that, for any $s \in \mathbb{P}_k\mathcal{A}$, the live prefixes of s are preserved by $\overline{f}$ in the following way

$$\mathbf{LP}(\overline{f}(s)) = \{\overline{f}(s^i) \mid s^i \in \mathbf{LP}(s)\}.$$

This means that for any related tuple $(s_1, \ldots, s_m) \in R^{\mathbb{P}_k\mathcal{A}}$ with $s_j \in \mathbf{LP}(s)$ for all $1 \leq j \leq m$ and $(\omega_{pos}(s_1), \ldots, \omega_{pos}(s_m)) \in R^{\mathcal{A}}$. Then we have that $\overline{f}(s_j) \in \mathbf{LP}(\overline{f}(s))$ for all $1 \leq j \leq m$ and

$$(\omega_{pos}(\overline{f}(s_1)), \ldots, \omega_{pos}(\overline{f}(s_m))) = (f(\omega_{pos}(s_1)), \ldots, f(\omega_{pos}(s_m)))$$

which is an element of $R^{\mathcal{B}}$ because f is a homomorphism.

Having defined the data that make up a comonad in Kleisli form we now complete the proof that this structure is indeed a comonad by verifying that it satisfies the comonad laws from Chapter 2.

Part II: Checking the comonad laws Recall the comonad laws in for a comonad $\mathbb{P} = (\mathbb{P}, \epsilon, \bar{\cdot})$ in Kleisli form from Definition 2.8:

1. $\overline{\epsilon_A} = \mathbf{id}_{\mathbb{P}A}$ for any structure $\mathcal{A} \in \mathcal{R}(\sigma)$
2. $\epsilon_B \circ \overline{f} = f$ for any winning strategy $f \colon \mathbb{P}\mathcal{A} \to \mathcal{B}$
3. $\overline{g \circ \overline{f}} = \overline{g} \circ \overline{f}$ for any winning strategies $f \colon \mathbb{P}\mathcal{A} \to \mathcal{B}$ and $g \colon \mathbb{P}\mathcal{B} \to \mathcal{C}$

Before we prove that the Kleisli triple $(\mathbb{P}_k, \epsilon, \bar{\cdot})$, defined in the last section, satisfies these conditions, let's reflect briefly on what these conditions are saying for the composition of sequential strategies for Duplicator in the existential k-pebble game. For two strategies f, g in the existential k-pebble games from $\mathcal{A}$ to $\mathcal{B}$ and from $\mathcal{B}$ to $\mathcal{C}$ respectively we have seen that their composition is given by the strategy $g \circ \overline{f}$ in the game from $\mathcal{A}$ to $\mathcal{C}$. The comonad rules can now be interpreted as ensuring that this strange-looking composition does actually form a category. In particular, the first two rules are saying that if we compose the identity strategy on the left or right of any strategy f we get no more or less than f. The final rule ensures that composition of strategies is associative. We now prove that these hold in the case of the triple $(\mathbb{P}_k, \epsilon, \bar{\cdot})$ defined above.

Lemma 3.7. *The triple* $(\mathbb{P}_k, \epsilon, \bar{\cdot})$ *satisfies the comonad laws.*

Proof. 1. For any $s \in \mathbb{P}_k\mathcal{A}$, $\epsilon_A(s) = \omega_{pos}(s)$. When we lift this using $\bar{\cdot}$ we have that, for any $s = [(a_1, p_1), \ldots, (a_m, p_m)]$

$$\overline{\epsilon_A}(s) = [(\omega_{pos}(s_1), p_1), \ldots, (\omega_{pos}(s_m), p_m)]$$

where $s_i = [(a_1, p_1), \ldots, (a_i, p_i)]$. From the definition, for any $1 \leq i \leq m$, $\omega_{pos}(s_i) = a_i$ and so $\overline{\epsilon_A}(s) = s$ and $\overline{\epsilon_A} = \mathbf{id}_A$, as required.

2. For the same s with prefixes $s_1, s_2, \ldots, s_m$ as above we have that $\epsilon_A(\overline{f}(s)) = f(s_m)$. However, s_m is the prefix of the first m elements of s which has only m elements, so $s_m = s$ and $\epsilon_A(\overline{f}(s)) = f(s)$, as required.

3. To prove this we note first that for any $f \colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, $\overline{f}$ preserves the length and prefix relation of list in $\mathbb{P}_k\mathcal{A}$. These facts are both clear from the definition which leaves the sequence of pebble indices intact and applies f component-wise to prefixes. The important consequence of this is that for any $s = [(a_1, p_1), \ldots, (a_m, p_m)]$ with prefixes $s_1, s_2, \ldots, s_m$ the list $\overline{f}(s)$ has prefixes $\overline{f}(s_1), \overline{f}(s_2), \ldots, \overline{f}(s_m)$. Now when we consider the left hand side of the identity we are trying to prove we have

$$\overline{g \circ \overline{f}}(s) = [(g(\overline{f}(s_1)), p_1), \ldots, (g(\overline{f}(s_m)), p_m)]$$

and, as the prefixes of $\overline{f}(s)$ are exactly $\overline{f}(s_1), \overline{f}(s_2), \ldots, \overline{f}(s_m)$, this is the same as $\overline{g}(\overline{f}(s))$, as required. □

Now we conclude this section by stating the important consequence of this result which is effectively a restatement of Theorem 4 and Proposition 10 from [6].

Theorem 3.8. $(\mathbb{P}_k, \epsilon, \bar{\cdot})$ *is a comonad on* $\mathcal{R}(\sigma^+)$ *where the subcategory on* I*-structures of the Kleisli category of* $\mathbb{P}_k$ *is equivalent to a category where objects are* σ*-structures and maps between* $\mathcal{A}$ *and* $\mathcal{B}$ *are winning sequential strategies for Duplicator in the existential* k*-pebble game between* $\mathcal{A}$ *and* $\mathcal{B}$

Proof. This is a direct consequence of Lemmas 3.7 and 3.5. □

In this section, we have arrived at the comonadic approach to Spoiler-Duplicator games as a consequence of formalising a very natural notion of composition for Duplicator winning strategies. We now see the rather unexpected power of this natural construction to connect seemingly unrelated notions in finite model theory by reviewing the results of Abramsky, Dawar and Wang's seminal paper [6] and the work which has followed in this research programme.

3.2 $\mathbb{P}_k$: the prototypical game comonad

In this section, we study the properties of the comonad $\mathbb{P}_k$ derived in the previous section. This section is largely a recollection of the main results of Abramsky, Dawar and Wang. This example is the prototype for several other game comonads and we see in the next section that the same properties of different game comonads also yield interesting analogues from finite model theory.

3.2.1 Definition of $\mathbb{P}_k$

In the original paper introducing the pebbling comonad [6], Abramsky, Dawar and Wang give a different presentation to that given above. Instead of defining the comonad in terms of the composition of Duplicator strategies in the k-pebble game they define the comonad in terms of a counit and comultiplication which are dual to the more familiar unit and multiplication used to define monads. This means giving a triple $(\mathbb{P}_k, \epsilon, \delta)$ where $\mathbb{P}_k$ is an endofunctor on the category $\mathcal{R}(\sigma)$ and $\epsilon\colon \mathbb{P}_k \implies 1$ and $\delta\colon \mathbb{P}_k \implies \mathbb{P}_k\mathbb{P}_k$ are natural transformations satisfying the comonad laws from Definition 2.8. We call this an Eilenberg-Moore triple and call this style of definition the Eilenberg-Moore (EM) form.

The data for $\mathbb{P}_k$ and ϵ are exactly the same as given in the last section which we restate here along with the definition of δ.

Definition 3.9 (Definition of $\mathbb{P}_k$ in EM form)**.** *The k-pebbling comonad is defined by the Eilenberg-Moore triple $(\mathbb{P}_k, \epsilon, \delta)$ consisting of*

- *an **endofunctor** $\mathbb{P}_k$ on $\mathcal{R}(\sigma)$ where*
  - *$\mathbb{P}_k\mathcal{A}$ has the underlying set $(A \times [k])^+$ and a tuple $(s_1, \dots, s_n)$ is in $R^{\mathbb{P}_k\mathcal{A}}$ if and only is $(\omega_{pos}(s_1), \dots \omega_{pos}(s_n))$ is in $R^{\mathcal{A}}$ and for each $i, j \in [n]$ either s_i is a live prefix of s_j or vice versa; and*
  - *$\mathbb{P}_k f$ for a homomorphism $f\colon \mathcal{A} \to \mathcal{B}$ applies f point-wise to the first component of each element of a list in $\mathbb{P}_k\mathcal{A}$,*
- *a **counit** $\epsilon\colon \mathbb{P}_k \implies 1$ where $\epsilon_{\mathcal{A}}\colon \mathbb{P}_k\mathcal{A} \to \mathcal{A}$ sends $s \in \mathbb{P}_k\mathcal{A}$ to $\omega_{pos}(s)$; and*
- *a **comultiplication** $\delta\colon \mathbb{P}_k \implies \mathbb{P}_k\mathbb{P}_k$ where $\delta_{\mathcal{A}}$ sends a list $s = [(a_1, p_1), \dots, (a_m, p_m)] \in \mathbb{P}_k\mathcal{A}$ to the list $[(s[: 1], p_1), \dots, (s[: m], p_m)] \in \mathbb{P}_k\mathbb{P}_k\mathcal{A}$.*

With this presentation, it remains to check that this EM triple satisfies the comonad laws. Abramsky, Dawar and Wang do exactly this in proving the following theorem.

Theorem 3.10 (Theorem 4 of [6])**.** *For any positive integer k, $(\mathbb{P}_k, \epsilon, \delta)$ defines a comonad on the category $\mathcal{R}(\sigma)$ for any finite relational signature σ.*

In the definition of this comonad there are two free parameters which alter the comonad in question. The first is the resource parameter k. In the definition above it is easy to see that k influences the amount of local structure from $\mathcal{A}$ which is lifted to each list in $\mathbb{P}_k\mathcal{A}$. This gives the impression that as k increases the functor $\mathbb{P}_k$ preserves more (or rather forgets less) of the information in the original structure. Abramsky, Dawar and Wang make explicit this refinement as k increases by showing that $(\mathbb{P}_k)_{k\in\omega}$ is, in fact, a graded comonad with inclusion maps giving the natural transformations $\iota^{l,k}\colon \mathbb{P}_l \implies \mathbb{P}_k$ for every $l \leq k$. As we'll see in the next sections, this resource parameter also determines the fragments of logic and structural parameters obtained by other aspects of the comonad. This grading of the comonad then represents, in a formal way, a grading of these related objects.

The second parameter is the signature σ. In the original pebbling comonad paper this is fixed and there is no attempt to formally relate instances of the comonad for different signatures. When we come to talk about interpretations in Chapter 5 we see some possible formal approaches to this parameter.

For our purposes, the following in an important proposition. It establishes that the pebbling comonad defined by Abramsky, Dawar and Wang as given in Definition 3.9 is equivalent to that given in the previous section in Definition 3.2.

Proposition 3.11. *The EM triple $(\mathbb{P}_k, \epsilon, \delta)$ and the Kleisli triple $(\mathbb{P}_k, \epsilon, \bar{\cdot})$ both define the same comonad on $\mathcal{R}(\sigma)$.*

Proof. We recall from Chapter 2 that any comonad can be represented equivalently as an EM or Kleisli triple. As such, for any comonad given in EM form we can define its Kleisli form as follows. If $\mathbb{T} = (\mathbb{T}, \epsilon^{\mathbb{T}}, \delta^{\mathbb{T}})$ is a comonad in EM form then the Kleisli coextension in $\mathbb{T}$ for a Kleisli map $f \colon \mathbb{T}A \to B$ is $\overline{f}^{\mathbb{T}} = \mathbb{T}f \circ \delta^{\mathbb{T}}_A$. Applying this construction to the EM triple $\mathbb{P}_k = (\mathbb{P}_k, \epsilon, \delta)$ we get the coextension $\overline{f}^{\mathbb{P}_k}$ which acts as follows on the element $s = [(a_1, p_1), \ldots, (a_m, p_m)] \in \mathbb{P}_k\mathcal{A}$

$$\begin{aligned}\overline{f}^{\mathbb{P}_k}(s) &= \mathbb{P}_k f(\delta_A(s)) \\ &= [(f(s[:1]), p_1), \ldots (f(s[:1]), p_1)]\end{aligned}$$

which is exactly the definition of $\overline{f}$ given in the previous section. As $\mathbb{P}_k$ and ϵ are common to both triples it is immediate that the definition of a comultiplication from $(\mathbb{P}_k, \epsilon, \bar{\cdot})$ would give exactly δ. □

Both of the constructions we have seen for the comonad $\mathbb{P}_k$ are inspired by the question of how to give a category-theoretic interpretation to the winning strategies for Duplicator in the k-pebble game. Indeed, we have seen in detail in the last section how to do this for one form of the construction. What then is the advantage of this structure being a comonad, in particular, and not any other type of construction? One significant advantage is that any comonad comes with some naturally related categories which in the case of a game comonad provide fruitful ways to study the underlying games.

As noted in any standard treatment of comonads, for example [70], any comonad arises from an adjunction between the underlying category of the comonad and some other category. In fact there is a whole category, $\mathbf{Adj}(\mathbb{P}_k)$ of such adjuctions. Two standard constructions exist in category theory which give the initial and terminal adjunctions. The associated categories are called the Kleisli and Eilenberg-Moore categories and we see in the next two parts of this section that these are extremely interesting in the case of $\mathbb{P}_k$.

3.2.2 Kleisli category of $\mathbb{P}_k$

The first of these categories which we look at is the category used in the initial adjunction of $\mathbf{Adj}(\mathbb{P}_k)$, which is usually known as the Kleisli category. We see in this section that this category for $\mathbb{P}_k$ relates the comonad to the expressive power of various fragments of logic where the resource parameter k controls the power of those fragments by limiting the number of variables. As a result we sometimes refer to this category, following the commentary of Abramsky and Shah [11] as the *Power category of* $\mathbb{P}_k$.

This category is defined as follows.

Definition 3.12 (The Kleisli Category of $\mathbb{P}_k$)**.** *For any integer k and finite signature σ, the category $\mathcal{K}(\mathbb{P}_k)$, for the pebbling comonad $\mathbb{P}_k$ on $\mathcal{R}(\sigma)$ has the same objects as $\mathcal{R}(\sigma)$ and for any two structures $\mathcal{A}, \mathcal{B}$, the morphisms between them are the homomorphisms $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$. Composition of morphisms $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ and $g\colon \mathbb{P}_k\mathcal{B} \to \mathcal{C}$ is given by $g \circ \overline{f}$ and the identity on $\mathcal{A}$ is ϵ_A.*

This is exactly the same category that we derived in Section 3.1. We saw in that section that the morphisms in this category were equivalent to Duplicator winning strategies in the one-way pebble game for $\exists^+\mathcal{L}^k$. This fact about the Kleisli category of $\mathbb{P}_k$ was also known in the original pebbling comonad paper. We call this result the *Morphism Power Theorem for* $\mathbb{P}_k$.

Theorem 3.13 (Morphism Power Theorem of $\mathbb{P}_k$)**.** *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following are equivalent for any positive integer k:*

1. *There is a Kleisli morphism $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$,*
2. *Duplicator has a winning strategy in $\exists^+\mathbf{Peb}_k(\mathcal{A}, \mathcal{B})$, and*
3. *$\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$.*

As we showed earlier in this chapter, the comonad $\mathbb{P}_k$ arises naturally out of trying to construct a category which handles the composition of Duplicator strategies in this game, so this result is not entirely surprising. The next result however shows us the first tangible benefit of this approach, namely that the isomorphisms in the Kleisli category give us a novel characterisation for a totally different game. This game is the k-pebble bijection game, $\mathbf{Bij}_k$, which captures equivalence in the k-variable infinitary counting logic $\mathcal{C}^k$. We call the following result, which was first proved by Abramsky, Dawar and Wang [6], the *Isomorphism Power Theorem for* $\mathbb{P}_k$.

Theorem 3.14 (Isomorphism Power Theorem of $\mathbb{P}_k$)**.** *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following are equivalent for any positive integer k:*

1. *There is a Kleisli isomorphism $\mathcal{A} \cong_{\mathcal{K}(\mathbb{P}_k)} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$,*

2. *Duplicator has a winning strategy in* $\mathbf{Bij}_k(\mathcal{A}, \mathcal{B})$*, and*

3. $\mathcal{A} \equiv_{\mathcal{C}^k} \mathcal{B}$.

The restriction to I-structures in both of these results is a common theme throughout the story of game comonads. As outlined earlier, this is done to ensure that the positions in the pebble game define partial functions. If we dropped this restriction and considered all homomorphisms in $\mathcal{K}(\mathbb{P}_k)[\sigma]$, the analogous "Morphism Power Theorem" would characterise the $\Rrightarrow$ relation for the logic $\exists^+\mathcal{L}^k$ *without equality* and, similarly, isomorphisms would capture $\equiv$ for $\mathcal{C}^k$ without equality. To see this, consult, for example [67]. As these logics without equality are less frequently studied objects in finite model theory we tend to stick to the subcategory on I-structures of $\mathcal{K}(\mathbb{P}_k)[\sigma^+]$. Abramsky and Shah [11] demonstrated that this category is the Kleisli category of a *relative comonad* $\mathbb{P}_k^+ = \mathbb{P}_k \circ J$ where $J \colon \mathcal{R}(\sigma) \to \mathcal{R}(\sigma^+)$. We call this category the *strict Power category of* $\mathbb{P}_k$ and write it $\mathcal{K}(\mathbb{P}_k^+)$.

The two results in this section relate the Kleisli category of $\mathbb{P}_k$ to the logics $\exists^+\mathcal{L}^k$ and $\mathcal{C}^k$ which previously have not been studied as closely related logics. We also see that the resource parameter k in $\mathbb{P}_k$ controls the expressive power of these logics by controlling the number of variables in each. We now see that at the other extreme of $\mathbf{Adj}(\mathbb{P}_k)$ we get get a very different category which has deep relations to structural decompositions.

3.2.3 Eilenberg-Moore category of $\mathbb{P}_k$

The second important category related to $\mathbb{P}_k$ is the category used in the terminal adjunction in $\mathbf{Adj}(\mathbb{P}_k)$. This is traditionally called the Eilenberg-Moore category of the comonad, which we write $\mathbf{EM}(\mathbb{P}_k)$. We see in this section that this category relates $\mathbb{P}_k$ to treewidth, a well-studied structural parameter on on relational structures. Because of this relation we sometimes refer to $\mathbf{EM}(\mathbb{P}_k)$ as the *Structure category of* $\mathbb{P}_k$. It is defined as follows.

Definition 3.15 (Coalgebras and the Eilenberg-Moore category of $\mathbb{P}_k$)**.** *For any integer k and finite signature σ, for the pebbling comonad $\mathbb{P}_k$ on $\mathcal{R}(\sigma)$ a* coalgebra *of $\mathbb{P}_k$ is a morphism $\alpha \colon \mathcal{A} \to \mathbb{P}_k\mathcal{A}$ satisfying the following counit and comultiplication laws:*

- *(Counit)* $\epsilon_A \circ \alpha = \mathbf{id}_A$*,*
- *(Comultiplication)* $\delta_A \circ \alpha = (\mathbb{P}_k\alpha) \circ \alpha$*.*

A coalgebra morphism *from $\alpha \colon \mathcal{A} \to \mathbb{P}_k\mathcal{A}$ to $\beta \colon \mathcal{B} \to \mathbb{P}_k\mathcal{B}$ is a homomorphism $f \colon \mathcal{A} \to \mathcal{B}$ such that $\mathbb{P}_k f \circ \alpha = \beta \circ f$.*

The category $\mathbf{EM}(\mathbb{P}_k)$ has as objects the coalgebras of $\mathbb{P}_k$ and as morphisms the coalgebra transformations between them. The identity on any $\alpha \colon \mathcal{A} \to \mathbb{P}_k\mathcal{A}$ is the identity map $\mathbf{id}_A$

and the composition of coalgebra morphisms is simply composition in σ which commutes with the coalgebra morphism condition.

In the last section, we showed how morphisms in the Power category related $\mathbb{P}_k$ to the expressive power of different logics but that the objects in this category were not especially interesting. In the case of the Structure category, we find the situation reversed and it is the objects, i.e. the coalgebras, which capture the important links to structural parameters. Indeed, the existence of a $\mathbb{P}_k$-coalgebra $\alpha\colon \mathcal{A} \to \mathbb{P}_k\mathcal{A}$ can be seen as proof that the structure A is sufficiently simple to be embedded, in a recoverable way into the unfolded structure $\mathbb{P}_k$. Indeed, Abramsky, Dawar and Wang show that the existence of such a coalgebra corresponds to the existence of a tree decomposition of width $k-1$. We state now the many equivalent conditions implied by this characterisation, as outlined in Chapter 2.

Theorem 3.16 (Coalgebra Structure Theorem for $\mathbb{P}_k$)**.** *For a relational structure $\mathcal{A}$, the following are equivalent for any positive integer k:*

- *There is a coalgebra $\alpha\colon \mathcal{A} \to \mathbb{P}_k\mathcal{A}$,*
- *There is a tree decomposition of $\mathcal{A}$ with bags of size at most k,*
- *k cops with helicopters can win the cops and robbers game on $\mathcal{A}$, and*
- *The treewidth of $\mathcal{A}$ is less than k.*

It is easy to see that as the resource parameter k increases the equivalent conditions in the above result are easier to satisfy for any given $\mathcal{A}$. Indeed, it is well known that any finite structure $\mathcal{A}$ of size n has treewidth $\leq n$. This result motivates the following definition of a structural parameter for $\mathbb{P}_k$ which appears in Abramsky, Dawar and Wang's original paper.

Definition 3.17. *For relational structure $\mathcal{A}$ the* pebble number *(or* $\mathbb{P}$-parameter*) of $\mathcal{A}$ is defined as*

$$f_{\mathbb{P}}(\mathcal{A}) := \min\{k \mid \mathcal{A} \textit{ has a } \mathbb{P}_k\textit{-coalgebra}\}.$$

An easy corollary of Theorem 3.16 is the following result relating this parameter to treewidth.

Theorem 3.18 (Parameter Structure Theorem for $\mathbb{P}_k$)**.** *For any relational structure $\mathcal{A}$,*

$$\mathbf{tw}(\mathcal{A}) = f_{\mathbb{P}}(\mathcal{A}) - 1$$

It is important to highlight that the two structure theorems here relate the *existence* of $\mathbb{P}_k$-coalgebras to bounds on treewidth but that there is no one-to-one correspondence between, for example, $\mathbb{P}_k$-coalgebras and tree decompositions of bag size k. Abramsky and Shah give such an exact characterisation of individual coalgebras, as k-pebble forest covers of the structure $\mathcal{A}$. We revisit this notion when we introduce new comonads later in this thesis.

We have seen in this section that the construction of $\mathbb{P}_k$ as a comonad has created a number of interesting connections in both logic and finite model theory. The two "Power" theorems relate Kleisli morphisms of $\mathbb{P}_k$ to different fragments of infinitary fixed-variable logics and the two "Structure" theorems relate the coalgebras of $\mathbb{P}_k$ to tree decompositions and to the important structural parameter treewidth. These connections are important motivation for finding further game comonads which follow this pattern. In the next section, we review some of the relevant comonads which have been found to follow the mould set by $\mathbb{P}_k$.

3.3 Other game comonads

Since the discovery of the $\mathbb{P}_k$ comonad, other similar "game comonads" have been developed, giving categorical semantics to different model comparison games in finite model theory. Structure and Power theorems for these comonads, analogous to those for $\mathbb{P}_k$ have established new links between different structural decompositions and logical fragments. A recent summary is given by Abramsky in [2].

As this thesis focuses on game comonads for descriptive complexity theory and the hierarchy of logics approaching `PTIME`, we look only at a number of relevant examples. In particular, given the importance of the logics $\mathcal{C}^k$ and $\mathcal{L}^k$ from the Power category of $\mathbb{P}_k$ in this hierarchy, we limit our attention in this section to the game comonads which are directly comparable (via comonad morphisms) with $\mathbb{P}_k$. There are three such comonads which we review in this section, as highlighted in Figure 3.1. These are

- $\mathbb{E}_k$, a comonad for the k round Ehrenfeucht-Fraïssé game,
- $\mathbb{PR}_k$, a comonad for Dalmau's k pebble relation game, and
- $\mathbb{P}_{n,k}$, a comonad for the n round, k pebble game.

$\mathbb{E}_k$ was first introduced by Abramsky and Shah in [11], $\mathbb{PR}_k$ by Montacute and Shah in [77], and $\mathbb{P}_{n,k}$ by Paine in [81]. In a sense, all of these comonads are different weakenings of $\mathbb{P}_k$ as the games captured are all easier for Duplicator to win and so the logics described by their Power categories are less expressive. Prior to this thesis, no comonads were known to strictly extend the power of $\mathbb{P}_k$. The first such comonad is the major contribution of Chapter 6.

$$\begin{tikzcd}
& \mathbb{P}_k & \\
\mathbb{E}_k \arrow[ur] & \mathbb{PR}_k \arrow[u] & \mathbb{P}_{n,k} \arrow[ul]
\end{tikzcd}$$

Figure 3.1: Three game comonads based on $\mathbb{P}_k$. Arrows are comonad morphisms.

Other game comonads have been introduced for different logics which are not directly comparable with infinitary k-variable logics. In particular, there are comonads for modal logic by Abramsky and Shah [11], hybrid logic by Abramsky and Marsden [9], and guarded fragments of first-order logic also by Abramsky and Marsden [8]. There have also been recent efforts made by Jakl, Marsden and Shah to extend game comonads to two-sorted structures and to second-order logics [62]. These comonads are outside the scope of this thesis.

In the rest of this section we briefly review the game comonads $\mathbb{E}_k$, $\mathbb{PR}_k$ and $\mathbb{P}_{n,k}$ and compare them with $\mathbb{P}_k$.

3.3.1 $\mathbb{E}_k$: the Ehrenfeucht-Fraïssé comonad

Introduced first by Abramsky and Shah [11], the $\mathbb{E}_n$ comonad gives a comonadic semantics to a family of k-round Ehrenfeucht-Fraïssé games. The existential or one-way Ehrenfeucht-Fraïssé game ($\exists\mathbf{EF}_k$, see Chapter 2) between structures $\mathcal{A}$ and $\mathcal{B}$ relaxes the existential k-pebble game by preventing the pebbles from being moved once they've been placed. Instead, Spoiler and Duplicator take turns placing pebbles on their respective structures and when all pebbles are placed the game ends. Duplicator wins if she makes it to the end of the game without Spoiler winning. Here we define the comonad, state its relations to logics of *quantifier depth* k and structures of *treewidth* bounded by k, and construct a comonad morphism $\nu^{\mathbb{E}} \colon \mathbb{E}_k \to \mathbb{P}_k$.

Definition of $\mathbb{E}_n$ The definition of this comonad is very similar to that of $\mathbb{P}_k$, where we send a structure $\mathcal{A}$ to an appropriate structure on the set of histories of Spoiler moves in this game. As they are no longer relevant, we can forget the labels on the pebbles and so lists of Spoiler moves on $\mathcal{A}$ are the simply non-empty lists $A^+_{\leq k}$. Our definition follows that of Abramsky and Shah.

Definition 3.19 (Definition of $\mathbb{E}_k$ in Kleisli form)**.** *The k-round Ehrenfeucht-Fraïssé comonad is defined by the Kleisli triple* $(\mathbb{E}_k, \epsilon^{\mathbb{E}}, \bar{\cdot})$ *consisting of*

- *an* ***endofunctor*** $\mathbb{E}_k$ *on* $\mathcal{R}(\sigma)$ *where*

- $\mathbb{E}_k\mathcal{A}$ *has the underlying set* $A^+_{\leq k}$ *and a tuple* $(s_1, \ldots, s_n)$ *is in* $R^{\mathbb{E}_k\mathcal{A}}$ *if and only is* $(\omega(s_1), \ldots \omega(s_n))$ *is in* $R^{\mathcal{A}}$ *and for each* $i, j \in [k]$ *either* s_i *is a prefix of* s_j *or vice versa; and*
- $\mathbb{E}_k f$ *for a homomorphism* $f\colon \mathcal{A} \to \mathcal{B}$ *applies* f *point-wise to each element of a list in* $\mathbb{E}_k\mathcal{A}$,

- *a* ***counit*** $\epsilon^{\mathbb{E}}\colon \mathbb{E}_n \implies 1$ *where* $\epsilon^{\mathbb{E}}_{\mathcal{A}}\colon \mathbb{E}_n\mathcal{A} \to \mathcal{A}$ *sends* $s \in \mathbb{E}_n A$ *to* $\omega(s)$*; and*
- *a* ***Kleisli coextension*** $\bar{\cdot}$ *which sends each Kleisli morphism* $f\colon \mathbb{E}_n\mathcal{A} \to \mathcal{B}$ *to the morphism* $\overline{f} : \mathbb{E}_n\mathcal{A} \to \mathbb{E}_n\mathcal{B}$ *defined on* $s = [a_1, \ldots, a_m] \in \mathbb{E}_n A$ *as*

$$\overline{f}(s) = [f(s[: 1], \ldots, f(s[: m]))].$$

Informally, we should see the action of $\mathbb{E}_n$ on a structure $\mathcal{A}$ as unfolding $\mathcal{A}$ into a forest of trees of depth n, whose nodes are labelled with elements of A. Each tree in this unfolding is rooted at a singleton list $[a_1] \in \mathbb{E}_n\mathcal{A}$ and each element, $s = [a_1, a_2, \ldots, a_m] \in \mathbb{E}_n\mathcal{A}$, describes a path in this tree from the root to an element with label $\epsilon_{\mathcal{A}}(s) = a_m$. The relations in $\mathbb{E}_n\mathcal{A}$ are defined so that the relational stuructre on each branch, $\{s_1 = [a_1], s_2 = [a_1, a_2], \ldots, s_n = [a_1, a_2, \ldots a_n]\}$, of each tree is the same as that on the induced substructure of $\mathcal{A}$ on the elements $\{\epsilon_{\mathcal{A}}(s_1), \ldots, \epsilon_{\mathcal{A}}(s_n)\}$.

Power for $\mathbb{E}_n$ Analogously to the $\mathbb{P}_k$ Power theorems, we can prove that Kleisli morphisms and isomorphisms for $\mathbb{E}_k$ between I-structures correspond to Duplicator winning strategies in a one-way and bijective game. The games here are the Ehrenfeucht-Fraïssé games on k rounds $\exists\mathbf{EF}_k$ and $\mathbf{BijEF}_k$. In turn, these games are known to capture, respectively, logical entailment over $\exists^+\mathcal{L}_k$ and logical equivalence over $\mathcal{C}_k$.

Structure for $\mathbb{E}_k$ The $\mathbb{E}_k$-coalgebras for a structure $\mathcal{A}$ are shown to be equivalent to depth k forest covers of $\mathcal{A}$ and as a result the coalgebra number $f_{\mathbb{E}}$ is equal to the *treedepth* of the structure, as defined in Chapter 2.

Comparison with $\mathbb{P}_k$ As stated above, the games captured by $\mathbb{E}_k$ are simpler versions of the pebble games captured by $\mathbb{P}_k$. We can make this comparison formally between the respective comonads with the following comonad morphism from $\mathbb{E}_k$ to $\mathbb{P}_k$.

Definition 3.20. *We define* $\nu^{\mathbb{E}}\colon \mathbb{E}_k \to \mathbb{P}_k$ *to be the natural transformation which sends* $s = [a_1, \ldots a_l] \in \mathbb{E}_k A$ *to* $\nu^{\mathbb{E}}_{\mathcal{A}}(s) := [(a_1, 1), \ldots, (a_l, l)]$.

3.3.2 $\mathbb{PR}_k$: the pebble-relation comonad

The pebble-relation comonad was introduced by Montacute and Shah [77] to give a comonadic semantics to the parameter of pathwidth and the pebble-relation game of

Dalmau which was used to study restricted conjunction fragments of first-order logic [31]. The construction of this comonad from $\mathbb{P}_k$ is more technically sophisticated than that of $\mathbb{E}_k$. The construction relies on the fact that Dalmau's game is equivalent to a so-called "all-in-one" pebble game of Stewart where Spoiler decides to play for some finite number of rounds and announces all of his moves ahead of time. Here we present a definition of this comonad, following that of Montacute and Shah, and state their results in relating this comonad to *restricted conjunction logics* and *pathwidth*.

Definition of $\mathbb{PR}_k$ To represent histories of Spoiler moves up to a certain round of the k-pebble all-in-one game we use pairs (s, i) in $\mathbb{P}_k\mathcal{A} \times \mathbb{N}$ such that $i \leq |s|$, where s is the series of moves that Spoiler declared at the start of the game and i is the number of the current round. The following definition shows how to turn this into a comonad.

Definition 3.21 (Definition of $\mathbb{PR}_k$ in Kleisli form)**.** *The k-pebble-relation comonad is defined by the Kleisli triple $(\mathbb{PR}_k, \epsilon^{\mathbb{PR}}, \bar{\cdot})$ consisting of*

- *an* ***endofunctor*** *$\mathbb{PR}_k$ on $\mathcal{R}(\sigma)$ which sends $\mathcal{A}$ to a structure with the underlying set $\prod_{s \in \mathbb{P}_k\mathcal{A}}[|s|]$ where the related tuples $R^{\mathbb{PR}_k\mathcal{A}}$ are precisely the tuples $((s, i_1), \ldots (s, i_n))$ such that $(s[: i_1], \ldots s[: i_n]) \in R^{\mathbb{P}_k\mathcal{A}}$,*

- *a* ***counit*** *$\epsilon^{\mathbb{PR}}$ which sends (s, i) to $\omega(s[: i])$, and*

- *a* ***Kleisli coextension*** *$\bar{\cdot}$ which sends any $f \colon \mathbb{PR}_k\mathcal{A} \to \mathcal{B}$ to the map $\overline{f} \colon \mathbb{PR}_k\mathcal{A} \to \mathbb{PR}_k\mathcal{B}$ which sends $(s, i) = ([(a_1, p_1), \ldots, (a_m, p_m)], i)$ to $([(f((s, 1)), p_1), \ldots (f((s, m)), p_m)], i)$.*

Structure and Power of $\mathbb{PR}_k$ The Structure and Power theorems for this comonad, as proved by Montacute and Shah, follow the same pattern as for $\mathbb{P}_k$ and $\mathbb{E}_k$.

On the Power side, the Kleisli morphisms and isomorphism between I-structures capture Duplicator winning strategies in the one-way and bijective k-pebble all-in-one games. Montacute and Shah show that the one-way game captures the k-variable restricted conjunction logic and they define a counting version of this logic $\#\curlywedge\mathcal{L}^k$ which is captured by the bijective all-in-one game.

On the Structure side, $\mathbb{PR}_k$-coalgebras over a structure $\mathcal{A}$ are equivalent to path decompositions of $\mathcal{A}$ of bag size k and the coalgebra number $f_{\mathbb{PR}}$ characterises the pathwidth of a structure.

Montacute and Shah also show that the natural transformation $\nu^{\mathbb{PR}} : \mathbb{PR}_k \to \mathbb{P}_k$, which sends any pair $(s, i) \in \mathbb{PR}_k\mathcal{A}$ to $s \in \mathbb{P}_k\mathcal{A}$, is a comonad morphism.

	Morphism Power		Isomorphism Power	
Comonad	**Game**	**Logic**	**Game**	**Logic**
$\mathbb{P}_k$	$\exists\mathbf{Peb}_k$	$\exists^+\mathcal{L}^k$	$\mathbf{Bij}_k$	$\mathcal{C}^k$
$\mathbb{E}_k$	$\exists\mathbf{EF}_k$	$\exists^+\mathcal{L}_k$	$\mathbf{BijEF}_k$	$\mathcal{C}_k$
$\mathbb{PR}_k$	$\exists\mathbf{PR}_k$	$\exists^+\curlywedge\mathcal{L}^k$	$\mathbf{BijPR}_k$	$\#\curlywedge\mathcal{L}^k$
$\mathbb{P}_{n,k}$	$\exists\mathbf{Peb}_{n,k}$	$\exists^+\mathcal{L}^k_n$	$\mathbf{Bij}_{n,k}$	$\mathcal{C}^k_n$

Table 3.1: Summary of Power Theorems

3.3.3 $\mathbb{P}_{n,k}$: the k pebble n round comonad

The $\mathbb{P}_{n,k}$ comonad aims to capture a version of the normal k pebble game which ends after a fixed number, $n \geq k$, or rounds. This construction, due to Paine [81] is the simplest of the variations on $\mathbb{P}_k$ and is actually constructed as a subcomonad of $\mathbb{P}_k$. This is done by defining $\mathbb{P}_{n,k}\mathcal{A}$ for any $\mathcal{A}$ as the induced substructure of $\mathbb{P}_k\mathcal{A}$ on the set $(A \times [k])^{\leq n}$ of Spoiler histories of length less than or equal to n. As any Kleisli coextension $\overline{f}$ of a Kleisli map $f \colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ preserves the length of Spoiler histories on $\mathcal{A}$, we can easily see that the triple $(\mathbb{P}_{n,k}, \epsilon, \bar{\cdot})$ defines a comonad on $\mathcal{R}(\sigma)$. Furthermore, the inclusion maps $\mathbb{P}_{n,k}\mathcal{A} \hookrightarrow \mathbb{P}_k\mathcal{A}$ define a comonad morphism.

Paine establishes connections between Kleisli morphisms and Duplicator winning strategies for the existential and bijective n-round k-pebble games. In turn, these are seen to capture the k-variable quantifier depth n logics $\exists^+\mathcal{L}^k_n$ and $\mathcal{C}^k_n$. The $\mathbb{P}_{n,k}$-coalgebras, on the other hand, are shown to capture a notion of depth n k-pebble forest covers which effectively restrict the depth of tree decompositions of bag size k. For any fixed n, the coalgebra number $f_{\mathbb{P}_n}$ is at most the treewidth of the structure as the tree decomposition of smallest width won't in general also have the smallest depth.

As we saw in this section, there are various weakenings of the $\mathbb{P}_k$ which appear in the game comonads literature. Each of them provides interesting connections to different games and logics from finite model theory as well as strucutral decompositions and combinatorial parameters. These connections can be stated in terms of Power Theorems in the mould of Theorems 3.13 and 3.14 and Structure Theorems like Theorems 3.16 and 3.18. The corresponding results from the comonads surveyed in this section are summarised in Tables 3.1 and 3.2.

In the next section, we look beyond the constructions of different game comonads to some steps towards general theory and applications before concluding this review.

Comonad	Coalgebra	Parameter
$\mathbb{P}_k$	tree decomposition of bag size k	treewidth
$\mathbb{E}_k$	forest cover of depth k	treedepth
$\mathbb{PR}_k$	path decomposition of bag size k	pathwidth
$\mathbb{P}_{n,k}$	k-pebble forest cover of depth n	depth n treewidth

Table 3.2: Summary of Structure Theorems

3.4 Other topics in game comonads

In this chapter we have seen how a comonad arises naturally from considering the compositional nature of certain model comparison games in finite model theory. We have seen that this categorical construction can be adapted to a variety of different games and that investigating the different components of these comonads provides fruitful connections to different aspects of logical power and combinatorial structure which are well-studied in logic and finite model theory. This, however, just touches on just the beginning of an emerging area of research into comonadic methods in finite model theory. Before concluding this chapter, we review the recent developments and current directions of research in this area. These include further connections between games and comonads, using comonads to prove new model theoretic results and generalise old ones, and developing an abstract comonadic theory of resources for finite model theory.

Further connections While the above examples have shown how to capture several different model comparison games and structural parameters using comonads, this by no means gives a complete account of compositionality for games and parameters in finite model theory.

On the one hand, there are many types of games and parameters on finite structures which are not given a semantics by the above constructions which are being studied using similar methods. Notable examples of this include the monadic semantics of non-local games developed by Abramsky, Barbosa, de Silva and Zapata [3] and the use of comonads in the theory of open parity games presented by Asada, Eberhart, Hasuo and Watanabe [89].

On the other hand, there are even simple variations of the above games which appear to be missing from the account above. One such omission is that of the back-and-forth Ehrenfeucht-Fraïssé games which capture logical equivalence over fragments of first-order logic (without counting). Abramsky and Shah [11] show that Duplicator strategies in these games can be witnessed as "spans of open pathwise embeddings" in the Kleisli category of game comonads above. This notion borrows from Joyal, Nielsen and Wynskel's characterisation of bisimulation in terms of open maps [64].

Proving and generalising results Another direction of research in this area is in finding new category theoretic proofs for theorems in finite model theory which rely on model comparison games. These new proofs can clarify the sometimes ad hoc constructions that arise in game-based proofs and make it easier to generalise such results between different logics and games.

Perhaps the most important example of such a result in the recent game comonads literature is the comonadic proof of a general Lovasz-type theorem which generalises Lovasz's classical characterisation of isomorphism between two relational structures in terms of the the equality of homomorphism-count vectors [74]. Dawar, Jakl and Reggio [38] show the following important result which holds for many game comonads.

Theorem 3.22 (The Dawar-Jakl-Reggio Lovasz-type Theorem)**.** *Given a game comonad $\mathbb{C}$ which satisfies certain conditions, then the following are equivalent for any structures $\mathcal{A}, \mathcal{B}$*

- $\mathcal{A} \cong_{\mathcal{K}(\mathbb{C})} \mathcal{B}$*, and*
- *for all $\mathcal{C}$ which admit a $\mathbb{C}$-coalgebra,*

$$|\mathbf{hom}(\mathcal{C}, \mathcal{A})| = |\mathbf{hom}(\mathcal{C}, \mathcal{B})|.$$

By the Structure and Power theorems for the relevant comonads, this result can be used to derive classifications of equivalences over different bounded logical fragments in terms of homomorphism-count vectors from structures bounded in some parameter. This results in a single standardised proof for several well-known results in finite model theory including those of Grohe [51] and Dvořàk [40]. In addition it has been used to derive novel results such as, in the case of Montacute and Shah's comonad $\mathbb{PR}_k$, a characterisation of homomorphism-count vectors over structures of bounded pathwidth in terms of $\# \curlywedge \mathcal{L}^k$ equivalence.

Aside from this, progress has been made towards comonadic accounts of preservation theorems [81] and Courcelle's theorem [62].

Towards a theory of resources Finally, the discovery of these graded game comonads which give a novel way of simultaneously controlling logical resources and structural parameters through a single "resource parameter" has inspired work towards a general comonadic theory of resources for finite structures.

An important step in this direction was the axiomatic theory given by Abramsky and Reggio [10] which has presented game comonads as arising from resource-indexed adjunctions with so-called arboreal categories. In addition to this, work has been done on finding general constructions of game comonads with Abramsky, Jakl and Paine [7] using discrete density comonads to construct comonads for any given structural parameter.

In the rest of this thesis, we see contributions to all three of these directions in game comonads research. Chapters 4 to 6 focus on expanding the connections made so far by this programme and by deriving new results from them. Chapters 7 and 8 highlight some apparent limitations of the comonadic approach to certain logical resources and present alternative category-theoretic constructions for efficient computation beyond $\mathcal{C}^k$.

Chapter 4

Quantifiers in the Kleisli Category

In the last chapter, we saw in Theorems 3.13 and 3.14 that morphisms and isomorphisms of the Kleisli category of $\mathbb{P}_k$ relate in a deep way to the logics $\exists^+\mathcal{L}^k_\infty$ and $\mathcal{L}^k_\infty(\#)$, respectively. These results are summarised in Figure 4.1 where the Hasse diagram on the left represents types of maps in $\mathcal{K}(\mathbb{P}_k)$ and that on the right gives the related logics ordered by inclusion. In this chapter, we deepen this relation significantly showing that *branch-injective, branch-surjective* and *branch-bijective* Kleisli maps, which were introduced by Abramsky, Dawar and Wang in the seminal paper on game comonads [6], relate in a similar way to logics which are intermediate between $\exists^+\mathcal{L}^k_\infty$ and $\mathcal{L}^k_\infty(\#)$. This more complicated correspondence between Kleisli maps and logics is summarised in Figure 4.2.

From a category-theoretic perspective it is interesting to ask how these intermediate maps relate to the monomorphisms and epimorphisms of the category $\mathcal{K}(\mathbb{P}_k)$. In the second part of this chapter, we compare these notions with branch-injective and branch-surjective maps, showing that monomorphisms and epimorphisms are strictly more permissive notions. We also provide some progress towards repairing this inequivalence in the case of monomorphisms by tweaking the definition of $\mathbb{P}_k$.

$$\begin{tikzcd}
\cong_{\mathcal{K}(\mathbb{P}_k)} \arrow[dd, dash] & & & \mathcal{L}^k_\infty(\#) \arrow[dd, dash] \\
 & & & \\
\to_{\mathcal{K}(\mathbb{P}_k)} & & & \exists^+\mathcal{L}^k_\infty
\end{tikzcd}$$

Figure 4.1: Morphisms and isomorphisms in $\mathbb{P}_k$ and their related logics.

$$
\begin{tikzcd}[column sep=small]
 & \to^{b}_{\mathcal{K}(\mathbb{P}_k)} \arrow[dl, dash] \arrow[dr, dash] & & & & \exists^{+}\mathcal{L}^{k}_{\infty}(\exists^{\geq m}, \forall^{\leq m}) \arrow[dl, dash] \arrow[dr, dash] & \\
\to^{i}_{\mathcal{K}(\mathbb{P}_k)} \arrow[dr, dash] & & \to^{s}_{\mathcal{K}(\mathbb{P}_k)} \arrow[dl, dash] & & \exists^{+}\mathcal{L}^{k}_{\infty}(\exists^{\geq m}) \arrow[dr, dash] & & \exists^{+}\mathcal{L}^{k}_{\infty}(\forall^{\leq m}) \arrow[dl, dash] \\
 & \to_{\mathcal{K}(\mathbb{P}_k)} & & & & \exists^{+}\mathcal{L}^{k}_{\infty} &
\end{tikzcd}
$$

Figure 4.2: Branch-injective, surjective, and bijective maps in $\mathcal{K}(\mathbb{P}_k)$ and their related logics.

$$
\begin{tikzcd}
\cong_{\mathbb{P}^{*}_{k}} \arrow[d, dash] & & \mathcal{L}^{k}_{\infty}(\#) \arrow[d, dash] \\
\rightarrowtail_{\mathbb{P}^{*}_{k}} \arrow[d, dash] & & \exists^{+}\mathcal{L}^{k}_{\infty}(\exists^{\geq m}) \arrow[d, dash] \\
\to_{\mathbb{P}^{*}_{k}} & & \exists^{+}\mathcal{L}^{k}_{\infty}
\end{tikzcd}
$$

Figure 4.3: Morphisms, monomorphisms, and isomorphisms in $\mathcal{K}(\mathbb{P}^{*}_{k})$ and their related logics.

4.1 Branch-injective and branch-surjective strategies

The concepts of branch-injectivity and branch-surjectivity of Kleisli maps $f : \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ were first used by Abramsky, Dawar and Wang as a step towards proving the Isomorphism Power Theorem for $\mathbb{P}_k$. The key definitions are as follows.

Definition 4.1. *For a Kleisli map $f : \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, a list $s \in (A \times [k])^*$, and an index $i \in [k]$, the i^{th}* branch map of f at s *is the function $f_{s,i} : A \to B$ defined as*

$$f_{s,i}(a) = f(s; (a, i)).$$

We say that f is branch-injective *when all branch maps $f_{s,i}$ are injective and we write $\mathcal{A} \to^{i}_{k} \mathcal{B}$ when there exists such an f.*

Similarly, f is branch-surjective *when all $f_{s,i}$ are surjective and we write $\mathcal{A} \to^{s}_{k} \mathcal{B}$ when there exists such an f.*

If f is both branch-injective and branch surjective we say it is branch-bijective *and write $\mathcal{A} \to^{b}_{k} \mathcal{B}$.*

While these definitions are useful in the development of the original Power Theorems in [6], they have not yet been studied in their own right. In particular, an interesting question is whether these maps have a correspondence with logic. As the maps $\rightarrow_k^{\mathrm{i}}, \rightarrow_k^{\mathrm{s}}$ and $\rightarrow_k^{\mathrm{b}}$ are all intermediate between $\rightarrow_k$ and $\cong_{\mathcal{K}(\mathbb{P}_k)}$, we expect that any such logic would be intermediate between $\exists^+\mathcal{L}_\infty^k$ and $\mathcal{L}_\infty(\#)$. To answer this question precisely, we introduce the following counting quantifiers which are both expressible in $\mathcal{L}_\infty^k(\#)$.

Definition 4.2. *For any non-negative integer m, the quantifiers $\exists^{\geq m}$ and $\forall^{\leq m}$ each bind a single variable in any formula. Their semantics on a structure $\mathcal{A}$ is defined as follows*

$$\mathcal{A}, \mathbf{a} \models \exists^{\geq m} x.\ \phi(\mathbf{x}, x) \iff |\{a \in A \ \mid\ \mathcal{A}, \mathbf{a}, a \models \phi(\mathbf{x}, x)\}| \geq m$$
$$\mathcal{A}, \mathbf{a} \models \forall^{\leq m} x.\ \phi(\mathbf{x}, x) \iff |\{a \in A \ \mid\ \mathcal{A}, \mathbf{a}, a \not\models \phi(\mathbf{x}, x)\}| \leq m.$$

For this reason, we refer to $\exists^{\geq m}$ as the "there exists at least m" *quantifier and $\forall^{\leq m}$ as the* "for all except m" *quantifier.*

In this section we settle the question of finding logics for branch-injective, branch-surjective and branch-bijective maps by proving the following theorem and thus establishing the structure on Kleisli maps laid out in Figure 4.2.

Theorem 4.3 (Branch Morphism Power Theorem). *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following equivalences hold for any positive integer k:*

1. *$\mathcal{A} \rightarrow_k^i \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\exists^{\geq m})} \mathcal{B}$,*

2. *$\mathcal{A} \rightarrow_k^s \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\forall^{\leq m})} \mathcal{B}$, and*

3. *$\mathcal{A} \rightarrow_k^b \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\exists^{\geq m}, \forall^{\leq m})} \mathcal{B}$*

We prove this theorem in two parts. Firstly, we introduce a new system of k-pebble games which generalise the existential k-pebble game of Kolaitis and Vardi [67] and show, in Proposition 4.5 that Duplicator winning strategies for these games correspond exactly to Kleisli maps with the desired branch map conditions. We then prove, in Proposition 4.6, that Duplicator winning strategies in these games correspond to the desired logical relations between structures. This completes the proof of Theorem 4.3.

4.1.1 Branch maps and functional games

For the first part of the proof of Theorem 4.3 we need to introduce versions of the existential k-pebble game which capture the various restrictions we can place on the branch maps in $\mathcal{K}(\mathbb{P}_k)$. We do this in the following definition by altering the rules of the ordinary existential k-pebble game so that Duplicator responds, in any round, not with a single element but with a function describing their response to any of Spoiler's potential moves.

$$
\begin{tikzcd}[column sep=small]
& \to_k^{\mathrm{b}} \arrow[dl, dash] \arrow[dr, dash] & \\
\to_k^{\mathrm{i}} \arrow[dr, dash] & & \to_k^{\mathrm{s}} \arrow[dl, dash] \\
& \to_k &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
& +\mathbf{Bij}^k \arrow[dl, dash] \arrow[dr, dash] & \\
+\mathbf{Inj}^k \arrow[dr, dash] & & +\mathbf{Surj}^k \arrow[dl, dash] \\
& +\mathbf{Fun}^k &
\end{tikzcd}
$$

Figure 4.4: Summary of equivalence between Duplicator winning strategies in various functional games and Kleisli maps satisfying certain branch conditions proved in Proposition 4.5.

Definition 4.4 (Functional k-pebble games)**.** *For two relational structures $\mathcal{A}$, $\mathcal{B}$, the functional k-pebble game, $+\mathbf{Fun}^k(\mathcal{A},\mathcal{B})$ is played by Spoiler and Duplicator. Prior to the jth round the position consists of partial maps $\pi^a_{j-1} : [k] \rightharpoonup A$ and $\pi^b_{j-1} : [k] \rightharpoonup B$. In Round j*

- *Spoiler chooses a pebble $i_j \in I$.*
- *Duplicator provides a function $f^j : A \to B$ such that for each $i \in [k] \setminus \{i_j\}$, $f_j(\pi^a_{j-1}(i)) = \pi^b_{j-1}(i)$.*
- *Spoiler chooses a pebble $a'_{i_j} \in A$.*
- *The updated position is given by $\pi^a_j(i_j) = a'_{i_j}$ and $\pi^b_j(i_j) = f^j(a'_{i_j})$; and $\pi^a_j(i)$ and $\pi^b_j(i)$ are unchanged for all $i \neq i_j$.*
- *Spoiler has won the game if there is some $R \in \sigma$ and $(i_1, \dots i_r) \in [k]^r$ with $(\pi^a_j(i_1), \dots, \pi^a_j(i_r)) \in R^{\mathcal{A}}$ but $(\pi^b_j(i_1), \dots, \pi^b_j(i_r)) \notin R^{\mathcal{B}}$.*

Duplicator wins by preventing Spoiler from winning.

The games $+\mathbf{Inj}^k(\mathcal{A},\mathcal{B})$, $+\mathbf{Surj}^k(\mathcal{A},\mathcal{B})$ and $+\mathbf{Bij}^k(\mathcal{A},\mathcal{B})$ are defined in the same way except that the maps f^j are required to be injective, surjective and bijective, respectively.

The following Proposition demonstrates the usefulness of these games by showing that Duplicator strategies for them are equivalent to the existence of Kleisli maps which satisfy the various branch map conditions outlined in Definition 4.1. A summary of these correspondences is provided in Figure 4.4. One simple consequence of this result is the equivalence of $\exists^+\mathbf{Peb}^k$ and $+\mathbf{Fun}^k$, which shows that the above definition is really just a restating of Kolaitis and Vardi's original game [67].

Proposition 4.5. *For two I-structures $\mathcal{A}$ and $\mathcal{B}$, we have the following list of equivalences:*

1. *Duplicator has a winning strategy for* $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \to_k \mathcal{B}$,
2. *Duplicator has a winning strategy for* $+\mathbf{Inj}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \to_k^i \mathcal{B}$,
3. *Duplicator has a winning strategy for* $+\mathbf{Surj}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \to_k^s \mathcal{B}$, *and*
4. *Duplicator has a winning strategy for* $+\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \to_k^b \mathcal{B}$.

Proof. At the heart of this equivalence is a very straightforward equivalence between the existence of Duplicator winning strategies of $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$ and the existence of Kleisli maps $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$ between the respective I-structures. Indeed given a Kleisli map $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, the branch maps $f_{s,i}$ for each Spoiler history $s \in (\mathcal{A} \times [k])^*$ and each pebble index $i \in [k]$ describe a Duplicator strategy for $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$ where $f_{s,i}$ is the function provided by Duplicator if Spoiler picks up pebble i at the beginning of the round following the sequence of moves s. A similar translation works in reverse where Duplicator's responses in each round of the game give the branch maps of a function $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$. It is not hard to see that such a map is a homomorphism if, and only if, the respective strategy is winning for $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$.

For the equivalences 2 to 4 above, we use the same translation between games and strategies as described above. The only difference is that the rules of the game in each case guarantee that the branch maps are all injective, surjective and bijective respectively. The same argument as above gives equivalence of homomorphisms and winning strategies. $\square$

This completes the first part of the proof of Theorem 4.3. In the next part, we establish a connection between winning strategies for these new games and logics extended by counting quantifiers.

4.1.2 Functional games and counting quantifiers

Part 2 of the proof of Theorem 4.3 requires us to relate Duplicator winning strategies for the games introduced in Part 1 to the logical fragments named in the theorem. We do this by proving the following proposition which is summarised in Figure 4.5.

Proposition 4.6. *For two relational structures $\mathcal{A}$ and $\mathcal{B}$, we have the following list of equivalences:*

1. *Duplicator has a winning strategy for* $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k} \mathcal{B}$,
2. *Duplicator has a winning strategy for* $+\mathbf{Inj}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\exists_{\geq m})} \mathcal{B}$,

$$
\begin{tikzcd}[arrows=dash]
 & +\mathbf{Bij}^k \arrow[dl] \arrow[dr] & \\
+\mathbf{Inj}^k \arrow[dr] & & +\mathbf{Surj}^k \arrow[dl] \\
 & +\mathbf{Fun}^k &
\end{tikzcd}
\qquad
\begin{tikzcd}[arrows=dash]
 & \exists^+\mathcal{L}^k(\forall^{\leq m}, \exists^{\geq m}) \arrow[dl] \arrow[dr] & \\
\exists^+\mathcal{L}^k(\exists^{\geq m}) \arrow[dr] & & \exists^+\mathcal{L}^k(\forall^{\leq m}) \arrow[dl] \\
 & \exists^+\mathcal{L}^k &
\end{tikzcd}
$$

Figure 4.5: Summary of result in Proposition 4.6.

3. *Duplicator has a winning strategy for* $+\mathbf{Surj}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\forall^{\leq m})} \mathcal{B}$*, and*

4. *Duplicator has a winning strategy for* $+\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ *iff* $\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\forall^{\leq m}, \exists^{\geq m})} \mathcal{B}$*.*

In the rest of this section we prove this proposition with a series of lemmata. The first two of these, Lemma 4.8 and Lemma 4.10, prove that Duplicator winning strategies for the functional games and logical entailment over the desired fragments of logic are equivalent, respectively, to the existence of appropriate *forth systems* and sets of *logical homomorphisms*. The final lemma, Lemma 4.11, establishes the equivalence of these two notions, thus completing the proof of Proposition 4.6.

Following the terminology of Kolaitis and Vardi [67], who use *forth systems* to represent Duplicator winning strategies in $\exists\mathbf{Peb}^k$, we make the following definition. These systems consist of positions in the k pebble game which are given by partial homomorphisms $f\colon \mathcal{A} \rightharpoonup \mathcal{B}$ with $|f| \leq k$. We write $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ for the set of all such partial homomorphisms and $\mathbf{hom}(\mathcal{A}, \mathcal{B})$ for the union $\bigcup_{k\geq 0} \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.

Definition 4.7. *A set* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ *is a* forth system *if it satisfies the following properties:*

- ***Downwards closure****: If* $f \in \mathcal{S}$ *then* $g \in \mathcal{S}$ *for any* $g \subset f$

- ***Forth property*** *For any* f *in* $\mathcal{S}$ *s.t.* $|f| < k$*, there exists a function* $\phi_f : A \to B$*, such that for all* $a \in A$*,* $f \cup \{(a, \phi_f(a))\} \in \mathcal{S}$*. For each such* f *we call this property* $\mathbf{Forth}(S, f)$*.*

If we can always choose the functions ϕ_f *to be injections, surjections or bijections respectively, we call* $\mathcal{S}$ *an injective, surjective or bijective forth system.*

It is not hard to see that these forth systems are essentially another way of presenting Duplicator winning strategies for the functional games of Definition 4.4. Indeed, we can think of the elements of a forth system as positions in these games from which Duplicator can play forever. The closure and forth properties ensure that in each round of the game Duplicator can force the game's position to remain inside their chosen forth system. This argument is formalised in the following result.

Lemma 4.8. *For any $\mathcal{A}$ and $\mathcal{B}$, the following equivalences hold:*

1. *There is Duplicator winning strategy for* $+\mathbf{Fun}^k(\mathcal{A}, \mathcal{B})$ *if, and only if, there is a non-empty forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.
2. *There is Duplicator winning strategy for* $+\mathbf{Inj}^k(\mathcal{A}, \mathcal{B})$ *if, and only if, there is a non-empty injective forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.
3. *There is Duplicator winning strategy for* $+\mathbf{Surj}^k(\mathcal{A}, \mathcal{B})$ *if, and only if, there is a non-empty surjective forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.
4. *There is Duplicator winning strategy for* $+\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ *if, and only if, there is a non-empty bijective forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.

Proof. Given a non-empty forth system $\mathcal{S}$ (of the appropriate kind), the translation into a Duplicator strategy in the respective game is relatively straightforward. By downward-closure, the empty partial function $\emptyset$ is in $\mathcal{S}$. As this is the starting position at the beginning of the respective functional game we can play this game as follows. Suppose $s \in \mathcal{S}$ is the position at the start of some round of the game. When Spoiler picks up some pebble i, this reduces the position to $s' \subset s$ and by downward-closure $s' \in \mathcal{S}$. The forth property of $\mathcal{S}$ says that there is a function $\phi_{s'}$, such that if Duplicator responds with this function then the positions at the end of the round are also in $\mathcal{S}$. Any strategy which always picks these functions $\phi_{s'}$ is winning for the appropriate game.

To go the other direction, we let $\mathcal{S}_{pos}$ be the set of reachable positions in the game when played according to Duplicators winning strategy. As there is no position where Duplicator loses we must have $\mathcal{S}_{pos} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$, so it remains to prove the downward-closed and forth properties. For downward closure, note that $\emptyset \in \mathcal{S}_{pos}$ as it is the starting position. Then for any $s \in \mathcal{S}_{pos}$ and any non-empty $s' \subset s$ we can reach s' from s as follows. Choose $(a, b) \in s'$ and let $i_1, \dots i_l \in [k]$ be the pebble indices which appear in s but not in s'. Then, starting from the position s, Spoiler should play the next l rounds by picking up pebble i_j and, regardless of Duplicator's reply, placing it down on a. The resulting position is s. The forth property is slightly more straightforward. If $s \in \mathcal{S}_{pos}$ and $|s| < k$ then if Spoiler chooses a pebble which does not appear in s, Duplicator's response must be a function ϕ_s (which is injective, surjective or bijective as required) and the condition that Duplicator does not lose by playing this function is exactly the respective forth condition. $\square$

Now we define the notion of a logical homomorphism which at first appears to generalise logical entailment relations of the form $\mathcal{A} \Rrightarrow_{\mathcal{L}^k} \mathcal{B}$ but will seen to be equivalent in the proof of Proposition 4.6. To do this, we recall from Chapter 2 that for some logic $\mathcal{L}$, two structures $\mathcal{A}$ and $\mathcal{B}$ and tuples $\mathbf{a} = (a_1, \dots, a_m)$ and $\mathbf{b} = (b_1, \dots, b_m)$, we write $\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}} \mathcal{B}, \mathbf{b}$ if for every formula $\phi(x_1, \dots x_m) \in \mathcal{L}$ if $\mathcal{A}, \mathbf{a} \models \phi(x_1, \dots x_m)$ then $\mathcal{B}, \mathbf{b} \models \phi(x_1, \dots x_m)$. We use this to build the following definition.

Definition 4.9. *For a k-variable logic $\mathcal{L}^k$, and two structures $\mathcal{A}$ and $\mathcal{B}$, a* logical homomorphism *for $\mathcal{L}^k$ between $\mathcal{A}$ and $\mathcal{B}$ is a partial homomorphism $p \in \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ such that for every tuple of elements $\mathbf{a}$ in $\mathbf{dom}(p)$ of size $\leq k$ we have*

$$\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}^k} \mathcal{B}, p(\mathbf{a}).$$

We denote the set of all logical homomorphisms for $\mathcal{L}^k$ between $\mathcal{A}$ and $\mathcal{B}$ by $\mathcal{S}_{\mathcal{L}^k}(\mathcal{A}, \mathcal{B}) \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$.

It is clear from this definition that the empty partial function $\emptyset$ is a logical homomorphism if and only if $\mathcal{A} \Rrightarrow_{\mathcal{L}} \mathcal{B}$. This allows us to state the following result linking logical homomorphisms and $\Rrightarrow_{\mathcal{L}}$.

Lemma 4.10. *For any logic $\mathcal{L}^k$ (in the statement of Theorem 4.3) and any structures $\mathcal{A}$ and $\mathcal{B}$ we have that $\mathcal{A} \Rrightarrow_{\mathcal{L}^k} \mathcal{B}$ if, and only if, there exists a logical homomorphism for $\mathcal{L}^k$ between $\mathcal{A}$ and $\mathcal{B}$.*

Proof. As noted before this proof, the condition that $\mathcal{A} \Rrightarrow_{\mathcal{L}^k} \mathcal{B}$ implies that $\emptyset$ is a logical homomorphism.

To go the other direction, suppose we have some logical homomorphism $p \in \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ where the domain of p is $\mathbf{a}$. Then, for any sentence $\phi \in \mathcal{L}^k$ such that $\mathcal{A} \models \phi$, we have trivially that $\mathcal{A}, \mathbf{a} \models \phi$ and so $\mathcal{B}, p(\mathbf{a}) \models \phi$. As ϕ has no free variables this is the same as saying that $\mathcal{B} \models \phi$. Thus we have $\mathcal{A} \Rrightarrow_{\mathcal{L}^k} \mathcal{B}$, as required. $\square$

We can now complete the proof of Proposition 4.6 by showing that, for the k-variable logics we're interested in, the existence of a logical homomorphism for that logic is equivalent to the existence of a non-empty forth system for the corresponding game. We do this by proving the following lemma.

Lemma 4.11. *For any structures $\mathcal{A}$ and $\mathcal{B}$ and any positive integer k we have the following equivalences.*

1. *There is a non-empty forth system $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ if, and only if, $\mathcal{S}_{\exists^+\mathcal{L}^k_\infty}(\mathcal{A}, \mathcal{B})$ is non-empty.*

2. *There is a non-empty injective forth system $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ if, and only if, $\mathcal{S}_{\exists^+\mathcal{L}^k_\infty(\exists_{\geq m})}(\mathcal{A}, \mathcal{B})$ is non-empty.*

3. *There is a non-empty surjective forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ *if, and only if,* $\mathcal{S}_{\exists^+\mathcal{L}^k_\infty(\forall^{\leq m})}(\mathcal{A}, \mathcal{B})$ *is non-empty.*

4. *There is a non-empty bijective forth system* $\mathcal{S} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ *if, and only if,* $\mathcal{S}_{\exists^+\mathcal{L}^k_\infty(\exists^{\geq m}, \forall^{\leq m})}(\mathcal{A}, \mathcal{B})$ *is non-empty.*

Proof. For the forward direction of each of these equivalences we aim to show that given such a forth system, we have that $\mathcal{S} \subset \mathcal{S}_{\mathcal{L}}(\mathcal{A}, \mathcal{B})$ for the appropriate logic $\mathcal{L}$. We do this by showing that any $p \in \mathcal{S}$ is a logical homomorphism for $\mathcal{L}$, i.e. that p preserves any formula $\phi(\mathbf{s}) \in \mathcal{L}$ which holds in $\mathbf{dom}(p)$. We do this by structural induction on the formula $\phi(\mathbf{s})$, noting that for any atomic formula $\phi(\mathbf{x})$ is preserved because each $p \in \mathcal{S}$ is a partial homomorphism. Furthermore, if $\phi(\mathbf{x}) = \bigwedge \phi_i(\mathbf{x}_i)$ or $\phi(\mathbf{x}) = \bigvee \phi_i(\mathbf{x}_i)$ then the preservation of each ϕ_i ensures the preservation of ϕ.

So the only cases of the induction which remain are those containing quantification. This is where the differentiation between the four different types of forth system becomes apparent.

Case 1 Suppose $\mathcal{S}$ is a forth system and let $\phi(\mathbf{x}) = \exists x.\phi'(x, \mathbf{x})$. If we have that $\mathcal{A}, \mathbf{a} \models \phi(\mathbf{x})$ for some $\mathbf{a} \subset \mathbf{dom}(p)$ then there is some $a \in A$ with $\mathcal{A}, a, \mathbf{a} \models \phi(x, \mathbf{x})$. The forth property of $\mathcal{S}$ at p implies that there is some $b \in B$ such that $p \cup \{(a, b)\} \in \mathcal{S}$. Induction implies that this partial homomorphism preserves ϕ' and so $\mathcal{B}, b, p(\mathbf{a}) \models \phi'(x, \mathbf{x})$, thus $\mathcal{B}, p(\mathbf{a}) \models \exists x.\phi(x, \mathbf{x})$ and so p preserves ϕ. Now by induction p is a logical homomorphism for $\exists^+\mathcal{L}^k_\infty$ as required.

Case 2 Suppose $\mathcal{S}$ is an injective forth system and let $\phi(\mathbf{x}) = \exists^{\geq m} x.\phi'(x, \mathbf{x})$. If we have that $\mathcal{A}, \mathbf{a} \models \phi(\mathbf{x})$ for some $\mathbf{a} \subset \mathbf{dom}(p)$ then there are m distinct elements $a_1, \ldots, a_m \in A$ such that $\mathcal{A}, a_i, \mathbf{a} \models \phi'(x, \mathbf{x})$. Now the injective forth property of $\mathcal{S}$ implies that there is an injection $f \colon A \to B$ such that for all $a \in A$ $p \cup \{(a, f(a))\} \in \mathcal{S}$. By induction all of these partial homomorphisms preserve ϕ' and so in particular we have that for each a_i $\mathcal{B}, f(a_i), p(\mathbf{a}) \models \phi'(x, \mathbf{x})$. The injectivity of f implies that $f(a_1), \ldots f(a_m)$ are all distinct and so $\mathcal{B}, p(\mathbf{a}) \models \exists^{\geq m} x.\phi'(x, \mathbf{x})$ and p preserves ϕ. By induction and noting that the induction in Case 1 also applies here, we have that every $p \in \mathcal{S}$ is a logical homomorphism for $\exists^+\mathcal{L}^k_\infty(\exists^{\geq m})$, as required.

Case 3 Suppose $\mathcal{S}$ is a surjective forth system and let $\phi(\mathbf{x}) = \forall^{\leq n} x.\phi'(x, \mathbf{x})$. We show that p preserves ϕ by proving the contrapositive. Suppose that $\mathcal{B}, p(\mathbf{a}) \not\models \phi(\mathbf{x})$. Then there must be distinct elements $b_1, \ldots b_{m+1}$ such that $\mathcal{B}, b_i, p(\mathbf{a}) \not\models \phi'(x, \mathbf{x})$. Now consider that the surjective forth condition of $\mathcal{S}$ at p ensures that there is a surjection $f \colon A \to B$ such that, for every $a \in A$, $p \cup \{(a, f(a))\} \in \mathcal{S}$. By induction each of these functions preserve ϕ'. As f is a surjection, we must have distinct elements $a_1, \ldots a_{m+1} \in A$ such that $f(a_i) = b_i$ for each $i \in [m+1]$. So by preservation of ϕ' we have that, for each $i \in [m+1]$, $\mathcal{A}, a_i, \mathbf{a} \not\models \phi'(x, \mathbf{x})$ and so $\mathcal{A}, \mathbf{a} \not\models \forall^{\leq m}.\phi'(x, \mathbf{x})$. This shows that p preserves ϕ and so by induction all partial homomorphisms in $\mathcal{S}$ are logical homomorphisms for $\exists^+\mathcal{L}^k_\infty(\forall^{\leq m})$.

Case 4 When $\mathcal{S}$ is a bijective forth system we have that, in particular, it is also an injective and surjective forth system. This means that the induction steps in the three previous cases all apply here and so by induction any $p \in \mathcal{S}$ is a logical homomorphism $\exists^+\mathcal{L}^k_\infty(\exists^{\geq m}, \forall^{\leq m})$, as required.

This completes the forward direction of the proof.

We now prove the other direction by establishing that the set of logical homomorphisms $\mathcal{S}_\mathcal{L}(\mathcal{A}, \mathcal{B})$ one of the four logics above is also a forth system of the appropriate type, as stated in the following claim.

Claim 4.12. *For any two structures $\mathcal{A}$ and $\mathcal{B}$, the sets of logical homomorphisms between $\mathcal{A}$ and $\mathcal{B}$ for the logics $\exists^+\mathcal{L}^k$, $\exists^+\mathcal{L}^k(\exists^{\geq k})$, $\exists^+\mathcal{L}^k(\forall^{\leq k})$ and $\exists^+\mathcal{L}^k(\forall^{\leq k}, \exists^{\geq k})$ are respectively a forth system, an injective forth system, a surjective forth system and a bijective forth system.*

Proof of claim. It is firstly clear from the definition that any set $\mathcal{S}_\mathcal{L}(\mathcal{A}, \mathcal{B})$ of logical homomorphisms is downward-closed. Indeed the relation $\mathcal{A}, a_1, \ldots, a_m \Rrightarrow_\mathcal{L} \mathcal{B}, b_1, \ldots, b_m$ implies $\mathcal{A}, a_{i_1}, \ldots, a_{i_l} \Rrightarrow_\mathcal{L} \mathcal{B}, b_{i_1}, \ldots, b_{i_l}$ for any subset of the chosen tuples. So it remains to show in each of the cases that $\mathcal{S}_\mathcal{L}(\mathcal{A}, \mathcal{B})$ satisfies the required forth property. To this end, we consider the following structure.

For any pair of tuples $\mathbf{a}$ in A and $\mathbf{b}$ in B we define the bipartite graph $\mathcal{G}^\mathcal{L}_{\mathbf{a},\mathbf{b}}$ to have two sets of vertices A and B and the edge relation

$$E(\mathcal{G}^\mathcal{L}_{\mathbf{a},\mathbf{b}}) := \{(a, b) \mid \mathcal{A}, \mathbf{a}, a \Rrightarrow_\mathcal{L} \mathcal{B}, \mathbf{b}, b\}.$$

To realise this in the logic $\mathcal{L}$, we define the $\mathcal{L}$-type of $a \in A$ with respect to $\mathbf{a}$ as the set

$$\Phi^\mathcal{L}_{\mathbf{a},a} = \{\phi(\mathbf{x}, x) \in \mathcal{L} \mid \mathcal{A}, \mathbf{a}, a \models \phi(\mathbf{x}, x)\}.$$

Then the, in general infinite, formula $\phi^\mathcal{L}_{\mathbf{a},a}(\mathbf{x}, x) := \bigwedge_{\phi \in \Phi^\mathcal{L}_{\mathbf{a},a}} \phi(\mathbf{x}, x)$ can be used to pick out the neighbourhood $\mathcal{N}(a)$ of a in $\mathcal{G}^\mathcal{L}_{\mathbf{a},\mathbf{b}}$ as

$$b \in \mathcal{N}(a) \iff \mathcal{B}, \mathbf{b}, b \models \phi^\mathcal{L}_{\mathbf{a},a}(\mathbf{x}, x).$$

We drop the superscript $\mathcal{L}$ when the logical fragment is clear from context.

Now if the partial function $\mathbf{a} \mapsto \mathbf{b}$ is a logical homomorphism for the logic $\mathcal{L}$ between $\mathcal{A}$ and $\mathcal{B}$, the various forth properties correspond to different types of matching in the bipartite graph $\mathcal{G}^\mathcal{L}_{\mathbf{a},\mathbf{b}}$. In particular,

- the forth property says that the graph contains a function from A to B,
- the injective forth property says that the graph has a matching which is total on A (which is also an injective function from A to B),
- the surjective forth property says that the graph contains a function from A to B and has a matching which is total on B (which is not necessarily a function from A to B), and
- the bijective forth property says that the graph has a perfect matching.

Now we show that for $\mathcal{L} = \exists^+\mathcal{L}^k$, $\exists^+\mathcal{L}^k(\exists^{\geq k})$, $\exists^+\mathcal{L}^k(\forall, \exists^{\leq k})$ or $\exists^+\mathcal{L}^k(\forall, \exists^{\leq k}, \exists^{\geq k})$ these matchings are guaranteed for tuples $\mathbf{a}, \mathbf{b}$ of size less than k by the relation $\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}} \mathcal{B}, \mathbf{b}$.

Case 1: $\mathcal{L} = \exists^+\mathcal{L}^k_\infty$ In this case, we need to show that for each a there is an element $b \in \mathcal{N}(a)$. However, as $\mathcal{A}, \mathbf{a}, a \models \phi_{\mathbf{a},a}(\mathbf{x}, x)$ we have that $\mathcal{A}, \mathbf{a} \models \exists x.\ \phi_{\mathbf{a},a}(\mathbf{x}, x)$. Now, as this formula is defined in $\exists^+\mathcal{L}^k$ the relation $\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}} \mathcal{B}, \mathbf{b}$ guarantees that $\mathcal{B}, \mathbf{b} \models \exists x.\ \phi_{\mathbf{a},a}(\mathbf{x}, x)$ and so any witness b for the outer existential of this formula is in $\mathcal{N}(a)$, as required.

Case 2: $\mathcal{L} = \exists^+\mathcal{L}^k_\infty(\exists^{\geq m})$ In this case, we need to show that $\mathcal{G}^{\mathcal{L}}_{\mathbf{a},\mathbf{b}}$ has a matching which is total on A. By Hall's Marriage Theorem, this is equivalent to the condition that for any subset $W \subset A$ the joint neighbourhood $\mathcal{N}(W) := \bigcup_{a \in W} \mathcal{N}(a)$ satisfies the following inequality

$$|W| \leq |\mathcal{N}(W)|.$$

This can be expressed in a formula of $\exists^+\mathcal{L}^k(\exists^{\geq m})$ as follows. Let $\phi_W(\mathbf{x}, x) := \bigvee_{a \in W} \phi_{\mathbf{a},a}(\mathbf{x}, x)$. As with $\phi_{\mathbf{a},a}$, it is clear that for any $b \in B$, $\mathcal{B}, \mathbf{b}, b \models \phi_W(\mathbf{x}, x)$ if and only if $b \in \mathcal{N}(W)$. However, using the $\exists^{\geq m}$ quantifier, we have that

$$\mathcal{A}, \mathbf{a} \models \exists^{\geq |W|} x.\ \phi_W(\mathbf{x}, x)$$

and so, as $\mathcal{A}, \mathbf{a} \Rrightarrow_{\exists^+\mathcal{L}^k(\exists^{\geq m})} \mathcal{B}, \mathbf{b}$, we have that $\mathcal{B}, \mathbf{b} \models \exists^{\geq |W|} x.\ \phi_W(\mathbf{x}, x)$. This gives that $|\mathcal{N}(W)| \geq |W|$ as required.

Case 3: $\mathcal{L} = \exists^+\mathcal{L}^k_\infty(\forall^{\leq m})$ For this direction we need a slight variation on the argument in Case 2. The condition from Hall's Marriage Theorem for $\mathcal{G}$ to have a matching which is total on B is that, for any $V \subset B$, $|\mathcal{N}(V)| \geq |V|$. To do this we want to take disjunction over types which are not in the neighbourhood $\mathcal{N}(V)$ given by

$$\phi_{\mathcal{N}(V)^c}(\mathbf{x}, x) = \bigvee_{a \notin \mathcal{N}(V)} \phi_{\mathbf{a},a}(\mathbf{x}, x).$$

We then consider the formula

$$\forall^{\leq |\mathcal{N}(V)|} x.\ \phi_{\mathcal{N}(V)^c}(\mathbf{x}, x).$$

This is clearly satisfied by $\mathcal{A}, \mathbf{a}$ and the preservation of $\mathcal{L}$ formulas in this case ensures that it is satisfied by $\mathcal{B}, \mathbf{b}$. However, for every $b \in V$, we must have that $\mathcal{B}, \mathbf{b}, b \not\models \phi_{\mathcal{N}(V)^c}(\mathbf{x}, x)$ because if this were the case there would be some $a \in \mathcal{N}(V)^c$ such that $\mathcal{B}, \mathbf{b}, b \models \phi_{\mathbf{a},a}(\mathbf{x}, x)$ which would mean that a is in the neighbourhood of b and thus is in $\mathcal{N}(V)$, a contradiction. This gives that $|V| \leq |\mathcal{N}(V)|$, as required.

Case 4: $\mathcal{L} = \exists^+\mathcal{L}^k(\exists^{\geq m}, \forall^{\leq m})$ This case combines Cases 2 and 3. As all of the formulas defined above are preserved by the relation $\mathcal{A}, \mathbf{a} \Rrightarrow_{\mathcal{L}} \mathcal{B}, \mathbf{b}$ in this case, we have that the the bipartite graph contains a matching which is total on both A and B and so is a perfect matching. □

□

This lemma completes the proof of Proposition 4.6 which, together with Proposition 4.5, completes the proof of Theorem 4.3. This establishes that the branch-injective, branch-surjective and branch-bijective maps of the Kleisli category of $\mathbb{P}_k$ can also be used to capture logical relations between structures. In the next section, we compare these maps with the category theoretic notions of monomorphisms and epimorphisms in $\mathcal{K}(\mathbb{P}_k)$.

4.2 Monomorphisms and epimorphisms in $\mathcal{K}(\mathbb{P}_k)$

In Section 4.1, we provided a new characterisation of morphisms in the category $\mathcal{K}(\mathbb{P}_k)$ based on the game-theoretic notions of branch-injectivity and branch-surjectivity. In this section, we aim to study these morphisms based on their category-theoretic rather than game-theoretic properties. In particular, we want to classify the monomorphisms and epimorphisms of $\mathcal{K}(\mathbb{P}_k)$.

Recall that for a general category the definitions of epimorphism and monomorphism are as follows.

Definition 4.13. *For a category* $\mathbf{C}$*, a map* $m : A \to B$ *in* $\mathbf{C}$ *is a* monomorphism *if for every object* X *in* $\mathbf{C}$ *and any two morphisms* $g, h : X \to A$ $m \circ g = m \circ h$ *implies* $g = h$*. If there exists such an* m *we write* $A \rightarrowtail B$*.*

Dually, a map $e : A \to B$ *in* $\mathbf{C}$ *is an* epimorphism *if for every object* Y *in* $\mathbf{C}$ *and any two morphisms* $g, h : B \to Y$ $g \circ e = h \circ e$ *implies* $g = h$*. If there exists such an* e *we write* $A \rightharpoonup B$*.*

The monomorphisms and epimorphisms of $\mathcal{R}(\sigma)$ for any finite signature σ are, respectively, the injective and surjective homomorphisms. It can thus be reasonably hoped that the monomorphisms and epimorphisms in the Kleisli category of $\mathbb{P}_k$ are branch-injective and branch-surjective respectively. We see in this section that this hope is not quite realised in $\mathcal{K}(\mathbb{P}_k)$. Instead, we compare these notions directly by showing, in Proposition 4.14, that branch-injective and branch-surjective maps are respectively monomorphisms and epimorphisms. We also prove, however, in Proposition 4.15 that the converse of this entailment is false.

In the second part of this section however we show that some hope can be recovered by slightly tweaking $\mathbb{P}_k$. To this end, we introduce a new comonad $\mathbb{P}_k^*$ which satisfies exactly the same morphism and isomorphism power theorems as $\mathbb{P}_k$ but additionally satisfies a new monomorphism power theorem relating monomorphisms and branch-injective strategies. This new comonad can be thought of as removing some redundant information in the game described by the comonad $\mathbb{P}_k$. A similar construction relating epimorphisms and branch-surjective strategies has proven more elusive.

4.2.1 Branch-injective $\neq$ monomorphic in $\mathcal{K}(\mathbb{P}_k)$

Here we compare the game-theoretic notions of branch-injectivity and branch-surjectivity with the category-theoretic notions of monomorphic and epimorphic in the Kleisli category of $\mathbb{P}_k$. It is not hard to see that the branch-injective maps are indeed monomorphisms and branch-surjective maps are indeed epimorphisms in the category $\mathcal{K}(\mathbb{P}_k)$, as we establish in the following proposition.

Proposition 4.14. *Given two relational structures $\mathcal{A}$ and $\mathcal{B}$ and a Kleisli map $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, if f is branch-injective then f is a monomorphism in $\mathcal{K}(\mathbb{P}_k)$ and if it is branch-surjective then f is an epimorphism in $\mathcal{K}(\mathbb{P}_k)$.*

Proof. We prove the cases for branch-injective and branch-surjective separately.

Branch-injective $\implies$ monomorphic Suppose f is branch-injective and let $g, h\colon \mathbb{P}_k\mathcal{C} \to \mathcal{A}$ be two distinct Kleisli maps into $\mathcal{A}$. This means that there is some $s \in \mathbb{P}_k\mathcal{C}$ which is minimal in length such that $g(s) \neq h(s)$. Write $s = s'; (a, i)$ where $s' \in (A \times [k])^*$ is some (potentially) empty history of Spoiler moves. If s' is empty then the injectivity of $f_{\epsilon,i}$ ensures that $f \circ g \neq f \circ h$, as $(f \circ g)([(a, i)])) = [(f_{\epsilon,i}(g([(a, i)])), i)]$ while $(f \circ h)([(a, i)])) = [(f_{\epsilon,i}(h([(a, i)])), i)]$ and $g([(a, i)]) \neq h([(a, i)])$. If s' is not empty then, by minimality, $g^*(s') = h^*(s') = t$ and we can repeat a similar argument using the injectivity of $f_{t,i}$.

Branch-surjective $\implies$ epimorphic For this direction, it is sufficient to show that the Kleisli coextension $f^*\colon \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{B}$ is surjective. This is because if any Kleisli maps $g, h\colon \mathbb{P}_k\mathcal{B} \to \mathcal{C}$ differ on some $s \in \mathbb{P}_k\mathcal{B}$ then the surjectivity of f^* guarantees that there is a $t \in \mathbb{P}_k\mathcal{A}$ such that $f^*(t) = s$ and so $g \circ f$ and $h \circ f$ differ on this t.

We now establish the surjectivity of f^* from branch-surjectivity by induction on the length of elements in $\mathbb{P}_k\mathcal{B}$. Surjectivity for all i of $f_{\epsilon,i}$ guarantees that for any $[(b,i)] \in \mathbb{P}_k B$ there is an $a_i \in A$ such that $f^*([a_i, i]) = [(b,i)]$. Now assume that for every $s \in \mathbb{P}_k B$ of length $< r$ there is a $t \in \mathbb{P}_k A$ such that $f^*(t) = s$. Take any s' of length r and write this as $s' = s; (b,i)$. As s has length $< r$ by induction there is a t such that $f^*(t) = s$. Now the surjectivity of the branch map $f_{t,i}$ implies that there is an $a_i \in A$ such that $f_{t,i}(a_i) = b$ and so $f^*(t; (a_i, i)) = s'$, as required.

□

A tempting first guess for characterising the monomorphisms and epimorphisms of $\mathcal{K}(\mathbb{P}_k)$ would be to try and prove a converse of Proposition 4.14. However, we now see that this is not the case by showing that there are monomorphisms and epimorphisms which are not branch-injective or branch-surjective. The key point is that, as shown in Proposition 4.14, branch-injective and branch-surjective maps detect a difference between two different strategies g and h in the round that this difference occurs. It is possible, however, to devise Kleisli maps which can differentiate any g and h at some later round than the one in which the difference occurs. These maps are examples of monomorphisms and epimorphisms which are not branch-injective or surjective. We provide an example of such a map in the following proposition.

Proposition 4.15. *For any k and any relational structure σ, there are monomorphisms and epimorphisms in the category $\mathcal{K}(\mathbb{P}_k)$ which are neither branch-injective nor branch-surjective, where $\mathbb{P}_k$ is considered as a comonad over $\mathcal{R}(\sigma)$.*

Proof. For our main counterexample we consider only the 1-pebble game and the corresponding comonad $\mathbb{P}_1$ and only structures over the empty relational signature. However, the problematic behaviour observed in this case can be embedded into games with more pebbles and more complex signatures.

The example Consider the two sets **1Bit** $:= \{0,1\}$ and **2Bit** $= \{00, 01, 10, 11\}$. The example in this section is a Kleisli morphism in $\mathcal{K}(\mathbb{P}_1)$ from **2Bit** to **1Bit** which is both an epimorphism and a monomorphism but is neither branch-injective nor branch-surjective. The idea of this strategy is the following. After r rounds of the 1-pebble game the history s of Spoiler moves on **2Bit** is simply a list of r elements from the set $\{00, 01, 10, 11\}$. This can be seen as a binary string w_s of length $2r$. After each Spoiler move Duplicator is required to provide an element of **1Bit**. As the sets have no relations and the I-relation does not apply as there are fewer than 2 pebbles, this choice is entirely unrestricted. This means we can define a Duplicator strategy `readout` which for any r-round Spoiler history s returns the r^{th} bit of w_s. This is clearly not a branch-injective or branch-surjective strategy as the branch map $\texttt{readout}_{s,1} \colon \mathbf{2Bit} \to \mathbf{1Bit}$ is a constant function outputting the r^{th} bit of w_s which is neither injective nor surjective. We now prove, however, that `readout` is both a monomorphism and an epimorphism.

We prove this by showing separately that `readout` is a monomorphism and an epimorphism.

`readout` is a monomorphism For this part, we take any morphisms $g, h \colon \mathbb{P}_1 C \to \mathbf{2Bit}$ such that $g \neq h$ and we need to show that compositions $\texttt{readout} \circ g$ and $\texttt{readout} \circ h$ are not equal. The fact that $g \neq h$ means that there is a history $t \in \mathbb{P}_1 C$ of length r such that $g(t) \neq h(t)$ and so the images of the Kleisli coextensions g^* and h^* also disagree on t. Write $s_1 := g^*(t)$ and $s_2 := h^*(t)$ for these distinct histories in $\mathbb{P}_1\mathbf{2Bit}$ of length r. Now we know that, as $g(t) \neq h(t)$, the strings w_{s_1} and w_{s_2} are strings of $2r$ bits which disagree on bit $2r-1$ or bit $2r$. This means that we can distinguish $\texttt{readout} \circ g$ and $\texttt{readout} \circ h$ as follows. Consider let t_1 and t_2 be any elements of $\mathbb{P}_1 C$ of lengths $r-1$ and r respectively. Then it is not hard to see that the output of $\texttt{readout} \circ g$ and $\texttt{readout} \circ h$ on $t; t_1$ and $t; t_2$ are respectively the $2r-1^{\text{th}}$ and $2r^{\text{th}}$ bits of s_1 and s_2. As s_1 and s_2 differ on these bits we have that $\texttt{readout} \circ g$ and $\texttt{readout} \circ h$ are distinct.

`readout` is an epimorphism This direction is somewhat easier as, by the argument employed in the proof of Proposition 4.14, it is sufficient to show that the Kleisli coextension $\texttt{readout}^* \colon \mathbb{P}_1\mathbf{2Bit} \to \mathbb{P}_1\mathbf{1Bit}$ is surjective.

To see that $\texttt{readout}^*$ is indeed surjective, note that for any sequence $t \in \mathbb{P}_1\mathbf{2Bit}$, $\texttt{readout}^*(t)$ is, by design, the Spoiler history marking the first r bit of the $2r$ bit string w_t. So to hit any $s \in \mathbb{P}_1\mathbf{1Bit}$ we just write any bit string w which is twice the length of w_s, the first half of which is exactly w_s. We can then choose a sequence of elements of $\mathbf{2Bit}$ whose concatenation is w. This sequence of elements is exactly the history t such that $\texttt{readout}^*(t) = s$ and we are done.

□

Proposition 4.15 has established that in the category $\mathcal{K}(\mathbb{P}_k)$ monomorphisms are not necessarily branch-injective and epimorphisms are not necessarily branch-surjective. This appears to leave open the interesting question of whether there *exists* a branch-injective map between two structures exactly whenever there *exists* a monomorphism between (and dually for branch-surjective maps and epimorphisms). For monomorphisms, the example provided in Proposition 4.15 precludes this possibility as $|\mathbf{2Bit}| > |\mathbf{1Bit}|$ so there can be no branch-injective map. The question remains open for branch-surjective maps and epimorphisms. In the next section, we exhibit a new comonadic construction $\mathbb{P}_k^*$ where branch-injective maps (i.e. Duplicator winning strategies in $+\mathbf{Inj}^h$) are exactly monomorphisms.

4.2.2 $\mathbb{P}_k^*$ and the Monomorphism Power Theorem

In this section, we introduce $\mathbb{P}_k^*$, a modified version of the $\mathbb{P}_k$ comonad which captures a variant of the k-pebble game where the reuse of single pebbles is more tightly controlled. As we will show, this modification does not affect the logical relations captured by Kleisli morphisms and isomorphisms between I-structures but it does result in a converse of Proposition 4.14 in this new setting, in the form of the following theorem, which is the main aim of this section.

Theorem 4.16 (Monomorphism Power Theorem for $\mathbb{P}_k^*$)**.** *For any two relational structures $\mathcal{A}$ and $\mathcal{B}$, the following are equivalent for any positive integer k:*

1. *There is a Kleisli monomorphism $m\colon \mathbb{P}_k^*\mathcal{A} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$,*
2. *Duplicator has a winning strategy in the game $+\mathbf{Inj}^k(\mathcal{A}, \mathcal{B})$, and*
3. *$\mathcal{A} \Rrightarrow_{\exists^+\mathcal{L}^k(\exists^{\geq m})} \mathcal{B}$.*

To understand the modifications we want to make to $\mathbb{P}_k$ for this theorem to hold, we need to understand the types of Spoiler moves in the k-pebble game we are trying to exclude. These moves communicate no extra information between Spoiler and Duplicator but form the basis for the pathological monomorphism we saw in the proof of Proposition 4.15. The moves fall into two types which we call *procrastination moves* and *prevarication moves.* In a procrastination move, Spoiler picks up a pebble labelled i which is placed at some element $a \in A$ and places it right back down on the same element. In a prevarication move, Spoiler moves the same pebble two or more times in a row to potentially different elements. We define these formally as follows.

Definition 4.17. *For any k and any structure $\mathcal{A}$ we call a Spoiler history $s \in \mathbb{P}_k\mathcal{A}$ a* procrastination move *if s is of the form $s'; (a, i); m; (a, i)$ where s' and m are (potentially empty) sequences of Spoiler moves such that the pebble index i does not appear in m.*

We call s a prevarication move *if it is of the form $s'; m; (a, i)$ where the sequence m is non-empty and the only pebble index appearing in m is i.*

In the Kleisli category of the new comonad $\mathbb{P}_k^*$ we want Duplicator strategies to essentially ignore these procrastination and prevarication moves. To do this we want to take a quotient of the structure $\mathbb{P}_k\mathcal{A}$ by an equivalence relation which equates Spoiler histories to all equivalent histories which have been extended by procrastination and prevarication. To this end, we give the following definition.

Definition 4.18. *For any structure $\mathcal{A}$, we define the equivalence relation $\approx_*$ on $\mathbb{P}_k\mathcal{A}$ to be the reflexive, transitive, symmetric closure of the relation which contains*

- *all pairs* (s,t) *where* s *is a procrastination move of the form* $s'; (a,i); m; (a,i)$ *and* $t = s'; (a,i)$ *and*
- *all pairs* (s,t) *where* s *is a prevarication move of the form* $s'; m; (a,i)$ *and* $t = s'; (a,i)$.

Recalling the notion of quotienting a relational structure described in 2, this relation allows us to define a structure $\mathbb{P}_k^*\mathcal{A} := \mathbb{P}_k\mathcal{A}/\approx_*$ which has as elements the equivalence classes of $\approx_*$ and has all the relations required to make the quotient map $q\colon \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{A}/\approx_*$ a homomorphism. For brevity, we write $q(s)$ as $[s]$ in this section.

To apply this quotient to the entire comonad $(\mathbb{P}_k, \epsilon, \delta)$ we need to do more than quotienting the relational structure $\mathbb{P}_k\mathcal{A}$. In particular, we also need to define maps of the following forms:

- $\mathbb{P}_k^*f\colon \mathbb{P}_k^*\mathcal{A} \to \mathbb{P}_k^*\mathcal{B}$, for every $f\colon \mathcal{A} \to \mathcal{B}$,
- $\epsilon_{\mathcal{A}}^*\colon \mathbb{P}_k^*\mathcal{A} \to \mathcal{A}$, for every $\mathcal{A}$, and
- $\delta_{\mathcal{A}}^*\colon \mathbb{P}_k^*\mathcal{A} \to \mathbb{P}_k^*(\mathbb{P}_k^*\mathcal{A})$, for every $\mathcal{A}$.

We construct these maps in the following lemma.

Lemma 4.19. *For any structure* $\mathcal{A}$ *and any homomorphism* $f\colon \mathcal{A} \to \mathcal{B}$ *the following are well-defined homomorphisms of the types given above:*

1. $\mathbb{P}_k^*f([s]) := [\mathbb{P}_kf(s)]$,
2. $\epsilon_{\mathcal{A}}^*([s]) := \epsilon_{\mathcal{A}}(s)$, *and*
3. $\delta_{\mathcal{A}}^*([s]) := [\mathbb{P}_kq(\delta_{\mathcal{A}}(s))]$.

Proof. In all of these cases to check that a function F is well-defined we need to show that for any $s \approx_{*t}$ we have $F([s]) = F([t])$. As $\approx_*$ is defined in Definition 4.18 as the transitive closure of so-called procrastination and prevarication pairs, it suffices to check that if (s,t) is such a pair then $F([s]) = F([t])$. We now proceed to check each of the functions defined above.

Of the three functions to check, 2 is by far the easiest. This is because if (s,t) is one of the pairs defined in Definition 4.18 then clearly $\epsilon_{\mathcal{A}}(s) = \epsilon_{\mathcal{A}}(t)$.

For 1, we note that, given some Spoiler history s, the function $\mathbb{P}_kf$ returns a history with exactly the same sequence of pebble indices moved but with positions of the pebbles changed by applying the function f. Now suppose (s,t) is one of the pairs from Definition 4.18, then $s = s'; (a,i); m; (a,i)$ or $s = s'; m; (a,i)$ where m contains either no pebble indices i or consists entirely of pebble indices i and $t = s'; (a, i)$. In either case, the images of s under $\mathbb{P}_k f$ are $\mathbb{P}_k f(s) = \mathbb{P}_k f(s'); (f(a), i); \mathbb{P}_k f(m); (f(a), i)$ or $s = \mathbb{P}_k f(s'); \mathbb{P}_k f(m); (f(a), i)$ respectively where the pebble indices in $\mathbb{P}_k f(m)$ are unchanged. So, $(\mathbb{P}_k f(s), \mathbb{P}_k f(t))$ is still a related pair. This means that $\mathbb{P}_k f(s) \approx_* \mathbb{P}_k f(t)$, and so $\mathbb{P}^*_k f(s) = \mathbb{P}^*_k f(t)$ as required.

To check that $\delta^*_{\mathcal{A}}$ is well-defined, we have to deal with two layers of quotienting. Indeed, given (s, t) a procrastination or prevarication pair in $\mathbb{P}_k\mathcal{A}$, we need to show that $\mathbb{P}_k q(\delta_{\mathcal{A}}(s))$ and $\mathbb{P}_k q(\delta_{\mathcal{A}}(t))$ are related by $\approx_*$ as elements of $\mathbb{P}_k(\mathbb{P}^*_k\mathcal{A})$. To see this let $s = s'; (a, i); m; (a, i)$ or $s = s'; m; (a, i)$ (where m contains no indices i or only indices i respectively) and let $t = s'; (a, i)$. Then the images of these histories under $\delta_{\mathcal{A}}$ are $\delta_{\mathcal{A}}(s) = \delta_{\mathcal{A}}(s'); (t, i); \tilde{m}; (s, i)$ or $\delta_{\mathcal{A}}(s) = \delta_{\mathcal{A}}(s'); \tilde{m}; (s, i)$ where $\tilde{m}$ contains no indices i or only indices i respectively) and $\delta_{\mathcal{A}}(t) = \delta_{\mathcal{A}}(s'); (t, i)$. These histories do not in general, satisfy the conditions to be related by $\approx_*$ in $\mathbb{P}_k\mathbb{P}_k\mathcal{A}$ as $s \neq t$. However, applying $\mathbb{P}_k q$ to each, replaces the occurences of s and t with $q(s)$ and $q(t)$ which are equal as $s \approx_* t$. Therefore, in $\mathbb{P}_k\mathbb{P}_k\mathcal{A}$, $(\mathbb{P}_k q(\delta_{\mathcal{A}}(s)), \mathbb{P}_k q(\delta_{\mathcal{A}}(t)))$ is a related pair, as required. □

Having given well-defined liftings of the functor $\mathbb{P}_k$ and the counit and comultiplication, ϵ and δ we now prove that these new maps define a new comonad.

Proposition 4.20. *For any signature σ, the triple $(\mathbb{P}^*_k, \epsilon^*, \delta^*)$ is a comonad on $\mathcal{R}(\sigma)$.*

Proof. We prove this result in two parts. First we establish that $\mathbb{P}^*_k$ is a functor and ϵ^* and δ^* are natural transformations and we then show that they satisfy the comonad laws from Definition 2.8. Throughout, we use the fact that the definition of $\mathbb{P}^*_k$ has been chosen such that, for the quotient map q, we have $(\mathbb{P}^*_k f) \circ q = q \circ (\mathbb{P}_k f)$.

Part 1 To show that $\mathbb{P}^*_k$ is a functor, we need to know that $\mathbb{P}^*_k(\mathbf{id}_{\mathcal{A}}) = \mathbf{id}_{\mathbb{P}^*_k\mathcal{A}}$ and for any $f : \mathcal{A} \to \mathcal{B}$ and $g : \mathcal{B} \to \mathcal{C}$ we have $\mathbb{P}^*_k(g \circ f) = \mathbb{P}^*_k(g) \circ (\mathbb{P}^*_k(f))$. Both follow immediately from the definitions and the functoriality of $\mathbb{P}_k$ as $\mathbb{P}^*_k(\mathbf{id}_{\mathcal{A}})([s]) = [\mathbb{P}_k(\mathbf{id}_{\mathcal{A}})(s)] = [s]$ and $\mathbb{P}^*_k(g \circ f)([s]) = [\mathbb{P}_k(g \circ f)(s)] = [\mathbb{P}_k(g) \circ \mathbb{P}_k(f)(s)] = \mathbb{P}^*_k(g)([\mathbb{P}_k(f)(s)]) = \mathbb{P}^*_k(g)(\mathbb{P}^*_k(f)([s]))$.

To show that ϵ and δ are natural transformations we need that for any $f : \mathcal{A} \to \mathcal{B}$ and any $s \in \mathbb{P}_k\mathcal{A}$ we have

$$f(\epsilon^*_{\mathcal{A}}([s])) = \epsilon^*_{\mathcal{B}}(\mathbb{P}^*_k f([s]))$$

and

$$\mathbb{P}^*_k\mathbb{P}^*_k f(\delta^*_{\mathcal{A}}([s])) = \delta^*_{\mathcal{B}}(\mathbb{P}^*_k f([s])).$$

The derivations for these are fairly mechanical and rely on the fact that ϵ and δ are natural transformations, $\mathbb{P}_k$ and $\mathbb{P}^*_k$ are functors and the definitions of $\mathbb{P}^*_k f$, ϵ^* and δ^*. We present these derivations below for completeness. For the first this goes as follows

$$\begin{aligned}
f(\epsilon^*_{\mathcal{A}}([s])) &= f(\epsilon_{\mathcal{A}}(s)) \\
&= \epsilon_{\mathcal{B}}(\mathbb{P}_k f(s)) \\
&= \epsilon^*_{\mathcal{B}}([\mathbb{P}_k f(s)]) \\
&= \epsilon^*_{\mathcal{B}}(\mathbb{P}^*_k f(s)).
\end{aligned}$$

And the second is derived in the following way, recalling that $\mathbb{P}^*_k f \circ q = q \circ \mathbb{P}_k f$,

$$\begin{aligned}
\mathbb{P}^*_k\mathbb{P}^*_k f(\delta^*_{\mathcal{A}}([s])) &= \mathbb{P}^*_k\mathbb{P}^*_k f([\mathbb{P}_k q(\delta_{\mathcal{A}}(s))]) \\
&= [\mathbb{P}_k(\mathbb{P}^*_k f)(\mathbb{P}_k q(\delta_{\mathcal{A}}(s)))] \\
&= [\mathbb{P}_k((\mathbb{P}^*_k f) \circ q)(\delta_{\mathcal{A}}(s))] \\
&= [\mathbb{P}_k(q \circ (\mathbb{P}_k f))(\delta_{\mathcal{A}}(s))] \\
&= [\mathbb{P}_k q(\mathbb{P}_k\mathbb{P}_k f(\delta_{\mathcal{A}}(s)))] \\
&= [\mathbb{P}_k q(\delta_{\mathcal{B}}(\mathbb{P}_k f(s)))] \\
&= \delta^*_{\mathcal{B}}(\mathbb{P}^*_k f([s])).
\end{aligned}$$

Part 2 We now show that $\mathbb{P}^*_k$ satisfies the counit and comultiplication laws which are respectively

$$\mathbb{P}^*_k\epsilon^*_{\mathcal{A}} \circ \delta^*_{\mathcal{A}} = \epsilon^*_{\mathbb{P}^*_k\mathcal{A}} \circ \delta^*_{\mathcal{A}} = \mathbf{id}_{\mathbb{P}^*_k}$$

and

$$\mathbb{P}^*_k\delta^*_{\mathcal{A}} \circ \delta^*_{\mathcal{A}} = \delta^*_{\mathbb{P}^*_k\mathcal{A}} \circ \delta^*_{\mathcal{A}}.$$

As in the last part we now give the detailed derivation of these using simple rewriting rules given by the above definitions, naturality of ϵ and δ and the counit and comultiplication rules for $\mathbb{P}_k$.

We first prove that for any $s \in \mathbb{P}_k\mathcal{A}$, $\mathbb{P}^*_k\epsilon^*_{\mathcal{A}} \circ \delta^*_{\mathcal{A}}([s]) = [s]$.

$$\begin{aligned}
\mathbb{P}^*_k\epsilon^*_{\mathcal{A}} \circ \delta^*_{\mathcal{A}}([s]) &= \mathbb{P}^*_k\epsilon^*_{\mathcal{A}}([\mathbb{P}_k q(\delta_{\mathcal{A}}(s))]) \\
&= [\mathbb{P}_k\epsilon^*_{\mathcal{A}}(\mathbb{P}_k q(\delta_{\mathcal{A}}(s)))] \\
&= [\mathbb{P}_k(\epsilon^*_{\mathcal{A}} \circ q)(\delta_{\mathcal{A}}(s))] \\
&= [\mathbb{P}_k(\epsilon_{\mathcal{A}})(\delta_{\mathcal{A}}(s))] \\
&= [\mathbf{id}_{\mathcal{A}}(s)] = [s].
\end{aligned}$$

For the other counit identity we show that $\epsilon^*_{\mathbb{P}^*_k\mathcal{A}} \circ \delta^*_{\mathcal{A}}([s]) = [s]$.

$$\begin{aligned}
\epsilon^*_{\mathbb{P}^*_k\mathcal{A}}(\delta^*_{\mathcal{A}}([s])) &= \epsilon^*_{\mathbb{P}^*_k\mathcal{A}}([\mathbb{P}_k q(\delta_{\mathcal{A}}(s))]) \\
&= \epsilon_{\mathbb{P}_k\mathcal{A}} \circ (\mathbb{P}_k q)(\delta_{\mathcal{A}}(s)) \\
&= q \circ (\epsilon_{\mathbb{P}_k\mathcal{A}})(\delta_{\mathcal{A}}(s)) \\
&= [\epsilon_{\mathbb{P}_k\mathcal{A}}(\delta_{\mathcal{A}}(s))] \\
&= [s].
\end{aligned}$$

We now conclude by deriving the comultiplication law, i.e. that, for any s, $\mathbb{P}^*_k\delta^*_{\mathcal{A}}(\delta^*_{\mathcal{A}}([s])) = \delta^*_{\mathbb{P}^*_k\mathcal{A}}(\delta^*_{\mathcal{A}}([s]))$.

$$\begin{aligned}
\mathbb{P}^*_k\delta^*_{\mathcal{A}}(\delta^*_{\mathcal{A}}([s])) &= [\mathbb{P}_k\delta^*_{\mathcal{A}}(\mathbb{P}_k q(\delta_{\mathcal{A}}(s)))] \\
&= [\mathbb{P}_k(\delta^*_{\mathcal{A}} \circ q)(\delta_{\mathcal{A}}(s))] \\
&= [\mathbb{P}_k(q \circ (\mathbb{P}_k q \circ \delta_{\mathcal{A}}))(\delta_{\mathcal{A}}(s))] \\
&= [\mathbb{P}_k(q \circ \mathbb{P}_k q)(\mathbb{P}_k\delta_{\mathcal{A}}(\delta_{\mathcal{A}}(s)))] \\
&= [\mathbb{P}_k(q \circ \mathbb{P}_k q)(\delta_{\mathbb{P}_k\mathcal{A}}(\delta_{\mathcal{A}}(s)))] \\
&= [(\mathbb{P}_k q \circ \mathbb{P}_k\mathbb{P}_k q \circ (\delta_{\mathbb{P}_k\mathcal{A}} \circ \delta_{\mathcal{A}})(s)] \\
&= [(\mathbb{P}_k q \circ (\delta_{\mathbb{P}^*_k\mathcal{A}} \circ \mathbb{P}_k q \circ \delta_{\mathcal{A}})(s)] \\
&= \delta^*_{\mathbb{P}^*_k}(\delta^*_{\mathcal{A}}([s])).
\end{aligned}$$

This completes the proof of the proposition.

□

We now prove an important result which establishes that the morphism and isomorphism power theorems for this new comonad are the same as those proved in Chapter 3 for $\mathbb{P}_k$

Proposition 4.21. *For any two structures $\mathcal{A}, \mathcal{B}$, there is a $\mathbb{P}^*_k$ Kleisli morphism (resp. isomorphism) between the I-structures $\mathcal{A}, \mathcal{B}$ if and only if there is a $\mathbb{P}_k$ Kleisli morphism (resp. isomorphism) between the I-structures $\mathcal{A}, \mathcal{B}$*

Proof. We first prove the correspondence for morphisms. One direction is easy. If there is a Kleisli map $f \colon \mathbb{P}^*_k\mathcal{A} \to \mathcal{B}$ then the composite map $f \circ q \colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ is a Kleisli map for $\mathbb{P}_k$.

For the other direction, we recall that a Kleisli map $f \colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ is a winning strategy for the existential k-pebble game from $\mathcal{A}$ to $\mathcal{B}$. This, in turn, is equivalent to the existence of a forth system $f \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$, as shown in Lemma 4.8. We can now use this to construct a homomorphism $f' \colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ such that for any $s \approx_* t$, we have $f'(s) = f'(t)$. Such a map would be a well-defined homomorphism $f' \colon \mathbb{P}^*_k\mathcal{A} \to \mathcal{B}$ and so we would be done. To construct f' we define the response to each Spoiler history in order of length, using the set $\mathcal{S}_f$. For each s, if s is either a prevarication or procrastination move, then we give the same response assigned to the earlier move to which s is related. Otherwise, we use the forth property of $\mathcal{S}_f$ and the current position of the game after s has been played to determine the next moves.

For the case of isomorphisms we can make a similar case based on the bijective forth systems which are equivalent to Duplicator winning strategies in the k-pebble bijection game which are captured by Kleisli isomorphisms in $\mathbb{P}_k$. □

With this more mundane set-up out of the way, we come at last to the true object of interest in this section: a characterisation of the Duplicator winning strategies for $+\mathbf{Inj}^k$ as the monomorphisms of the category $\mathcal{K}(\mathbb{P}_k^*)$. One direction of this characterisation is given in Proposition 4.14, which is easily lifted to the setting of $\mathbb{P}_k^*$. So it remains to show that any monomorphism in $\mathcal{K}(\mathbb{P}_k^*)$ is branch-injective. We first outline informally the strategy for proving this direction before concluding with a full proof of Theorem 4.16.

In Proposition 4.15, the example of a monomorphism in $\mathcal{K}(\mathbb{P}_k)$ which was not branch-injective had a slightly subtle way of distinguishing between different sequences of Spoiler moves. In branch-injective maps, the corresponding Duplicator strategy must distinguish between the sequences of Spoiler moves at the earliest round where a difference occurs. However, the `readout` function did not do this, instead encoding the difference in its response to late moves. The strategy for proving Theorem 4.16 is to show that in $\mathbb{P}_k^*\mathcal{A}$ we can construct sequences of Spoiler moves which *force* Duplicator to distinguish any differences in the round in which they occur. We now set up the definition required for this.

Definition 4.22. *For Spoiler histories $s, s' \in \mathbb{P}_k A$ in the existential k-pebble game we say that s'* collapses to *s if $s \sqsubset s'$ and there is some prefix $s'' \sqsubset s$ such that either $s'' \approx_* s'$ or there exists a sequence $s'' = s_0 \sqsubset s_1 \sqsubset \ldots \sqsubset s_l = s'$ where for each $0 \leq i < l$, $(s_i, s_{i+1}) \in I^{\mathbb{P}_k \mathcal{A}}$. We call s'' a* witness *of the collapse from s' to s.*

We say that a Duplicator strategy $g\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, collapses after *$s$ if for every Spoiler history s' such that $s \sqsubset s'$ the history $g^*(s')$ collapses to $g^*(s)$. We say that a collection of Spoiler histories $(s_t)_{t \in \mathbb{P}_k A}$ witnesses the collapse of g after s if for each $t \in \mathbb{P}_k A$, s_t witnesses the collapse of $g^*(s;t)$ to $g^*(s)$.*

It is not hard to see that when we have some $s, s', s'' \in \mathbb{P}_k A$ such that s'' witnesses the collapse of s' to s then for any strategy $f\colon \mathbb{P}_k^*\mathcal{A} \to \mathcal{B}$ the Duplicator response $f([s'])$ is equal to $f([s''])$. Furthermore, if a strategy $g\colon \mathbb{P}_k\mathcal{C} \to A$ collapses after s then the output of $f \circ g$ on any $s' \sqsupset s$ is determined by the series of responses $f^*(g^*(s))$, with the witness for each s' indicating the relevant prefix.

We now see how to use this notion of collapse to prove that monomorphisms between I-structures in the category $\mathcal{K}(\mathbb{P}_k^*)$ are also branch-injective. This allows us to complete the proof of the Monomorphism Power Theorem.

Proof of Theorem 4.16. Several parts of this equivalence have already been established earlier in this chapter. Indeed, the equivalence of 2 and 3 is one part of Proposition 4.6 and the implication 2 $\implies$ 1 follows from Propositions 4.5 and 4.14. Thus we focus in this proof on the implication from 1 to 2.

By the equivalence of Duplicator winning strategies for $+\mathbf{Inj}^k(\mathcal{A}, \mathcal{B})$ and branch-injective maps in $\mathcal{K}(\mathbb{P}_k)$ (Proposition 4.5) it suffices to show that any Kleisli monomorphism $m\colon \mathbb{P}_k^* \mathcal{A} \to \mathcal{B}$ between I-structures is also branch-injective when considered as a map in $\mathcal{K}(\mathbb{P}_k)$. To do this we show that for any $s \in \mathbb{P}_k A$ and $i \in [k]$ the branch map $m_{s,i}$ is injective. For this we show that for any $a, a' \in A$ if $a \neq a'$ then $m_{s,i}(a) \neq m_{s,i}(a')$ and we do this by choosing a structure $\mathcal{C}$ and an appropriate pair $g, h\colon \mathbb{P}_k\mathcal{C} \to \mathcal{A}$ of distinct strategies such that the inequality $m \circ g \neq m \circ h$ guaranteed by m being a monomorphisms allows us to conclude that $m_{s,i}(a) \neq m_{s,i}(a')$. We can do this as follows.

Firstly, choose $\mathcal{C}$ to be the structure $\mathcal{A}_0$ which has the exact same underlying set A as $\mathcal{A}$ but for every relation R in the signature of $\mathcal{A}$, let $R^{\mathcal{A}_0} = \emptyset$. This allows us to construct Duplicator strategies from $\mathcal{A}_0$ to $\mathcal{A}$ which have no restrictions other than those imposed by the I-relations and, in the case of the category $\mathcal{K}(\mathbb{P}_k^*)$, the $\approx_*$ relation. This choice also has the convenient side effect that, as sets, we have $\mathbb{P}_k A_0 = \mathbb{P}_k A$. Now we choose the strategies g and h as follows:

- For any $t \in \mathbb{P}_k A$ such that either $t \sqsubset s$ or t is incomparable with $s; (a, i)$, define both $g^*(t) = h^*(t) = t$.
- For $t = s; (a, i)$ define $g(t) = a$ and $h(t) = a'$.
- For any $t \sqsupset s'; (a, i)$, with $\omega(t) = (x, j)$ there are three cases. If there is a $t' \sqsubset t$ such that $(t', t) \in I^{\mathbb{P}_k \mathcal{A}}$ then let $g(t) = g(t')$ and $h(t) = h(t')$. Otherwise, if j appears as the last pebble of some earlier prefix of t then choose the maximal such prefix t' and let $g(t) = g(t')$ and $h(t) = h(t')$. Otherwise, let t' be the maximal prefix of t such that $\omega_{peb}(t') = i$ (we know that one exists as $t \sqsupset s; (a, i)$).

This definition ensures that for any $t \in \mathbb{P}_k \mathcal{A}_0$ the strategies g and h agree on every Spoiler history up to s or incomparable with $s; (a, i)$ and that they both collapse after $s; (a, i)$. Furthermore this collapse occurs in such a way that for any $s' \sqsupset s; (a, i)$ there is a common witness $t \sqsubset s; (a, i)$ for the collapse of g and h. This means that for any f, $f \circ g \neq f \circ h$ if and only if $f(g^*(s; (a, i))) \neq f(h^*(s; (a, i)))$ which, by definition of $g(s; (a, i))$ and $h(s; (a, i))$ is equivalent to saying $f_{s,i}(a) \neq f_{s,i}(a')$. Applying this to the strategy m, where $m \circ g \neq m \circ h$ is guaranteed by the property of being a monomorphism, we have the desired conclusion. □

This result concludes this chapter by providing both category-theoretic and logical characterisations of branch-injective maps. While we know from Proposition 4.6 that the logical characterisation extends to branch-surjective maps, a similar relationship to epimorphisms is not yet known and is left to future work.

In the next chapter we focus on the logical characterisation of the maps in $\mathcal{K}(\mathbb{P}_k)$ by exploring the links between the quantifiers described in this chapter and the notion of generalised quantifiers which appear throughout finite model theory.

Chapter 5

Kleisli maps and generalised quantifiers

We have seen that the comonad $\mathbb{P}_k$ captures the most expressive logics of all the existing game comonads. A central aim of this thesis is to expand the remit of these compositional methods to even more expressive logics. Generalised quantifiers, which date back to Lindström [73], provide a framework for constructing such logics. In this chapter, we see how these relates to the logics captured by $\mathbb{P}_k$.

To do this we generalise two important results in the finite model theory of generalised quantifiers. The first is a result of Kolaitis and Väänänen [68] which showed that $\mathcal{L}^k_\infty(\#)$ has the same expressive power as $\mathcal{L}^k_\infty$ expanded by all unary generalised quantifiers. We extend this to show that all of the fragments of logic captured by maps in $\mathcal{K}(\mathbb{P}_k)$ live double lives as logics expanded by unary generalised quantifiers of different kinds. The second result is Hella's game-theoretic characterisation of logics with higher arity generalised quantifiers [58]. We generalise this by providing a system of related games which bound the expressive power of logics with different subsets of higher arity generalised quantifiers. In proving these new results, we uncovered a subtle difference between the ways in which generalised quantifier are used by Hella, and Kolaitis and Väänänen which has been overlooked in the published literature. We conclude the chapter by correcting this difference in the unary case.

5.1 Generalised quantifiers

The idea behind a generalised quantifier is to add expressive power to some logic $\mathcal{L}$ by taking an isomorphism-closed class $\mathbf{K}$ of τ-structures and introducing a quantifier Q_K which can identify structures which belong to $\mathbf{K}$. For this, we need a way to construct τ-structures using formulas from $\mathcal{L}$ over an arbitrary signature σ. This can be done in many ways, but the one considered in the results which we generalise in this chapter is the following notion of an *interpretation*.

Definition 5.1. *For a logic $\mathcal{L}$ and any two (finite) relational signatures σ and $\tau = \{R_1, \ldots, R_m\}$ an $\mathcal{L}$-interpretation of τ in σ is a collection of $\mathcal{L}[\sigma]$ formulas*

$$I(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}, \mathbf{z}) = (\phi_R(\mathbf{x}_R, \mathbf{z}_R))_{R \in \tau}$$

where the tuple $\mathbf{x}_R$ has $ar(R)$ distinct elements and $\mathbf{z}$ is the tuple of parameters consisting of all unique variables appearing in $\mathbf{z}_R$ for some $R \in \tau$. For each R, $\mathbf{x}_R$ and $\mathbf{z}_R$ are disjoint from each other but note that the variables in $\mathbf{x}_R$ may be reused in $\mathbf{z}_{R'}$ for some other R'.

For any such interpretation, we have a mapping which sends any σ-structure $\mathcal{A}$ with an assignment $\mathbf{b}$ the variables in $\mathbf{z}$ to a τ-structure $I(\mathcal{A}, \mathbf{b})$ with the same underlying set A as $\mathcal{A}$ and with related tuples, for each $R \in \tau$, given by

$$R^{I(\mathcal{A},\mathbf{b})} = \{\mathbf{a} \in A^{ar(R)} \mid \mathcal{A}, \mathbf{a}, \mathbf{b} \models \phi_R(\mathbf{x}_R, \mathbf{z}_R)\}.$$

We now define the syntax and semantics of generalised quantifiers.

Definition 5.2. *Let $\mathcal{L}$ be a logic and $\mathbf{K}$ a class of τ-structures with $\tau = \{R_1, ..., R_m\}$. The extension $\mathcal{L}(Q_K)$ of $\mathcal{L}$ by the generalised quantifier for the class $\mathbf{K}$ is obtained by extending the syntax of $\mathcal{L}$ by the following formula formation rule:*

> *Let $I(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}, \mathbf{z}) = (\phi_{R_1}, ..., \phi_{R_m})$ be formulas in $\mathcal{L}(Q_K)$ that form an interpretation of τ in σ with parameters $\mathbf{z}$. Then $\psi(\mathbf{z}) = Q_K(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}).I(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}, \mathbf{z})$ is a formula in $\mathcal{L}(Q_K)$ over the signature σ. The semantics of the formula is given by $(\mathcal{A}, \mathbf{a}) \models Q_K(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}).I(\mathbf{x}_{R_1}, \ldots, \mathbf{x}_{R_m}, \mathbf{z})$, if, and only if, $\mathcal{B} := I(\mathcal{A}, \mathbf{a})$ is defined and $\mathcal{B}$ is in K.*

In this way, adding the generalised quantifier Q_K to the logic $\mathcal{L}$ is the most direct way to make the class $\mathbf{K}$ definable in $\mathcal{L}$. Formally, if $\mathcal{L}$ is a regular logic in the sense of [41], then its extension by Q_K is the minimal regular logic that can also define $\mathbf{K}$.

While these generalised quantifiers have a quite different syntax and semantics to the classical first-order quantifiers, $\exists$ and $\forall$, the following observation shows that $\exists$ and $\forall$ can be derived as generalised quantifiers.

Observation 5.3. *Let $\phi(x, \mathbf{z})$ be a formula in L. This determines an interpretation into τ_1 the signature with a single unary relation U. The classes $\mathbf{K}_\exists = \{\mathcal{A} \mid U^{\mathcal{A}} \neq \emptyset\}$ and $\mathbf{K}_\forall = \{\mathcal{A} \mid U^{\mathcal{A}} = A\}$ are isomorphism-closed classes of τ_1 structures. Now, the generalised quantifiers $Q_{K_\exists}$ and $Q_{K_\forall}$ have the same formula formation rules as $\exists$ and $\forall$ and the formulas $Q_{K_\exists} x\phi(x, \mathbf{z})$ and $Q_{K_\forall} x\phi(x, \mathbf{z})$ have the same semantics as $\exists x\phi(x, \mathbf{z})$ and $\forall x\phi(x, \mathbf{z})$.*

Throughout this chapter we consider three different ways of restricting the power of generalised quantifier logics. These are number of variables, *arity* of the quantifiers, and closure-properties of the underlying classes.

Throughout this chapter, we write $\mathcal{L}^k(Q_K)$ for the k variable fragment of a logic $\mathcal{L}(Q_K)$, meaning specifically those formulas of $\mathcal{L}(Q_K)$ which use at most k variable symbols in total. We note that this is subtly different from the logic that would be obtained by extending the logic $\mathcal{L}^k$ using the formula formation rule in Definition 5.2. For example, the rule above would allow the following formula in FO^2 extended by Q_K, despite using 3 variables in total.

$$\psi(y_1, y_2) = Q_K(x, x).(E(x, y_1), E(x, y_2)).$$

In addition to restricting variable count, we can also restrict the kinds of generalised quantifier we add to our logic. We define the *arity* of the quantifier Q_K to be the maximum arity of any relation in τ. Note that this is the number of variables bound by the quantifier. We write $\mathcal{L}^k_\infty(\mathbf{Q}_n)$ for the k-variable fragment of $\mathcal{L}_\infty$ extended with *all* quantifiers of arity n. This is only of interest when $n \leq k$. As we see in Section 5.2, Kolaitis and Väänänen [68] relate the logic $\mathcal{L}^k_\infty(\mathbf{Q}_1)$ to the k-variable counting logic $\mathcal{L}^k_\infty(\#)$. However, allowing quantifiers of higher arity gives logics of considerably more expressive power. In particular, if σ is a signature with all relations of arity at most n, then *any* property of σ-structures is expressible in $\mathcal{L}^n_\infty(\mathbf{Q}_n)$. Thus, all properties of graphs, for instance, are expressible in $\mathcal{L}^2_\infty(\mathbf{Q}_2)$. It is notable, however, that for any fixed positive integer n there are properties of finite structures (exhibited by Hella [58]) which are decidable in `PTIME` but not expressible in $\mathcal{L}^\omega_\infty(\mathbf{Q}_n)$.

In the case of $n = 1$, we note an important difference in the way the logic $\mathcal{L}^k_\infty(\mathbf{Q}_1)$ is defined in the two main sources for this chapter, namely Hella [58], and Kolaitis and Väänänen [68]. Hella's version is precisely the logic $\mathcal{L}^k_\infty(\mathbf{Q}_1)$ defined so far in this section. Kolaitis and Väänänen, however, place an extra implicit assumption on the types of unary interpretation $I(x_1, \ldots, x_m, \mathbf{z})$ allowed. In particular, they insist that none of the variables x_i appears in the tuple of parameters $\mathbf{z}$. We call this fragment KV $\mathcal{L}^k_\infty(\mathbf{Q}_1)$. In Section 5.4, we explore the difference between this logic and the full (Hella) $\mathcal{L}^k_\infty(\mathbf{Q}_1)$.

Finally, we are interested in studying the expressive power of various subfamilies of $\mathbf{Q}_n$. In particular, we say that a class $\mathbf{K}$ is homomorphism-closed (or injective, surjective or bijective homomorphism-closed) when for any $\mathcal{A} \in \mathbf{K}$, if $f : \mathcal{A} \to \mathcal{B}$ is a homomorphism (or injective, surjective or bijective homomorphism respectively) then $\mathcal{B} \in \mathbf{K}$. We write $\mathcal{L}^k_\infty(\mathbf{Q}^{\mathrm{h}}_n), \mathcal{L}^k_\infty(\mathbf{Q}^{\mathrm{i}}_n), \mathcal{L}^k_\infty(\mathbf{Q}^{\mathrm{s}}_n)$ and $\mathcal{L}^k_\infty(\mathbf{Q}^{\mathrm{b}}_n)$ for the respective logics extended by these corresponding families of n-ary quantifiers.

We finish this section with a brief comment about some easy-to-overlook nullary (arity 0) generalised quantifiers and what they say about the size of structures. We assume that these quantifiers are present in every $\mathcal{L}^k_\infty(\mathbf{Q}_n)$. For any relational signature σ let $\mathcal{R}(\sigma)^{=M}$ denote the collection of σ-structures whose universe has exactly M elements.

Let $\mathcal{R}(\sigma)^{\geq M} = \bigcup_{m \geq M} \mathcal{R}(\sigma)^{=m}$ and similarly $\mathcal{R}(\sigma)^{\leq M} = \bigcup_{m \leq M} \mathcal{R}(\sigma)^{=m}$. It is obvious that $\mathcal{R}(\sigma)^{=M}$ is bijection-closed, $\mathcal{R}(\sigma)^{\geq M}$ is injection-closed and $\mathcal{R}(\sigma)^{\leq M}$ is surjection-closed. When $\sigma = \emptyset$ is the empty signature this gives us classes of sets $\mathcal{K}^{=M}$, $\mathcal{K}^{\geq M}$ and $\mathcal{K}^{\leq M}$ which are closed under bijections, injections and surjections respectively. As any signature σ admits an empty interpretation into the empty signature which sends any σ-structure to its underlying set, we can create sentences B_m, I_m, and S_m by binding the nullary quantifier $\mathcal{Q}_K$, for $K = \mathcal{K}^{=M}$, $\mathcal{K}^{\geq M}$ and $\mathcal{K}^{\leq M}$ respectively, to this empty interpretation. As noted in the following observation these sentences are important for comparing the sizes of structures, in any signature.

Observation 5.4. *For all $n, k, m \in \mathbb{N}$ there are sentences B_m, I_m, and S_m in $+\mathcal{L}^k(\mathbf{Q}_n^b), +\mathcal{L}^k(\mathbf{Q}_n^i)$, and $+\mathcal{L}^k(\mathbf{Q}_n^s)$ respectively, such that*

$$\begin{aligned} \mathcal{A} \models B_m &\iff |A| = m \\ \mathcal{A} \models I_m &\iff |A| \geq m \\ \mathcal{A} \models S_m &\iff |A| \leq m \end{aligned}$$

As a direct result of this we have that

$$\begin{aligned} \mathcal{A} \Rrightarrow_{+\mathcal{L}^k(\mathbf{Q}_n^b)} \mathcal{B} &\implies |A| = |B| \\ \mathcal{A} \Rrightarrow_{+\mathcal{L}^k(\mathbf{Q}_n^i)} \mathcal{B} &\implies |A| \leq |B| \\ \mathcal{A} \Rrightarrow_{+\mathcal{L}^k(\mathbf{Q}_n^s)} \mathcal{B} &\implies |A| \geq |B|. \end{aligned}$$

5.2 Kolaitis and Väänänen's result in $\mathcal{K}(\mathbb{P}_k)$

Having defined the notion of adding generalised quantifiers to a logic, we would like to be able to compare the expressive power of these newly extended logics with logics we are familiar with. As observed in the last section, generalised quantifiers are in general a very flexible way to add expressive power to a logic. It is therefore very interesting from a descriptive complexity point of view to show that all generalised quantifiers of a certain arity or kind can be emulated by some simpler set of quantifiers. In this section we prove several new results of this type, relating logics with unary generalised quantifiers to the logics which were shown in Chapter 4 to arise naturally in the Kleisli category of $\mathbb{P}_k$

Kolaitis and Väänänen, proved one of the original results in this direction by showing that the expressive power of infinitary k-variable logic extended with all unary quantifiers is equivalent to that of the same logic extended by counting quantifiers. This is summarised in the following theorem where we recall the notion of equivalence of logic from Chapter 2.

Theorem 5.5 (Kolaitis & Väänänen, 1995 [68]). *For any positive integer k,*

$$KV\mathcal{L}^k_\infty(\mathbf{Q}_1) \equiv \mathcal{L}^k_\infty(\#)$$

$$
\begin{tikzcd}
 & \exists^+\mathcal{L}^k(\exists^{\geq n}, \forall^{\leq m}) \arrow[dl, dash] \arrow[dr, dash] & \\
\exists^+\mathcal{L}^k(\exists^{\geq n}) \arrow[dr, dash] & & \exists^+\mathcal{L}^k(\forall^{\leq m}) \arrow[dl, dash] \\
 & \exists^+\mathcal{L}^k &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \exists^+\mathcal{L}^k(\mathbf{Q}_1^{\mathrm{b}}) \arrow[dl, dash] \arrow[dr, dash] & \\
\exists^+\mathcal{L}^k(\mathbf{Q}_1^{\mathrm{i}}) \arrow[dr, dash] & & \exists^+\mathcal{L}^k(\mathbf{Q}_1^{\mathrm{s}}) \arrow[dl, dash] \\
 & \exists^+\mathcal{L}^k(\mathbf{Q}_1^{\mathrm{h}}) &
\end{tikzcd}
$$

Figure 5.1: Each Hasse diagram contains extensions of $\mathcal{L}_\infty$ of increasing expressive power. Theorem 5.7 establishes equivalence at each level of these diagrams.

This result is interesting from the perspective of game comonads because of the role that $\mathcal{L}^k_\infty(\#)$ plays in $\mathcal{K}(\mathbb{P}_k)$. Indeed, recalling the Isomorphism Power Theorem for $\mathbb{P}_k$ (Theorem 3.14), we have the following corollary of Theorem 5.5.

Corollary 5.6. *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following are equivalent for any positive integer k:*

- *There is a Kleisli isomorphism $\mathcal{A} \cong_{\mathcal{K}(\mathbb{P}_k)} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$*
- *$\mathcal{A} \equiv_{\mathcal{L}^k_\infty(\mathbf{Q}_1)} \mathcal{B}$.*

We now show that this correspondence between generalised quantifiers and Kleisli maps for $\mathbb{P}_k$ can be extended to cover all the types of map considered in Chapter 4. In particular, we prove the following theorem whose set of logical equivalences is summed up in Figure 5.1.

Theorem 5.7. *For every positive integer k, we have the following equivalences of logics:*

- *KV $\exists^+\mathcal{L}^k_\infty(\mathbf{Q}_1^h) \equiv \exists^+\mathcal{L}^k_\infty$,*
- *KV $\exists^+\mathcal{L}^k_\infty(\mathbf{Q}_1^i) \equiv \exists^+\mathcal{L}^k_\infty(\exists^{\geq n})$,*
- *KV $\exists^+\mathcal{L}^k_\infty(\mathbf{Q}_1^s) \equiv \exists^+\mathcal{L}^k_\infty(\forall^{\leq m})$, and*
- *KV $\exists^+\mathcal{L}^k_\infty(\mathbf{Q}_1^b) \equiv \exists^+\mathcal{L}^k_\infty(\exists^{\geq n}, \forall^{\leq m})$.*

Before proving this result we state the following important corollary which links these unary generalised quantifiers to branch-injective, branch-surjective and branch-injective maps in $\mathcal{K}(\mathbb{P}_k)$. It follows from the classification of the logics on the right-hand side of Figure 5.1 provided in Chapter 4.

Corollary 5.8. *For two I-structures $\mathcal{A}$ and $\mathcal{B}$, we have the following list of equivalences:*

1. $\mathcal{A} \Rrightarrow_{KV\ \exists^+\mathcal{L}^k_\infty(\mathbf{Q}^h_1)} \mathcal{B}$ *iff* $\mathcal{A} \rightarrow_k \mathcal{B}$,

2. $\mathcal{A} \Rrightarrow_{KV\ \exists^+\mathcal{L}^k_\infty(\mathbf{Q}^i_1)} \mathcal{B}$ *iff* $\mathcal{A} \rightarrow^i_k \mathcal{B}$,

3. $\mathcal{A} \Rrightarrow_{KV\ \exists^+\mathcal{L}^k_\infty(\mathbf{Q}^s_1)} \mathcal{B}$ *iff* $\mathcal{A} \rightarrow^s_k \mathcal{B}$, *and*

4. $\mathcal{A} \Rrightarrow_{KV\ \exists^+\mathcal{L}^k_\infty(\mathbf{Q}^b_1)} \mathcal{B}$ *iff* $\mathcal{A} \rightarrow^b_k \mathcal{B}$.

We also make the following observation about simplifying formulas in the logical fragment KV $\mathcal{L}^k_\infty(\mathbf{Q}_1)$ to a convenient form.

Observation 5.9. *Recall that in the fragment KV $\mathcal{L}^k_\infty(\mathbf{Q}_1)$, any formula of the form*

$$Q_K(x_1, \ldots, x_m).(\phi_1(x_1, \mathbf{z}_{R_1}), \ldots, \phi_m(x_m, \mathbf{z}_{R_m}))$$

has the property that x_i does not appear in $\mathbf{z}_{R_j}$ for any j. Thus we can replace this formula with the equivalent formula

$$Q_K(x, \ldots, x).(\phi_1(x, \mathbf{z}_{R_1}), \ldots, \phi_m(x, \mathbf{z}_{R_m})).$$

We write this as $Q_K x.\ I(x, \mathbf{z})$ where $I(x, \mathbf{z})$ is the interpretation $(\phi(x, \mathbf{z}_R))_{R\in\tau_m}$. Applying this rewriting recursively, we can write any formula of KV $\mathcal{L}^k_\infty(\mathbf{Q}_1)$ using these more simple unary quantifiers and we do this throughout this section.

We now provide the details of the proof of Theorem 5.7.

Proof of Theorem 5.7. To prove the right to left implication in each of the cases above it suffices to show that the quantifiers $\exists$, $\exists^{\geq n}$, and $\forall^{\leq m}$ can be written as unary generalised quantifiers which are in $\mathbf{Q}^{\mathrm{h}}_1, \mathbf{Q}^{\mathrm{i}}_1$ and $\mathbf{Q}^{\mathrm{s}}_1$, respectively. We saw in Observation 5.3 how to do this for $\exists$ by constructing a class $\mathbf{K}_\exists$. It is not difficult to see that $\mathbf{K}_\exists$ is closed under homomorphisms. For $\exists^{\geq n}$, and $\forall^{\leq m}$, we proceed in a similar fashion by defining classes $\mathbf{K}_{\geq n}$ and $\mathbf{K}_{\leq m}$ as

$$\mathbf{K}_{\geq n} = \{\mathcal{A} \in \mathcal{R}(\tau_1) \ \mid\ |R^{\mathcal{A}}| \geq n\}$$

and

$$\mathbf{K}_{\leq m} = \{\mathcal{A} \in \mathcal{R}(\tau_1) \ \mid\ |R^{\mathcal{A}}| \geq |A| - m\}.$$

It is easy to see that these classes are closed under injective and surjective homomorphisms respectively and that for any formula $\phi(x, \mathbf{y})$, the formula $\exists^{\geq n}x.\ \phi(x, \mathbf{y})$ is equivalent to $Q_{\mathbf{K}_{\geq n}}x.\ (\phi(x, \mathbf{y}))$ and the formula $\forall^{\leq n}x.\ \phi(x, \mathbf{y})$ is equivalent to $Q_{\mathbf{K}_{\geq n}}x.\ (\phi(x, \mathbf{y}))$.

For the other direction, we need to show that logics extended by the relevant set of counting quantifiers are sufficient to define any unary generalised quantifier with the corresponding closure properties. In particular, we need to show that for any formula $\xi(\mathbf{y})$ of the form $Q_K x.\ I(x, \mathbf{y})$ we can define an equivalent formula $\overline{\xi}(\mathbf{y})$ which replaces all unary generalised quantifiers with counting quantifiers of the correct type and uses no more variables than ξ. Our strategy to do this is to construct for each $\mathcal{A}$ in $\mathcal{R}(\tau_r)$ formulas $\phi^h_{I,\mathcal{A}}(\mathbf{y}), \phi^i_{I,\mathcal{A}}(\mathbf{y}), \phi^s_{I,\mathcal{A}}(\mathbf{y})$, and $\phi^b_{I,\mathcal{A}}(\mathbf{y})$ which are satisfied on a structure $\mathcal{B}$ with choice of parameters $\mathbf{c}$ if and only if there is a homomorphism $f\colon \mathcal{A} \to I(\mathcal{B}, \mathbf{c})$ which is injective, surjective or bijective respectively in the case of the latter three formulas. Given these formulas we construct $\overline{\xi}$ as the following large disjunction where x is h, i, s or b depending on the closure properties of $\mathbf{K}$

$$\overline{\xi}(\mathbf{y}) = \bigvee_{\mathcal{A} \in \mathbf{K}} \overline{\phi^x_{I,\mathcal{A}}}(\mathbf{y}). \tag{*}$$

The equivalence of $\overline{\xi}$ and ξ is proven as follows. For one direction, if $\mathcal{B}, \mathbf{c} \models \xi(\mathbf{y})$ then $I(\mathcal{B}, \mathbf{c}) \in \mathbf{K}$ and so $\phi^h_{I,I(\mathcal{B},\mathbf{c})}(\mathbf{y})$ appears in the disjunction in $\overline{\xi}$ and this is trivially modelled by $\mathcal{B}, \mathbf{c}$. In the other direction if $\mathcal{B}, \mathbf{c} \models \overline{\xi}(\mathbf{y})$ then there exists some $\mathcal{A} \in \mathbf{K}$ such that $\mathcal{B}, \mathbf{c} \models \phi^x_{I,\mathcal{A}}(\mathbf{y})$ and so there is a homomorphism of the appropriate kind from $\mathcal{A}$ to $I(\mathcal{B}, \mathbf{c})$. By the closure of $\mathbf{K}$ under the appropriate kind of homomorphisms, this gives us that $I(\mathcal{B}, \mathbf{c}) \in \mathbf{K}$ and so $\mathcal{B}, \mathbf{c} \models \xi(\mathbf{y})$.

We complete this proof by showing how to construct the formulas $\phi^h_{I,\mathcal{A}}(\mathbf{y}), \phi^i_{I,\mathcal{A}}(\mathbf{y}), \phi^s_{I,\mathcal{A}}(\mathbf{y})$, and $\phi^b_{I,\mathcal{A}}(\mathbf{y})$. This relies on the following claim about unary structures which makes up the main technical body of this proof

Claim 5.10. *For any unary signature τ_r and any τ_r-structure $\mathcal{A}$ there are 1-variable sentences $\phi^h_{\mathcal{A}}, \phi^i_{\mathcal{A}}, \phi^s_{\mathcal{A}}$, and $\phi^b_{\mathcal{A}}$ in the respective logics such that for any of these*

$$\mathcal{B} \models \phi \iff \exists f\colon \mathcal{A} \to \mathcal{B} \textit{ of the appropriate kind.}$$

Proof of claim. To construct these sentences we observe that when constructing homomorphisms between unary structures, the image at each point can be chosen independently of the image at all other points. Let $\mathcal{A}$ and $\mathcal{B}$ be two τ_r-structures. For any element $a \in A$ let the *unary type of a in $\mathcal{A}$* be defined as the set of all unary predicates satisfied by a, which we call $\mathbf{u\text{-}type}_{\mathcal{A}}(a)$. We say that $b \in B$ is a *candidate image* of $a \in A$ if $\mathbf{u\text{-}type}_{\mathcal{A}}(a) \subset \mathbf{u\text{-}type}_{\mathcal{B}}(b)$, i.e. that all unary predicates in τ_r which are satisfied by a, are also satisfied by b. We write $a \dashrightarrow b$ for this relation and for any element $a \in A$ write $\mathcal{N}_{\dashrightarrow}(a)$ for its neighbourhood under $\dashrightarrow$ and $\mathcal{N}^c_{\dashrightarrow}(a)$ for $B \setminus \mathcal{N}_{\dashrightarrow}(a)$. Write $\mathcal{N}_{\dashleftarrow}(b)$ and $\mathcal{N}^c_{\dashleftarrow}(b)$ for, respectively, the set $\{a \mid a \dashrightarrow b\}$ and its complement in A.

In this presentation, we have a relatively simple definition of a homomorphism between two τ_r-structures. Indeed a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ is simply a choice of $b \in \mathcal{N}_{\dashrightarrow}(a)$ for each $a \in A$. This means that we can define $\mathcal{N}_{\dashrightarrow}(a)$ by the single-variable formula $\psi_{a\dashrightarrow}(x) = \bigwedge_{U \in \mathbf{u\text{-}type}_{\mathcal{A}}(a)} U(x)$. And so the required sentence $\phi^h_{\mathcal{A}}$ is

$$\phi^h_{\mathcal{A}} := \bigwedge_{a \in A} \exists x.\ \psi_{a\dashrightarrow}(x).$$

For injective or surjective homomorphisms we consider matchings in the bipartite graph defined by $\dashrightarrow$ which are total on A and B respectively. These can be expressed in terms of Hall's conditions, as we saw in the proof of Proposition 4.6.

For matchings total on A, which correspond in this case with injective homomorphisms from $\mathcal{A}$, this condition is

$$\forall S \subset A.\ |S| \leq |\mathcal{N}_{\dashrightarrow}(S)|$$

which is easily converted into a corresponding formula in $\exists^+\mathcal{L}^k_\infty(\exists^{\geq n})$ as

$$\phi^i_{\mathcal{A}} = \bigwedge_{S \subset A} \exists^{\geq |S|} x.\ \bigvee_{a \in S} \psi_{a \dashrightarrow}(x).$$

For matchings which are total on B, the condition is

$$\forall S \subset B.\ |S| \leq |\mathcal{N}_{\dashleftarrow}(S)|. \tag{SM1}$$

To turn this into a sentence identifying surjective homomorphisms from a structure $\mathcal{A}$ we need to be careful of two things. Firstly, a matching which is total on B does not have to be a function from A to B. We overcome this by noting that if $\dashrightarrow \subset A \times B$ contains both a function and a matching which is total on B then it contains a surjective function. This means we can define $\phi^s_{\mathcal{A}}$ as the conjunction of $\phi^h_{\mathcal{A}}$ and a formula $\phi^{sm}_{\mathcal{A}}$ defining when $\dashrightarrow \subset A \times B$ has a matching which is total on B.

The second issue is in defining $\phi^{sm}_{\mathcal{A}}$. In particular, we need to construct a sentence that is parametrised entirely in terms of $\mathcal{A}$ but the condition (SM1) quantifies over subsets of B rather than subsets of A. To overcome this, we first show that (SM1) is equivalent to the condition

$$\forall S' \subset A.\ |\mathcal{N}^c_{\dashrightarrow}(S')| \leq |A \setminus S'|. \tag{SM2}$$

Once we have this equivalence, we can write $\phi^{sm}_{\mathcal{A}}$ as

$$\phi^{sm}_{\mathcal{A}} = \bigwedge_{S' \subset A} \forall^{\leq |S'|} x.\ \left[\bigvee_{a \in S'} \psi_{a \dashrightarrow}(x) \right]$$

and so $\phi^s_{\mathcal{A}} = \phi^h_{\mathcal{A}} \wedge \phi^{sm}_{\mathcal{A}}$ is the required sentence in $\exists^+\mathcal{L}^k_\infty(\forall^{\leq m})$.

Now we prove the equivalence of (SM1) and (SM2). Firstly, we show that (SM1) implies (SM2). Given any $S' \subset A$, let $S = \mathcal{N}^c_{\dashrightarrow}(S') \subset B$. Now (SM1) implies that $|S| \leq |S''|$ where $S'' = \mathcal{N}_{\dashleftarrow}(S) = \mathcal{N}_{\dashleftarrow}(\mathcal{N}^c_{\dashrightarrow}(S'))$. We know by the definition of $\dashrightarrow$ that $S' \cap \mathcal{N}_{\dashleftarrow}(\mathcal{N}^c_{\dashrightarrow}(S')) = \emptyset$ and so $S'' \subset A \setminus S'$ and thus we have $|S| \leq |S''| \leq |A \setminus S'|$ as required.

Now we show the other direction. Given $S \subset B$, let $S' = \mathcal{N}^c_{\dashleftarrow}(S)$. (SM2) gives that

$$|B \setminus S''| \leq |A \setminus S'|$$

where $S'' = \mathcal{N}_{\dashrightarrow}(S') \subset B$. Now as $S'' \cap S = \emptyset$ by definition and $A \setminus S' = A \setminus \mathcal{N}^c_{\dashleftarrow}(S) = \mathcal{N}_{\dashleftarrow}(S)$. So we have

$$|S| \leq |B \setminus S''| \leq |A \setminus S'| = |\mathcal{N}_{\dashleftarrow}(S)|$$

as required.

Finally, to define $\phi^b_{\mathcal{A}}$ we simply note that as we care only about finite structures it suffices to check if there are both injective and surjective homomorphisms and so $\phi^b_{\mathcal{A}} = \phi^i_{\mathcal{A}} \wedge \phi^s_{\mathcal{A}}$.

□

To complete the proof of the theorem, it remains to define the formulas $\phi^h_{I,\mathcal{A}}(\mathbf{y})$, $\phi^i_{I,\mathcal{A}}(\mathbf{y})$, $\phi^s_{I,\mathcal{A}}(\mathbf{y})$, and $\phi^b_{I,\mathcal{A}}(\mathbf{y})$. To do this we take the sentences $\phi^h_{\mathcal{A}}, \phi^i_{\mathcal{A}}, \phi^s_{\mathcal{A}}$, and $\phi^b_{\mathcal{A}}$ defined in the proof of the claim and replace each unary atom $R(x)$ with the appropriate $\phi_R(x, \mathbf{y}_R)$ from the interpretation I. By the proof of the claim above these new formulas are satisfied by some structure $\mathcal{B}$ and choice of parameters $\mathbf{b}$ if, and only if, there is a homomorphism $f\colon \mathcal{A} \to I(\mathcal{B}, \mathbf{b})$, of the appropriate kind. Finally, as the variable x does not appear in the tuples $\mathbf{y}_R$ (by the assumption that $I(x, \mathbf{y}))$ is permitted in KV $\mathcal{L}^k_\infty(\mathbf{Q}_1)$), this can be done in k variables without any relabelling. This means that the formula $\overline{\xi}$ defined in (*) contains the same number of variables as ξ and so we are done.

□

Theorem 5.7 and Corollary 5.8 establish a deep connection between unary generalised quantifiers and the Kleisli category of $\mathbb{P}_k$. It is natural now to ask if this correspondence between comonadic semantics and generalised quantifiers lifts to higher arities. In the next chapter, we see a complete answer to this question but first, in the next section we introduce a new system of pebble games which bridges the gap between the logical and comonadic worlds for higher arity generalised quantifiers.

5.3 Hella's games for generalised quantifiers

Corollary 5.8 in the last section showed that the comonad $\mathbb{P}_k$ effectively captures, in a fairly robust way, the unary level of the hierarchy of generalised quantifiers. This suggests an intriguing question of whether there is a similar comonadic semantics for higher arities. This is precisely the question which is answered in Chapter 6. An important prerequisite for this work is to find appropriate games which generalise the system of pebble games studied in Chapter 4 and capture the logical relations relevant to higher arity generalised quantifiers.

One such family of games is that of Hella's n-bijective k-pebble games, $\mathbf{Bij}^n_k$. These are model-comparison games which capture equivalence of relational structures over the logic $\mathcal{L}^k_\infty(\mathbf{Q}_n)$, i.e. k-variable infinitary logic where the allowed quantifiers are all generalised quantifiers with arity $\leq n$. This game generalises a variant of the bijection game $\mathbf{Bij}^k$ which captures equivalence over $\mathcal{L}^k_\infty(\#)$ which we saw in the last section is equivalent to Kolaitis and Väänänen's unary fragment $KV\mathcal{L}^k(\mathbf{Q}_1)$.

In this section, we introduce a family of games which relax the rules of $\mathbf{Bij}^n_k$ and prove a generalisation of Hella's result by showing the correspondence of these games to different fragments of $\mathcal{L}^k_\infty(\mathbf{Q}_n)$, with the new games and their corresponding logics being summarised in Figure 5.2. Throughout this section, we will state our main results in terms of a quantifier-free version of $\mathcal{L}^k_\infty$ which restricts negation to the atomic formulas. This allows for a clearer distinction between the different classes of generalised quantifiers to be studied and is defined as follows.

Definition 5.11. *For any signature σ, we denote by $+\mathcal{L}^k[\sigma]$, the class of positive infinitary k-variable quantifier-free formulas over σ. That means the k variable fragment of the class of formulas*

$$+\mathcal{L}[\sigma] ::= R(x_1, \dots x_m) \mid \bigwedge_{\mathcal{I}} \phi \mid \bigvee_{\mathcal{J}} \psi$$

for any $R \in \sigma$. We use $\mathcal{L}^k[\sigma]$ to denote a similar class of formulas but with negation permitted on atoms.

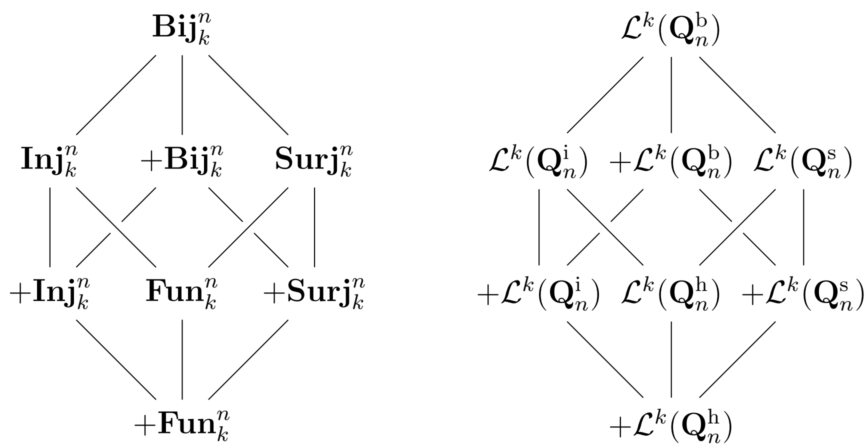


Figure 5.2: Hasse diagrams of new games (ordered by difficulty for Duplicator) and corresponding logics (ordered by expressive power).

5.3.1 Relaxing $\mathbf{Bij}^n_k$

Recall from Chapter 2 that each round of $\mathbf{Bij}^n_k(\mathcal{A}, \mathcal{B})$ involves Duplicator selecting a bijection $f : A \to B$ and ends with a test of whether for the pebbled positions $(a_i, b_i)_{i \in [k]}$ it is the case that for any $\{i_1, \dots i_r\} \subset [k]$

$$(a_{i_1}, \dots a_{i_r}) \in R^{\mathcal{A}} \iff (b_{i_1}, \dots b_{i_r}) \in R^{\mathcal{B}}$$

where Duplicator loses if the test is failed. For the rest of the round, Spoiler rearranges up to n pebbles on $\mathcal{A}$ with the corresponding pebbles on $\mathcal{B}$ moved according to f.

A crucial difference between this game and the bijective k-pebble game $\mathbf{Bij}^k$ is the order in which Spoiler and Duplicator make their moves. This difference results in a subtle difference in the logics captured by each game as is explored more comprehensively in Section 5.4.

To create from $\mathbf{Bij}^n_k$ a "one-way" game from $\mathcal{A}$ to $\mathcal{B}$ we need to relax the condition that the f provided by Duplicator needs to be a bijection and the $\iff$ in the final test. We do the by taking inspiration from the relaxation of $\mathbf{Bij}^k$ to $+\mathbf{Fun}^k$ described in Chapter 4. As in the case of the bijection games, the main difference between the following definition and that in Definition 4.4 is the difference in the order of Spoiler and Duplicator moves.

Definition 5.12. *For two relational structures $\mathcal{A}$, $\mathcal{B}$, the positive k-pebble n-function game, $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ is played by Spoiler and Duplicator. Prior to the jth round the position consists of partial maps $\pi^a_{j-1} : [k] \rightharpoonup A$ and $\pi^b_{j-1} : [k] \rightharpoonup B$. In Round j*

- *Duplicator provides a function $h_j : A \to B$ such that for each $i \in [k]$, $h_j(\pi^a_{j-1}(i)) = \pi^b_{j-1}(i)$.*
- *Spoiler picks up to n distinct* pebbles, *i.e. elements $p_1, \dots p_m \in [k] (m \leq n)$ and m elements $x_1, \dots x_m \in A$.*
- *The updated position is given by $\pi^a_j(p_l) = x_l$ and $\pi^b_j(p_l) = h_j(x_l)$ for $l \in [m]$; and $\pi^a_j(i) = \pi^a_{j-1}(i)$ and $\pi^b_j(i) = \pi^b_{j-1}(i)$ for $i \notin \{p_1, \dots, p_m\}$.*
- *Spoiler has won the game if there is some $R \in \sigma$ and $(i_1, \dots i_r) \in [k]^r$ such that $(\pi^a_j(i_1), \dots, \pi^a_j(i_r)) \in R^{\mathcal{A}}$ but $(\pi^b_j(i_1), \dots, \pi^b_j(i_r)) \notin R^{\mathcal{B}}$.*

Duplicator wins by preventing Spoiler from winning.

We saw in Chapter 4 that the positive function game $+\mathbf{Fun}^k$ is equivalent to the existential k-pebble game for which we motivated the pebbling comonad construction in Chapter 3. This makes $+\mathbf{Fun}^n_k$ a good candidate for the one-way game corresponding to $\mathbf{Bij}^n_k$. The similarity with the games in Chapter 4 also motivates the following definitions of positive n-injective, n-surjective and n-bijective games.

Definition 5.13. *For two relational structures $\mathcal{A}$, $\mathcal{B}$, the positive k-pebble n-injection (resp. surjection, bijection) game, $+\mathbf{Inj}^n_k(\mathcal{A}, \mathcal{B})$ (resp. $+\mathbf{Surj}^n_k(\mathcal{A}, \mathcal{B})$, $+\mathbf{Bij}^n_k(\mathcal{A}, \mathcal{B})$) is played by Spoiler and Duplicator. Prior to the jth round the position consists of partial maps $\pi^a_{j-1} : [k] \rightharpoonup A$ and $\pi^b_{j-1} : [k] \rightharpoonup B$. In Round j*

- *Duplicator provides an injection (resp. a surjection, bijection) $h_j : A \to B$ such that for each $i \in [k]$, $h_j(\pi^a_{j-1}(i)) = \pi^b_{j-1}(i)$.*

- *Spoiler picks up to n distinct* pebbles, *i.e. elements $p_1, \dots p_m \in [k] (m \leq n)$ and m elements $x_1, \dots x_m \in A$.*
- *The updated position is given by $\pi_j^a(p_l) = x_l$ and $\pi_j^b(p_l) = h_j(x_l)$ for $l \in [m]$; and $\pi_j^a(i) = \pi_{j-1}^a(i)$ and $\pi_j^b(i) = \pi_{j-1}^b(i)$ for $i \notin \{p_1, \dots, p_m\}$.*
- *Spoiler has won the game if there is some $R \in \sigma$ and $(i_1, \dots i_r) \in [k]^r$ such that $(\pi_j^a(i_1), \dots, \pi_j^a(i_r)) \in R^{\mathcal{A}}$ but $(\pi_j^b(i_1), \dots, \pi_j^b(i_r)) \notin R^{\mathcal{B}}$.*

Duplicator wins by preventing Spoiler from winning.

Strengthening the test condition in each round so that Spoiler wins if there is some $R \in \sigma$ and $(i_1, \dots i_r) \in [k]^r$ such that $(\pi_j^a(i_1), \dots, \pi_j^a(i_r)) \in R^{\mathcal{A}}$ if, and only if, $(\pi_j^b(i_1), \dots, \pi_j^b(i_r)) \notin R^{\mathcal{B}}$, we get the definitions for the games $\mathbf{Fun}_k^n$, $\mathbf{Inj}_k^n$, $\mathbf{Surj}_k^n$ and $\mathbf{Bij}_k^n$ where the latter is precisely the n-bijective k-pebble game of Hella.

We now show that these games each correspond to a logic extended by n-ary quantifiers, generalising a result of Hella and extending the connection between pebble games and generalised quantifiers which we saw in the last section.

5.3.2 Generalising Hella's Theorem

Having introduced games which relax Hella's n-bijective k-pebble game, we now show that these games capture the expressive power of interesting fragments of $\mathcal{L}_{\infty\omega}(\mathbf{Q}_n)$, the logic originally studied in Hella's paper [58]. In this paper, Hella proved the following theorem relating Duplicator winning strategies in $\mathbf{Bij}_k^n$ and logical equivalence of structures over $\mathcal{L}_{\infty}^k(\mathbf{Q}_n)$.

Theorem 5.14 (Hella's Theorem)**.** *For all $n, k \in \mathbb{N}$ the following are equivalent:*

- *Duplicator has a winning strategy for $\mathbf{Bij}_k^n(\mathcal{A}, \mathcal{B})$*
- $\mathcal{A} \equiv_{\mathcal{L}_{\infty}^k(\mathbf{Q}_n)} \mathcal{B}$
- $\mathcal{A} \equiv_{\mathbf{FO}^k(\mathbf{Q}_n)} \mathcal{B}$

The goal of this section is to prove a generalisation of this result which relates the games and logics in Figure 5.2. While the generalised quantification here is in the style of Hella rather than Kolaitis and Väänänen, this structure reinforces the notion that these games are in a sense the "correct" generalisation of the games related to $\mathbb{P}_k$, which were given their own characterisation in terms of generalised quantifier logics in the last section.

In order to present the proof of this in a uniform fashion, we label the corners of these cubes by three parameters $x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}} \in \{0, 1\}$ as indicated in Figure 5.3. These parameters signal the presence or absence of certain rules in the corresponding games. x_{i} and x_{s} indicate if the function provided by Duplicator in each round is required to be injective or surjective respectively. x_{n} indicates if Spoiler wins when negated atoms are not preserved by the partial map defined at the end of a round.

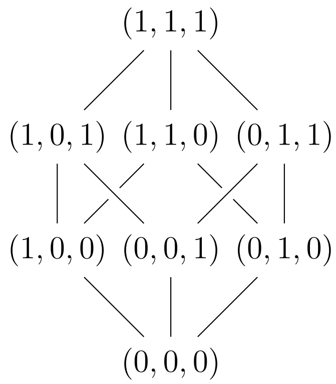


Figure 5.3: Cube of parameters

Now we define the aliases of each of the games which modify $\mathbf{Fun}^n_k$ as follows, with the games defined lining up with the games defined in Chapter 4.

Definition 5.15. *For two σ-structures $\mathcal{A}$ and $\mathcal{B}$, the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ is played by Spoiler and Duplicator in the same fashion as the game $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ with the following additional rules:*

1. *When Duplicator provides a function $f : A \to B$ at the beginning of a round, f is required to be*
   - *injective if $x_{\mathsf{i}} = 1$ and*
   - *surjective if $x_{\mathsf{s}} = 1$.*
2. *If $x_{\mathsf{n}} = 1$, Spoiler wins at move j if the partial map taking $\pi^a_j(i)$ to $\pi^b_j(i)$ fails to preserve negated atoms as well as atoms.*

Similarly, we define parameterised aliases for the related logics. To lighten our notational burden, we use $\mathcal{H}^{n,k}$ to denote the logic $+\mathcal{L}^k(\mathbf{Q}^{\mathrm{h}}_n)$ throughout this section.

Definition 5.16. *For any $\mathbf{x} = (x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$, we define $\mathcal{H}^{n,k}_{\mathbf{x}}$ to be the logic $\mathcal{H}^{n,k}$ extended by*

1. *all n-ary generalised quantifiers closed by all homomorphisms which are:*
   - *injective, if $x_{\mathsf{i}} = 1$; and*
   - *surjective, if $x_{\mathsf{s}} = 1$*

2. if $x_{\mathsf{n}} = 1$, negation on atoms.

For example, $\mathcal{H}^{n,k}_{001}$ extends $\mathcal{H}^{n,k}$ with negation on atoms but contains no additional quantifiers as all n-ary quantifiers closed under homomorphisms are already in $\mathcal{H}^{n,k}$. On the other hand, $\mathcal{H}^{n,k}_{110}$ does not allow negation on atoms but allows all quantifiers that are closed under bijective homomorphisms.

We can now state the main theorem of this section as follows.

Theorem 5.17 (Generalised Hella's Theorem)**.** *For $\mathbf{x} \in \{0,1\}^3$ and all $n, k \in \mathbb{N}$ the following are equivalent:*

- *Duplicator has a winning strategy for $\mathbf{x}$-$\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$*
- $\mathcal{A} \Rrightarrow_{\mathcal{H}^{n,k}_{\mathbf{x}}} \mathcal{B}$
- $\mathcal{A} \Rrightarrow_{\exists^+\mathbf{FO}^{n,k}_{\mathbf{x}}} \mathcal{B}$

To truly claim that this theorem as stated is a generalisation of Theorem 5.14 we need to show that the case of $\mathbf{x} = (1,1,1)$ above yields exactly Hella's result. To do this we show that the relations $\equiv_{\mathcal{L}^k_\infty(\mathbf{Q}_n)}$ and $\Rrightarrow_{\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)}$ are the same. As $\mathcal{L}^k_\infty(\mathbf{Q}_n)$ contains arbitrary negations it is easy to see that $\equiv_{\mathcal{L}^k_\infty(\mathbf{Q}_n)}$ is the same as the $\Rrightarrow_{\mathcal{L}^k_\infty(\mathbf{Q}_n)}$ relation. So all that remains to show is that $\mathcal{L}^k_\infty(\mathbf{Q}_n)$ is equivalent to $\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)$, as is done in the following proposition.

Proposition 5.18. *For all $n, k \in \mathbb{N}$, $\mathcal{L}^k(\mathbf{Q}^b_n) \equiv \mathcal{L}^k_\infty(\mathbf{Q}_n)$.*

Proof. To prove this equivalence we need to overcome two differences between these logics. Firstly, the class $\mathbf{Q}^{\mathrm{b}}_n$ of bijective-homomorphism-closed n-ary quantifiers is a proper subclass of $\mathbf{Q}_n$ of all isomorphism-closed n-ary quantifiers. The following observation provides a way of replacing general isomorphism-closed classes with bijective-homomorphism-closed ones by modifying the signature.

Observation 5.19. *For $\mathbf{K}$ an isomorphism-closed class of τ-structures, if $\tau' = \tau \cup \{\overline{R} \mid R \in \tau\}$ then*

$$\mathbf{K}' = \{\mathcal{A} \in \mathcal{R}(\tau') \mid \langle A, (R^{\mathcal{A}})_{R\in\tau}\rangle \in \mathbf{K} \text{ and } \forall R \in \tau,\ \overline{R}^{\mathcal{A}} = A^{ar(R)} \setminus R^{\mathcal{A}}\}$$

is a bijective-homomorphism closed class of τ' structures.

An important consequence of this is that for any such K, the formula

$$\phi(\mathbf{y}) = \mathcal{Q}_K(\mathbf{x}_R)_{R\in\tau}.\ (\psi_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau}$$

is equivalent to the formula

$$\phi'(\mathbf{y}) = \mathcal{Q}_{K'}(\mathbf{x}_R)_{R\in\tau'}.\ (\psi'_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau'},$$

where for any $R \in \tau$ $\psi'_R = \psi_R$ and $\psi'_{\overline{R}} = \neg\psi_R$.

The second difference between these two logics is the role of negation. As defined in this section, $\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)$ only allows negation on atoms, whereas $\mathcal{L}^k_\infty(\mathbf{Q}_n)$ allows negation throughout formulas. The following observation is important for dealing with this difference.

Observation 5.20. *A class of τ-structures K is isomorphism-closed if, and only if, its complement K^c is.*

This implies that the formula $\phi(\mathbf{y}) = \neg\mathcal{Q}_K(\mathbf{x}_R)_{R\in\tau}.\ (\psi_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau}$ is equivalent to $\phi'(\mathbf{y}) = \mathcal{Q}_{K^c}(\mathbf{x}_R)_{R\in\tau}.\ (\psi_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau}$.

Clearly $\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)$ is contained in $\mathcal{L}^k_\infty(\mathbf{Q}_n)$, so we focus on translating a formula $\phi(\mathbf{y}) \in \mathcal{L}^k_\infty(\mathbf{Q}_n)$ to an equivalent $\tilde{\phi}(\mathbf{y})$ in $\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)$. This can be done by induction on the quantifier depth of ϕ. For quantifier depth 0, there are no quantifiers to be replaced and any negation is either on atoms or can be assumed to be on atoms by appropriately distributing over conjunction or disjunction.

Now we assume ϕ has quantifier depth q. Without loss of generality, we can assume that ϕ is of the form $\mathcal{Q}_K(\mathbf{x}_R)_{R\in\tau}.\ (\psi_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau}$ for some isomorphism-closed class K of τ-structures. Indeed, if ϕ contains a leading negation we can use Observation 5.20 to remove the negation by replacing K with K^c. Note that the formulas ψ_R and $\neg\psi_R$ have quantifier depth strictly less than q and so by induction they have equivalents $\tilde{\psi}_R$ and $\neg\tilde{\psi}_R$ in $\mathcal{L}^k(\mathbf{Q}^{\mathrm{b}}_n)$. Now, using the consequence of Observation 5.19 mentioned above, we can define $\tilde{\phi}$ as $\mathcal{Q}_{K'}(\mathbf{x}_R)_{R\in\tau'}.\ (\tilde{\psi}'_R(\mathbf{x}_R, \mathbf{y}_R))_{R\in\tau'}$ $\square$

5.3.3 Proof of Theorem 5.17

Now to prove the desired correspondence between $\mathbf{x}$-$\mathbf{Fun}^n_k$ and $\mathcal{H}^{n,k}_{\mathbf{x}}$, we adapt a proof from Hella [58] to work for this parameterised set of games.

For this we need the language of *forth* systems which Hella uses as an explicit representation of a Duplicator winning strategy[1]. We provide the appropriate generalised definition here:

Definition 5.21. *Let $\boldsymbol{Part}^k_{x_{\mathsf{n}}}(\mathcal{A}, \mathcal{B})$ be the set of all partial functions $A \rightharpoonup B$ which preserve atoms (i.e. are partial homomorphisms) and, if $x_{\mathsf{n}} = 1$ additionally preserve negated atoms.*

A set $\mathcal{S} \subset \boldsymbol{Part}^k_{x_{\mathsf{n}}}(\mathcal{A}, \mathcal{B})$ is a forth system for the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ if it satisfies the following properties:

- ***Downwards closure**: If $f \in \mathcal{S}$ then $g \in \mathcal{S}$ for any $g \subset f$*

[1] These are called "k-variable n-bijective back-and-forth sets" in Hella's paper, where the "back" condition is implicit in the use of bijections. We drop that in the present generalisation.

- *($x_{\mathsf{i}}, x_{\mathsf{s}}$)-**forth property** For any f in $\mathcal{S}$ s.t. $|f| \leq k$, there exists a function $\phi_f : A \to B$, which is injective if $x_{\mathsf{i}} = 1$ and surjective if $x_{\mathsf{s}} = 1$ s.t. for every $C \subset dom(f), D \subset A$ with $|D| \leq n$ and $|C \cup D| \leq k$ we have $(f \downharpoonleft C) \cup (\phi_f \downharpoonleft D) \in \mathcal{S}$.*

As this definition is essentially an unravelling of a Duplicator winning strategy for the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ we can prove the following useful lemma.

Lemma 5.22. *There is a non-empty forth system $\mathcal{S}$ for the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ if, and only if, Duplicator has a winning strategy for the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$*

Proof. For the forward direction we note that if the pebbled position at the beginning of some round of $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ describes a partial homomorphism $f \in \mathcal{S}$ then the forth condition on $\mathcal{S}$ guarantees that if Duplicator plays $\phi_f : A \to B$ in this round then, whatever move Spoiler chooses in response, the pebbled position at the end of the round is some $f' \in \mathcal{S}$. As $\mathcal{S} \subset \mathbf{Part}_{x_{\mathsf{n}}}^k(\mathcal{A}, \mathcal{B})$ we know that such a move does not result in Duplicator losing the game. So if $\emptyset \in \mathcal{S}$, Duplicator can use $\mathcal{S}$ to play indefinitely without losing.
For the other direction, we note that the set of possible positions when playing the game $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ according to some winning Duplicator strategy Φ forms a forth system $\mathcal{S}_\Phi$. $\square$

Following Hella, we define the *canonical* forth system for a game as follows:

Definition 5.23. *The canonical forth system for $(x_{\mathsf{i}}, x_{\mathsf{s}}, x_{\mathsf{n}})$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ is denoted $I_{\mathbf{x}}^{n,k}(\mathcal{A}, \mathcal{B})$ and is given by the intersection $\bigcap_m I_{\mathbf{x}}^{n,k,m}(\mathcal{A}, \mathcal{B})$, whose conjuncts are defined inductively as follows:*

1. *$I_{\mathbf{x}}^{n,k,0}(\mathcal{A}, \mathcal{B}) := \boldsymbol{Part}_{x_{\mathsf{n}}}^k(\mathcal{A}, \mathcal{B})$.*
2. *$I_{\mathbf{x}}^{n,k,m+1}(\mathcal{A}, \mathcal{B})$ is the set of $\rho \in I_{\mathbf{x}}^{n,k,m}(\mathcal{A}, \mathcal{B})$ such that ρ satisfies the $(x_{\mathsf{i}}, x_{\mathsf{s}})$-forth condition with respect to the set $I_{\mathbf{x}}^{n,k,m}(\mathcal{A}, \mathcal{B})$*

It is not difficult to see that for any forth system $\mathcal{S}$ for $\mathbf{x}$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ we have $\mathcal{S} \subset I_{\mathbf{x}}^{n,k}(\mathcal{A}, \mathcal{B})$. This means that there is a winning strategy for Duplicator in the game $\mathbf{x}$-$\mathbf{Fun}_k^n(\mathcal{A}, \mathcal{B})$ if, and only if, $I_{\mathbf{x}}^{n,k}(\mathcal{A}, \mathcal{B})$ is not empty.

To complete the vocabulary needed to emulate Hella's proof in this setting we introduce the following generalisations of Hella's definitions.

Definition 5.24. *For any $\rho \in \boldsymbol{Part}_{x_{\mathsf{n}}}^k(\mathcal{A}, \mathcal{B})$ and $\phi(\mathbf{y})$ a formula in some logic, we say that ρ* preserves *$\phi(\mathbf{y})$ if for any $\mathbf{a} \subset dom(\rho)$ of the same length as $\mathbf{y}$ we have that $\mathcal{A}, \mathbf{a} \models \phi(\mathbf{y}) \implies \mathcal{B}, \rho(\mathbf{a}) \models \phi(\mathbf{y})$.*

Denote by $J_{\mathbf{x}}^{n,k}(\mathcal{A}, \mathcal{B})$ the set of all $\rho \in \boldsymbol{Part}_{x_{\mathsf{n}}}^k(\mathcal{A}, \mathcal{B})$ which preserve all $\mathcal{H}_{\mathbf{x}}^{n,k}$ formulas. Let $\exists^+\mathbf{FO}_{\mathbf{x}}^{n,k}$ denote the fragment of $\mathcal{H}_{\mathbf{x}}^{n,k}$ with only finitary conjunctions and disjunctions.

Denote by $K^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$ the set of all $\rho \in \boldsymbol{Part}^k_{x_{\mathsf{n}}}(\mathcal{A},\mathcal{B})$ which preserve all $\exists^+\mathbf{FO}^{n,k}_{\mathbf{x}}$ formulas.

Now, we directly modify Hella's argument to prove the following:

Lemma 5.25. *For $\mathcal{A},\mathcal{B}$ finite relational structures and all choices of n, k and $\mathbf{x}$,*

$$I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) = J^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) = K^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$$

Proof. We prove the result by showing that

$$I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset J^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset K^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$$

The inclusion $J^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset K^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$ is obvious so we focus on proving

1. $I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset J^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$; and
2. $K^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B}) \subset I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$

Proof of 1. Given $\rho \in I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$ we prove by structural induction on $\phi \in \mathcal{H}^{n,k}_{\mathbf{x}}$ that p preserves ϕ. Clearly as ρ is a partial homomorphism, it preserves atoms and, if $x_{\mathsf{n}} = 1$, negated atoms. The inductive cases for $\vee$ and $\wedge$ are easy so we focus on the cases ($\mathbf{y} \subset \mathrm{dom}(\rho)$) where

$$\phi(\mathbf{y}) = Q_K(\mathbf{z}_1, \ldots \mathbf{z}_m).(\psi_1(\mathbf{y}_1, \mathbf{z}_1), \ldots \psi_m(\mathbf{y}_m, \mathbf{z}_m))$$

Now $\rho \in I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$ implies the existence of a map $f : A \to B$ such that for all $C \subset \mathrm{dom}(\rho), D \subset A$ with $|D| \leq n$ we have $(\rho \downharpoonright C) \cup (f \downharpoonright D) \in I^{n,k}_{\mathbf{x}}(\mathcal{A},\mathcal{B})$, so using the induction hypothesis we have that for all i,

$$\mathcal{A}, \mathbf{a}_i, \mathbf{b}_i \models \psi_i(\mathbf{y}_i, \mathbf{z}_i) \implies \mathcal{B}, \rho\mathbf{a}_i, f\mathbf{b}_i \models \psi_i(\mathbf{y}_i, \mathbf{z}_i)$$

This means that f is a homomorphism

$$f : \langle A, \psi_1(\mathbf{a}_1, \cdot), \ldots \psi_m(\mathbf{a}_m, \cdot)\rangle \to \langle B, \psi_1(\rho\mathbf{a}_1, \cdot), \ldots \psi_m(\rho\mathbf{a}_m, \cdot)\rangle$$

Furthermore, in the cases where $(x_{\mathsf{i}}, x_{\mathsf{s}}) = (1,0), (0,1)$ or $(1,1)$ this homomorphism is injective, surjective and bijective respectively and the class $\mathbf{K}$ is closed under injective-homomorphism, surjective-homomorphism or bijective-homomorphism so in all of these cases

$$\langle A, \psi_1(\mathbf{a}_1, \cdot), \ldots \psi_m(\mathbf{a}_m, \cdot)\rangle \in \mathbf{K} \implies \langle B, \psi_1(\rho\mathbf{a}_1, \cdot), \ldots \psi_m(\rho\mathbf{a}_m, \cdot)\rangle \in \mathbf{K}$$

and so $\mathcal{A}, \mathbf{a} \models \phi(\mathbf{y}) \implies \mathcal{B}, \rho\mathbf{a} \models \phi(\mathbf{y})$ and we are done with the proof of 1.

Proof of 2. Suppose that we have $p \in K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$. We have that $p \in I^{n,k,0}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ by definition, so we prove by induction that $p \in I^{n,k,m}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$, for all m and any $p \in K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$. Indeed, suppose this is true for $m' < m$ but that $p \notin I^{n,k,m}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ for some $p \in K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$. Then it must be the case that for every $f : A \to B$ (injective if $x_{\mathsf{i}} = 1$, surjective if $x_{\mathsf{s}} = 1$) there is some choice of tuples $\mathbf{a}_f$ from $\mathrm{dom}(p)$ and $\mathbf{b}_f$ from A with $|\mathbf{b}_f| \leq n$ and $|\mathbf{a}_f \cup \mathbf{b}_f| \leq k$ such that $(p \downharpoonright \mathbf{a}_f) \cup (f \downharpoonright \mathbf{b}_f) \notin I^{n,k,m-1}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$. By induction, this means that $(p \downharpoonright \mathbf{a}_f) \cup (f \downharpoonright \mathbf{b}_f) \notin K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ and so there is a formula $\psi_f(\mathbf{y}, \mathbf{z})$ such that $\mathcal{A}, \mathbf{a}_f, \mathbf{b}_f \models \psi_f(\mathbf{y}, \mathbf{z})$ but $\mathcal{B}, p\mathbf{a}_f, f\mathbf{b}_f \not\models \psi_f(\mathbf{y}, \mathbf{z})$.

Let $F_{\mathbf{x}}$ denote the set of functions $f : A \to B$ which are injective if $x_{\mathsf{i}} = 1$ and surjective if $x_{\mathsf{s}} = 1$. Recall from Observation 5.4, the existence of p implies that $F_{\mathbf{x}}$ is non-empty. Now we define two structures $\mathcal{A}_p = \langle A, (\psi_f(\mathbf{a}_f, \cdot))_{F_{\mathbf{x}}} \rangle$ and $\mathcal{B}_p = \langle B, (\psi_f(p\mathbf{a}_f, \cdot))_{F_{\mathbf{x}}} \rangle$. We have by construction that no $f \in F_{\mathbf{x}}$ is a homomorphism from $\mathcal{A}_p \to \mathcal{B}_p$, meaning that we can define a class $\mathbf{K}$ with $\mathcal{A}_p \in \mathbf{K}$ and $\mathcal{B}_p \notin \mathbf{K}$ which is closed under:

- all homomorphisms, if $(x_{\mathsf{i}}, x_{\mathsf{s}}) = (0, 0)$
- all injective homomorphisms, if $(x_{\mathsf{i}}, x_{\mathsf{s}}) = (1, 0)$
- all surjective homomorphisms, if $(x_{\mathsf{i}}, x_{\mathsf{s}}) = (0, 1)$
- all bijective homomorphisms, if $(x_{\mathsf{i}}, x_{\mathsf{s}}) = (1, 1)$

So in all cases, the quantifier Q_K is allowed in $\mathcal{H}^{n,k}_{\mathbf{x}}$ so

$$\phi(\mathbf{y}) = Q_K(\mathbf{z}_f)_{f \in F_{\mathbf{x}}}.(\psi_f(\mathbf{y}_f, \mathbf{z}_f))_{f \in F_{\mathbf{x}}}$$

is in $\exists^+\mathbf{FO}^{n,k}_{\mathbf{x}}$ and is true on $(\mathcal{A}_p, \mathbf{a})$ but false on $(\mathcal{B}_p, p\mathbf{a})$. However, this contradicts that $p \in K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ and so preserves the truth of all such formulas.

□

We conclude this section by showing how to put together the results of this section to get the desired correspondence for the whole family of games and logics, stated in Theorem 5.17.

Proof of Theorem 5.17. First note that by the definition of the canonical forth system, Duplicator wins $\mathbf{x}\text{-}\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ if, and only if, $\emptyset \in I^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$. Furthermore, $J^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ and $K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ are defined as the sets of partial maps ρ which preserve any $\mathcal{H}^{n,k}_{\mathbf{x}}$ or $\exists^+\mathbf{FO}^{n,k}_{\mathbf{x}}$ formulas respectively which hold on the domain of ρ. So $\emptyset \in J^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ or $K^{n,k}_{\mathbf{x}}(\mathcal{A}, \mathcal{B})$ if, and only if, all *sentences* in these logics which are true $\mathcal{A}$ are also true in $\mathcal{B}$, i.e. $\mathcal{A} \Rrightarrow_{\mathcal{H}^{n,k}_{\mathbf{x}}} \mathcal{B}$ or $\mathcal{A} \Rrightarrow_{\exists^+\mathbf{FO}^{n,k}_{\mathbf{x}}} \mathcal{B}$. Applying the result of Lemma 5.25 proves the equivalence of these three. □

5.4 Discord between Hella and Kolaitis-Väänänen

So far in this chapter, we have proved new generalisations of classic results about generalised quantifiers due to Kolaitis and Väänänen [68] and Hella [58]. In this section, we pause to reflect on a difference between the approaches to generalised quantifiers taken in these two seminal papers. As noted in Section 5.1, the definitions given by Hella and Kolaitis and Väänänen of how generalised quantifiers bind to interpretations are slightly different. In Hella's paper, however, this difference is assumed not to have any impact on the expressiveness of the logic. Indeed, Hella claims that in the case of $n = 1$, Theorem 5.14 can be combined with Kolaitis and Väänänen's Theorem, given above as Theorem 5.5, to prove that Duplicator wins the 1-bijective k-pebble game between two structures $\mathcal{A}$ and $\mathcal{B}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ are equivalent over $\mathcal{L}^k_\infty(\#)$. As we know that the k-pebble bijection game, $\mathbf{Bij}^k$, given in Chapter 2 and defined, for example, in Chapter 13 of Immerman's book [59], captures equivalence in k-variable counting logic we can restate Hella's claim as follows.

Claim 5.26 (Hella's claim)**.** *For any k and any structures $\mathcal{A}$ and $\mathcal{B}$, Duplicator has a winning strategy for $\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ if and only if Duplicator has a winning strategy for $\mathbf{Bij}^1_k(\mathcal{A}, \mathcal{B})$.*

In the following subsections, we disprove this claim but show that it can be amended by changing the number of pebbles in one of the games.

5.4.1 Showing that $\mathcal{L}^k_\infty(\mathbf{Q^1}) \not\equiv \mathcal{L}^k_\infty(\#)$

To disprove the claim above, we exhibit a pair of structures $\mathcal{A}$ and $\mathcal{B}$ where Duplicator has a winning strategy for the game $\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ but not for $\mathbf{Bij}^1_k(\mathcal{A}, \mathcal{B})$. This can be done with the following simple counterexample for $k = 2$. The same example is used by Immerman and Lander [60] to exhibit two graphs which are non-isomorphic but have identical $\mathcal{L}^k_\infty(\#)$ theories.

Proposition 5.27.

$$\mathcal{L}^2(\mathbf{Q}_1) \not\equiv \mathcal{L}^2(\#)$$

Proof. Take the graphs $\mathcal{G}$ and $\mathcal{H}$ to be, respectively, a pair of triangles and a hexagon, as illustrated in Figure 5.4. Immerman and Lander [60] show that these are indistinguishable in $\mathcal{L}^2(\#)$. Now consider the following strategy for Spoiler in Hella's 2-pebble 1-bijective game.

First, place the two pebbles down on the endpoints of any edge (a_1, a_2) of $\mathcal{G}$ and assume Duplicator has responded with some edge (b_1, b_2) of $\mathcal{H}$. Now ask Duplicator for their bijection f from $\mathcal{G}$ to $\mathcal{H}$. Consider the third vertex a_3 of the triangle in $\mathcal{G}$ containing

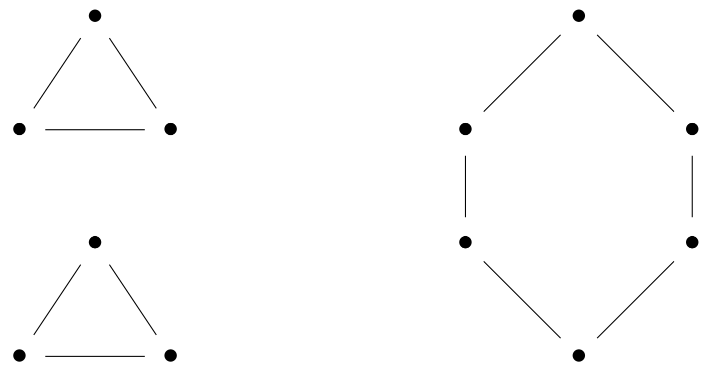


Figure 5.4: Two graphs which are equivalent in $\mathcal{L}^2(\#)$ but not in $\mathcal{L}^2(\mathbf{Q}^1)$

the edge (a_1, a_2). a_3 is in the intersection of the neighbourhoods of a_1 and a_2. Now ask where does a_3 get mapped under Duplicator's bijection? As the intersection of the neighbourhoods of b_1 and b_2 is necessarily empty, it must get mapped into at most one of these neighbourhoods. Suppose, without loss of generality, that $f(a_3)$ is not in $N(b_2)$, then Spoiler should pick up pebble 1 and place it on a_3. Duplicator's response cannot be in the neighbourhood of b_2 so the map $(a_2, a_3) \mapsto (b_2, f(a_3))$ cannot be a partial isomorphism and so Spoiler wins. $\square$

Recalling Theorem 5.5 that $\mathcal{L}^k_\infty(\#)$ is equivalent to Kolaitis and Väänänen's version of $\mathcal{L}^k_\infty(\mathbf{Q}^1)$, we have established with this example that the definitions given by Hella and Kolaitis and Väänänen of a k-variable logic extended by all unary generalised quantifiers cannot be the same for all k. In the next section we show that they are related by an "off-by-one" correction in the number of variables.

5.4.2 Showing that $\mathcal{L}^k_\infty(\mathbf{Q}^1) \equiv \mathcal{L}^{k+1}_\infty(\#)$

In this section, we prove a proposition which relates Hella's version of $\mathcal{L}^k_\infty(\mathbf{Q}^1)$ with that of Kolaitis and Väänänen. We do this by showing that Duplicator has a winning strategy for Hella's 1-bijective k-pebble game if and only if they have a winning strategy for the $k+1$-pebble bijective game. As we saw in the main theorems of Hella and Kolaitis and Väänänen which we generalised earlier in this chapter, such strategies correspond to the equivalence of structures over Hella's $\mathcal{L}^k_\infty(\mathbf{Q}^1)$ and Kolaitis and Väänänen's $\mathcal{L}^{k+1}(\mathbf{Q}^1)$ respectively. So, we can conclude that these two logics are related by an "off-by-one" shift in variable count, proving the following proposition.

Proposition 5.28. *For any two structures $\mathcal{A}$ and $\mathcal{B}$ over a signature σ of arity at most k, Duplicator has a winning strategy for $\mathbf{Bij}^1_k(\mathcal{A}, \mathcal{B})$ if, and only if, Duplicator has a winning strategy for $\mathbf{Bij}^{k+1}(\mathcal{A}, \mathcal{B})$.*

This result, in turn, allows us to relate the expressive powers of three logics which have featured prominently in this chapter, namely counting logic, and the two variants of infinitary first-order logic extended by generalised quantifiers, due respectively to Kolaitis and Väänänen, and Hella.

Corollary 5.29. *For any two structures $\mathcal{A}$ and $\mathcal{B}$ over a signature σ of arity at most k,*

$$\mathcal{A} \equiv_{\mathcal{L}^k_\infty(\mathbf{Q}^1)} \mathcal{B} \iff \mathcal{A} \equiv_{\mathcal{L}^{k+1}_\infty(\#)} \mathcal{B} \iff \mathcal{A} \equiv_{KV\mathcal{L}^{k+1}_\infty(\mathbf{Q}^1)} \mathcal{B}$$

We now build towards a proof of Proposition 5.28.

In order to do this easily we first introduce an alternative form of the $k+1$ pebble game which uses only k pebbles. We first state this game and then show that it is equivalent to the original game.

The first step towards proving this result involves a restatement of the ordinary $k+1$-pebble bijection game, as an equivalent k pebble game with slightly different rules. This restatement is originally due to Grohe and Otto [52] who use it in a similar way to compare it to the Sherali-Adams hierarchy which exhibits a similar “off-by-one” relationship with counting logic.

Definition 5.30. *The k-pebble Duplicator-first bijection game between $\mathcal{A}$ and $\mathcal{B}$ is played as follows. Prior to the jth round the position, π_{j-1}, consists of partial maps $\pi^a_{j-1} : [k] \rightharpoonup A$ and $\pi^b_{j-1} : [k] \rightharpoonup B$.*

- *Duplicator provides a bijection $b_j : A \to B$ such that for each $i \in [k]$, $b_j(\pi^a_{j-1}(i)) = \pi^b_{j-1}(i)$.*
- *Spoiler chooses an element $a \in A$ and wins the game if*

$$\{(\pi^a_{j-1}(i), \pi^b_{j-1}(i))\}_{i\in[k]} \cup \{(a, b_j(a))\}$$

  is not a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

- *If not, the game continues with Spoiler choosing an index $l \in [k]$ and the new position to π_j consisting of π^a_j and π^b_j where $\pi^a_j(i) = a$ and $\pi^b_j(i) = b_j(a)$ and for $i \neq l$ $\pi^a_j(i) = \pi^a_{j-1}(i)$ and $\pi^b_j(i) = \pi^b_{j-1}(i)$.*

Duplicator wins by preventing Spoiler from winning.

The equivalence of the Duplicator-first k-pebble bijection game and the ordinary (Spoiler-first) $k+1$ pebble game is proved by Grohe and Otto [52] but the reasoning is relatively simple. Indeed, the difference between the two games is exactly the difference between whether Spoiler picks up a pair of pebbles at the beginning of Round j, as in the ordinary version, or the end of Round $j-1$, as in the Duplicator-first version. When the size of the position is less than k this distinction doesn't matter and so in particular the one-round versions of these games are clearly equivalent and induction on the number of rounds played completes the proof.

Now that we have an equivalent version of the $k+1$-pebble, bijection game where Duplicator plays first we can more easily compare this to the 1-bijective game of Hella. In Hella's game however, there is a slight subtlety in the size of the partial functions checked in the winning condition, which are only of size at most k. We overcome this difference by slightly restricting the signature of the structures on which we play these games, giving us the following result relating Hella's game to the ordinary bijection game.

Proof of Proposition 5.28. In the proof we replace the game $\mathbf{Bij}^k(\mathcal{A}, \mathcal{B})$ with the equivalent k-pebble Duplicator-first bijection game discussed in the previous paragraph.

It is not hard to see that in both games the choices made by Spoiler and Duplicator are the same in each round, namely, a bijection b_f from Duplicator, and an element $a \in A$ and a pebble index $l \in [k]$ from Spoiler. Similarly, fixing a starting position in a round and given the choices of Spoiler and Duplicator the end position of that round is equal in either game. Now we say that a bijection b_j is *acceptable* (in either game) for a position π_j if there is no Spoiler move which would result in Spoiler winning in that round of the game should Duplicator play b_j.

We now show that for a fixed position π_j, the bijection b_j is acceptable in the 1-bijective game if, and only if, it is acceptable in the Duplicator-first k-pebble game. The backwards direction is easier. Suppose b_j is not acceptable in the 1-bijective game. This means that there is an $l \in [k]$ and an $a \in A$ such that the set $\{(\pi_j^a(i), \pi_j^b(i)\}_{i \in [k] \setminus \{l\}} \cup \{(a, b_j(a))\}$ is not a partial isomorphism. However this is a subset of $\{(\pi_j^a(i), \pi_j^b(i)\}_{i \in [k]} \cup \{(a, b_j(a))\}$ which is tested in the Duplicator-first game and so b_j is not acceptable for this game either. In the other direction we note that if $S_j = \{(\pi_j^a(i), \pi_j^b(i)\}_{i \in [k]} \cup \{(a, b_j(a))\}$ is not a partial isomorphism then there must be some relational symbol R in the signature and some tuple $(a_1, \ldots, a_m)$ from $\{\pi_j^a(i)\}_{i \in [k]} \cup \{a\}$ with image $(b_1, \ldots, b_m)$ under the partial bijection defined S_j such that one of these tuples is in the R relation of its respective structure and the other is not. However as we have insisted that the arity of the signature is at most k this means that $m \leq k$. As we know that $\{(\pi_j^a(i), \pi_j^b(i)\}_{i \in [k]}$ must be a partial isomorphism as it is the position at the start of a round in the game, we can deduce that there must be an l such that $a_n \neq \pi_j^a(l)$ for any $n \in [m]$ and so $\{(\pi_j^a(i), \pi_j^b(i)\}_{i \in [k] \setminus \{l\}} \cup \{(a, b_j(a))\}$ is not an isomorphism. This means that b_j is also not acceptable in the 1-bijective game.

Now we know that the reachable positions in each game under any Duplicator strategy are exactly the same in each of these games and so there is a Duplicator strategy for one if, and only if, there is a Duplicator strategy for the other.

□

In this chapter, we have demonstrated how the comonad $\mathbb{P}_k$ relates to generalised quantifiers by showing that the system of logics which underlies the Kleisli category of $\mathbb{P}_k$, as explored in the last chapter, is equivalent to a system of logics extended by classes of injective, surjective and bijective homomorphism-closed unary generalised quantifiers. We showed that a well-known game of Hella for capturing logics extended by all higher arity generalised quantifiers can be relaxed to capture logics extended by higher arity injective and surjective homomorphism-closed generalised quantifiers. We also clarified a subtlety in relating the Hella's logic to counting logic in the unary case, showing that there is an "off-by-one" relationship which was overlooked in the past.

This progress sets the scene for the next chapter which shows how to generalise the $\mathbb{P}_k$ comonad construction to capture generalised quantifier logics of all arities, introducing new techniques for constructing new comonads from old and identifying a natural new structural parameter related to generalised quantifiers.

Chapter 6

Game comonads and generalised quantifiers

In the last two chapters, we have deepened our understanding of the $\mathbb{P}_k$ comonad in two ways. Firstly, with the Branch Morphism Power Theorem (Theorem 4.3) of Chapter 4, we uncovered new relationships between maps in the Kleisli category, $\mathcal{K}(\mathbb{P}_k)$, and fragments of infinitary k-variable logic. Secondly, we related each of these fragments of logic to k-variable logics extended with unary generalised quantifiers in Theorem 5.7. However, even after these new results, it remains the case that the most expressive logic to be captured by game comonads is $\mathcal{L}^{\omega}_{\infty}(\#)$ which is characterised by Theorem 3.14 which in turn is equivalent to $\mathcal{L}^{\omega}_{\infty}(\mathbf{Q}_1)$ by Kolaits and Väänänen [68].

In this chapter, we go further than this for the first time by constructing a comonad $\mathbb{H}_{n,k}$ whose Kleisli isomorphisms correspond to Duplicator winning strategies in Hella's n-bijective k-pebble game which we recalled in the last chapter. These strategies characterise equivalence in the logic $\mathcal{L}^{k}_{\infty}(\mathcal{Q}_n)$, which is strictly stronger than $\mathcal{L}^{\omega}_{\infty}(\#)$ for any $n > 1$. The relaxations of Hella's game which we also introduced there will be important to the construction. Indeed, we construct $\mathbb{H}_{n,k}$ in the first place to give a comonadic semantics to Duplicator winning strategies for the $+\mathbf{Fun}^n_k$ game. We call this comonad the *Hella comonad* to honour Hella's contribution to games for logics with generaslised quantifiers.

In the first section of this chapter, we define the comonad $\mathbb{H}_{n,k}$. To do this, we introduce a technique for constructing new game comonads from old, obtaining $\mathbb{H}_{n,k}$ as a quotient of $\mathbb{P}_k$. This quotient is informed by a game-theoretic connection between $\exists\mathbf{Peb}^k$ and $+\mathbf{Fun}^n_k$ and is chosen so that we can lift the Morphism Power Theorem for $\mathbb{P}_k$ to one for $\mathbb{H}_{n,k}$.

In the second part of this chapter, we show that the surprising confluence of structure and power results observed in Chapter 3 for $\mathbb{P}_k$, is replicated for the new $\mathbb{H}_{n,k}$. On the "Power" side, we prove versions of the Branch Morphism Power Theorem (Theorem 4.3) and Isomorphism Power Theorem (Theorem 3.14) for $\mathbb{H}_{n,k}$, showing how various maps in $\mathcal{K}(\mathbb{H}_{n,k})$ capture Duplicator winning strategies for each of the relaxations of Hella's

k-pebble n-bijective game, introduced in Chapter 5. Together with Theorem 5.17, this establishes a deep connection between $\mathbb{H}_{n,k}$ and the full range of k-variable logics extended with n-ary quantifiers studied in Chapter 5. On the "Structure" side, we look to generalise the Coalgebra Structure Theorem (Theorem 3.16) which showed a direct correspondence between coalgebras of $\mathbb{P}_k$ and tree decompositions of width k.

We conclude the chapter by reflecting on the wider applicability of the methods developed in this chapter to build comonadic semantics for other pebble games which capture logics more powerful than $\mathbb{P}_k$.

6.1 Constructing the Hella Comonad

In this section we will define a comonad which gives a compositional semantics to Duplicator strategies for the k-pebble n-function game, $+\mathbf{Fun}^n_k$ which we introduced in the last chapter. As we saw in Theorem 5.17, this game serves to bound the expressive power of $+\mathcal{L}^k(\mathbf{Q}^{\mathrm{h}}_n)$ which is a natural generalisation of the logic $\exists^+\mathcal{L}^k_\infty$ whose expressiveness is bounded by the existential k-pebble game. Thus the ultimate aim of this section will be the following theorem.

Theorem 6.1. *For all finite relational signatures σ and all positive integers n, k with $n \leq k$, there is a comonad $(\mathbb{H}_{n,k}, \epsilon^{n,k}, \delta^{n,k})$ on the category $\mathcal{R}(\sigma)$ such that for all structures $\mathcal{A}, \mathcal{B}$ the following are equivalent.*

- *There is a homomorphism $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$.*
- *There is a Duplicator winning strategy for the game $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$.*
- *$\mathcal{A} \Rrightarrow_{\mathcal{L}^k_\infty(\mathbf{Q}^h_n)} \mathcal{B}$.*

To prove this theorem we would like to show that the Duplicator strategies for the game $+\mathbf{Fun}^n_k$ can be represented as Kleisli maps in the same way that we did for $\mathbb{P}_k$ in Chapter 3. However, there are fundamental differences between these games which make it difficult to repeat the arguments used for $\mathbb{P}_k$. In particular, in $+\mathbf{Fun}^n_k$ Duplicator plays first, providing a function $A \to B$ and the number of pebbles moved in each round is allowed to vary.

To overcome these difficulties we introduce a new strategy for creating a game comonad. This involves first showing, in Section 6.1.1 that Duplicator winning strategies for $+\mathbf{Fun}^n_k$ can be translated into a special kind of winning strategy for $\exists\mathbf{Peb}^k$. In Section 6.1.2, we then show that these special strategies can be identified by taking a quotient of the structure $\mathbb{P}_k$. We then show in the final section of this chapter, how this new construction can be extended to a comonad on the category $\mathcal{R}(\sigma)$.

6.1.1 Translating Duplicator strategies

In this section, we study the Duplicator strategies for the game $+\mathbf{Fun}_k^n$ and show how to relate them to strategies for the game $\exists\mathbf{Peb}^k$. As $+\mathbf{Fun}_k^n$ is a strictly more difficult game for Duplicator to win in general we find that it is easy to translate a strategy for $+\mathbf{Fun}_k^n$ to a strategy for $\exists\mathbf{Peb}^k$. What is more surprising however is that we can identify a subset of winning strategies, called *n-consistent* strategies for $\exists\mathbf{Peb}^k$ for which this is reversible. This leads to the following important lemma along the way to constructing the desired comonad from Theorem 6.1.

Lemma 6.2. *Duplicator has an n-consistent winning strategy in $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ if, and only if, it has a winning strategy in $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$.*

In what follows, we build up towards the proof of this lemma by defining the strategy translations in question and fixing notation that will be important for the rest of the construction of $\mathbb{H}_{n,k}$.

Recall from Chapter 3 how the pebbling comonad $\mathbb{P}_k$ is obtained by defining a structure for each $\mathcal{A}$ whose universe consists of (non-empty) lists in $(A \times [k])^*$ which we think of as sequences of moves by Spoiler in a game $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$, with $\mathcal{B}$ unspecified. We call a sequence in $(A \times [k])^*$ a *k-history* (allowing the empty sequence). In contrast, a move in the $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$ involves Spoiler moving up to n pebbles and therefore a history of Spoiler moves is a sequence in $((A \times [k])^{\leq n})^*$. We call such a sequence an *n,k-history.* With this set-up, deterministic sequential Duplicator strategies are given by functions

$$(A \times [k])^+ \to B$$

for $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ and

$$((A \times [k])^{\leq n})^* \to (A \to B)$$

for $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$. The mismatch in the outputs of these games appears initially to be an obstacle to translating between them. However, recalling the definition of Duplicator strategies in terms of branch maps from Chapter 4, we can represent Duplicator strategies for $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ as maps of the following type

$$((A \times [k])^* \times [k]) \to (A \to B)$$

and we will do so throughout this chapter. Noting that $(A\times[k])^+ = (A\times[k])^* \times (A\times[k])$, we can see that this is just an application of Currying a function.

It is easy to see that a strategy for Duplicator in $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$ can always be translated into one for $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$. Indeed, any k-history $s = [(a_1, p_1), \dots (a_l, p_l)]$ can simply be viewed as a degenerate n,k-history $s = [[(a_1, p_1)], \dots [(a_l, p_l)]]$ where Spoiler has chosen to move only one pebble in each round. Further, it is not hard to see that the resulting positions at the end of each round will be the same as those obtained in the n-function game and so if such a strategy is winning for Duplicator in $+\mathbf{Fun}_k^n$ then it is winning in $\exists\mathbf{Peb}^k$. We aim now to establish conditions for when a translation can be made in the reverse direction, from $\exists\mathbf{Peb}^k$ strategies to $+\mathbf{Fun}_k^n$ strategies. For this, it is useful to establish the following machinery.

There is a natural *flattening* operation that takes n,k-histories to k-histories. We denote the operation by F, so that $F([s_1, s_2, \ldots, s_m]) = s_1 \cdot s_2 \cdots s_m$, is the concatenation of the s_i where $s_i \in (A \times [k])^{\leq n}$. Of course, the function F is not injective and has no inverse. However, it is worth considering functions G from k-histories to n,k-histories that are right-inverse to F in the sense that $F(G(t)) = t$. One obvious such function takes a k-history $s_1, \ldots, s_m$ to the n,k-history $[[s_1], \ldots, [s_m]]$, i.e. the sequence of one-element sequences. This is, in some sense, minimal in that it imposes the minimal amount of structure on $G(t)$. We are interested in a *maximal* such function. For this, recall that the Spoiler moves in any n,k-history are sequences in $(A \times [k])^{\leq n}$ which have length at most n and do not have a repeated index from $[k]$. We want a function which splits up a k-history s into a sequence of maximal such blocks. This leads us to the following definition.

Definition 6.3. *A list $s \in (A \times [k])^*$ is called n-*basic *if it contains fewer than or equal to n pairs and the pebble indices are all distinct.*

The n-structure function $S_n : (A \times [k])^* \to ((A \times [k])^{\leq n})^*$ is defined recursively as follows:

- $S_n(s) = [s]$ *if s is n-basic*
- *otherwise, $S_n(s) = [a]; S_n(t)$ where $s = a \cdot t$ such that a is the largest n-basic prefix of s.*

This function should be seen as taking a k-history s and placing square brackets into it to group sequential single pebble moves into a legal Spoiler moves in the $+\mathbf{Fun}_k^n$ game. Thus it should be clear that applying the flattening operation simply removes these square brackets and so $F(S_n(s)) = s$ for any s. It is useful to characterise the range of the function S_n, which we do through the following definition.

Definition 6.4. *An n,k-history t is* structured *if whenever s and s' are successive elements of t, then either s has length exactly n or s' begins with a pair (a,p) such that p occurs in s.*

It is immediate from the definitions that $S_n(s)$ is structured for all k-histories s and that an n,k-history is structured if, and only if, $S_n(F(s)) = s$.

We are now ready to characterise those Duplicator winning strategies for $\exists\mathbf{Peb}^k$ that can be lifted to strategies in $+\mathbf{Fun}_k^n$. First, we define a function that lifts a position in $\exists\mathbf{Peb}^k$ that Duplicator must respond to, i.e. a pair (s,p) where s is a k-history and p a pebble index, to a position in $+\mathbf{Fun}_k^n$, i.e. an n,k-history.

Definition 6.5. *Suppose s is a k-history and s' is the last n-basic list in $S_n(s)$, so $S_n(s) = t; s'$. Let $p \in [k]$ be a pebble index.*

*Define the n-*structuring *$\alpha_n(s,p)$ of (s,p) by*

$$\alpha_n(s,p) = \begin{cases} t; s' & \text{if } |s'| = n \text{ or } p \text{ occurs in } s' \\ t & \text{otherwise.} \end{cases}$$

Now that we have a translation α_n from positions in the $\exists\mathbf{Peb}^k$ to positions in the $+\mathbf{Fun}_k^n$, it is clear that we can lift a strategy from the k-pebble to the n-function game along this translation if Duplicator's response at any position (s,i) relies only on the translated position $\alpha_n(s,i)$. This gives the following important definition.

Definition 6.6. *Say that a Duplicator strategy $\Psi : ((A \times [k])^* \times [k]) \to (A \to B)$ in $\exists\mathbf{Peb}^k$ is n-consistent if for all k-histories s and s' and all pebble indices p and p':*

$$\alpha_n(s,p) = \alpha_n(s',p') \quad \Rightarrow \quad \Psi(s,p) = \Psi(s',p').$$

Intuitively, an n-consistent Duplicator strategy in the game $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ is one where Duplicator plays the same function in all moves that could be part of the same Spoiler move in the game $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$. We are now ready to prove the main result of this subsection.

Proof of Lemma 6.2. The reverse direction is easy. Suppose first that $\Psi : ((A \times [k])^{\leq n})^* \to (A \to B)$ is a Duplicator winning strategy in $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$. Define the strategy Ψ' in $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ such that for a k-history s and a pebble index $p \in [k]$, $\Psi'(s,p) = \Psi(\alpha_n(s,p))$. This is easily seen to be n-consistent and winning.

For the other direction we deal with the case of $n = 1$ separately.

For $n = 1$, all $1,k$-histories are structured. Indeed, for any k-history s and any pebble index p, $\alpha_1(s,p) = G(s)$. This means that, for any p and p', $\alpha_1(s,p) = \alpha_1(s,p')$ and the 1-consistent winning strategies are precisely those such that for any k-history s, pebble indices p and p' and elements a, b if $a = b$ then $f([s;(a,p)]) = f([s;(b,p')])$. This is the same as saying that the branch maps $\phi^f_{s,p}$ and $\phi^f_{s,p'}$ are equal for every history s and every pair of pebble indices p and p'. We denote the common branch map at s by ϕ^f_s. This then gives a strategy in the game $+\mathbf{Fun}_k^1(\mathcal{A},\mathcal{B})$ where after every $1,k$-history t, Duplicator provides the function $\phi^f_{F(t)}$.

When $n \geq 2$, suppose Ψ is an n-consistent winning strategy for Duplicator in $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$. We construct from this a winning strategy Ψ' for Duplicator in $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$. If t is a structured n,k-history and p is the last pebble index occurring in it, we can just take $\Psi'(t) = \Psi(F(t),p)$. To extend this to unstructured n,k-histories, we first define the structured companion of an n,k-history.

Suppose t is an n, k-history that is not structured and let $s, s' \in (A \times [k])^{\leq n}$ be a pair of consecutive sequences witnessing this. We call such a pair a *bad* pair. Let (a, p) be the last pair occurring in s and (a', p') the first pair occurring in s'. Let t' be the prefix of t ending with s and let κ be the last element of A such that (κ, p') appears in $F(t)$ if there is any. We now obtain a new n, k-history from t by replacing the pair s, s' by $s, \mathtt{link}, s'$ where

$$\mathtt{link} = \begin{cases} [(a,p),(\kappa,p')] & \text{if defined} \\ [(a,p),(a,p')] & \text{otherwise.} \end{cases}$$

It is clear that in this n, k-history, neither of the pairs $s, \mathtt{link}$ or $\mathtt{link}, s'$ is bad, so it has one fewer bad pairs than t. Also, this move is chosen so that responding to the moves $F(\mathtt{link})$ according to Ψ does not change the partial function defined by the pebbled position after responding to the moves $F(s)$. Repeating the process, we obtain a structured n, k-history which we call $\tilde{t}$, the structured companion of t.

We can now formally define the Duplicator strategy by saying for any n, k-history t, $\Psi'(t) = \Psi(F(\tilde{t}), p)$ where $\tilde{t}$ is the structured companion of t and p is the last pebble index occurring in t. To see why Ψ' is a winning strategy, we note that as responding with Ψ to the $\mathtt{link}$ moves does not alter the partial function defined by the pebbled position, the function defined after responding to $\tilde{t}$ according to Ψ is the same as that defined after responding to t according to Ψ'. So if there is a winning n, k-history t for Spoiler against Ψ' then $F(\tilde{t})$ is a winning k-history for Spoiler against Ψ, a contradiction.

□

6.1.2 Structural quotients and morphism power

Central to Abramsky, Dawar and Wang's construction [6] of the pebbling comonad $\mathbb{P}_k$ is the Morphism Power Theorem (Theorem 3.13) which states that for any two structures $\mathcal{A}$ and $\mathcal{B}$ there is a Duplicator winning strategy for $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$ if and only if there is a homomorphism $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$ between the I-structures $\mathcal{A}$ and $\mathcal{B}$. Building on the equivalence between n-consistent Duplicator winning strategy for $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$ and winning strategies for $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ proved in Lemma 6.2, the aim of this section is to show that we can identify these special strategies using homomorphisms from a quotiented version of $\mathbb{P}_k\mathcal{A}$. The result is the following Lemma which is central to the construction of the comonad $\mathbb{H}_{n,k}$.

Lemma 6.7. *For all $n, k \in \mathbb{N}$ with $n \leq k$, there is an equivalence relation $\approx_n$ on all structures $\mathbb{P}_k\mathcal{A}$ such that for any structures $\mathcal{A}$ and $\mathcal{B}$, the following are equivalent:*

- *There is a homomorphism $f\colon \mathbb{P}_k\mathcal{A}/{\approx_n} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$.*
- *There is an n-consistent Duplicator winning strategy for $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$.*

In order to define this equivalence relation $\approx_n$, we recall that an n-consistent strategy for $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$ is a map $f\colon \mathbb{P}_k\mathcal{A} \to \mathcal{B}$ where the branch maps $f_{s,i}$ and $f_{t,j}$ are equal if and only if $\alpha_n(s,i)$ and $\alpha_n(t,j)$. This inspired the following definition which identifies k-histories which must have the same response under any n-consistent strategy.

Definition 6.8. *For $n \in \mathbb{N}$ and $\mathcal{A}$ a relational structure. Define $\approx_n$ on the universe of $\mathbb{P}_k\mathcal{A}$ as*

$$s;(a,i) \approx_n t;(b,j) \iff a = b \textbf{ and } \alpha_n((s,i)) = \alpha_n((t,j)).$$

Recalling the definition of a structured n,k-history from Section 6.1.1, for any structured n,k-history t, we write $[t|a]$ to denote the $\approx_n$-equivalence class of an element $s;(a,i) \in \mathbb{P}_k\mathcal{A}$ with $\alpha_n(s,i) = t$.

As defined $\approx_n$ is clearly an equivalence relation as it is defined entirely in terms of the equality relation and so reflexivity, symmetry and transitivity are inherited from $=$.

We now would like to define the quotient structure $\mathbb{P}_k\mathcal{A}/\approx_n$ which identifies elements which are equivalent under $\approx_n$. However, it can be seen that the relation $\approx_n$ is not a congruence of the structure $\mathbb{P}_k$, as defined in Chapter 2. So, there is not a canonical definition of the relations on this quotient. Indeed, given an arbitrary equivalence relation $\sim$ over a relational structure $\mathcal{M}$, there are two standard ways to define relations in a quotient $\mathcal{M}/\!\sim$. We could say that a tuple $(c_1, \ldots c_r)$ of equivalence classes is in a relation $R^{\mathcal{M}/\sim}$ if, and only if, *every* choice of representatives is in $R^{\mathcal{M}}$ or if *some* choice of representatives is in $R^{\mathcal{M}}$. The latter definition has the advantage that the quotient map from $\mathcal{M}$ to $\mathcal{M}/\!\sim$ is a homomorphism and it is this definition that we assume for the rest of the chapter.

This definition of the relational structure $\mathbb{P}_k\mathcal{A}/\approx_n$ has the very useful consequence that the quotient map $q_n\colon \mathbb{P}_kA \to \mathbb{P}_kA/\approx_n$ which sends an element of $\mathbb{P}_kA$ to its $\approx_n$-equivalence class is a homomorphism between the relational structures $\mathbb{P}_k\mathcal{A}$ and $\mathbb{P}_k\mathcal{A}/\approx_n$. With this we can establish the following useful property.

Lemma 6.9.

$$f : \mathbb{P}_k\mathcal{A}/\approx_n \to \mathcal{B} \text{ is a homomorphism} \iff f \circ q_n : \mathbb{P}_k\mathcal{A} \to \mathcal{B} \text{ is a homomorphism}$$

Proof. As we observed above that q_n is a homomorphism, the direction from left to right simply follows from the composition of homomorphisms.

For the other direction, we prove the contrapositive. Note that for any related tuple $(c_1, \ldots, c_m) \in R^{\mathbb{P}_k\mathcal{A}/\approx_n}$ there must be a tuple of elements $(s_1, \ldots, s_n) \in R^{\mathbb{P}_k\mathcal{A}}$ such that $q_n(s_i) = c_i$ for each i. So if f fails to preserve some such related tuple from $\mathbb{P}_k\mathcal{A}/\approx_n$ then the corresponding tuple in $\mathbb{P}_k\mathcal{A}$ is not preserved by $f \circ q_n$.

□

We can now complete this section by providing a proof of the important Lemma 6.7.

Proof of Lemma 6.7. Taking the equivalence relation $\approx_n$ to be that described in Definition 6.8 and the structure $\mathbb{P}_k\mathcal{A}/\approx_n$ and quotient map q_n to be defined as above, we prove the equivalence between homomorphism and strategies as follows.

Suppose we have an n-consistent winning strategy Ψ for Duplicator in $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$. The n-consistency condition implies that the Duplicator response to a Spoiler play $s;(a,i) \in (A\times[k])^*$ is determined by $\alpha_n((s,i))$ and a only. So the corresponding homomorphism $f_\Psi : \mathbb{P}_kA \to B$ (given by Theorem 3.13) respects $\approx_n$ and $f_\Psi \circ q_n$ is a well-defined homomorphism $f : \mathbb{P}_k\mathcal{A}/\approx_n \to B$.

For the other direction, given a homomorphism $f\colon \mathbb{P}_k\mathcal{A}/\approx_n \to \mathcal{B}$ note that $f \circ q_n$ defines a Duplicator winning strategy for $\exists\mathbf{Peb}^k(\mathcal{A},\mathcal{B})$ which is n-consistent, as required.

$\square$

Recalling the equivalence between Duplicator strategies for $+\mathbf{Fun}^n_k$ and n-consistent strategies for $\exists\mathbf{Peb}^k$, the previous result motivates us to define the structure $\mathbb{H}_{n,k}\mathcal{A}$ as the quotient structure $\mathbb{P}_k\mathcal{A}/\approx_n$. In the next section, we complete the proof of Theorem 6.1, by showing that $\mathbb{H}_{n,k}$ can be made into a comonad.

6.1.3 Definition of $\mathbb{H}_{n,k}$

Lemmas 6.2 and 6.7 from the last two sections established that for any structure $\mathcal{A}$ we can construct a structure $\mathbb{H}_{n,k}\mathcal{A}$ which represents the moves of Spoiler in the game $+\mathbf{Fun}^n_k$ in the same way that $\mathbb{P}_k\mathcal{A}$ does for $\exists\mathbf{Peb}^k$. In this section, we prove Theorem 6.1 by showing that this construction can be used to build a comonad on the category $\mathcal{R}(\sigma)$ with the desired properties. To do this we need to do the following three things.

1. Define $\mathbb{H}_{n,k}$ as an endofunctor on $\mathcal{R}(\sigma)$.
2. Define natural transformations $\epsilon^{n,k}\colon \mathbb{H}_{n,k} \implies 1$ and $\delta^{n,k}\colon \mathbb{H}_{n,k}\mathbb{H}_{n,k} \implies \mathbb{H}_{n,k}$.
3. Show that $(\mathbb{H}_{n,k}, \epsilon^{n,k}, \delta^{n,k})$ satisfies the comonad laws.

Defining the $\mathbb{H}_{n,k}$ functor To show that $\mathbb{H}_{n,k}$ can be made into a functor we need to define how it acts on homomorphisms $f\colon \mathcal{A} \to \mathcal{B}$ in a way that respects composition. Recalling the definition of $\approx_n$ from Section 6.1.2, the $\approx_n$-equivalence class of any element $s;(a,i) \in \mathbb{P}_k\mathcal{A}$ is represented by pair $[\alpha_n(s,i)|a]$. We can then define the function $\mathbb{P}_kf/\approx_n\colon \mathbb{P}_k\mathcal{A}/\approx_n \to \mathbb{P}_k\mathcal{B}/\approx_n$ which sends $[\alpha_n(s,i)|a]$ to $[\alpha_n(\mathbb{P}_kf(s),i)|f(a)]$. This is well-defined because the shape of the sequence $\alpha_n(s,i)$ relies on the sequence of pebble indices in s rather than the pebbled elements. So $\alpha_n(s,i) = \alpha_n(t,j)$ implies that

$\alpha_n(\mathbb{P}_k f(s), i) = \alpha_n(\mathbb{P}_k f(t), j)$ and $\mathbb{P}_k f/\approx_n$ is a homomorphism because $\mathbb{P}_k f$ is a homomorphism. We can now use the functoriality of $\mathbb{P}_k$ to deduce that

$$\mathbb{P}_k(f \circ g)/\approx_n = (\mathbb{P}_k f/\approx_n) \circ (\mathbb{P}_k g/\approx_n).$$

So we can define the functor $\mathbb{H}_{n,k}$ as follows.

Definition 6.10. *For $n, k \in \mathbb{N}$, $k \geq n$ and σ a relational signature, we define the functor $\mathbb{H}_{n,k} : \mathcal{R}(\sigma) \to \mathcal{R}(\sigma)$ by:*

- ***On objects*** $\mathbb{H}_{n,k}\mathcal{A} := \mathbb{P}_k\mathcal{A}/\approx_n$.
- ***On morphisms*** $\mathbb{H}_{n,k}f := \mathbb{P}_k f/\approx_n$.

Importantly for the results of the next section, we show in the following lemma that the quotient map q_n is a natural transformation between $\mathbb{P}_k$ and the new comonad $\mathbb{H}_{n,k}$.

Lemma 6.11. *$q_n : \mathbb{P}_k \Rightarrow \mathbb{H}_{n,k}$ is a natural transformation.*

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be relational structures over the same signature and $f : \mathcal{A} \to \mathcal{B}$ be a homomorphism. To show that q_n is natural we need to establish the equality $q_n \circ \mathbb{P}_k f = \mathbb{H}_{n,k}f \circ q_n$. Fix an element $s; (a, i) \in \mathbb{P}_k\mathcal{A}$. On the right hand side, we have that $q_n(s; (a, i)) = [\alpha_n(s, i)|a]$ and so $\mathbb{H}_{n,k}f \circ q_n = [\alpha_n(\mathbb{P}_k f(s), i)|f(a)]$. On the left hand side, $\mathbb{P}_k f(s; (a, i)) = \mathbb{P}_k f(s); (f(a), i)$ and so $q_n \circ \mathbb{P}_k f(s; (a, i)) = [\alpha_n(\mathbb{P}_k f(s), i)|f(a)]$ as required.

□

Defining $\epsilon^{n,k}$ and $\delta^{n,k}$ Having defined the functor $\mathbb{H}_{n,k}$ as a quotient of the functor $\mathbb{P}_k$ with respect to the equivalence relation $\approx_n$, we now show how the counit ϵ and comultiplication δ for $\mathbb{P}_k$ can be lifted to $\mathbb{H}_{n,k}$. We do this in a similar way to the definition of the counit and comultiplication of $\mathbb{P}^*_k$ in Section 4.2.2 by proving the following lemma.

Lemma 6.12. *The counit ϵ and comultiplication δ for $\mathbb{P}_k$ lift to well-defined natural transformations for $\mathbb{H}_{n,k}$ which are defined as follows.*

1. $\epsilon^{n,k}_{\mathcal{A}}([s]) := \epsilon_{\mathcal{A}}(s)$, *and*
2. $\delta^{n,k}_{\mathcal{A}}([s]) := [\mathbb{P}_k q_n(\delta_{\mathcal{A}}(s))]$.

Proof. Suppose $s; (a, i) \approx_n t; (b, j) \in \mathbb{P}_k\mathcal{A}$. Then by the definition of $\approx_n$, we have $a = b$ and so $\epsilon_{\mathcal{A}}(s; (a, i)) = a = b = \epsilon_{\mathcal{A}}(t; (b, j))$. Thus $\epsilon^{n,k}_{\mathcal{A}}$ is well-defined. Furthermore, as $\epsilon_{\mathcal{A}} = \epsilon^{n,k}_{\mathcal{A}} \circ q_n$, we have that $\epsilon^{n,k}_{\mathcal{A}}$ is a homomorphism by Observation 6.9 and by Lemma 6.11 it is natural.

The argument is slightly more complicated for $\delta^{n,k}$. For the function above to be well-defined we need that, for any $s;(a,i) \approx_n t;(a,j)$,

$$(\mathbb{P}_k q_n \circ \delta_{\mathcal{A}})(s;(a,i)) \approx_n (\mathbb{P}_k q_n \circ \delta_{\mathcal{A}})(t;(a,j)),$$

as elements of $\mathbb{P}_k(\mathbb{H}_{n,k}\mathcal{A})$. Firstly, by definition $\delta_{\mathcal{A}}(s;(a,i)) = \delta_{\mathcal{A}}(s);(s;(a,i),i)$ and so $\mathbb{P}_k q_n \circ \delta_{\mathcal{A}}(s;(a,i)) = \mathbb{P}_k q_n(\delta_{\mathcal{A}}(s));(q_n(s;(a,i)),i)$. We can write similar expressions for $t;(a,j)$.

As we have that $\alpha_n(s,i) = \alpha_n(t,j)$, we can deduce that $\alpha_n(\mathbb{P}_k q_n(\delta_{\mathcal{A}}(s)),i) = \alpha_n(\mathbb{P}_k q_n(\delta_{\mathcal{A}}(t)),j)$. This is because $\mathbb{P}_k q_n$ only changes the pebbled elements of a k-history on $\mathbb{P}_k\mathcal{A}$ leaving the pebble indices unchanged and α_n is based only on the pebble indices of a k-history. So, by the definition of $\approx_n$, $\mathbb{P}_k q_n \circ \delta_{\mathcal{A}}(s;(a,i)) \approx_n \mathbb{P}_k q_n \circ \delta_{\mathcal{A}}(t;(a,j))$ if $q_n(s;(a,i)) = q_n(t;(a,j))$, which is precisely the statement that $s;(a,i) \approx_n t;(a,j)$. Naturality for $\delta^{n,k}$ follows from the naturality of q_n and δ. □

Proof of Theorem 6.1 We now put together the results of Section 6.1 to prove that $(\mathbb{H}_{n,k}, \epsilon^{n,k}, \delta^{n,k})$ is a comonad which captures the Duplicator winning strategies of $+\mathbf{Fun}^n_k$ as Kleisli morphisms. The following proof concludes this section.

Proof of Theorem 6.1. To prove that $(\mathbb{H}_{n,k}, \epsilon^{n,k}, \delta^{n,k})$ is a comonad it remains to check that this triple satisfies the counit identities

$$\mathbb{H}_{n,k}\epsilon^{n,k}_{\mathcal{A}} \circ \delta^{n,k}_{\mathcal{A}} = 1_{\mathbb{H}_{n,k}\mathcal{A}} = \epsilon^{n,k}_{\mathbb{H}_{n,k}\mathcal{A}} \circ \delta^{n,k}_{\mathcal{A}}$$

and comultiplication identity

$$\delta^{n,k}_{\mathbb{H}_{n,k}\mathcal{A}} \circ \delta^{n,k}_{\mathcal{A}} = \mathbb{H}_{n,k}\delta^{n,k}_{\mathcal{A}} \circ \delta^{n,k}_{\mathcal{A}}.$$

One way to do this would be to prove these equalities directly as in Lemma 4.20. Here we provide an alternative by lifting these identities from the comonad $\mathbb{P}_k$ as follows.

We claim the stronger result that for any equation E built from composing ϵ, δ and $\mathbb{P}_k$, if E is true then the equation $\tilde{E}$, obtained by replacing ϵ by $\epsilon^{n,k}$, δ with $\delta^{n,k}$ and $\mathbb{P}_k$ with $\mathbb{H}_{n,k}$ is also true. We show this as follows. By the naturality of q_n, we have that $q_n \circ \mathbb{P}_k q_n = \mathbb{H}_{n,k} q_n \circ q_n$ and so, for any m and any $t \in (\mathbb{P}_k)^m\mathcal{A}$, there is a well-defined notion of "the" equivalence class of t, $\mathbf{q_n}(t) \in (\mathbb{H}_{n,k})^m\mathcal{A}$. We now see that $\epsilon^{n,k}$ and $\delta^{n,k}$ have been defined such that $\epsilon^{n,k} \circ \mathbf{q_n} = \mathbf{q_n} \circ \epsilon$ and $\delta^{n,k} \circ \mathbf{q_n} = \mathbf{q_n} \circ \delta$. This means that for any term $T\colon (\mathbb{P}_k)^m\mathcal{A} \to (\mathbb{P}_k)^r\mathcal{A}$ formed out of $\mathbb{P}_k$, ϵ and δ by composition we have that the term $\tilde{T}\colon (\mathbb{H}_{n,k})^m\mathcal{A} \to (\mathbb{H}_{n,k})^r\mathcal{A}$ formed by the replacements above satisfies $\mathbf{q_n}(T(t)) = \tilde{T}(\mathbf{q_n}(t))$. Thus we can lift any equation in $(\mathbb{P}_k, \epsilon, \delta)$ to one over $(\mathbb{H}_{n,k}, \epsilon^{n,k}, \delta^{n,k})$. This works in particular for the counit and comultiplication laws.

Finally we show that for any structures $\mathcal{A}$ and $\mathcal{B}$ there is a homomorphism $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, there is a winning strategy for Duplicator in the game $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$. This is simply the combination of two lemmas proved earlier in this section. In particular, Lemma 6.2, tells us that there is a winning strategy for Duplicator in $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$ if and only if there is an n-consistent winning strategy for Duplicator in $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$ and Lemma 6.7 tells us that this is equivalent to the existence of the required homomorphism.

□

6.2 Structure and power of $\mathbb{H}_{n,k}$

Having shown in the last section that we can create a game comonad, $\mathbb{H}_{n,k}$ for the k-pebble n-function game, we now prove a number of results which connect this comonad to other related games, logics and structural decompositions. These results show that the surprising system of connections uncovered by the pebbling comonad and other game comonads reviewed in Chapter 3 also exists in the more expressive setting of logics with n-ary generalised quantifiers.

In Section 6.2.1, we show that the Kleisli category of this new comonad, can be used to identify logical relations over other k-variable n-ary generalised quantifier logics. In particular, we show that Kleisli isomorphisms correspond to winning strategies in Hella's original k-pebble n-bijective game and that fragments of $\mathcal{L}_\infty(\mathbf{Q}_n)$ restricted to injective, surjective and bijective homomorphism-closed quantifiers can be identified by restricting Kleisli morphisms in a similar manner to the techniques employed in Chapter 4. In this way, we see that $\mathbb{H}_{n,k}$ captures the entire hierarchy of games and logics introduced in the Generalised Hella's Theorem of the last chapter.

In Section 6.2.2, we classify the coalgebras of the new comonad showing that they correspond to a previously unstudied but interesting generalisation of tree decompositions. We prove some preliminary results about this new decomposition and suggest some future directions for investigation of this new concept.

6.2.1 Kleisli maps of $\mathbb{H}_{n,k}$

As we have seen throughout this thesis, the maps in the Kleisli category of the pebbling comonad $\mathbb{P}_k$ have deep connections to pebble games and k-variable logics. In particular, Abramsky, Dawar and Wang proved the Morphism and Isomorphism Power Theorems (Theorems 3.13 and 3.14) which show how the morphisms and isomorphisms in $\mathcal{K}(\mathbb{P}_k)$ relate to the games $\exists\mathbf{Peb}^k$ and $\mathbf{Bij}^k$ and the logics $\exists^+\mathcal{L}^k_\infty$ and $\mathcal{L}^k_\infty(\#)$. In the Branch Morphism Power Theorem (Theorem 4.3), we showed that placing conditions on the branch maps of morphisms in $\mathcal{K}(\mathbb{P}_k)$ exactly captured intermediate games and logics.

In this section, we show that Kleisli maps of the new comonad $\mathbb{H}_{n,k}$ have similar relations to games and logics. We do this by proving the following three theorems which are respectively Morphism, Isomorphism and Branch Morphism Power Theorems for $\mathbb{H}_{n,k}$.

The first is simply a combination of Theorem 6.1 from the last section which connects Kleisli morphisms to the positive n-function game and the first case of Theorem 5.17 from Chapter 5 which relates this game to k-variable infinitary logic extended by all homomorphism-closed generalised quantifiers.

Theorem 6.13 (Morphism Power Theorem for $\mathbb{H}_{n,k}$). *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following are equivalent for any positive integers n, k with $n \leq k$:*

1. *There is a Kleisli morphism $\mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$,*
2. *Duplicator has a winning strategy in $+\mathbf{Fun}^n_k(\mathcal{A}, \mathcal{B})$, and*
3. *$\mathcal{A} \Rrightarrow_{+\mathcal{L}^k_\infty(\mathbf{Q}^h_n)} \mathcal{B}$.*

In addition to this we prove the following two theorems. The Isomorphism Power Theorem for $\mathbb{H}_{n,k}$ connects the comonad with Hella's original k-pebble n-bijective game and the logic $\mathcal{L}^k_\infty(\mathbf{Q}_n)$, as follows.

Theorem 6.14 (Isomorphism Power Theorem for $\mathbb{H}_{n,k}$). *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following are equivalent for any positive integers n, k with $n \leq k$:*

1. *There is a Kleisli isomorphism $\mathcal{A} \cong_{\mathcal{K}(\mathbb{H}_{n,k})} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$,*
2. *Duplicator has a winning strategy in $\mathbf{Bij}^n_k(\mathcal{A}, \mathcal{B})$, and*
3. *$\mathcal{A} \equiv_{\mathcal{L}^k_\infty(\mathbf{Q}_n)} \mathcal{B}$.*

Along the way to proving Theorem 6.14 we introduce a notion of branch maps and define branch-injectivity, branch-surjectivity and branch-bijectivity which we denote by $\to^{\mathrm{i}}_{n,k}$, $\to^{\mathrm{s}}_{n,k}$ and $\to^{\mathrm{b}}_{n,k}$ in the following theorem.

Theorem 6.15 (Branch Morphism Power Theorem for $\mathbb{H}_{n,k}$). *For two relational structures $\mathcal{A}$ and $\mathcal{B}$ the following equivalences hold for any positive integers n, k with $n \leq k$:*

1. *$\mathcal{A} \to^{i}_{n,k} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{+\mathcal{L}^k_\infty(\mathbf{Q}^i_n)} \mathcal{B}$,*
2. *$\mathcal{A} \to^{s}_{n,k} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{+\mathcal{L}^k_\infty(\mathbf{Q}^s_n)} \mathcal{B}$, and*
3. *$\mathcal{A} \to^{b}_{n,k} \mathcal{B}$ for the I-structures $\mathcal{A}$ and $\mathcal{B}$ if, and only if, $\mathcal{A} \Rrightarrow_{+\mathcal{L}^k_\infty(\mathbf{Q}^b_n)} \mathcal{B}$*

In Chapter 4, we followed Abramsky, Dawar and Wang in classifying the morphisms of $\mathcal{K}(\mathbb{H}_{n,k})$ according to whether their *branch maps* are injective, surjective or bijective. Here, we extend these notions to the comonad $\mathbb{H}_{n,k}$. This gives us a way of classifying the morphisms to match the classification of strategies given in Theorem 5.17.

Definition 6.16. *For $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ a Kleisli morphism of $\mathbb{H}_{n,k}$, the* branch maps *of f are defined as the following collection of functions $A \to B$, indexed by the structured n,k-histories $t \in ((A \times [k])^{\leq n})^*$:*

$$\phi_t^f(x) = f([t|x]).$$

We say that such an f is

- branch-injective *if for every t, ϕ_t^f is injective,*
- branch-surjective *if for every t, ϕ_t^f is surjective, and*
- branch-bijective *if for every t, ϕ_t^f is bijective.*

If such a map exists we write, respectively, $\mathcal{A} \to_{n,k}^{i} \mathcal{B}$, $\mathcal{A} \to_{n,k}^{s} \mathcal{B}$, and $\mathcal{A} \to_{n,k}^{b} \mathcal{B}$.

We now prove Theorem 6.15 by showing these restrictions on the Kleisli morphisms are enough to satisfy the extra restrictions on Duplicator in the positive n-injection, n-surjection and n-bijection games from Chapter 5.

Proof of Theorem 6.15. In Theorem 6.1, we saw that the existence of Kleisli maps $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ between I-structures $\mathcal{A}$ and $\mathcal{B}$ is equivalent to the existence of Duplicator winning strategies for the game $+\mathbf{Fun}_k^n(\mathcal{A},\mathcal{B})$. This worked in two steps. First, Lemma 6.2 gave a translation between deterministic strategies Ψ for $+\mathbf{Fun}_k^n$ and n-consistent strategies Ψ' for $\exists\mathbf{Peb}^k$. Secondly, Lemma 6.7 gave a translation between n-consistent strategies for $\exists\mathbf{Peb}^k$ and Kleisli morphisms of $\mathbb{H}_{n,k}$.

The proof of the first equivalence in the theorem at hand follows from the observation that the two translations above preserve the injectivity, surjectivity or bijectivity of the maps provided by Duplicator at each round of the respective games.

The second equivalence in this theorem, between the games and logics, follows from Theorem 5.17 of the last chapter.

□

To complete the correspondence with the strategies and logics in Theorem 5.17, we define the following Kleisli maps in Definition 6.17 and summarise the relationships between maps, strategies and games in Figure 6.1.

Definition 6.17. *We say a Kleisli map $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ between I-structures $\mathcal{A}$ and $\mathcal{B}$ is*

- strongly branch-injective *if it is branch-injective and the related strategy for $+\mathbf{Inj}_k^n(\mathcal{A},\mathcal{B})$ is also winning for $\mathbf{Inj}_k^n(\mathcal{A},\mathcal{B})$,*

- strongly branch-surjective *if it is branch-surjective and the related strategy for* $+\mathbf{Surj}_k^n(\mathcal{A}, \mathcal{B})$ *is also winning for* $\mathbf{Surj}_k^n(\mathcal{A}, \mathcal{B})$, *and*

- strongly branch-bijective *if it is branch-bijective and the related strategy for* $+\mathbf{Bij}_k^n(\mathcal{A}, \mathcal{B})$ *is also winning for* $\mathbf{Bij}_k^n(\mathcal{A}, \mathcal{B})$.

If such a maps exists we write, respectively, $\mathcal{A} \rightarrowtriangle^{i}_{n,k} \mathcal{B}$, $\mathcal{A} \rightarrowtriangle^{s}_{n,k} \mathcal{B}$, *and* $\mathcal{A} \rightarrowtriangle^{b}_{n,k} \mathcal{B}$.

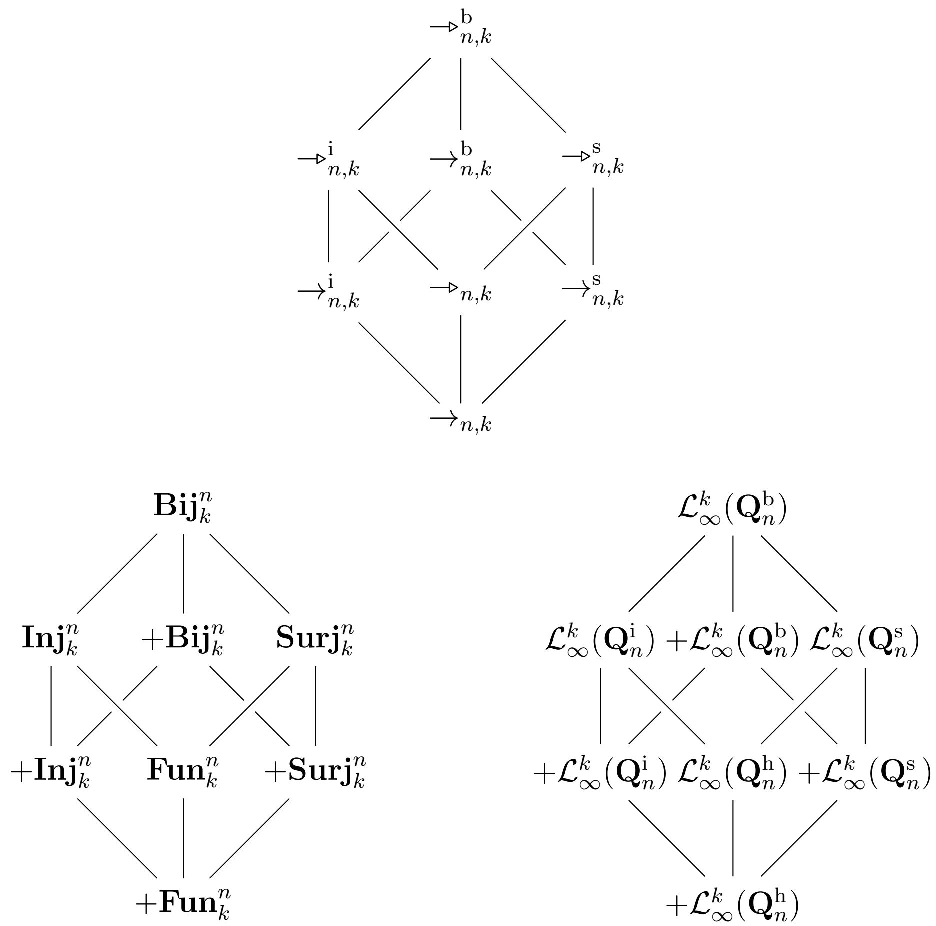


Figure 6.1: Hasse diagrams of the types of Kleisli maps in $\mathcal{K}(\mathbb{H}_{n,k})$ ordered by restrictions on branch maps and their corresponding games (ordered by difficulty for Duplicator) and logics (ordered by expressive power).

With the language of branch maps firmly lifted to the Kleisli category of $\mathbb{H}_{n,k}$, we prove the following generalisation of a lemma of Abramsky, Dawar and Wang [6] which is central to the proof of Theorem 6.14.

Lemma 6.18. *For* $\mathcal{A}, \mathcal{B}$ *finite relational* I*-structures,*

$$\mathcal{A} \rightleftarrows^{i}_{n,k} \mathcal{B} \iff \mathcal{A} \rightleftarrows^{s}_{n,k} \mathcal{B} \iff \mathcal{A} \rightarrowtriangle^{b}_{n,k} \mathcal{B} \iff \mathcal{A} \cong_{\mathcal{K}(\mathbb{H}_{n,k})} \mathcal{B}$$

Proof. As A and B are finite, the existence of an injection $A \to B$ implies that $|A| \leq |B|$. So, $\mathcal{A} \rightleftarrows^{\mathrm{i}}_{n,k} \mathcal{B}$ implies that $|A| = |B|$ and thus any injective map between the two is also surjective and vice versa. This means the first equivalence is trivial and further both of these imply $\mathcal{A} \rightleftarrows^{\mathrm{b}}_{n,k} \mathcal{B}$

For the second equivalence, we first introduce some notation. Let $P^m_{\mathcal{A}}$ be the finite substructure of $\mathbb{H}_{n,k}\mathcal{A}$ induced on the elements $\{[s|a] \mid s \in ((A \times [k])^{\leq n})^{\leq m}\}$. Note that for any $f\colon \mathcal{A} \to^{\mathrm{b}}_{n,k} \mathcal{B}$, the Kleisli completion f^* restricts to a bijective homomorphism $P^m_{\mathcal{A}} \to P^m_{\mathcal{B}}$ for each m. So if $f\colon \mathbb{H}_{n,k}\mathcal{A} \to \mathcal{B}$ and $g\colon \mathbb{H}_{n,k}\mathcal{B} \to \mathcal{A}$ are branch-bijective, we have for each m a pair of bijective homomorphisms $P^m_{\mathcal{A}} \rightleftarrows P^m_{\mathcal{B}}$. As these are finite structures we can deduce that these are indeed isomorphisms and so f is a strategy for $\mathbf{Bij}^n_k(\mathcal{A}, \mathcal{B})$.

For the final equivalence, if f witnesses $\mathcal{A} \rightarrowtriangle^{\mathrm{b}}_{n,k} \mathcal{B}$ then we have, by induction, that f^* is an isomorphism from $P^m_{\mathcal{A}}$ to $P^m_{\mathcal{B}}$ for each m,. So $f^* : \mathbb{H}_{n,k}\mathcal{A} \to \mathbb{H}_{n,k}\mathcal{B}$ is an isomorphism witnessing $\mathcal{A} \cong_{\mathcal{K}(\mathbb{H}_{n,k})} \mathcal{B}$. For the converse we suppose that there is an isomorphism $h^* : \mathbb{H}_{n,k}\mathcal{A} \to \mathbb{H}_{n,k}\mathcal{A}$. Then the Kleisli map $h = \epsilon^{n,k}_{\mathcal{B}} \circ h^*$ is a strongly branch-bijective strategy. □

The final equivalence in Lemma 6.18 gives us a relationship between strongly branch-bijective maps and isomorphisms in the Kleisli category of $\mathbb{H}_{n,k}$. We now finish off the proof of Theorem 6.14 as follows.

Proof of Theorem 6.14. By Lemma 6.18, we have that $\mathcal{A} \cong_{\mathcal{K}(\mathbb{H}_{n,k})} \mathcal{B}$ if and only if there is a strongly branch-bijective map $f\colon \mathcal{A} \rightarrowtriangle^{\mathrm{b}}_{n,k} \mathcal{B}$. By the translation to Duplicator strategies in Theorem 6.15 and the definition of $\rightarrowtriangle^{\mathrm{b}}_{n,k}$ in Definition 6.17, the existence of such an f is equivalent to the existence of a winning strategy for Duplicator in the game $\mathbf{Bij}^n_k(\mathcal{A}, \mathcal{B})$, completing the first equivalence in the theorem.

To prove the correspondence with logic we invoke Hella's theorem from [58] which is stated above as Theorem 5.14. □

6.2.2 Coalgebras of $\mathbb{H}_{n,k}$

The Coalgebra Structure Theorem for $\mathbb{P}_k$ (Theorem 3.16) proved by Abramsky, Dawar and Wang [6] showed that the coalgebras of the comonad $\mathbb{P}_k$ have a surprising correspondence with objects of great interest to finite model theorists. That is, any coalgebra $\alpha : \mathcal{A} \to \mathbb{P}_k\mathcal{A}$ gives a tree decomposition of $\mathcal{A}$ of width at most $k-1$ and any such tree decomposition can be turned into a coalgebra.

In this section, we define a new form of structural decomposition called an *extended tree decomposition* which is bounded by two parameters, width and arity, and we relate this to our new comonad by proving the following version of the Coalgebra Structure Theorem for $\mathbb{H}_{n,k}$.

Theorem 6.19 (Coalgebra Structure Theorem for $\mathbb{H}_{n,k}$). *For $\mathcal{A}$ a finite relational structure the following are equivalent:*

1. *there is a $\mathbb{H}_{n,k}$-coalgebra $\alpha \colon \mathcal{A} \to \mathbb{H}_{n,k}\mathcal{A}$*
2. *there is a structured extended tree decomposition of $\mathcal{A}$ with width at most k and arity at most n*

The equivalent result for $\mathbb{P}_k$ works because $\mathbb{P}_k\mathcal{A}$ has a treelike structure where any pebble history, or branch, $s \in \mathbb{P}_k\mathcal{A}$ only witnesses the relations from the $\leq k$ elements of $\mathcal{A}$ which remain pebbled after Spoiler plays the sequence of moves s. So a homomorphism $\mathcal{A} \to \mathbb{P}_k\mathcal{A}$ witnesses a sort of treelike k-locality of the relational structure $\mathcal{A}$ and the $\mathbb{P}_k$-coalgebra laws are precisely enough to ensure this can be presented as a tree decomposition (of width $< k$).

In lifting this comonad to $\mathbb{H}_{n,k}$, we have given away some of the restrictive k-local nature of $\mathbb{P}_k$ which makes this argument work. The structure $\mathbb{H}_{n,k}\mathcal{A}$ witnesses many more of $\mathcal{A}$'s relations than $\mathbb{P}_k\mathcal{A}$. Take, for example, the substructure induced on the elements $\{[\epsilon|x] \mid x \in \mathcal{A}\}$, where ϵ is the empty history. This witnesses all relations in $\mathcal{A}$ which have arity $\leq n$. So, in particular, if $\mathcal{A}$ contains no relations of arity greater than n, this substructure is just a copy of $\mathcal{A}$ and the obvious embedding $A \to \mathbb{H}_{n,k}A$ can be easily seen to be a $\mathbb{H}_{n,k}$-coalgebra. From this, we can see that if $\mathbb{H}_{n,k}$-coalgebras capture some notion of n-generalised tree decomposition, this should clearly be more permissive than the notion of tree decomposition, allowing a controlled amount of non-locality (parameterised by n) and collapsing completely for σ-structures with $n \geq \text{arity}(\sigma)$. In this section we define the appropriate generalisation of tree decomposition and show its relation with $\mathbb{H}_{n,k}$-coalgebras.

Generalising tree decomposition

Recall the following definition of a tree decomposition of a σ-structure for example from Definition 4.1.1 of [50].

Definition 6.20. *A* tree decomposition *of a σ-structure $\mathcal{A}$ is a pair (T, B) with T a tree and $B : T \to 2^A$ such that:*

1. *For every $a \in A$ the set $\{t \mid a \in B(t)\}$ induces a subtree of T; and*
2. *for all relational symbols $R \in T$ and related tuples $\mathbf{a} \in R^{\mathcal{A}}$, there exists a node $t \in T$ such that $\mathbf{a} \subset B(t)$.*

To arrive at a generalisation of tree decomposition which allows for the non-locality discussed above, we first introduce the following extension of ordinary tree decompositions.

Definition 6.21. *An* extended tree decomposition *of a σ-structure $\mathcal{A}$ is a triple (T, β, γ) with $\beta, \gamma : T \to 2^A$ such that:*

1. *(T, B) is a tree-decomposition of $\mathcal{A}$ where $B : T \to 2^A$ is defined by $B(t) := \beta(t) \cup \gamma(t)$; and*

2. *if $a \in \gamma(t)$ and $a \in B(t')$ then $t \leq t'$.*

In an extended tree decomposition, the *bags* B of the underlying tree decomposition are split into a *fixed bag* β and a *floating bag* γ. The second condition above ensures that $\gamma(t)$ contains only elements $a \in \mathcal{A}$ for which t is their *first*[1] appearance in (T, B). Width and arity are two important properties of extended tree decompositions which are defined as follows.

Definition 6.22. *Let $D = (T, \beta, \gamma)$ be an extended tree decomposition.*

The width, *$w(D)$, of D is* $\max_{t \in T} |\beta(t)|$.

The arity, *$ar(D)$, of D is the least $n \leq w(D)$ such that:*

1. *if $t < t'$ then $|\beta(t') \cap \gamma(t)| \leq n$; and*

2. *for every tuple $(a_1, \ldots, a_m)$ in every relation R of $\mathcal{A}$, there is a $t \in T$ such that $\{a_1, \ldots, a_m\} \subseteq B(t)$ and $|\{a_1, \ldots, a_m\} \cap \gamma(t)| \leq n$.*

We note that the definition of width here differs from the width of the underlying tree decomposition (T, B). However as we see in Lemma 6.27 having an ordinary tree decomposition of width k is equivalent to having an extended tree decomposition of width k and arity 1.

We are particularly interested in extended tree decompositions that are further well-structured, in a sense that is related to the definition of structured n, k-histories in Section 6.1.1.

Definition 6.23. *An extended tree decomposition with width k and arity n is* structured *if for every $a \in A$ there exists $t \in T$ s.t. $a \in \gamma(t)$, for every node t, $\gamma(t) \neq \emptyset$, for any child t' of t $\beta(t') \cap \gamma(t) \neq \emptyset$ and for any t'' a child of t' we have that either:*

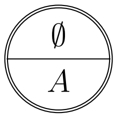
- *$|\beta(t') \cap \gamma(t)| = n$; or*
- *$|\beta(t')| < k$; or*
- *$\gamma(t) \cap \beta(t') \setminus \beta(t'') \neq \emptyset$*

[1] minimum in the tree order

Drawing extended tree decompositions and examples

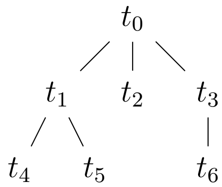
We draw extended tree decompositions as trees where the nodes have two labels, an upper label indicating the fixed bag at that node and the lower label denoting the floating bag. In this subsection, we give some simple examples of these decompositions.

Example 6.24. *Any structure $\mathcal{A}$ which has no relations of arity greater than n admits a trivial arity n, width 0 extended tree decomposition with a single node. This is drawn as:*

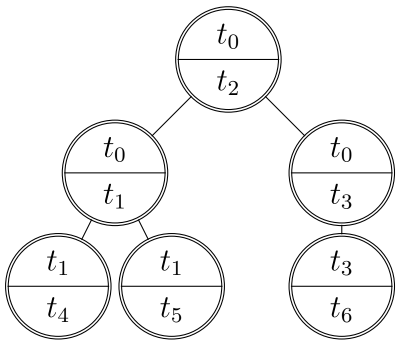


From this example we see that, in particular, any graph $\mathcal{G}$ has a trivial extended tree decomposition of arity 2. The next two examples show that for graphs, extended tree decompositions of arity 1 look similar to ordinary tree decompositions.

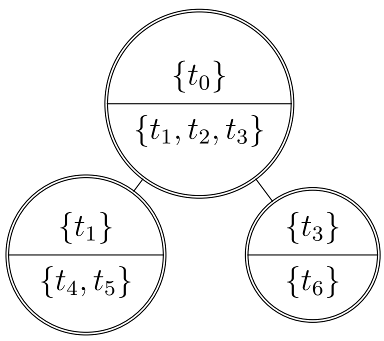
Example 6.25. *Consider the following tree $\mathcal{T}$ as a graph.*



As with ordinary tree decompositions a tree can be given a decomposition of width 1 by creating a bag for each edge. The corresponding extended tree decomposition of width 1 and arity 1 for $\mathcal{T}$ is the following:

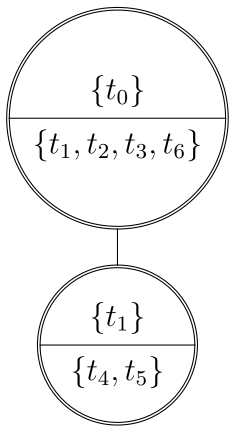


Unlike with ordinary tree decompositions, the floating bags in extended tree decompositions can be used to give more succinct decompositions (without changing the width). For example, the following is an extended decomposition of $\mathcal{T}$ again with width 1 and arity 1.



As we see in Lemma 6.27, the correspondence between ordinary tree decompositions and extended tree decompositions of arity 1 extends beyond trees to all relational structures. However, for signatures of arity higher than 2 increasing the arity of an extended tree decomposition can result in non-trivial decompositions of lower width as is shown by the following example.

Example 6.26. *Consider a hypergraph $\mathcal{T}'$ constructed from $\mathcal{T}$ above by adding ternary edges $\{t_0, t_1, t_2\}, \{t_0, t_1, t_3\}, \{t_0, t_2, t_3\}$ and $\{t_1, t_4, t_5\}$. Such a structure contains a 4-clique $\{t_0, t_1, t_2, t_3\}$ in its Gaifman graph, as defined, for example, in Definition 4.1 of Libkin's book [71]. So, it cannot have an ordinary tree decomposition of width less than 3. However, the following is an extended tree decomposition of width 1 and arity 2 for $\mathcal{T}'$:*



Preliminary results on extended tree decompositions

Before proving Theorem 6.19 we present two results which establish some basic facts about this new type of decomposition. The first establishes the equivalence of width k, arity 1 extended tree decompositions with ordinary tree decompositions of width k. This is interesting as we recall from Abramsky, Dawar and Wang [6] that tree decompositions of width k correspond to coalgebras of $\mathbb{P}_{k+1}$ whereas we will see in Theorem 6.19 that coalgebras of $\mathbb{H}_{1,k}$ give extended tree decompositions of arity 1 and width k. In this light, this result can be seen as demonstrating the extra strength of $\mathbb{H}_{1,k}$ over $\mathbb{P}_k$, in a manner that is consistent with the "off-by-one" difference in the logics studied in Section 5.4.2.

Lemma 6.27. *A relational structure $\mathcal{A}$ has a tree decomposition of width k if, and only if, it has an extended tree decomposition of width k and arity 1*

Proof. ($\implies$) Without loss of generality we can assume that (T, β) is a tree decomposition such that for all $t \in T$ $|\beta(t)| = k+1$ and if t' is a child of t in T then $|\beta(t) \cap \beta(t')| = k$. We now show how to transform such a tree decomposition into an extended decomposition (T', β', γ) of width k and arity 1.

Define the equivalence relation $\approx$ on T as

$$t' \approx t'' \iff t' \text{ and } t'' \text{ have the same parent } t \text{ in } T \textbf{ and } \beta(t) \cap \beta(t') = \beta(t) \cap \beta(t'')$$

Now we can define the extended decomposition as follows:

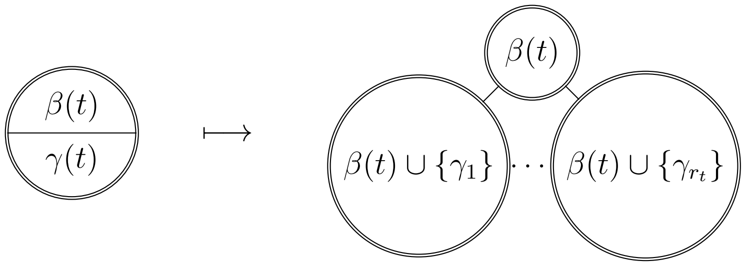
- $T' = T/\!\approx$
- $\beta'([t]) = \beta(t) \cap \beta(t_0)$ where t_0 is the common parent of the elements of $[t]$
- $\gamma([t]) = \bigcup_{t' \in [t]} \beta(t') \setminus \beta(t_0)$

For non-root nodes t in T both β' and γ are well-defined by the definition of $\approx$. For the singleton equivalence class $[r]$ containing the root of T we choose any $c_r \in \beta(r)$ and define $\beta'([r]) = \beta(r) \setminus \{c_r\}$ and $\gamma([r]) = \{c_r\}$.

Letting $B([t]) = \beta'([t]) \cup \gamma([t])$ we have that $B([t]) \supset \beta(t)$ and so (T', B) is a tree decomposition. Furthermore, $\gamma([t]) \cap \beta(t_0) = \emptyset$ by definition, so for any $[t'] < [t]$ we have $B([t']) \cap \gamma([t]) = \emptyset$ by the condition that $\beta^{-1}(x)$ is a connected subtree of T for any $x \in T$. So (T', β', γ) is an extended tree decomposition.

It is easy to see that the maximum size of $\beta'(t)$ is equal to k by design. So the width of (T', β', γ) is k. If $\mathbf{a}$ is a tuple in a relation of $\mathcal{A}$ we know that there is a node $t \in T$ such that $\mathbf{a} \subset \beta(t)$. By definition, $\beta(t) \subset B'(t)$ with $|\beta(t) \cap \gamma([t])| \leq 1$. So the arity of (T', β', γ) is 1, as required.

($\Longleftarrow$) To go backwards we take a width k, arity 1 extended tree decomposition (T, β, γ) and we construct a tree decomposition $(\tilde{T}, \tilde{\beta})$ by replacing each node $t \in T$ with the following spider H_t:



where the children of the leaf of H_t labelled by $\beta(t) \cup \{\gamma_i\}$ are the roots of the spiders $H_{t'}$ such that t' is a child of t in T and $\beta(t') \cap \gamma(t) = \{\gamma_i\}$. To see that this is a tree decomposition note firstly that $\tilde{T}$ is clearly a tree under this construction. Next, it is easy to see that for any $a \in \beta(t) \cup \gamma(t)$, a either appears in every bag of H_t or just in a single leaf. This means that the bags containing a in $\tilde{T}$ still form a connected subtree. Lastly, we need to show that each related tuple $\mathbf{a}$ in $\mathcal{A}$ is contained in some bag of $\tilde{T}$. This is guaranteed by the condition that (T, β, γ) has arity 1, which means any time $\mathbf{a} \subset \beta(t) \cup \gamma(t)$ there exists $\gamma_i \in \gamma(t)$ such that $\mathbf{a} \subset \beta(t) \cup \gamma(t) \cup \{\gamma_i\}$.

$\square$

Having established the connection between extended tree decompositions and ordinary tree decompositions we now relate extended tree decompositions to the $\mathbb{H}_{n,k}$ comonad introduced in Section 6.1 with the next easy but important result. It is noteworthy here that the extended tree decompositions admitted by the structures $\mathbb{H}_{n,k}\mathcal{A}$ are *structured*. This is important later in this section.

Lemma 6.28. *For any finite $\mathcal{A}$, there is a structured extended tree decomposition of $\mathbb{H}_{n,k}\mathcal{A}$ of width k and arity n*

Proof. Recall from Definition 6.8 that the underlying set of $\mathbb{H}_{n,k}\mathcal{A}$ consists of representatives $[s|a]$ of equivalence classes in $\mathbb{P}_k\mathcal{A}/\approx_n$ where $s \in ((A \times [k])^{\leq n})^*$ is a *structured* n,k-history and $a \in A$. We construct an extended tree decomposition where each node is an n,k-history s appearing in one of these representatives. The tree ordering is simply given by the prefix relation. The fixed bag at s, $\beta(s)$, contains up to k elements which represent the at most k elements which are pebbled after s is played. To describe these explicitly, let $\overline{s} \in \mathbb{P}_k\mathcal{A}$ be the flattening of the list s and for each $i \in [k]$ appearing as a pebble index in s and let s_i be the maximal prefix of $\overline{s}$ which ends in (a,i) for some $a \in A$. Then $\beta(s)$ contains the $\approx_n$-equivalence classes of each of the s_i. As there can be at most k elements in this set, our extended tree decomposition has width k. The floating bag is given, more simply as $\gamma(s) = \{[s|a] \mid a \in A\}$. From this description it is easy to see that for any $[s|a] \in \mathbb{H}_{n,k}\mathcal{A}$, if $[s|a]$ appears in $\beta(s')$ then s is a prefix of s' and for any s'' with $s \sqsubset s'' \sqsubset s'$ we have $[s|a] \in \beta(s'')$. This confirms that $B^{-1}([s|a])$ is a connected subtree of T and that $\gamma^{-1}([s|a])$ is a singleton containing the root of that subtree.

To show that (T,β,γ) defines an extended tree decomposition of $\mathbb{H}_{n,k}\mathcal{A}$ it now suffices to show that any related tuple $\mathbf{g} = ([s_1|a_1], \ldots [s_l|a_l]) \in R^{\mathbb{H}_{n,k}\mathcal{A}}$ appears in some bag. Because of the way relations are defined in $\mathbb{H}_{n,k}$ we can find $(t_1, \ldots t_l) \in R^{\mathbb{P}_k\mathcal{A}}$ s.t. $q(t_i) = [s_i|a_i]$. By the definition of relations in $\mathbb{P}_k\mathcal{A}$ we know that the t_i are totally ordered by the prefix relation. This means that the s_i are similarly totally ordered with largest element s. The related tuple is contained in $\beta(s) \cup \gamma(s)$. Furthermore, $\mathbf{g} \cap \gamma(s)$ contains the t_i for which $q(t_i) = [s|a_i]$. As these are linearly ordered by the prefix relation it would be impossible for there to be more than n distinct such lists. This means that (T,β,γ) is indeed an extended tree decomposition of width k and arity n.

To see that (T,β,γ) is structured we rely on the fact that the sequences $s \in ((A \times [k])^{\leq n})^*$ appearing in T are themselves structured in the sense of Definition 6.4. The proof is as follows. Suppose there is a node $s \in T$ with a child $s;x \in T$ where $x \in (A \times [k])^{\leq n}$ and suppose that $|\beta(s;x) \cap \gamma(s)| < n$ and $|\beta(s;x)| = k$. We now need to show that for any node $s;x;y \in T$ $\gamma(s) \cap \beta(s;x) \setminus \beta(s;x;y) \neq \emptyset$. Unpacking the definitions we have that $\gamma(s) \cap \beta(s;x)$ contains elements $[s|a]$ where (a,i) appears in x for some i. As we also know that $|\beta(s;x)| = k$, which means in particular that x does not contain two pairs (a,i) (a,j) for $i \neq j$ because if it did the contributions from pebbles i and j to $\beta(s;x)$ would both be $[s|a]$. These two facts together mean that the length of x must be strictly less than n. Thus as $s;x;y$ is a structured n,k-history we must have that the first element of y is

(b_y, i_y) where the index i_y appears in some pair (b_x, i_y) in x. It is not hard to see that $[s|b_x] \in \gamma(s) \cap \beta(s;x) \setminus \beta(s;x;y)$, completing our proof.

□

We now prove the main theorem of this section, that the $\mathbb{H}_{n,k}$-coalgebras are in correspondence with structured extended tree decompositions of width k and arity n.

Proof of Theorem 6.19. (1 $\implies$ 2) Let α be a coalgebra and, as $\epsilon \circ \alpha = \mathrm{id}_{\mathcal{A}}$, let $\alpha(a) = [s_a|a]$. Recall that by Lemma 6.28 there is a structured extended tree decomposition (T, β, γ) of $\mathbb{H}_{n,k}\mathcal{A}$ with arity n and width k where the nodes of T are labelled by structured n,k-histories $s \in ((A \times [k])^{\leq n})^*$. We use this decomposition to define a decomposition $(T_\alpha, \beta_\alpha, \gamma_\alpha)$ on $\mathcal{A}$ as follows:

- T_α is the tree T restricted to the set $\{s_a \mid a \in A\}$.
- $\beta_\alpha(s) := \{a \in A \mid \alpha(a) \in \beta(s)\}$.
- $\gamma_\alpha(s) := \{a \in A \mid \alpha(a) \in \gamma(s)\}$.

We now show, firstly, that this is an extended tree decomposition, secondly that it has width k and arity n and finally that it is structured.

$(T_\alpha, \beta_\alpha, \gamma_\alpha)$ **is an extended tree decomposition** First of all, this requires that T_α be a tree. For any $s \in T_\alpha$ we have some $a \in A$ with $\alpha(a) = [s|a]$. Suppose that $s = [l_1|l_2|\ldots|l_m]$. It is sufficient to show that $s_i \in T_\alpha$ for any prefix $s_i = [l_1|\ldots|l_i]$ of s (including the empty sequence). This fact can be deduced from the comultiplication law that for all a $\mathbb{H}_{n,k}\alpha(\alpha(a)) = \delta_A(\alpha(a))$. The left-hand side of this equation is $\mathbb{H}_{n,k}\alpha(\alpha(a)) = [\overline{s}|\alpha(a)]$ where $\overline{s} = [\overline{l_1}|\overline{l_2}|\ldots|\overline{l_m}]$ and the right-hand side is $\delta_A(\alpha(a)) = [\tilde{s}|\alpha(a)]$ where $\tilde{s} = [\tilde{l_1}|\tilde{l_2}|\ldots|\tilde{l_m}]$. Taking any $l_i = [(b_1, p_1), \ldots (b_{m_i}, p_{m_i})]$ it is not hard to see that $\overline{l_i} = [([\alpha(b_1)|b_1], p_1) \ldots ([\alpha(b_{m_i})|b_{m_i}], p_{m_i})]$ and $\tilde{l_i} = [([s_{i-1}|b_1], p_1), \ldots ([s_{i-1}|b_{m_i}], p_{m_i})]$. From this we can conclude that for any b appearing in l_i for any $1 \leq i \leq m$ we have that $\alpha(b) = [s_{i-1}|b]$ where s_0 is the empty sequence. This proves that all prefixes of s appear in T_α. Now we show that $B_\alpha := \beta_\alpha \cup \gamma_\alpha$ (T_α, B_α) defines a tree decomposition of $\mathcal{A}$. Indeed $B_\alpha^{-1}(a)$ is a subtree because it is really the intersection of two subtrees of the original T. Furthermore, for any $\mathbf{a} \in R^{\mathcal{A}}$, we have that $\alpha(\mathbf{a}) \in R^{\mathbb{H}_{n,k}\mathcal{A}}$. As (T, β, γ) is a tree decomposition, there is an $s \in T$ with $\alpha(\mathbf{a}) \subset \beta(s) \cup \gamma(s)$. You can assume $\alpha(\mathbf{a}) \cap \gamma(s) \neq \emptyset$ by taking the longest prefix of s which satisfies this[2]. This means that $s \in T_\alpha$ and $\mathbf{a} \subset \beta_\alpha(s) \cup \gamma_\alpha(s)$. This shows that $(T_\alpha, \beta_\alpha\gamma_\alpha)$ defines an extended tree decomposition.

[2]This works by noting that for s' a parent of s in T, $\beta(s) \setminus \beta(s') \subset \gamma(s)$

$(T_\alpha, \beta_\alpha\gamma_\alpha)$ **has width** k **and arity** n As α is injective by the coalgebra law $\epsilon \circ \alpha = \mathrm{id}_{\mathcal{A}}$, we know that for any $s \in T_\alpha$ $|\beta_\alpha(s)| \leq |\beta(s)|$. As (T, β, γ) has width k this means that $|\beta_\alpha(s)| \leq k$ for all $s \in T_\alpha$ and so $(T_\alpha, \beta_\alpha, \gamma_\alpha)$ has width k. For arity, we have that for any related tuple $\mathbf{a}$ in $\mathcal{A}$ the tuple $\alpha(\mathbf{a})$ is related in $\mathbb{H}_{n,k}\mathcal{A}$. As (T, β, γ) has arity n we know that, for any $s \in T$, $|\alpha(\mathbf{a}) \cap \gamma(s)| \leq n$. So again by the injectivity of α $|\mathbf{a} \cap \gamma_\alpha(s)| \leq n$ and so $(T_\alpha, \beta_\alpha, \gamma_\alpha)$ has arity n.

$(T_\alpha, \beta_\alpha, \gamma_\alpha)$ **is structured** Finally the extended tree decomposition is structured because (T, β, γ) is structured and the coalgebra laws guarantee that $|\beta_\alpha(s)| = |\beta(s)|$ and $|\beta_\alpha(s') \cap \gamma_\alpha(s)| = |\beta(s') \cap \gamma(s)|$ for any $s \in T_\alpha$ with child node s'. This first equation is deduced by noting that injectivitiy guarantees $|\beta_\alpha(s)| \leq |\beta(s)|$. The reverse inequality comes from the fact that any $t \in \beta(s)$ is the $\approx_n$ equivalence class of some prefix of s. As we saw before, the comultiplication law guarantees that such classes are realised as $\alpha(b)$ for an appropriate b so we have $|\beta_\alpha(s)| = |\beta(s)|$. The second equation follows from the same reasoning. Together these ensure that the conditions for being structured which are satisfied in (T, β, γ) are also satisfied in $(T_\alpha, \beta_\alpha, \gamma_\alpha)$.

(2 $\implies$ 1) Defining a $\mathbb{H}_{n,k}$ coalgebra from a structured extended tree decomposition (T, β, γ) of width k and arity n requires some careful bookkeeping which is presented explicitly here. Throughout we rely on the fact that our tree T comes with an order $\leq$ and so has a root which we call r. By the conditions of being structured, we have for each $a \in A$ a $\leq$-minimal node $c_a \in A$ where a appears in $B(c_a)$ and we have that $a \in \gamma(c_a)$. This means in particular that at the root $\beta(r) = \emptyset$.

The general strategy in defining the coalgebra α_T is to assign to each node $c \in T$ a structured n, k-history $s_c \in ((A \times [k])^{\leq n})^*$ which records the elements of A which have appeared in (T, β, γ) on the path from r to c. We then show that $\alpha_T(a) = [s_{c_a}|a]$ defines a $\mathbb{H}_{n,k}$ coalgebra for $\mathcal{A}$.

Defining s_c Starting at the root we define s_r to be the empty list. At each new node in $c \in T$ with parent c' we define $l_c \in (A \times [k])^{\leq n}$ to record the elements of A which appear in $\gamma(c')$ and persist in $\beta(c)$, assisted by a function $\iota_c : \beta(c) \to [k]$ which keeps track of the indices assigned to each element. As the arity of (T, β, γ) is n we know that $|\gamma(c') \cap \beta(c)| \leq n$. We now define s_c, l_c and ι_c inductively on the nodes of the tree T, such that for c a child of c', $\iota_c = \iota_{c'}$ on $\beta(c) \cap \beta(c')$ and s_c is formed by appending l_c to $s_{c'}$. This defines s_c on all the nodes of T.

Defining l_c in such a way as to ensure s_c is a structured n, k-history requires some care with assigning pebble indices from $[k]$ to the elements in $\gamma(c') \cap \beta(c)$. We say that a *live* prefix of s_c is a prefix s' of the flattened list $F(s_c) \in (A \times [k])^*$ with final element (b, i) such that no larger prefix of $F(s_c)$ ends with (b', i) for any $b' \in A$. We say that b is live in s_c if it appears at the end of some live prefix s'. The end goal is that s_c will be an n, k-history where the live elements are exactly those in $\beta(c)$ and that for each such element b there is a live prefix of s_c ending in the pair $(b, \iota_c(b))$.

At each c we partition $\beta(c)$ as $N_c \cup R_c$ where $N_c := \gamma(c') \cap \beta(c)$ is the set of *new* elements in β_c and $R_c := \beta(c) \cap \beta(c')$ is the set of elements *retained* from the parent node. Firstly, we define ι_c to be equal to $\iota_{c'}$ on R_c. As the width of (T, β, γ) is k we know that $|\beta(c)| \leq k$ and so the number of *free indices* $|[k] \setminus \iota_c(R_c)|$ is at least as big as the number of new elements $|N_c|$ so we can assign to each element b of N_c a distinct index $\iota_c(b)$ from $[k] \setminus \iota_c(R_c)$. In many cases this is enough and we can pick any ordering $b_1, \ldots b_m$ of the elements in N_c and set l_c to be the list $[(b_1, \iota_c(b_1)), \ldots (b_m, \iota_c(b_m))]$.

We now need to define some modifications to this to ensure that s_c is structured. Recall that an n, k-history s is structured if and only if for every pair of successive blocks l' appearing immediately before l in s we have that either $|l| = n$ or the first pebble index in l must have appeared in l'. To ensure this holds true for each s_c, we need to take extra care defining l_c in cases where $|l_{c'}|$ or $|N_c|$ are less than n.

If $|l_{c'}| < n$ then we must choose $\iota_c(b_1)$ to be an index which appeared in $l_{c'}$. To see that we can do this recall that (T, β, γ) is structured and so for each non-root node c' with child c we have (using our new language from this proof) that at least one of the following is true

1. $|N_{c'}| = n$,
2. $|\beta(c')| < k$; or
3. $R_c \setminus N_{c'} \neq \emptyset$.

In the first case, we have $|l_{c'}| = n$ so no action needs to be taken. In the second case, where $|N_{c'}| < n$ and $|\beta(c')| < k$ then there is a spare index $i \in [k] \setminus \iota_{c'}(\beta(c'))$ and we define $l_{c'}$ to be $[(b'_1, \iota_{c'}(b'_1)), \ldots (b'_m, \iota_{c'}(b'_{m'}), (b'_{m'}, i)]$ and we define $\iota_c(b_1) := i$. In the third case, there may not be a spare index i but instead there is some element $b \in N_{c'} \setminus R_c$ meaning that some element which appears in $l_{c'}$ does not need to be live after l_c. In this case we simply define $\iota_c(b_1) := \iota_{c'}(b)$. Collectively, these modifications ensure that s_c is structured and so the definition $\alpha_T(a) := [s_{c_a}|a]$ is well-defined. It remains to show that α_T is a coalgebra.

α_T is a coalgebra To show that α_T is a homomorphism, take any related tuple $\mathbf{a} \in R^{\mathcal{A}}$. As (T, β, γ) is an extended tree decomposition there is some c such that $\mathbf{a} \subset \beta(c) \cup \gamma(c)$. Now as the arity of the decomposition is n there are at most n elements $a \in \mathbf{a}$ with $a \in \gamma(c)$ and so $\alpha(a) = [s_c|a]$. For all the other elements $a' \in \mathbf{a}$ there must be some earlier c_0 with $a' \in \gamma(c_0)$ and a unique path $c_0 < c_1 < \cdots < c_q = c$ linking c_0 and c in T. We must have $a' \in \beta(c_1)$ and $a' \in R_{c_i}$ for all $1 \leq i \leq q$ so by the definition of s_c above we know that the index $\iota_{c_1}(a')$ used to pebble a' in l_{c_1} has not been reallocated by the end of s_c. From this it is easy to see that the tuple $\alpha(\mathbf{a})$ (with function application defined component-wise on the tuple) is related in $\mathbb{H}_{n,k}\mathcal{A}$. Finally, we verify that α satisfies the coalgebra laws. The counit law, $\epsilon \circ \alpha = \mathrm{id}_{\mathcal{A}}$ is satisfied by definition. For comultiplication, it suffices to check that for any $a, b \in A$, if b appears in $s_{c_a} = [l_{c_1}| \dots |l_{c_q}]$ then it appears in exactly one of the l_{c_i} and $\alpha(b) = [[l_{c_1}| \dots |l_{c_{i-1}}]|b]$. This can be seen to hold from the construction above, concluding our proof.

□

In this chapter, we have a achieved a central aim of the thesis by providing a comonadic semantics to a logic which is strictly more expressive than $\mathcal{L}^{\omega}_{\infty}(\#)$. We did this by constructing, for each n the family of comonads $(\mathbb{H}_{n,k})_{k \in \mathbb{N}}$ which together capture the n-ary generalised quantifier logic $\mathcal{L}^{\omega}_{\infty}(\mathbf{Q}_n)$ and other related logics, as well as new notions of structural decomposition for these logics. Additionally, by constructing this as a quotient of $\mathbb{P}_k$ and establishing that the quotient map q_n is a natural transformation which commutes with the counit and comultiplications of $\mathbb{P}_k$ and $\mathbb{H}_{n,k}$ we can extend the network of known game comonads from Figure 3.1 to the new state of affairs summarised in Figure 6.2.

$$\begin{tikzcd}
\mathbb{H}_{1,k} \arrow[r] & \dots \arrow[r] & \mathbb{H}_{n,k} \arrow[r] & \dots \\
& \mathbb{P}_k \arrow[ul] \arrow[ur] & & \\
\mathbb{E}_k \arrow[ur] & \mathbb{PR}_k \arrow[u] & \mathbb{P}_{n,k} \arrow[ul] &
\end{tikzcd}$$

Figure 6.2: A new hierarchy of game comonads. Arrows are comonad morphisms.

While $\mathcal{L}^{\omega}_{\infty}(\mathbf{Q}_n)$ is a very expressive logic, from the viewpoint of descriptive complexity theory it is somewhat unsatisfactory for two reasons. The first is that, for $n > 1$, even the first order version of this logic $\mathbf{FO}(\mathbf{Q}_n)$ is capable of expressing properties which are uncomputable. In particular, it can express any isomorphism-closed query on graphs. Secondly, it was shown by Hella [58] that for any fixed n there are `PTIME` properties which are not expressible in $\mathcal{L}^{\omega}_{\infty}(\mathbf{Q}_n)$. Despite these shortcomings, Dawar [32] showed that if there is a logic for `PTIME` then there is such a logic of the form $\mathbf{FO}(\mathbf{Q})$ where $\mathbf{Q}$ is some uniform family of generalised quantifiers.

Inspired by the inability of $\mathcal{L}^{\omega}_{\infty}(\#)$ to express solvability of linear equations over finite fields (see [14] for an overview), one source of candidate logics for `PTIME` has involved expanding **FO** by linear-algebraic generalised quantifiers. Such logics have included rank logic [35] and linear-algebraic logic [34]. Although both of these have been shown by Lichter [72, 36] to be insufficient to capture `PTIME`, we remain interested in finding a compositional approach to this kind of extension to counting logic. There are many open questions about applying the comonadic approach to the some of these logics and, in particular, a so far unfulfilled aim of the game comonads programme has been a game comonad for Dawar and Holm's invertible maps game [37]. In the next two chapters, however, we depart from comonads and show that different tools from category theory are useful for studying linear-algebraic approximations to homomorphism and isomorphism. In Chapter 7, we demonstate how monads can be used to capture well studied linear programming relaxations of the homomorphism problem and in Chapter 8 we see how presheaves and cohomology can help us expand the $\mathcal{L}^\omega_\infty(\#)$-equivalence to distinguish structures which differ on those properties which Lichter [72] showed not to be expressible in rank logic.

Chapter 7

Monads for approximating homomorphism

So far in this thesis, we have explored the effectiveness of comonads in capturing games which describe different logical relations approximating homomorphism and isomorphism on relational structures. We have shown that the remit of such comonadic methods extends far beyond the realm of k-variable logics with counting quantifiers which were central to Abramsky, Dawar and Wang's pebbling comonad [6] and other variations on this. However, as noted at the end of the last chapter, one area of finite model theory and descriptive complexity where these methods have so far struggled to find application is that of approximations to homomorphism and isomorphism which incorporate linear algebraic powers. As linear algebra is a source of problems which are solvable in `PTIME`, for example, computing ranks of matrices and solving systems of linear equations, these approximations are especially interesting for algorithms and descriptive complexity theory. In the next two chapters, we go beyond comonads in search of category-theoretic constructions which help us to reason compositionally about these linear-algebraic approximations to homomorphism and isomorphism.

In this chapter, we recall some simple examples of linear algebraic approximations to homomorphism which occur in the study of constraint satisfaction problems. These approximations involve relaxing the homomorphism condition $\mathcal{A} \to \mathcal{B}$ into a system of linear equations over some semiring $\mathbf{S}$ and the power of these approximations on finite structures and for certain values of $\mathbf{S}$ has been extensively studied, particularly in the promise CSP community.

We show that these approximations admit a category-theoretic semantics in terms of *monads* which is somewhat dual to the comonadic semantics of earlier chapters. In Chapter 3, we saw how the non-empty list comonad could be lifted to relational structures to represent Spoiler moves in model comparison games. In this chapter, we see how to lift the vector space and distribution monads, $\mathbb{V}_{\mathbf{S}}$ and $\mathbb{D}_{\mathbf{S}}$ to relational structures. In doing so, we prove an analogue of the Power theorems seen earlier for comonads (Theorem 3.13 & 6.13) by showing that Kleisli morphisms of the form $\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ correspond to satisfying assignments to systems of $\mathbf{S}$-linear equations which approximate the relation $\mathcal{A} \to \mathcal{B}$.

We note, in Section 7.2.3, that for some values of $\mathbf{S}$ the structures $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ have appeared before under different guises in the constraint satisfaction literature and without the monad structure. This helps us to establish links between Kleisli maps and various well-known CSP algorithms in Section 7.3.2 and between maps of the form $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ and structure polymorphisms in Section 7.4.1. The presentation in this chapter also allows us to see these examples in the wider context of their relation to the distribution monad for any semiring $\mathbf{S}$. This raises interesting questions about Kleisli isomorphisms and algebras for these monads, inspired by the parallel Power and Structure results for game comonads.

7.1 Linear programming relaxations for homomorphism and isomorphism

Since Tinhofer [87, 88], it has been common to study relaxations of homomorphism and isomorphism between graphs in terms of linear programming problems. This comes from the fact that the existence of a homomorphism between relational structures is equivalent to the a 0-1 solution to a system of linear equations as seen in Definition 7.1. While these problems are in general difficult to solve, the relaxations obtained by replacing the 0-1 valued matrices with those taking values in some semiring $\mathbf{S}$ may be easily computable and often provide approximations to $\to$ and $\cong$ which are interesting in finite model theory and descriptive complexity. Examples of these are fractional homomorphism [26] and isomorphism [84], the related Sherali-Adams hierarchy and the basic and affine linear programming relaxations of the constraint satisfaction problem. In this section, we present a unified and general framework for these relaxations, drawing largely on the presentation of Butti and Dalmau [26].

To provide a homomorphism $f\colon \mathcal{A} \to \mathcal{B}$ between two relational structures, we provide two pieces of data. Firstly for each element in the underlying set of A we provide an element in B and for each related tuple $\mathbf{a} \in R^{\mathcal{A}}$ we provide a related tuple $\mathbf{b} \in R^{\mathcal{B}}$ which agrees with the assignments on the elements. It has long been recognised that these interrelated conditions can be written down as the following system of equations where there is a homomorphism if and only if there is a solution in the set $\{0, 1\}$. The current presentation is due to Butti and Dalmau [26].

Definition 7.1. *We define the system of equations $LP_{\text{0-1}}(\mathcal{A}, \mathcal{B})$ as follows. Let there be variables $x_{a,b}$ for each $a \in A$ and $b \in B$ and $x^R_{\mathbf{a},\mathbf{b}}$ for each $R \in \sigma$ and tuples $\mathbf{a}$ and $\mathbf{b}$ in A and B respectively with length $ar(R)$.*

$$\forall a \in A, b \in B \; x_{a,b} \in \{0,1\}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, \mathbf{b} \in B^{ar(R)} \; x^R_{\mathbf{a},\mathbf{b}} \in \{0,1\}$$
$$\forall a \in A \qquad \sum_{b \in B} x_{a,b} = 1 \tag{Hom.1}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, \mathbf{b} \notin R^{\mathcal{B}} \qquad x^R_{\mathbf{a},\mathbf{b}} = 0 \tag{Hom.2}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, a \in \{\mathbf{a}\}, b \in B \qquad \sum_{\substack{f\colon \{\mathbf{a}\} \to B \\ f(a)=b}} x^R_{\mathbf{a},f(\mathbf{a})} = x_{a,b} \tag{Hom.3}$$

Solving this system of equations is equivalent to determining if there is a homomorphism $\mathcal{A} \to \mathcal{B}$. As noted in Chapter 2, any constraint satisfaction problem can be expressed as such a homomorphism problem and so solving this system of equations is `NP-Complete` in general. This motivates studying the relaxation of these equations to different algebraic domains which we define as follows.

Definition 7.2. *Let* $\mathbf{S} = (S, +, \cdot, 0_{\mathbf{S}}, 1_{\mathbf{S}})$ *be a semiring. We define the system of equations* $LP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ *as follows.*

$$\forall a \in A, b \in B \; x_{a,b} \in \mathbf{S}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, \mathbf{b} \in B^{ar(R)} \; x^R_{\mathbf{a},\mathbf{b}} \in \mathbf{S}$$
$$\forall a \in A \qquad \sum_{b \in B} x_{a,b} = 1_{\mathbf{S}} \tag{$LP_{\mathbf{S}}.1$}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, \mathbf{b} \notin R^{\mathcal{B}} \qquad x^R_{\mathbf{a},\mathbf{b}} = 0_{\mathbf{S}} \tag{$LP_{\mathbf{S}}.2$}$$
$$\forall \mathbf{a} \in R^{\mathcal{A}}, a \in \{\mathbf{a}\}, b \in B \qquad \sum_{\substack{g\colon \{\mathbf{a}\} \to B \\ g(a)=b}} x^R_{\mathbf{a},g(\mathbf{a})} = x_{a,b} \tag{$LP_{\mathbf{S}}.3$}$$

Another variant of this system which is considered by the CSP algorithms community, for example by Brakensiek, Guruswami, Wrochna and Živný in Section 2.3 of [23], involves replacing $LP_{\mathbf{S}}.3$ with the following condition. While it may not be obvious from the statement, this is a relaxtion of $LP_{\mathbf{S}}.3$ as we establish later in Theorem 7.16.

$$\forall \mathbf{a} \in R^{\mathcal{A}}, i \in [ar(R)], b \in B \qquad \sum_{\mathbf{b}[i]=b} x^R_{\mathbf{a},\mathbf{b}} = x_{\mathbf{a}[i],b} \tag{$BLP_{\mathbf{S}}.3$}$$

Taking the name from the Basic Linear Program which is exactly this system of equations with $\mathbf{S} = \mathbb{Q}_{\geq 0}$, we call this system $BLP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$.

We now recap over some well-known aliases of these systems of equations.

Observation 7.3. *Let $\mathbb{Q}_{\geq 0}$ and $\mathbb{Z}$ be, respectively, the semiring of non-negative rationals and the integers. Then the following cases of $LP_{\mathbf{S}}$ and $BLP_{\mathbf{S}}$ have the following well-known aliases.*

- *If $LP_{\mathbb{Q}_{\geq 0}}(\mathcal{A}, \mathcal{B})$ has a solution we also say that there is a* fractional homomorphism *from $\mathcal{A}$ to $\mathcal{B}$. $LP_{\mathbb{Q}_{\geq 0}}(\mathcal{A}, \mathcal{B})$ is also called $SA^1(\mathcal{A}, \mathcal{B})$ because it is the first level of a Sherali-Adams-inspired hierarchy. (Butti & Dalmau, [26])*
- *$BLP_{\mathbb{Q}_{\geq 0}}(\mathcal{A}, \mathcal{B})$ is also called the basic linear program (BLP) for the CSP instance $(\mathcal{A}, \mathcal{B})$. (Brakensiek, Guruswami, Wrochna & Živný, [23])*
- *$BLP_{\mathbb{Z}}(\mathcal{A}, \mathcal{B})$ is also called the affine integer relaxation (AIP) of the CSP instance $(\mathcal{A}, \mathcal{B})$. (Brakensiek, Guruswami, Wrochna & Živný, [23])*

In the next section, we construct a monad on relational structures inspired by these relaxations of homomorphism. In Section 7.3, we show that solutions to the systems of equations are equivalent to Kleisli morphisms of these monads.

7.2 Linear-algebraic monads

So far in this thesis, we have studied comonads on the category of relational structures. These are interesting in finite model theory and descriptive complexity because they can represent approximations to the homomorphism relation by some Spoiler-Duplicator game where Spoiler has only limited access to the structure $\mathcal{A}$. They do this by replacing $\mathcal{A}$ with some structure $\mathbb{C}_k\mathcal{A}$ such that $\mathbb{C}_k\mathcal{A} \to \mathcal{A}$ so that the existence of a map of the form $\mathbb{C}_k\mathcal{A} \to \mathcal{B}$ is a weaker condition in general than the existence of a map of the form $\mathcal{A} \to \mathcal{B}$.

We now wish to consider a somewhat dual method of approximating homomorphisms by replacing the right-hand-side structure, $\mathcal{B}$ with a construction $\mathbb{T}\mathcal{B}$ with $\mathcal{B} \to \mathbb{T}\mathcal{B}$ so that maps $\mathcal{A} \to \mathbb{T}\mathcal{B}$ approximate $\mathcal{A} \to \mathcal{B}$. Dually to the story told in Chapter 3 the structure required to make such maps compose nicely is that of a monad on the category of relational structures for any signature. This technique was first employed by Abramsky, Barbosa, de Silva and Zapata [3] who constructed a monad on the category of relational structures whose Kleisli morphisms characterised winning strategies in quantum-inspired non-local games.

In this section, we define two new families of monads on relational structures based on free linear-algebraic monads on **Set** which are parameterised by a semiring **S** in a similar way to the systems of equations in Section 7.1. The first monad is based on the vector space monad (or more generally the free left semimodule monad) and the second, a submonad of the first, is based on the distribution monad. We end this section by observing examples of these general constructions which have previously appeared in the logic and algorithms literature.

7.2.1 The vector space monad: $\mathbb{V}_{\mathbf{S}}$

Recall from Chapter 2 that a semiring $\mathbf{S}$ is an algebraic structure consisting of an underlying set S with binary operations $\cdot$ and $+$, called multiplication and addition respectively, and two constants 0 and 1 which are respectively the identity for $+$ and $\cdot$. In the examples we are concerned with both $\cdot$ and $+$ are commutative.

Given such a semiring $\mathbf{S}$ and a relational structure $\mathcal{A}$, we construct the relational structure $\mathbb{V}_{\mathbf{S}}\mathcal{A}$ on the set of formal $\mathbf{S}$-linear sums over elements of the underlying set A as follows.

Definition 7.4. *For a relational structure $\mathcal{A}$ over a finite signature σ and a semiring $\mathbf{S}$, the structure $\mathbb{V}_{\mathbf{S}}\mathcal{A}$ has as the underlying set all functions $\alpha\colon A \to \mathbf{S}$ with* finite support, *meaning that the set $\{a \mid \alpha(a) \neq 0\}$ is finite. We write the elements as formal $\mathbf{S}$-linear sums over the elements of A as follows, where we abbreviate $\alpha(a)$ to α_a,*

$$\mathbb{V}_{\mathbf{S}}A := \left\{ \sum_{a\in A} \alpha_a a \;\mid\; \alpha\colon A \to \mathbf{S} \right\}.$$

The related tuples for each relation $R \in \sigma$ are witnessed by functions $\gamma\colon R^{\mathcal{A}} \to \mathbf{S}$ of finite support and are written as follows, where we abbreviate $\gamma(\mathbf{a})$ to $\gamma_{\mathbf{a}}$,

$$R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}} := \left\{ \left(\sum_{\mathbf{a}\in R^{\mathcal{A}}} \gamma_{\mathbf{a}}\mathbf{a}[1], \ldots, \sum_{\mathbf{a}\in R^{\mathcal{A}}} \gamma_{\mathbf{a}}\mathbf{a}[m] \right) \;\mid\; \gamma\colon R^{\mathcal{A}} \to \mathbf{S} \right\}$$

where $R \in \sigma$ has arity m.

On the underlying sets this is in fact the free left-$\mathbf{S}$-semimodule monad (see, for example, Example 2.8 of [48]) which has the following associated definitions for $\mathbb{V}_{\mathbf{S}}f\colon \mathbb{V}_{\mathbf{S}}A \to \mathbb{V}_{\mathbf{S}}B$ for any function $f\colon A \to B$ and unit and multiplication natural transformations η and μ. When $\mathbf{S}$ is a field this is precisely the vector space monad.

Definition 7.5. *For any semiring $\mathbf{S}$, structures $\mathcal{A}$ and $\mathcal{B}$, and morphism $f\colon \mathcal{A} \to \mathcal{B}$, we define the functions $\mathbb{V}_{\mathbf{S}}f\colon \mathbb{V}_{\mathbf{S}}A \to \mathbb{V}_{\mathbf{S}}B$, $\eta_{\mathcal{A}}\colon A \to \mathbb{V}_{\mathbf{S}}A$ and $\mu_{\mathcal{A}}\colon \mathbb{V}_{\mathbf{S}}\mathbb{V}_{\mathbf{S}}A \to \mathbb{V}_{\mathbf{S}}A$ as follows.*

$$f(\sum_{a\in A} \alpha_a a) = \sum_{a\in A} \alpha_a f(a)$$

$$\eta_{\mathcal{A}}(a) = a$$

$$\mu_{\mathcal{A}}(\gamma_1[\sum_{a\in A} \alpha_a^1 a] + \ldots + \gamma_m[\sum_{a\in A} \alpha_a^m a]) = \sum_{i=1}^{m} \sum_{a\in A} \gamma_1 \alpha_a^1 a$$

As the triple $(\mathbb{V}_{\mathbf{S}}, \eta, \mu)$ inherits a monad structure on the underlying sets from the free left-semimodule monad, it only remains to check that these maps are relational homomorphisms to lift this monad structure to $\mathcal{R}(\sigma)$. We do this in the next result.

Proposition 7.6. *For any finite relational signature σ the triple $(\mathbb{V}_{\mathbf{S}}, \eta, \mu)$ defines a monad on $\mathcal{R}(\sigma)$.*

Proof. For this proof we write a generic element $\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}$ as

$$\theta = \sum_{a \in A} \alpha_a^\theta a.$$

Now, we show that the monad on **Set** defined above is also a monad on $\mathcal{R}(\sigma)$, by showing that $\mathbb{V}_{\mathbf{S}}f$, $\eta_{\mathcal{A}}$ and $\mu_{\mathcal{A}}$ are homomorphisms.

Given a homomorphism $f\colon \mathcal{A} \to \mathcal{B}$, to show that $\mathbb{V}_{\mathbf{S}}f$ is a homomorphism we must show that, for any $R \in \sigma$ and any tuple $(\theta_1, \dots \theta_m) \in (\mathbb{V}_{\mathbf{S}}\mathcal{A})^m$,

$$(\theta_1, \dots \theta_m) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}} \implies (\mathbb{V}_{\mathbf{S}}f(\theta_1), \dots \mathbb{V}_{\mathbf{S}}f(\theta_m)) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{B}}.$$

Suppose that $(\theta_1, \dots \theta_m) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}$. By definition this means there is a set of variables $\gamma_{\mathbf{a}}$ for each $\mathbf{a} \in R^{\mathcal{A}}$ such that, for each $i \in [m]$,

$$\theta_i = \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}} \mathbf{a}[i].$$

As f is a homomorphism, we must also have that $f(\mathbf{a}) \in R^{\mathcal{B}}$ and so we can set

$$\gamma_{\mathbf{b}}^f := \sum_{\substack{\mathbf{a} \in R^{\mathcal{A}} \\ f(\mathbf{a}) = \mathbf{b}}} \gamma_{\mathbf{a}}.$$

This then gives that

$$\sum_{\mathbf{b} \in R^{\mathcal{B}}} \gamma_{\mathbf{b}}^f \mathbf{b}[i] = \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}} f(\mathbf{a}[i]) = \mathbb{V}_{\mathbf{S}}f(\theta_i),$$

and so $(\mathbb{V}_{\mathbf{S}}f(\theta_1), \dots, \mathbb{V}_{\mathbf{S}}f(\theta_m)) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{B}}$, as required.

Showing that $\eta_{\mathcal{A}}$ is a homomorphism is much easier. Indeed, if $(a_1, \dots a_m) \in R^{\mathcal{A}}$ then setting $\gamma_{\mathbf{a}}^\eta = 1_{\mathbf{S}}$ if $\mathbf{a} = (a_1, \dots, a_m)$ and $\gamma_{\mathbf{a}}^\eta = 0_{\mathbf{S}}$ otherwise. We have that $(\eta_{\mathcal{A}}(a_1), \dots, \eta_{\mathcal{A}}(a_m)) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}$.

Finally, we show that $\mu_{\mathcal{A}}$ is a homomorphism. Suppose we have some related tuple in $R^{\mathbb{V}_{\mathbf{S}}\mathbb{V}_{\mathbf{S}}\mathcal{A}}$ and we write this as

$$\left(\sum_{\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}} \alpha_\theta^1 \theta, \dots, \sum_{\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}} \alpha_\theta^m \theta \right) \in R^{\mathbb{V}_{\mathbf{S}}\mathbb{V}_{\mathbf{S}}\mathcal{A}}.$$

This means that there is some function $\gamma\colon R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}} \to \mathbf{S}$ such that, for each $i \in [m]$,

$$\sum_{\theta \in R^{\mathbb{V}_{\mathbf{S}}}} \gamma_\theta \theta[i] = \sum_{\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}} \alpha_\theta^i \theta.$$

Now to show that $\mu_{\mathcal{A}}$ is a homomorphism we need to show that

$$\left(\sum_{\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}} \sum_{a \in \mathcal{A}} \alpha_\theta^1 \alpha_a^\theta a, \dots, \sum_{\theta \in \mathbb{V}_{\mathbf{S}}\mathcal{A}} \sum_{a \in \mathcal{A}} \alpha_\theta^m \alpha_a^\theta a \right) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}.$$

To do this note that for each $\theta \in R^{\mathbb{V}_\mathbf{S}}$ there is, by definition, a function $\gamma^\theta \colon R^\mathcal{A} \to \mathbf{S}$ such that, for each $i \in [m]$,

$$\sum_{\mathbf{a} \in R^\mathcal{A}} \gamma^\theta_\mathbf{a} \mathbf{a}[i] = \theta[i].$$

Then define the function $\gamma^\mu \colon R^\mathcal{A} \to \mathbf{S}$ as

$$\gamma^\mu_\mathbf{a} := \sum_{\theta \in R^{\mathbb{V}_\mathbf{S}\mathcal{A}}} \gamma_\theta \gamma^\theta_\mathbf{a}.$$

Then we can deduce from the equations above that, for each $i \in [m]$,

$$\begin{aligned}
\sum_{\mathbf{a} \in R^\mathcal{A}} \gamma^\mu_\mathbf{a} \mathbf{a}[i] &= \sum_{\mathbf{a} \in R^\mathcal{A}} \sum_{\theta \in R^{\mathbb{V}_\mathbf{S}\mathcal{A}}} \gamma_\theta \gamma^\theta_\mathbf{a} \mathbf{a}[i] \\
&= \sum_{\theta \in R^{\mathbb{V}_\mathbf{S}\mathcal{A}}} \gamma_\theta \sum_{\mathbf{a} \in R^\mathcal{A}} \gamma^\theta_\mathbf{a} \mathbf{a}[i] \\
&= \sum_{\theta \in R^{\mathbb{V}_\mathbf{S}\mathcal{A}}} \gamma_\theta \theta[i] \\
&= \sum_{\theta \in \mathbb{V}_\mathbf{S}\mathcal{A}} \alpha^i_\theta \theta \\
&= \sum_{\theta \in \mathbb{V}_\mathbf{S}\mathcal{A}} \sum_{a \in \mathcal{A}} \alpha^i_\theta \alpha^\theta_a a.
\end{aligned}$$

Which concludes the proof that $\mu_\mathcal{A}$ is a homomorphism. $\square$

While this construction yields a monad on relational structures the following observation shows that we can not hope for the Kleisli morphisms $\mathcal{A} \to \mathbb{V}_\mathbf{S}\mathcal{B}$ to yield any of the interesting relaxations observed in Section 7.1.

Observation 7.7. *The relational structure $\mathbb{V}_\mathbf{S}\mathcal{A}$ defined in Definition 7.4 is somewhat degenerate in the sense that it always contains an element on which there is a loop for every relation $R \in \sigma$. For any $\mathbf{S}$ and structure $\mathcal{A}$, consider the element $\mathbf{0} = \sum_{a \in A} 0_\mathbf{S} \cdot a$ obtained by setting each α_a to the additive identity $0_\mathbf{S}$ in $\mathbf{S}$. Now given any relation $R \in \sigma$, we have, by definition, that the tuple $(\sum_{\mathbf{a} \in R^\mathcal{A}} \gamma_\mathbf{a} \mathbf{a}[1], \ldots, \sum_{\mathbf{a} \in R^\mathcal{A}} \gamma_\mathbf{a} \mathbf{a}[m])$ is in $R^{\mathbb{V}_\mathbf{S}\mathcal{A}}$ for any choice of weightings $\gamma_\mathbf{a}$. So, setting $\gamma_\mathbf{a} = 0_\mathbf{S}$, we have that $(\mathbf{0}, \ldots, \mathbf{0}) \in R^{\mathbb{V}_\mathbf{S}\mathcal{A}}$, for all R. This means that the constant function $f_\mathbf{0} \colon A \to \mathbb{V}_\mathbf{S}B$ which maps all $a \in A$ to $\mathbf{0} \in \mathbb{V}_\mathbf{S}B$ is always a homomorphism.*

For the rest of this section we focus on the definition of another monad $\mathbb{D}_\mathbf{S}$ which avoids the pitfall pointed out in the observation above.

7.2.2 The distribution monad: $\mathbb{D}_\mathbf{S}$

For any semiring $\mathbf{S}$, there is another well-studied monad on **Set** which is a submonad of $\mathbb{V}_\mathbf{S}$. This is the *distribution monad* which restricts the underlying set to

$$\mathbb{D}_\mathbf{S}A := \left\{ \sum_{a \in A} \alpha_a a \in \mathbb{V}_\mathbf{S}A \;\middle|\; \sum_{a \in A} \alpha_a = 1_\mathbf{S} \right\}.$$

It is not hard to see that the functions $\mathbb{D}_{\mathbf{S}}f, \eta_A, \mu_A$ all preserve membership of this subset and so $(\mathbb{D}_{\mathbf{S}}, \eta, \mu)$ defines a monad on **Set**. We now define a relational structure on $\mathbb{D}_{\mathbf{S}}A$ and show that the monad lifts to $\mathcal{R}(\sigma)$.

Definition 7.8. *For a relational structure $\mathcal{A}$ over a finite signature σ and a semiring $\mathbf{S}$, the structure $\mathbb{D}_{\mathbf{S}}\mathcal{A}$ has the underlying set*

$$\left\{ \sum_{a \in A} \alpha_a a \in \mathbb{V}_{\mathbf{S}}A \mid \sum_{a \in A} \alpha_a = 1_{\mathbf{S}} \right\}$$

and relations

$$R^{\mathbb{D}_{\mathbf{S}}\mathcal{A}} := \left\{ \left(\sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}} \mathbf{a}[1], \ldots, \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}} \mathbf{a}[m] \right) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}} \mid \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}} = 1_{\mathbf{S}} \right\}$$

where $R \in \sigma$ has arity m.

We now show that the proof of Proposition 7.6 can be strengthened to show that $\mathbb{D}_{\mathbf{S}}$ also defines a monad on $\mathcal{R}(\sigma)$.

Proposition 7.9. *For any finite relational signature σ the triple $(\mathbb{D}_{\mathbf{S}}, \eta, \mu)$ defines a monad on $\mathcal{R}(\sigma)$.*

Proof. To prove this, we need to show that $\mathbb{D}_{\mathbf{S}}f$, $\eta_{\mathcal{A}}$ and $\mu_{\mathcal{A}}$ are all homomorphisms with respect to the new relational structures defined in Definition 7.8.

This can be done by showing that the functions γ^f, γ^η, and γ^μ, used in the proof of Proposition 7.6, each sum to $1_{\mathbf{S}}$, on the condition that the functions which witness relations in the antecedents of this proof also sum to $1_{\mathbf{S}}$. We check these as follows.

Firstly to show that $\mathbb{D}_{\mathbf{S}}f$ is a homomorphism we recall that for $\theta \in R^{\mathbb{D}_{\mathbf{S}}}$ witnessed by $\gamma \colon R^{\mathcal{A}} \to \mathbf{S}$ we defined

$$\gamma^f_{\mathbf{b}} := \sum_{\substack{\mathbf{a} \in R^{\mathcal{A}} \\ f(\mathbf{a}) = \mathbf{b}}} \gamma_{\mathbf{a}}.$$

We know that $\gamma^f \colon R^{\mathcal{B}} \to \mathbf{S}$ witnesses that $\mathbb{V}_{\mathbf{S}}f(\theta) \in R^{\mathbb{V}_{\mathbf{S}}}$, so it remains to show that $\sum_{\mathbf{b}} \gamma^f_{\mathbf{b}} = 1_{\mathbf{S}}$. To this end, note that we have the following equivalences

$$\begin{aligned} \sum_{\mathbf{b} \in R^{\mathcal{B}}} \gamma^f_{\mathbf{b}} &= \sum_{\mathbf{b} \in R^{\mathcal{B}}} \sum_{\substack{\mathbf{a} \in R^{\mathcal{A}} \\ f(\mathbf{a}) = \mathbf{b}}} \gamma_{\mathbf{a}} \\ &= \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma_{\mathbf{a}}. \end{aligned}$$

This final sum is equal to $1_{\mathbf{S}}$ by the assumption that $\theta \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{A}}$, so we have $\sum \gamma^f_{\mathbf{b}} = 1_{\mathbf{S}}$ and so $\mathbb{V}_{\mathbf{S}}f(\theta) \in R^{\mathbb{D}_{\mathbf{S}}}$, as required.

For $\eta_{\mathcal{A}}$, the proof is easy as $\gamma^{\eta_{\mathcal{A}}}$ is non-zero for exactly one tuple $(a_1, \dots, a_m)$ and takes the value $1_{\mathbf{S}}$ on this.

For $\mu_{\mathcal{A}}$, we assume that we have a function $\gamma\colon R^{\mathbb{D}_{\mathbf{S}}\mathcal{A}} \to \mathbf{S}$ witnessing that $(\phi_1, \dots \phi_m) \in R^{\mathbb{D}_{\mathbf{S}}\mathbb{D}_{\mathbf{S}}\mathcal{A}}$ and for each $\theta \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{A}}$, we have functions $\gamma^\theta\colon R^{\mathcal{A}} \to \mathbf{S}$ witnessing these relations. Now the argument in the proof of Proposition 7.6 shows that $(\mu_{\mathcal{A}}(\phi_1), \dots, \mu_{\mathcal{A}}(\phi_m)) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}$ is witnessed by $\gamma^\mu\colon R^{\mathcal{A}} \to \mathbf{S}$ defined as

$$\gamma^\mu_{\mathbf{a}} := \sum_{\theta \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}} \gamma_\theta \gamma^\theta_{\mathbf{a}}.$$

We now show that $(\mu_{\mathcal{A}}(\phi_1), \dots, \mu_{\mathcal{A}}(\phi_m)) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{A}}$ by showing that $\sum \gamma^\mu_{\mathbf{a}} = 1_{\mathbf{S}}$. This follows by rearranging finite sums as follows.

$$\begin{aligned}
\sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma^\mu_{\mathbf{a}} &= \sum_{\mathbf{a} \in R^{\mathcal{A}}} \sum_{\theta \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}} \gamma_\theta \gamma^\theta_{\mathbf{a}} \\
&= \sum_{\theta \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}} \gamma_\theta \sum_{\mathbf{a} \in R^{\mathcal{A}}} \gamma^\theta_{\mathbf{a}} \\
&= \sum_{\theta \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{A}}} \gamma_\theta (1_{\mathbf{S}}) \\
&= 1_{\mathbf{S}}.
\end{aligned}$$

So we have shown that $\mu_{\mathcal{A}}\colon \mathbb{D}_{\mathbf{S}}\mathbb{D}_{\mathbf{S}}\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{A}$ is a homomorphism and thus the $\mathbf{S}$ distribution monad on **Set** lifts to the $\mathbb{D}_{\mathbf{S}}$ monad on relational structures, completing the proof. □

7.2.3 Distribution constructions in constraint satisfaction

We now see how for certain values of $\mathbf{S}$ the construction $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ has appeared in various forms in the literature of constraint satisfaction problems. In particular, Chen, Dalmau and Grußien [30] considered the case of the Boolean semiring, $\mathbb{B}$ and Barto, Bulín, Krokhin, and Opršal [18] considered cases of the non-negative rationals, $\mathbb{Q}_{\geq 0}$; and the integers, $\mathbb{Z}$. Despite the equivalence of the structures in these works with $\mathbb{D}_{\mathbf{S}}$ neither consider the functorial or monadic nature of the construction.

Boolean semiring and the powerset construction

In [30], Chen, Dalmau and Grußien define a construction called the *power set* construction as follows.

Definition 7.10. *For any relational structure $\mathcal{B}$ over some signature σ, define the relational stucture $\mathcal{P}(\mathcal{B})$ on the set of non-empty subsets of B as follows. For each $R \in \sigma$, define the relation*

$$R^{\mathcal{P}(\mathcal{B})} := \{(\pi_1 S, \dots \pi_m S) \mid S \subset R^{\mathcal{B}}\},$$

where $\pi_i\colon B^m \to B$ is the projection map onto the i^{th} component.

Chen et al. use this construction to investigate the power of the well-known arc consistency algorithm. We define this algorithm and cite some of the applications from [30] later in this chapter. For now, we show that this construction is simply the distribution monad construction given above for $\mathbf{S} = \mathbb{B}$.

Proposition 7.11. *For any relational structure $\mathcal{B}$,*

$$\mathbb{D}_{\mathbb{B}}\mathcal{B} \cong \mathcal{P}(\mathcal{B})$$

Proof. The isomorphism between these two structures is straightforward. For any element $\alpha = \sum_b \alpha_b b \in \mathbb{D}_{\mathbb{B}}\mathcal{B}$, we construct the subset P_α of B as those $b \in B$ such that $\alpha_b = 1_{\mathbb{B}}$. As the semiring $\mathbb{B}$ has only two elements, this map is also invertible by mapping any subset P of B to $\alpha^S = \sum_b \alpha_b^S b$ where

$$\alpha_b^S = \begin{cases} 1_{\mathbb{B}} & \text{if } b \in S \\ 0_{\mathbb{B}} & \text{otherwise.} \end{cases}$$

It is not hard to see that the condition that $\sum_b \alpha_b = 1_{\mathbb{B}}$ is exactly the condition that S_α is non-empty and, using the same correspondence as above between $\mathbb{B}$-valued distributions and subsets, we see that related tuples in $R^{\mathbb{D}_{\mathbb{B}}\mathcal{B}}$ are given projections of by subsets of $R^{\mathcal{B}}$ exactly as in Chen et al.'s definition.

□

Two constructions for equations over rationals and integers

Two other constructions which we show to be cases of the distribution monad appear in the work of Barto, Bulín, Krokhin, and Opršal [18] and are used by those authors to bound the power of the basic linear program and affine integer relaxation, which we noted in Observation 7.3 to be equivalent to the systems of equations $BLP_{\mathbb{Q}_{\geq 0}}$ and $BLP_{\mathbb{Z}}$ respectively. Their results will be useful later on in this chapter. First, we define their constructions and show that they are isomorphic to those given above.

Definition 7.12 (Definition 7.10, [18])**.** *For any finite relational structure $\mathcal{B}$ over the signature σ, the structure $LP(\mathcal{B})$ has as elements the functions $\phi\colon B \to \mathbb{Q} \cap [0,1]$ such that $\sum_{b\in B} \phi(b) = 1$. For any m-ary relation $R \in \sigma$ the relation $R^{LP(\mathcal{B})}$ contains every tuple $(\phi_1, \ldots, \phi_m)$ such that there exists a function $\gamma\colon R^{\mathcal{B}} \to \mathbb{Q} \cap [0,1]$ satisfying*

$$\sum_{\mathbf{b}\in R^{\mathcal{B}}, \mathbf{b}[i]=b} \gamma(\mathbf{b}) = \phi_i(b).$$

In a very similar way Barto et al. also define the following structure.

Definition 7.13 (Definition 7.20, [18]). *For any finite relational structure $\mathcal{B}$ over the signature σ, the structure $IP(\mathcal{B})$ has as elements the functions $\phi\colon B \to \mathbb{Z}$ such that $\sum_{b\in B}\phi(b) = 1$. For any m-ary relation $R \in \sigma$ the relation $R^{IP(\mathcal{B})}$ contains every tuple $(\phi_1, \dots, \phi_m)$ such that there exists a function $\gamma\colon R^{\mathcal{B}} \to \mathbb{Z}$ satisfying*

$$\sum_{\mathbf{b}\in R^{\mathcal{B}}, \mathbf{b}[i]=b} \gamma(\mathbf{b}) = \phi_i(b).$$

It is easy to see that both of these constructions are isomorphic to those given above by the distribution monad.

Proposition 7.14. *For any relational structure $\mathcal{B}$,*

$$\mathbb{D}_{\mathbb{Q}_{\geq 0}}\mathcal{B} \cong LP(\mathcal{B})$$

and

$$\mathbb{D}_{\mathbb{Z}}\mathcal{B} \cong IP(\mathcal{B})$$

Proof. In both of these cases the isomorphism is simple. Indeed, the functions $\phi\colon B \to \mathbf{S}$ which define elements in each case can easily be seen as defining a formal sum $\sum_b \phi(b)b$ in the free left semimodule over B for the semirings $\mathbb{Q}_{\geq 0}$ and $\mathbb{Z}$ respectively. The condition that $\sum_b \phi(b) = 1$ also ensures that these are elements of $\mathbb{D}_{\mathbf{S}}B$. It is straightforward to check that the relation condition is also equivalent to that given in Definition 7.8. □

In their work, Barto et al. use these structures to classify the strength of the basic linear program and affine integer relaxation algorithms in two ways. Firstly they show that for any structures $\mathcal{A}$ and $\mathcal{B}$, the equations $BLP_{\mathbb{Q}_{\geq 0}}(\mathcal{A}, \mathcal{B})$ and $BLP_{\mathbb{Z}}(\mathcal{A}, \mathcal{B})$ are satisfiable if, and only if, there are homomorphism of the types $\mathcal{A} \to LP(\mathcal{B})$ and $\mathcal{A} \to IP(\mathcal{B})$, respectively. Secondly, they show that homomorphism of the type $LP(\mathcal{B}) \to \mathcal{B}$ and $IP(\mathcal{B}) \to \mathcal{B}$, characterise exactly the polymorphisms required by $\mathcal{B}$ to ensure that the respective linear programming approximations are sufficient to solve all constraint satisfaction problems with domain $\mathcal{B}$.

In the next two sections, we show that such results hold in general for the monads $\mathbb{D}_{\mathbf{S}}$ and that these are neat analogues to the Power and Structure theorems we saw in the case of monads.

7.3 Kleisli category of $\mathbb{D}_{\mathbf{S}}$

In this section, we prove Theorem 7.16, an important relationship between linear programming relaxations of homomorphisms from Section 7.1 and the monad $\mathbb{D}_{\mathbf{S}}$ from Section 7.2. We call this result the Morphism Power Theorem for $\mathbb{D}_{\mathbf{S}}$ as it establishes the power of the approximation to homomorphism provided by the Kleisli maps of $\mathbb{D}_{\mathbf{S}}$ in a way which is comparable to the similar results in Theorems 3.13 and 6.13 for the comonads $\mathbb{P}_k$ and $\mathbb{H}_{n,k}$.

The notion of an I^*-structure is needed to make the second correspondence work for reasons similar to those that lead to the introduction of I-structures for game comonads.

To state the main theorem of this section, we first need to define the notion of the I^*-structure for a relational structure $\mathcal{A}$. This is related to Definition 3.4 which introduced I-structures and the reasons for introducing this notion are similar to those in Chapter 3. There, I-structures were introduced in order to overcome the fact that if, for the game comonad $\mathbb{P}_k$, an arbitrary Kleisli morphism $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$ was translated into a Duplicator strategy for the k-pebble game, it could result in positions in the game which preserve all relations except the identity relation. In other words, positions which fail to be partial homomorphisms simply because they fail to be functions.

In translating from morphisms $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ to solutions to linear programs $LP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ we have a similar concern. In particular, for any related tuple $\mathbf{a} \in R^{\mathcal{A}}$, we have that the image $f(\mathbf{a})$ is a related tuple in $\mathbb{D}_{\mathbf{S}}\mathcal{B}$. This means, by the definition of the relations on $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ that $f(\mathbf{a})$ can be written as a weighted sum over related tuples in $\mathcal{B}$. However if there is some $i \neq j$ such that $\mathbf{a}[i] = \mathbf{a}[j]$, there is no way to guarantee that the sum of related tuples which witnesses $f(\mathbf{a}) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$ uses only tuples $\mathbf{b} \in R^{\mathcal{B}}$ such that $\mathbf{b}[i] = \mathbf{b}[j]$. However to satisfy the linear program $LP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$, it is required by ($LP_{\mathbf{S}}$.3) that the variable $x_{\mathbf{a},\mathbf{b}}$ is non-zero only if $\mathbf{b}$ is the image $f(\mathbf{a})$ of some function f. Thus any such $\mathbf{b}$ must satisfy the equality relations satisfied by $\mathbf{a}$.

For this reason, we define a new relational signature σ^* as follows. Write Π_k for the set of equivalence relations on the elements of the set $[k]$ and then for every $R \in \sigma$ and $\pi \in \Pi_{ar(R)}$, σ^* contains a relational symbol R_π with arity $ar(R)$. This allows us to define I^*-structures as follows.

Definition 7.15. *For any relational structure $\mathcal{A}$ over the signature σ* the I^*-structure $\mathcal{A}$ *is the unique relational structure over the signature σ^* which has the same underlying set A and, for each $R \in \sigma$ and $\pi \in \Pi_{ar(R)}$,*

$$R_\pi^{\mathcal{A}} = \left\{\mathbf{a} \in R^{\mathcal{A}} \;\mid\; \forall i, j \in [ar(R)].i\pi j \implies \mathbf{a}[i] = \mathbf{a}[j]\right\}.$$

We can now state the main theorem of this section.

Theorem 7.16 (Morphism Power Theorem for $\mathbb{D}_{\mathbf{S}}$)**.** *For any two relational structures $\mathcal{A}$ and $\mathcal{B}$ and a semiring $\mathbf{S}$ the following are equivalent:*

1. *There is a Kleisli morphism $\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$.*

2. *The linear program $BLP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ has a solution.*

Separately, we have that the following are equivalent:

1. *There is a Kleisli morphism $\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ for the I^*-structures $\mathcal{A}$ and $\mathcal{B}$.*

2. *The linear program $LP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ has a solution.*

This relationship encourages us to ask whether there is a matching notion of an Isomorphism Power Theorem for $\mathbb{D}_{\mathbf{S}}$, relating isomorphisms in $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$ to linear relaxations of isomorphism. We show in Section 7.3.3 that this hope is misleading for certain choices of $\mathbf{S}$ and gives rise to open problems in other cases.

We now prove Theorem 7.16 and we apply it to specific cases for $\mathbf{S}$.

7.3.1 Proof of Theorem 7.16

Proof of Theorem 7.16. The first equivalence is relatively straightforward. Given a homomorphism $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$, for each $a \in A$ write $f(a) := \sum_{b\in B} \alpha_{a,b} b$.

Then $f(a) \in \mathbb{D}_{\mathbf{S}}\mathcal{B}$ implies that setting $x_{a,b} := \alpha_{a,b}$, for each $a \in A$ and $b \in B$, satisfies the equation $LP_{\mathbf{S}}.1$. Furthermore, for every related tuple $\mathbf{a} \in R^{\mathcal{A}}$ there is function $\gamma^{\mathbf{a}}\colon R^{\mathcal{B}} \to \mathbf{S}$ witnessing that $f(\mathbf{a}) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$. Letting $x^R_{\mathbf{a},\mathbf{b}} := \gamma^{\mathbf{a}}_{\mathbf{b}}$ for $\mathbf{b} \in R^{\mathcal{B}}$ and setting it to $0_{\mathbf{S}}$ otherwise, we satisfy $LP_{\mathbf{S}}.2$ by definition. Finally, we see that $BLP_{\mathbf{S}}.3$ is satisfied because for each $\mathbf{a} \in R^{\mathcal{A}}$, $i \in [ar(R)]$ and $b \in B$ we have that

$$\sum_{\mathbf{b}[i]=b} x^R_{\mathbf{a},\mathbf{b}} = \sum_{\mathbf{b}[i]=b} \gamma^{\mathbf{a}}_{\mathbf{b}},$$

and because $\gamma^{\mathbf{a}}$ witnesses that $f(\mathbf{a}) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$, we must have that

$$\sum_{\mathbf{b}[i]=b} \gamma^{\mathbf{a}}_{\mathbf{b}} = \alpha_{\mathbf{a}[i],b}$$

which is the coefficient of b in $f(\mathbf{a}[i])$ and, by definition, the value of $x_{\mathbf{a}[i],b}$. This completes the forward direction of the first part.

For the other direction we want to use a solution to $BLP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ given as $(x_{a,b})_{a\in A, b\in B}$ and $(x^R_{\mathbf{a},\mathbf{b}})_{\mathbf{a}\in R^{\mathcal{A}}, \mathbf{b}\in B^{ar(R)}}$ to construct a homomorphism $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$. We claim that the function f which sends each $a \in A$ to

$$f(a) := \sum_{b\in B} x_{a,b} b$$

is exactly such a homomorphism. Firstly the range of this function is contained in $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ because the $x_{a,b}$ satisfy $LP_{\mathbf{S}}.1$ and so $\sum_b x_{a,b} = 1_{\mathbf{S}}$ and so $f(a) \in \mathbb{D}_{\mathbf{S}}\mathcal{B}$. To show that f is a homomorphism we take any $\mathbf{a} \in R^{\mathcal{A}}$ and show that $f(\mathbf{a})$ is in $R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$. To do this we claim that $(x^R_{\mathbf{a},\mathbf{b}})_{\mathbf{b}\in R^{\mathcal{B}}}$ witnesses this. Indeed, as $LP_{\mathbf{S}}.2$ implies that $x_{\mathbf{a},\mathbf{b}} = 0_{\mathbf{S}}$ whenever $\mathbf{b} \notin R^{\mathcal{B}}$, we can rephrase $BLP_{\mathbf{S}}.3$ as saying that for each $\mathbf{a} \in R^{\mathcal{A}}$, $i \in [ar(R)]$ and $b \in B$,

$$\sum_{\substack{\mathbf{b}\in R^{\mathcal{B}} \\ \mathbf{b}[i]=b}} x_{\mathbf{a},\mathbf{b}} = x_{\mathbf{a}[i],b}.$$

This means that for each i,

$$\sum_{\mathbf{b}\in R^{\mathcal{B}}} x^R_{\mathbf{a},\mathbf{b}}\mathbf{b}[i] = f(\mathbf{a}[i])$$

and so we have witnessed that $f(\mathbf{a}) \in R^{\mathbb{V}_{\mathbf{S}}\mathcal{B}}$. To finish the proof that f is a homomorphism of the desired type, we must show that for each $\mathbf{a} \in R^{\mathcal{A}}$, $\sum_{\mathbf{b}} x^R_{\mathbf{a},\mathbf{b}} = 1_{\mathbf{S}}$. This can be derived from $BLP_{\mathbf{S}}.3$ and $LP_{\mathbf{S}}.1$ as follows.

$$\begin{aligned}\sum_{\mathbf{b}\in R^{\mathcal{B}}} x_{\mathbf{a},\mathbf{b}} &= \sum_{b\in B} \sum_{\substack{\mathbf{b}\in R^{\mathcal{B}} \\ \mathbf{b}[1]=b}} x^R_{\mathbf{a},\mathbf{b}} \\ &= \sum_{b\in B} x_{\mathbf{a}[1],b} \\ &= 1_{\mathbf{S}}\end{aligned}$$

For the second part, we have the same maps between solutions of the equations and homomorphisms but we additionally have to show that satisfying the equational condition $LP_{\mathbf{S}}.3$ corresponds exactly to the relational condition of preserving the I^* structures.

Firstly, suppose we have a homomorphism $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ between I^*-structures. As before, we define the values $x_{a,b}$ by the coefficients $\alpha_{a,b}$ in $f(a)$ and this satisfies $LP_{\mathbf{S}}.1$. Now we need to make assignments to the variables $x^R_{\mathbf{a},\mathbf{b}}$ for $\mathbf{a} \in R^{\mathcal{A}}$ in the original structure $\mathcal{A}$ and $\mathbf{b} \in B^{ar(R)}$. Given such an $\mathbf{a}$, let $\pi_{\mathbf{a}}$ be the equivalence relation on $[ar(R)]$ such that $i\pi_{\mathbf{a}}j \iff \mathbf{a}[i] = \mathbf{a}[j]$. By definition, we know that $\mathbf{a} \in R^{\mathcal{A}}_{\pi_{\mathbf{a}}}$ in the I^*-structure $\mathcal{A}$. As f is a homomorphism, we have $f(\mathbf{a}) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}_{\pi_{\mathbf{a}}}$. This is witnessed by some function $\gamma^{\mathbf{a}}\colon R^{\mathcal{B}}_{\pi_{\mathbf{a}}} \to \mathbf{S}$ such that for each $i \in [ar(M)]$

$$f(\mathbf{a}[i]) = \sum_{\mathbf{b}\in R^{\mathcal{B}}_{\pi_{\mathbf{a}}}} \gamma^{\mathbf{a}}_{\mathbf{b}}\mathbf{b}[i].$$

Setting $x^R_{\mathbf{a},\mathbf{b}}$ to be $\gamma^{\mathbf{a}}_{\mathbf{b}}$ where defined above and $0_{\mathbf{S}}$ elsewhere. Then focusing on the coefficient of any $b \in B$, the above equation yields:

$$x_{\mathbf{a}[i],b} = \sum_{\substack{\mathbf{b}\in R^{\mathcal{B}}_{\pi_{\mathbf{a}}} \\ \mathbf{b}[i]=b}} x^R_{\mathbf{a},\mathbf{b}}.$$

However, it is easy to see from the conditions imposed by $\pi_{\mathbf{a}}$ that each $\mathbf{b} \in R^{\mathcal{B}}_{\pi_{\mathbf{a}}}$ is uniquely identified as the image $g(\mathbf{a})$ of some function $g\colon \{\mathbf{a}\} \to B$ so the above equation can be rewritten finally as, for each $a \in \{\mathbf{a}\}$ and $b \in B$:

$$x_{a,b} = \sum_{\substack{g\colon \{\mathbf{a}\}\to B \\ g(a)=b}} x^R_{\mathbf{a},g(\mathbf{a})}.$$

$\mathbf{S}$	$\mathbb{B}$	$\mathbb{Q}_{\geq 0}$	$\mathbb{Z}$
$\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$	AC	BLP	AIP

Table 7.1: Kleisli morphisms for the monad $\mathbb{D}_{\mathbf{S}}$ capture the expressive power of different algorithms for CSP as we vary $\mathbf{S}$.

So this assignment to the x variables satisfies $LP_{\mathbf{S}}.3$.

To go the other direction, we define the homomorphism from $\mathcal{A}$ to $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ as in the last part by setting $f(a) := \sum_b x_{a,b}b$. We now need to show that satisfying $LP_{\mathbf{S}}.3$ for each $\mathbf{a} \in R^{\mathcal{A}}$ is enough to show that f is a homomorphism between I^*-structures. To do this we need to show that for any $\mathbf{a} \in R_\pi^{\mathcal{A}}$ we have $f(\mathbf{a}) \in R_\pi^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$. For every function $g\colon \{\mathbf{a}\} \to B$, define the weighting $\gamma^{\mathbf{a}}_{g(\mathbf{a})}$ to be $x_{\mathbf{a},g(\mathbf{a})}$ as given in the solution to $LP_{\mathbf{S}}(\mathcal{A},\mathcal{B})$. As any function g preserves all equalities mandated by π and the assignment to the x variables satisfies $LP_{\mathbf{S}}.2$, we have that $\gamma^{\mathbf{a}}_{\mathbf{b}} \neq 0_{\mathbf{S}}$ only if $\mathbf{b} \in R_\pi^{\mathcal{B}}$. Now we can see that the function $\gamma^{\mathbf{a}}\colon R_\pi^{\mathcal{B}} \to \mathbf{S}$ witness $f(\mathbf{a}) \in R^{\mathbb{D}_{\mathbf{S}}\mathcal{B}}$ by deducing from $LP_{\mathbf{S}}.3$ that for each i

$$f(\mathbf{a}[i]) = \sum_{\mathbf{b} \in R_\pi^{\mathcal{B}}} \gamma^{\mathbf{a}}_{\mathbf{b}}\mathbf{b}[i]$$

and noting that, as before, $\sum_{\mathbf{b}} \gamma^{\mathbf{a}}_{\mathbf{b}} = 1_{\mathbf{S}}$.

$\square$

7.3.2 Algorithms and Kleisli morphisms

Theorem 7.16 shows that the monadic constructions given in the last section capture the approximations to homomorphism given by the linear equations in Section 7.1. While in general these systems of equations do not admit efficient solutions, for certain semirings $\mathbf{S}$ these maps capture well-known `PTIME` algorithms for approximating homomorphism. Here, we briefly recall some of these connections which have been studied for the the distribution constructions reviewed in Section 7.2.3. The relations between Kleisli maps and algorithms are summarised in Table 7.1.

For $\mathbf{S} = \mathbb{Q}_{\geq 0}$ and $\mathbf{S} = \mathbb{Z}$, we recall from Observation 7.3 that the systems of equations $BLP_{\mathbf{S}}$ are more commonly known as the basic linear program and affine integer relaxation for the homomorphism problem. These are used extensively in work on promise constraint satisfaction, for example in [23]. In [18], Barto et al. prove a version of Theorem 7.16 in these cases by showing that the existence of a homomorphism of the form $\mathcal{A} \to LP(\mathcal{B})$ or $\mathcal{A} \to IP(\mathcal{B})$ corresponds to the acceptance of the pair $(\mathcal{A},\mathcal{B})$ by the BLP or AIP algorithm respectively.

In the case of the boolean semiring, we can characterise the Kleisli morphisms in terms of an algorithm which doesn't immediately look like a system of linear equations. The *arc consistency* (AC) algorithm has been studied since Feder and Vardi's seminal work on constraint satisfaction [44] where it was called the $(1, k)$-consistency algorithm with k standing for the arity of the underlying signature. We give a definition of this algorithm here due to Chen et al. [30].

Definition 7.17. *Given two finite relational structures $\mathcal{A}$ and $\mathcal{B}$, we define the* arc consistency *algorithm on the pair $(\mathcal{A}, \mathcal{B})$ as follows:*

1. *For each $a \in A$ set $S_a := B$.*
2. *While there exists some $R \in \sigma$ and $(a_1, \ldots, a_m) \in R^{\mathcal{A}}$ and some j such that*
$$S_{a_j} \neq \mathbf{proj}_j \left(R^{\mathcal{B}} \cap (S_{a_1} \times \ldots \times S_{a_m}) \right),$$
modify S_{a_j} by setting it to $S'_{a_j} := \mathbf{proj}_j(R^{\mathcal{B}} \cap (S_{a_1} \times \ldots \times S_{a_m}))$.
3. *On exiting the loop in Step 2,*
   - *If there is any $S_a = \emptyset$, **reject** $(\mathcal{A}, \mathcal{B})$.*
   - *Otherwise, **accept**.*

Also in [30], Chen, Dalmau and Grussien show that the acceptance of a given pair $(\mathcal{A}, \mathcal{B})$ by this algorithm is equivalent to the existence of a homomorphism of the form $\mathcal{A} \to \mathcal{P}(\mathcal{B})$, where $\mathcal{P}(\mathcal{B})$ is the power set construction given in Section 7.2.3. As this construction is isomorphic to $\mathbb{D}_{\mathbb{B}}\mathcal{B}$ by Proposition 7.11, this gives an alternate characterisation of the Kleisli morphisms in this case and, additionally, via Theorem 7.16, shows that the arc consistency algorithm is equivalent to solving the basic linear program over the semiring $\mathbb{B}$.

7.3.3 Isomorphisms in $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$

Having comprehensively classified the morphisms in $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$, it is natural to ask whether an equally interesting classification exists for the isomorphisms. We saw for example in Theorems 3.14 and 6.14 how isomorphisms of the Kleisli categories of $\mathbb{P}_k$ and $\mathbb{H}_{n,k}$ capture winning strategies in appropriate bijective games.

The first result of this section shows that for two of the semirings that we are interested in above, namely $\mathbb{B}$ and $\mathbb{Q}_{\geq 0}$, there is no additional merit in considering Kleisli isomorphism of $\mathbb{D}_{\mathbf{S}}$. Indeed, in these cases, this notion can be seen to be equivalent to the normal isomorphism of structures as stated in the following result.

Proposition 7.18. *For any semiring $\mathbf{S} = (S, +, \cdot, 0_{\mathbf{S}}, 1_{\mathbf{S}})$, where the non-zero elements of $\mathbf{S}$ are closed under both $+$ and $\cdot$, we have that for any structures $\mathcal{A}$ and $\mathcal{B}$*
$$\mathcal{A} \cong_{\mathcal{K}(\mathbb{D}_{\mathbf{S}})} \mathcal{B} \iff \mathcal{A} \cong \mathcal{B}.$$

Proof. The backwards direction is trivial as any isomorphism lifts to being a Kleisli isomorphism.

For the other direction, let $f\colon \mathcal{A} \to \mathbb{D}_\mathbf{S}\mathcal{B}$ and $g\colon \mathcal{B} \to \mathbb{D}_\mathbf{S}\mathcal{A}$ be inverses in the Kleisli category of $\mathbb{D}_\mathbf{S}$ witnessing that $\mathcal{A} \cong_{\mathcal{K}(\mathbb{D}_\mathbf{S})} \mathcal{B}$. Writing $f(a) = \sum_b \alpha_{a,b} b$ and $g(b) = \sum_a \beta_{b,a} a$, we have that the composite functions $g^* \circ f$ and $f^* \circ g$ act as

$$g^* \circ f(a) = \sum_{a' \in A} \sum_{b \in B} \alpha_{a,b}\beta_{b,a'} a'$$

and

$$f^* \circ g(b) = \sum_{b' \in B} \sum_{a \in A} \alpha_{a,b'}\beta_{b,a} b'.$$

Now, as f and g are inverse to each other $\mathcal{K}(\mathbb{D}_\mathbf{S})$ we must have that $g^* \circ f(a) = a$ for all $a \in A$ and $f^* \circ g(b) = b$ for all $b \in B$. So the above equations imply that for any $a, a' \in A$,

$$\sum_{b \in B} \alpha_{a,b}\beta_{b,a'} = \begin{cases} 1_\mathbf{S} & \text{if } a = a' \\ 0_\mathbf{S} & \text{otherwise} \end{cases}$$

and for any $b, b' \in B$,

$$\sum_{a \in A} \alpha_{a,b'}\beta_{b,a} = \begin{cases} 1_\mathbf{S} & \text{if } b = b' \\ 0_\mathbf{S} & \text{otherwise.} \end{cases}$$

Closure of the nonzero elements of $\mathbf{S}$ under $+$ and $\cdot$ means that we have, for all $a \in A.b \in B$,

$$\alpha_{a,b} \neq 0_\mathbf{S} \implies \forall a' \in A \;\; \beta_{b,a'} = \begin{cases} 1_\mathbf{S} & \text{if } a = a' \\ 0_\mathbf{S} & \text{otherwise} \end{cases} \qquad \implies g(b) = a$$

and

$$\beta_{b,a} \neq 0_\mathbf{S} \implies \forall b' \in B \;\; \alpha_{a,b'} = \begin{cases} 1_\mathbf{S} & \text{if } b = b' \\ 0_\mathbf{S} & \text{otherwise} \end{cases} \qquad \implies f(a) = b.$$

Linking these two implications together we see that f and g are simply morphisms between $\mathcal{A}$ and $\mathcal{B}$ which are inverse to each other and so $\mathcal{A} \cong \mathcal{B}$.

□

This result has established that Kleisli isomorphisms for $\mathbb{D}_\mathbf{S}$ are essentially uninteresting when the conditions of the proposition hold. There are however many semirings where these conditions don't hold. In particular, any ring $\mathbf{S}$ has additive inverses for all elements and so non-zero elements are not closed under $+$.

In the rest of the section, we lay the groundwork for future investigations of Kleisli isomorphisms in $\mathcal{K}(\mathbb{D}_\mathbf{S})$ beyond the reach of Proposition 7.18. To do this, we first prove Proposition 7.20 which states that the composition of maps in $\mathcal{K}(\mathbb{D}_\mathbf{S})$ can be seen as the multiplication of left stochastic matrices with entries in $\mathbf{S}$. This is not a novel result as such matrices are exactly the morphisms in the category of $\mathbf{S}$-distribution, see for example [61] for the $\mathbf{S} = \mathbb{Q}_{\geq 0}$ case. However, we present the details here to aid the understanding of this new structure. We start by defining the matrix for any Kleisli map.

Definition 7.19. *For any Kleisli map $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$, which we write as $f(a) = \sum_b \alpha_{a,b} b$ for each $a \in A$, define the $B \times A$ matrix M^f where the rows and columns are indexed by the elements of B and A respectively, as*

$$M^f_{b,a} = \alpha_{a,b}.$$

Due to the condition that $\sum_b \alpha_{a,b} = 1_{\mathbf{S}}$ for every $a \in A$, we have that the matrix M^f is left stochastic, *meaning that the row sums are all equal to $1_{\mathbf{S}}$.*

Given such a $B \times A$ left stochastic matrix M, we can also define the unique function $f_M\colon A \to \mathbb{D}_{\mathbf{S}}B$ represented by M as

$$f(a) = \sum_{b \in B} M_{b,a} b.$$

As this definition makes clear there is one-to-one correspondence between functions of the type $A \to \mathbb{D}_{\mathbf{S}}B$ and left stochastic $B \times A$ matrices over $\mathbf{S}$. From now on we use such a function and its related matrix interchangeably. We now show that see that composition of maps in $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$ corresponds to matrix multiplication of these related matrices.

Proposition 7.20. *Given two Kleisli morphisms $f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ and $g\colon \mathcal{B} \to \mathbb{D}_{\mathbf{S}}\mathcal{C}$ we have that, writing the composition in the Kleisli category as $g^* \circ f\colon \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{C}$,*

$$M_{g^* \circ f} = M_g \cdot M_f.$$

Proof. First write the functions f and g as

$$f(a) = \sum_{b \in B} \alpha_{a,b} b$$

for each $a \in A$ and

$$g(b) = \sum_{c \in C} \beta_{b,c} c$$

for each $b \in B$. Then we have that the composition $g^* \circ f$, where the Kleisli extension g^* is defined from the monad structure as $\mu_{\mathcal{C}} \circ \mathbb{D}_{\mathbf{S}} g$, is given by

$$\begin{aligned} g^* \circ f(a) &= \sum_{b \in B} \alpha_{a,b} f(b) \\ &= \sum_{b \in B} \alpha_{a,b} \sum_{c \in C} \beta_{b,c} c \\ &= \sum_{c \in C} \sum_{b \in B} \alpha_{a,b} \beta_{b,c} c. \end{aligned}$$

So the matrix for $g^* \circ f$, is

$$(M_{g^* \circ f})_{c,a} := \sum_{b \in B} \alpha_{a,b}\beta_{b,c}$$

and so, $M_{g^* \circ f} = M_g \cdot M_f$, as required.

□

An easy consequence of this lemma is that Kleisli isomorphisms in $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$ are witnessed by pairs M, N of left stochastic matrices over $\mathbf{S}$ which are inverses of each other. So the proof of Proposition 7.18 can be seen as showing that any pair of left stochastic inverse matrices over a semiring $\mathbf{S}$ satisfying the conditions of the proposition must be permutation matrices.

The following example shows that for some semirings $\mathbf{S}$ there are inverse pairs of stochastic matrices which are not permutation matrices.

Example 7.21. *Over the ring of integers, $\mathbb{Z}$ the following is a pair of left stochastic matrices which are inverse to one another but are not permutation matrices:*

$$\begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix}$$

The existence of such examples proves that it would be possible for there to be an isomorphism of structures in a category $\mathcal{K}(\mathbb{D}_{\mathbf{S}})$ for some $\mathbf{S}$ between two non-isomorphic relational structures. Whether such a pair does in fact exist remains an open question for future reseach.

7.4 Algebras for $\mathbb{D}_{\mathbf{S}}$

Earlier in this thesis, we saw that the coalgebras of game comonads such as $\mathbb{P}_k$ and $\mathbb{H}_{n,k}$ corresponded to interesting structural decompositions, as captured in the so-called Structure theorems (Theorem 3.16 & 6.19). Thus, it is natural to ask what happens in the dual picture for the new monadic constructions in this chapter. This means investigating the algebras of the form $\phi\colon \mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ and their relationship to the structure of $\mathcal{B}$.

We approach this in two ways. First, we show that the existence of a simple homomorphism $f\colon \mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ is enough to prove that all constraint satisfaction problems over the domain $\mathcal{B}$ can be solved by $BLP_{\mathbf{S}}$. This result extends known results of Barto et al. [18] in the case of BLP and AIP. Further to this, we see that homomorphisms of this type correspond to families of polymorphisms of the underlying structures. Secondly, we ask whether the condition of being an algebra gives a strictly stronger notion of structure on $\mathcal{B}$ and answer this positively in the case of $\mathbb{B}$, by showing that the $\mathbb{D}_{\mathbb{B}}$-algebras correspond to semilattice operations.

7.4.1 Homomorphisms and polymorphisms

Here we show that homomorphisms of the form $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ are exactly the structural condition required to ensure that constraint satisfaction problems over the domain $\mathcal{B}$ are solvable if and only if the equations $BLP_{\mathbf{S}}$ are solvable.

Recall that we write $CSP(\mathcal{B})$ for the set of finite structures $\mathcal{A}$ such that $\mathcal{A} \to \mathcal{B}$. We say that $BLP_{\mathbf{S}}$ *decides* $CSP(\mathcal{B})$ when $\mathcal{A} \in CSP(\mathcal{B})$ if, and only if, there is a solution to the system of equations $BLP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$. This allows us to state the following result which is easy to prove but surprisingly effective.

Theorem 7.22. *For any* $\mathbf{S}$ *and any finite structure* $\mathcal{B}$*, we have that* $BLP_{\mathbf{S}}$ *decides* $CSP(\mathcal{B})$ *if, and only if, there is a homomorphism of the form* $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$*.*

Proof. By Theorem 7.16, we know that $BLP_{\mathbf{S}}(\mathcal{A}, \mathcal{B})$ has a solution if and only if there is a homomorphism $\mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$. This means that the statement that $BLP_{\mathbf{S}}$ decides $CSP(\mathcal{B})$ is equivalent to saying that, for any structure $\mathcal{A}$,

$$\mathcal{A} \to \mathcal{B} \iff \mathcal{A} \to \mathbb{D}_{\mathbf{S}}\mathcal{B},$$

i.e. that $CSP(\mathcal{B}) = CSP(\mathbb{D}_{\mathbf{S}}\mathcal{B})$.

We now show that this is equivalent to $\mathcal{B} \rightleftarrows \mathbb{D}_{\mathbf{S}}\mathcal{B}$, in the case where $\mathbf{S}$ is finite. The forward direction follows from the existence of identity maps $\mathcal{B} \to \mathcal{B}$ and $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$. Applying the equality $CSP(\mathcal{B}) = CSP(\mathbb{D}_{\mathbf{S}}\mathcal{B})$ to these gives that $\mathcal{B} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$ and $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$, as required. The other direction is gotten by composition. Suppose we have $\mathcal{C} \in CSP(\mathcal{B})$ and $\mathcal{D} \in CSP(\mathbb{D}_{\mathbf{S}}\mathcal{B})$ then post composing the respective homomorphisms with the homomorphic equivalence $\mathcal{B} \rightleftarrows \mathbb{D}_{\mathbf{S}}\mathcal{B}$ we have that

$$\mathcal{C} \to \mathcal{B} \to \mathbb{D}_{\mathbf{S}}\mathcal{B} \text{ and } \mathcal{D} \to \mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B},$$

as required.

Finally, as we always have the homomorphism $\eta_{\mathcal{B}} \colon \mathcal{B} \to \mathbb{D}_{\mathbf{S}}\mathcal{B}$, the condition that $\mathcal{B}$ and $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ are homomorphically equivalent reduces to the existence of a homomorphism of the form $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$.

When $\mathbf{S}$ is infinite, the forward argument above fails as $\mathbb{D}_{\mathbf{S}}\mathcal{B}$ is an infinite structure and so is not in $CSP(\mathbb{D}_{\mathbf{S}}\mathcal{B})$ as defined. However, we get around this by observing that for any finite substructure $\mathcal{C} \hookrightarrow \mathbb{D}_{\mathbf{S}}\mathcal{B}$ we have that $\mathcal{C} \in CSP(\mathbb{D}_{\mathbf{S}}\mathcal{B})$ and so if $BLP_{\mathbf{S}}$ solves $CSP(\mathcal{B})$, we have that $\mathcal{C} \to \mathcal{B}$. So, by compactness of $\mathcal{B}$ we have that $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$. $\square$

For the rings $\mathbb{B}$, $\mathbb{Q}_{\geq 0}$ and $\mathbb{Z}$, we can now show that this theorem recovers some previous results on the power of the respective CSP algorithms.

For $\mathbb{B}$, Chen et al. show that a homomorphism of the form $\mathcal{P}(\mathcal{B}) \to \mathcal{B}$ is equivalent to the existence of totally symmetric polymorphisms on $\mathcal{B}$ of all arities, where a polymorphism

f is totally symmetric if $f(x_1, \ldots, x_n)$ relies only on the set $\{x_1, \ldots, x_n\}$. Recalling the isomorphism between $\mathcal{P}(\mathcal{B})$ and $\mathbb{V}^*_{\mathbb{B}}\mathcal{B}$ from Proposition 7.11, this recovers an important characterisation of *width* 1 structures as a consequence of Theorem 7.22. From Feder and Vardi [44], a structure $\mathcal{B}$ is said to have width 1 if the arc consistency algorithm decides $CSP(\mathcal{B})$.

Corollary 7.23. *For a finite relational structure $\mathcal{B}$, the following are equivalent:*

- *$\mathcal{B}$ has width* 1
- *$BLP_{\mathbb{B}}$ decides $CSP(\mathcal{B})$*
- *$\mathcal{B}$ admits totally symmetric polymorphisms of all arities*
- *There exists a homomorphism $\mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$*

For the $\mathbb{Q}_{\geq 0}$ and $\mathbb{Z}$, recalling the isomorphism in Section 7.2.3, Barto et al. [18] show that the existence of morphisms of the form $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ can be characterised in terms of polymorphisms. They show, in particular, that there is a homomorphism $\mathbb{D}_{\mathbb{Q}_{\geq 0}}\mathcal{B} \to \mathcal{B}$ if, and only if, $\mathcal{B}$ admits *symmetric* polymorphism of every arity, where a polymorphism f is symmetric if for any permutation $\pi \in S_n$, $f(x_1, \ldots, x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$. They also show that there is a homomorphism $\mathbb{D}_{\mathbb{Z}}\mathcal{B} \to \mathcal{B}$ if, and only if, $\mathcal{B}$ admits an *alternating* polymorphism of every odd arity. A polymorphism is alternating if for every parity-preserving permutation $\pi \in S_n$, $f(x_1, \ldots x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$ and for any z and w, $f(x_1, \ldots, z, z) = f(x_1, \ldots, w, w)$. This gives us the following corollaries of Theorem 7.22 which, in fact, also follow from Barto et al.'s Theorems 7.9 and 7.19.

Corollary 7.24. *For a finite relational structure $\mathcal{B}$, the following are equivalent:*

- *$BLP_{\mathbb{Q}_{\geq 0}}$ decides $CSP(\mathcal{B})$*
- *$\mathcal{B}$ admits symmetric polymorphisms of all arities*
- *There exists a homomorphism $\mathbb{D}_{\mathbb{Q}_{\geq 0}}\mathcal{B} \to \mathcal{B}$*

Corollary 7.25. *For a finite relational structure $\mathcal{B}$, the following are equivalent:*

- *$BLP_{\mathbb{Z}}$ decides $CSP(\mathcal{B})$*
- *$\mathcal{B}$ admits alternating polymorphisms of all odd arities*
- *There exists a homomorphism $\mathbb{D}_{\mathbb{Z}}\mathcal{B} \to \mathcal{B}$*

We now show that requiring homomorphism $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$ to be algebras is a strictly stronger condition on $\mathcal{B}$.

7.4.2 Algebras and operations

We end this section by showing that for $\mathbb{B}$, the condition of a homomorphism $\phi\colon \mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$ being an algebra results in $\mathcal{B}$ admitting a more restrictive type of polymorphism than those given in Corollary 7.23. Indeed, we show the following result relating $\mathbb{D}_{\mathbb{B}}$-algebras and semilattice operations.

Proposition 7.26. *For any relational structure $\mathcal{B}$, there exists an algebra $\phi\colon \mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$ if, and only if, $\mathcal{B}$ admits a semilattice operation.*

A semilattice operation on $\mathcal{B}$ is a binary polymorphism $\wedge\colon B^2 \to B$ which is associative, commutative and idempotent, in the sense that it satisfies the following three equations for any $x, y, z \in B$:

- $\wedge(x, \wedge(y, z)) = \wedge(\wedge(x, y), z)$
- $\wedge(x, y) = \wedge(y, x)$
- $\wedge(x, x) = x$.

These have been studied in the theory of constraint satisfaction at least since the work of Jeavons, Cohen and Gyssens [63], who aptly called them ACI (associative, commutative and idempotent) operations and it is known that there are structures which have width 1 which do not admit such an operation. See, Example 1.6.2 from the notes of Zarathustra Brady [22]. This establishes the desired separation between algebras of the form $\mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$ and homomorphisms of the same type. We now prove the result.

Proof of Proposition 7.26. Recall that an algebra of the $\mathbb{D}_{\mathbb{B}}$ monad is a homomorphism $\phi\colon \mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$ which satisfies the following two rules which we call the identity law

$$\forall x \in B \quad \phi \circ \eta_{\mathcal{B}}(x) = x$$

and associativity law

$$\forall s \in \mathbb{D}_{\mathbb{B}}\mathbb{D}_{\mathbb{B}}B \quad \phi \circ \mathbb{D}_{\mathbb{B}}\phi = \phi \circ \mu_{\mathcal{B}},$$

where η and μ are the unit and multiplication of the monad.

Using the set notation for elements of $\mathbb{D}_{\mathbb{B}}\mathcal{B}$ which is permitted by Propostion 7.11, the rules above state precisely that for all x $\phi(\{x\}) = x$ and for all non empty subsets $P_1, \dots P_m \subset B$,

$$\phi(\bigcup_i P_i) = \phi(\{\phi(P_i) \mid i \in [m]\}).$$

Given such an algebra we define a binary operation $\wedge_\phi$ by $\wedge_\phi(x, y) = \phi(\{x, y\})$. This is a polymorphism because ϕ is a homomorphism and it satisfies the conditions of being a semilattice operation as follows. Idempotency follows directly from the unit law.

Commutativity is given by definition. Associativity follows as

$$\wedge_\phi(x, \wedge_\phi(y,z)) = \phi(\{x, \phi(\{y,z\})\})$$

which by the associativity law of the algebra is just $\phi(\{x,y,z\})$ which is equal, by a similar argument, to $\wedge_\phi(\wedge_\phi(x,y),z)$.

To go the other direction, we take a semilattice operation $\wedge$ and first note that the idempotency, commutativity and associativity of $\wedge$ together guarantee that for any elements $x_1, \dots x_m$ in B, the element $x_1 \wedge \dots \wedge x_m$ is well-defined and depends only on the set $\{x_1, \dots x_m\}$. We denote this element $\bigwedge_i x_i$ and use it to define a function $\phi_\wedge \colon \mathbb{D}_{\mathbb{B}}\mathcal{B} \to \mathcal{B}$ as $\phi_\wedge(\{x_1, \dots x_m\}) := \bigwedge_i x_i$. We now show that this is an algebra. Idempotency and associativity of $\wedge$ guarantee that the unit and associativity laws are satisfied. So, it remains to show that $\phi_\wedge$ is a homomorphism. Suppose that $(P_1, \dots P_m) \in R^{\mathbb{D}_{\mathbb{B}}\mathcal{B}}$ meaning that there is a set $S \subset R^{\mathcal{B}}$ such that P_i is the i^{th} projection of S. Writing S as a large array of elements $(s_{1,1}, \dots, s_{1,m}), \dots (s_{l,1}, \dots s_{l,m})$ we can think of applying $f_\wedge$ to each column $\{s_{1,i}, \dots s_{l,i}\} = P_i$ as repeatedly applying $\wedge$ on adjacent rows. As $\wedge$ is a polymorphism, all rows in the resulting array after each step will be in $R^{\mathcal{B}}$ and so the tuple $(f_\wedge(P_1), \dots, f_\wedge(P_m)) \in R^{\mathcal{B}}$, as required. $\square$

This example suggests the important questions of whether we can provide satisfying characterisation of the $\mathbb{D}_{\mathbf{S}}$ algebras for other semirings and whether such algebras have a more general significance to the constraint satisfaction problem as homomorphism do in Theorem 7.22. In this direction, it should be noted that over **Set**, the algebras of the **S**-distribution monad are known as convex spaces and have been widely studied in category theory. An interesting survey of these for the non-negative rational case is given by Fritz [47].

In this chapter, we have seen how monads can be used to provide a different approach to approximating homomorphism between relational structures to that seen in the game comonads of earlier chapters. We also saw that similar, somewhat dual results about Kleisli morphisms and monad algebras could be found connecting these new constructions to known notions in finite model theory. Despite these similarities this theory also has apparent differences to that of game comonads, notably in the areas of Kleisli isomorphisms and the seeming importance of non-algebraic homomorphisms over algebras. These differences provide fruitful directions for future work.

Despite the many interesting facets of this new theory, the developments of this chapter fall short of the aim expressed at the end of Chapter 6 to find a new category theoretic semantics for approximations to structure isomorphism which are both efficiently computable and go beyond the realm of fixed-point logic with counting. In the next and final chapter we tackle this problem using a new set of tools from algebraic topology.

Chapter 8

Cohomology for homomorphism and isomorphism

So far in this thesis, we have explored compositional semantics for two types of approximation to homomorphism and isomorphism. In Chapters 3 to 6, we saw that comonads give an interesting perspective on Duplicator strategies for a wide variety of k-pebble games. These comonads weaken the homomorphism relation $\mathcal{A} \to \mathcal{B}$ by replacing $\mathcal{A}$ with some resourced-indexed unfolding of $\mathbb{T}_k\mathcal{A}$. Then, homomorphisms $\mathbb{T}_k\mathcal{A} \to \mathcal{B}$ only require that the "k-local" structure of $\mathcal{A}$ is preserved, with the structure of $\mathbb{T}_k\mathcal{A}$ mediating how local patches are attached to one another. In Chapter 7, we used monads to weaken homomorphisms in a dual way. Instead replacing $\mathcal{B}$ with some semiring-indexed completion $\mathbb{W}_{\mathbf{S}}\mathcal{B}$. In this case, homomorphisms $\mathcal{A} \to \mathbb{W}_{\mathbf{S}}\mathcal{B}$ are "global" assignments of images to elements of $\mathcal{A}$ which take values in a generally more regular structure $\mathbb{W}_{\mathbf{S}}\mathcal{B}$.

In this chapter, we introduce a new approach to approximating homomorphism and isomorphism which encompasses elements from both of these approaches and gives a topological meaning to this intuitive language of "local" and "global" approximations. We do this by showing that local solutions to any instance of the homomorphism or isomorphism problems form a topological object known as a *presheaf*. With this new perspective, we establish that homomorphisms and isomorphisms are *global sections* of these presheaves and approximating these relations efficiently can be seen as computing obstructions to these global sections. This approach owes owes a great deal by means of inspiration to the sheaf-theoretic approach to quantum contextuality introduced by Abramsky and Brandenburger [5]. We show that well-known local approximation, k-consistency ($\to_k$) and k-Weisfeiler-Leman equivalence[1] ($\equiv_k$), can be recovered as greatest fixed points of presheaf operators. Furthermore, we show how invariants from sheaf cohomology can be used to find further obstacles to combining local homomorphisms and isomorphisms into

[1]The algorithm we call "k-Weisfeiler-Leman" is more commonly called "$(k-1)$-Weisfeiler-Leman" in the literature, see for example [27]. We prefer "k-Weisfeiler-Leman" to emphasise its relationship to k-variable logic and sets of k-local isomorphisms.

global ones. We use these to construct new efficient extensions to the k-consistency and k-Weisfeiler-Leman algorithms computing relations $\to_k^{\mathbb{Z}}$ and $\equiv_k^{\mathbb{Z}}$ which refine $\to_k$ and $\equiv_k$.

In Section 8.1, we recount known relationships between Kleisli homomorphisms and isomorphisms of $\mathbb{P}_k$ and the k-consistency and k-Weisfeiler-Leman algorithms. In Section 8.2, we show how k-local homomorphisms and isomorphisms form a presheaf and that the relations $\to_k$ and $\equiv_k$ correspond to the existence of certain natural subpresheaves of this. In Section 8.3, we show how to use the cohomology of these presheaves to create new efficiently-computable algorithms which compute new relations $\to_k^{\mathbb{Z}}$ and $\equiv_k^{\mathbb{Z}}$, refining $\to_k$ and $\equiv_k$. In Section 8.4, we establish the power of these new cohomological algorithms, showing that $\to_k^{\mathbb{Z}}$ decides all affine CSPs and that $\equiv_k^{\mathbb{Z}}$ can distinguish structures which differ on many variants of the CFI property including a recent variant due to Lichter [72] which is inexpressible in rank logic.

8.1 Local methods for homomorphism and isomorphism

In this chapter, we use presheaves and their algebraic invariants to extend two classic algorithms for approximating homomorphism and isomorphism. These algorithms are the k-consistency algorithm which was introduced as the (k, k)-consistency algorithm by Feder and Vardi [44], and the k-Weisfeiler-Leman algorithm which is a generalisation of the naïve vertex classification algorithm (see, for example, Read and Corneil [85]) introduced for all dimensions k by Babai and named for Weisfeiler and Leman's algebraic contributions to the $k = 3$ case [90]. For a full history of the k-Weisfeiler-Leman algorithm refer to Cai, Fürer and Immerman [27]. Explicit modern presentations of these algorithms can be seen, for example, in [13] and [66]. We instead focus on equivalent formulations in terms of positional strategies for Duplicator in the k-pebble games which were featured extensively in earlier chapters of this thesis.

In this section, we fix definitions for these algorithms and recap some of the known limitations of these approximations, which motivate the later constructions of this chapter.

8.1.1 Local algorithms and forth systems

Let $\mathcal{A}$ and $\mathcal{B}$ be two relational structures over a common signature σ and let k be a positive integer. We assume throughout this chapter that k is greater than the arity of σ. As in Chapter 4, we write $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ for the set of partial homomorphisms from $\mathcal{A}$ to $\mathcal{B}$ of size at most k, we call these the k-local homomorphisms from $\mathcal{A}$ to $\mathcal{B}$. Additionally we write $\mathbf{isom}_k(\mathcal{A}, \mathcal{B})$ for the set of all partial isomorphisms in $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$. The k-consistency and k-Weisfeiler-Leman algorithms can now be presented in the following ways.

Classical k-consistency algorithm

Recall from Definition 4.7 that a forth system is a set of partial homomorphisms $S \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ which is downward-closed and such that every $f \in S$ with $|\mathbf{dom}(f)| < k$ satisfies the property $\mathbf{Forth}(S, f)$ which states that there is a function $\phi_f \colon A \to B$ such that for any $a \in A$, $f \cup \{(a, f(a))\} \in S$. We saw in Lemma 4.8 that the existence of a non-empty forth system is equivalent to a Duplicator strategy in the game $\exists\mathbf{Peb}^k(\mathcal{A}, \mathcal{B})$. This result is known to Kolaitis and Vardi [67] who also show that this condition is equivalent to the pair $(\mathcal{A}, \mathcal{B})$ being accepted by the k-consistency algorithm. This allows us to give the following presentation of k-consistency.

Given $S \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ which is downward-closed, we define $\overline{S}$ to be the largest subset of S which is downwards-closed and has the forth property. Note that $\emptyset$ satisfies these conditions, so such a set always exists. For a fixed k there is a simple algorithm for computing $\overline{S}$ from S. This is done by starting with $S_0 = S$ and then entering the following loop with $i = 0$

1. Initialise S_{i+1} as being equal to S_i.
2. For each $s \in S_i$, check if $\mathbf{Forth}(S_i, s)$ holds and if not remove it from S_{i+1} along with all $s' > s$.
3. If none fail this test, halt and output S_i.
4. Otherwise, increment i by one and repeat.

It is easily seen that this runs in polynomial time in $|A|^k|B|^k$.

Now for a pair of structures $\mathcal{A}, \mathcal{B}$ we say that the pair $(\mathcal{A}, \mathcal{B})$ is k consistent if $\overline{\mathbf{hom}_k(\mathcal{A}, \mathcal{B})} \neq \emptyset$. We denote this by writing $\mathcal{A} \to_k \mathcal{B}$ and the algorithm above shows how to decide this relation in polynomial time for fixed k. This relation has many equivalent logical and algorithmic definitions as seen in [44], and [19].

Classical k-Weisfeiler-Leman algorithm

In a similar way to above, we exploit a well-known relationship between k-variable logics and the k-Weisfeiler-Leman algorithm to give the following presentation. Indeed, Immerman and Lander [60] first established that two structures are k-WL-equivalent if and only if they satisfy the same formulas of infinitary k-variable logic with counting quantifiers (written $\mathcal{A} \equiv_{\mathcal{L}^k_\infty(\#)} \mathcal{B}$).

A modified version of Lemma 4.8 (which was known to Hella [58]), shows that this is true if and only if there is a non-empty set $S \subset \mathbf{isom}_k(\mathcal{A}, \mathcal{B})$ which is downward-closed and has the *bijective forth property*, meaning that each $f \in S$ with $|\mathbf{dom}(f)| < k$ satisfies

$\mathbf{BijForth}(S, f)$, which is the forth property where the witnessing function ϕ_f can be chosen to be a bijection. Whether such a bijection exists can be determined efficiently given $\mathcal{A}, \mathcal{B}, S$ and f by determining if the bipartite graph with vertices $A \sqcup B$ and edges $\{(a, b) \mid f \cup \{(a, b)\} \in S\}$ has a perfect matching.

For downward-closed $S \subset \mathbf{isom}_k(\mathcal{A}, \mathcal{B})$, let $\overline{\overline{S}}$ be the largest subset of S which is downward-closed and satisfies the bijective forth property. For fixed k this can be computed in polynomial time in the sizes of $\mathcal{A}$ and $\mathcal{B}$ and so an alternative polynomial time algorithm for determining $\equiv_{k-WL}$ is computing $\overline{\overline{\mathbf{isom}_k(\mathcal{A}, \mathcal{B})}}$ and checking if it is non-empty.

8.1.2 Limitations of local methods

The limitations of these algorithms are well-known. For k-consistency, we recall from Chapter 7 that the question of whether there is a homomorphism $\mathcal{A} \to \mathcal{B}$ is the same as an instance of the constraint satisfaction problem $CSP(\mathcal{B})$. In this context Feder and Vardi [44], showed that k-consistency decides $CSP(\mathcal{B})$ if, and only if, $\mathcal{B}$ has *bounded width*. While this is a strictly wider class of structures than the *width* 1 structures of Corollary 7.23, there are many domains which have unbounded width. An important class of examples for this chapter are domains which encode systems of linear equations over finite fields.

For k-Weisfeiler-Leman equivalence, Cai, Fürer and Immerman showed that k-WL can only distinguish structures which differ on properties expressible in $\mathcal{C}^k$ and in the same paper they demonstrated an efficiently decidable graph property which is not expressible in $\mathcal{C}^k$ for any k. This property is known as the Cai-Fürer-Immerman (CFI) property and forms the basis of many constructions used for lower bounds in descriptive complexity. One such construction of particular interest to this chapter is that of Lichter [72] which was used very recently to distinguish rank logic from PTIME. We revisit these constructions later in the chapter.

8.2 Presheaves for homomorphism and isomorphism

The two algorithms in the last section for computing $\to_k$ and $\equiv_k$ have a similar flavour. Both of them make use of local information about the homomorphism and isomorphism problem to try and approximate the global problem. This situation of accessing some object, in this case the space of homomorphisms, only through some local representations of it is common in other areas of mathematics, in particular, algebraic topology. There and in the wider mathematical world the notion of a *presheaf* has been employed in the study of such situations. In this section, we show how to apply this terminology to the homomorphism and isomorphism problems on finite relational structures and how to recover the k-consistency and k-Weisfeiler-Leman algorithms in this framework. First we recall some basic definitions about presheaves. For a complete account of the background of presheaves we refer to Chapter 2 of MacLane and Moerdijk's textbook [76].

For any categories $\mathbf{C}$ and $\mathbf{S}$, an $\mathbf{S}$-valued presheaf over $\mathbf{C}$ is a contravariant functor $\mathcal{F}\colon \mathbf{C}^{op} \to \mathbf{S}$. In the cases of interest to us $\mathbf{C}$ is some subset of the powerset of some set X with subset inclusion as the morphisms. We call X the underlying space of $\mathbf{C}$. For this reason, when $U' \subset U$ in $\mathbf{C}$ we write $(\cdot)_{|_{U'}}$ for the restriction map $\mathcal{F}(U' \subset U)\colon \mathcal{F}(U) \to \mathcal{F}(U')$. When $\mathbf{S} = \mathbf{AbGrp}$, we call $\mathcal{F}$ an *abelian presheaf* and when $\mathbf{S} = \mathbf{Set}$ we just call $\mathcal{F}$ a presheaf or a presheaf of sets where there is ambiguity. These are the only cases we will consider. A *global section* of $\mathcal{F}$ is a natural transformation $s\colon \mathbb{I} \implies \mathcal{F}$, where $\mathbb{I}$ is the terminal presheaf which sends all objects to the terminal object in $\mathbf{S}$, i.e. the singleton set in $\mathbf{Set}$ or the trivial group in $\mathbf{AbGp}$. We represent such an s as a collection $\{s_U \in \mathcal{F}(U)\}_{U \in \mathbf{C}}$ where naturality implies that for any subsets U and U' in $\mathbf{C}$ $(s_U)_{|_{U \cap U'}} = (s_{U'})_{|_{U \cap U'}}$. In the theory of abelian presheaves, an important concept is that of *exact sequences*. For any sequence of abelian groups $A_1, A_2, \ldots$ and group homomorphisms $d_i\colon A_i \to A_{i+1}$, we say the sequence is *exact at i* if $\mathrm{im}(d_{i-1}) = \ker(d_i)$ where the *kernel*, $\ker(d_i)$, is defined as the subgroup of A_i consising of all $a \in A_i$ such that $d_i(a) = 0$. The whole sequence is said to be *exact* if it is exact at each i. By analogy to this situation, a sequence of abelian presheaves $\mathcal{F}_1, \mathcal{F}_2, \ldots$ over the the same category $\mathbf{C}$ and natural transformations $\delta^i\colon \mathcal{F}_i \implies \mathcal{F}_{i+1}$ is said to be exact if for each object $C \in \mathbf{C}$, the sequence of abelian groups $\mathcal{F}_1(C), \mathcal{F}_2(C) \ldots$ and homomorphisms $\delta^i_C\colon \mathcal{F}_i(C) \to \mathcal{F}_{i+1}(C)$ is exact. We now see how to make sense of this abstract structure in the case of homomorphisms and isomorphisms between relational structures.

8.2.1 Defining presheaves of local solutions

The sets $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ and $\mathbf{isom}_k(\mathcal{A}, \mathcal{B})$ were central to our definition of the k-consistency and k-Weisfeiler-Leman algorithms. We now show that they can be given the natural structure of presheaves on the underlying space A. Indeed, we define the *presheaf of homomorphisms from $\mathcal{A}$ to $\mathcal{B}$* $\mathcal{H}(\mathcal{A}, \mathcal{B})\colon \mathbf{P}(\mathbf{A})^{op} \to \mathbf{Set}$ as $\mathcal{H}(\mathcal{A}, B)(U) = \{s \in \mathbf{hom}(\mathcal{A}, \mathcal{B}) \mid \mathbf{dom}(s) = U\}$ with restriction maps $\mathcal{H}(\mathcal{A}, \mathcal{B})(U' \subset U)$ given by the restriction of partial homomorphisms $(\cdot)_{|_{U'}}$. Similarly, let $\mathcal{I}(\mathcal{A}, \mathcal{B})$ be the subpresheaf of $\mathcal{H}(\mathcal{A}, \mathcal{B})$ containing only partial isomorphisms. Now, consider the cover of A by subsets of size at most k, written $A^{\leq k} \subset P(A)$. We take this set to contain the empty set $\emptyset \subset A$. We define the *presheaves of k-local homomorphisms and isomorphisms* $\mathcal{H}_k(\mathcal{A}, \mathcal{B})$ and $\mathcal{I}_k(\mathcal{A}, \mathcal{B})$ as the functors $\mathcal{H}(\mathcal{A}, \mathcal{B})$ and $\mathcal{I}(\mathcal{A}, \mathcal{B})$ restricted to the subcategory $(\mathbf{A}^{\leq \mathbf{k}})^{op} \subset \mathbf{P}(\mathbf{A})^{op}$. For $\emptyset$, $\mathcal{H}_k(\mathcal{A}, \mathcal{B})(\emptyset)$ and $\mathcal{I}_k(\mathcal{A}, \mathcal{B})(\emptyset)$ are both singleton sets containing the unique empty function which we call ϵ. This will be important in Section 8.4. We now see that these presheaves encode the full homomorphism and isomorphism problems respectively.

8.2.2 Global sections and full solutions

A crucial fact about these presheaves of k-local solutions to the homomorphism and isomorphism problem is that, for large enough k, these presheaves encode the answer to the full homomorphism and isomorphism problems. In fact, we see in the following result that there is a correspondence between the existence of a homomorphism or isomorphism and the existence of a global section in the corresponding presheaf of local solutions.

Lemma 8.1. *For a relational structures $\mathcal{A}, \mathcal{B}$ over the same signature, σ, and k at least the arity of σ then*

$$\mathcal{A} \to \mathcal{B} \iff \mathcal{H}_k \textit{ has a global section}$$

and if $|A| = |B|$ then

$$\mathcal{A} \cong \mathcal{B} \iff \mathcal{I}_k \textit{ has a global section.}$$

Indeed, in each case, there is a bijection between the set of morphisms and the set of global sections.

Proof. ($\implies$) This direction is easy. In the case of homomorphisms the argument proceeds as follows. Suppose that $h \colon \mathcal{A} \to \mathcal{B}$ is a homomorphism. Consider the collection of maps $\{h_U\}_{U \in A^{\leq k}}$ defined by $h_U = h_{|_U}$. This forms global section of $\mathcal{H}_k$ because, firstly, $h_U \in \mathcal{H}_k(U)$ as the restriction of a homomorphism is a partial homomorphism and, secondly, the naturality condition is satisfied as $(h_U)_{|_{U'}} = h_{|_{U'}}$ for any $U' \subset U$. The argument follows similarly for isomorphism and $\mathcal{I}_k$.

($\impliedby$) For this direction, in the case of homomorphisms, let $s : \mathbb{I} \implies \mathcal{H}_k$ be a global section. Firstly, we claim that there is a single function $h : A \to B$ such that $s_U = h_{|_U}$ for all $U \in A^{\leq k}$. Indeed, this is the function h which sends any element $a \in A$ to the element $h(a) := s_{\{a\}}(a) \in B$. This satisfies the required property that for any U, $s_U = h_{|_U}$, by the naturality of s along the inclusion $\{u\} \subset U$ which ensures that $s_U(u) = s_{\{u\}}(u) = h(u)$. To show that h is a homomorphism, take any related tuple $(a_1, \ldots, a_m) \in R^{\mathcal{A}}$. Then, $U = \{a_1, \ldots a_m\}$ has size at most k and so s_U is a homomorphism and $(s_U(a_1), \ldots, s_U(a_m)) \in R^{\mathcal{B}}$.

In the case of isomorphisms, we can define the map h in the same way. As its projections $s_U = h_{|_U}$ are all partial isomorphisms, h will be injective and so is bijective by the assumption on sizes of A and B. So applying the above, h will be a bijective homomorphism. To show that it is indeed an isomorphism we show that it reflects related tuples in B. Suppose $(b_1, \ldots, b_m) \in R^{\mathcal{B}}$ then consider the set $V = \{b_1, \ldots, b_m\}$. As h is bijective there is a set $U = h^{-1}(V) \in A^{\leq k}$. Then $h_{|_U} = s_U$ is an isomorphism between U and V and so $(h^{-1}(b_1), \ldots, h^{-1}(b_m)) \in R^{\mathcal{A}}$ as required. $\square$

Noting that $\mathcal{H}_k$ and $\mathcal{I}_k$ can be computed for any relational structures A and B in $\mathcal{O}(\text{poly}(|A| \cdot |B|))$, this also gives us an interesting starting point for designing efficient algorithms for approximating homomorphism and isomorphism. In particular, any efficient algorithms which find obstacles to the existence of global sections in $\mathcal{H}_k$ and $\mathcal{I}_k$ will provide a tractable approximation to homomorphism and isomorphism. We now see how this approach can be used to capture some classical approximations of these problems.

8.2.3 Flasque subpresheaves and local consistency

In this section, we show that the relations $\to_k$ and $\equiv_k$ computed respectively by the k-consistency and k-Weisfeiler-Leman algorithms can be seen as the absence of certain local obstructions to global sections in $\mathcal{H}_k(\mathcal{A}, \mathcal{B})$ and $\mathcal{I}_k(\mathcal{A}, \mathcal{B})$. In particular, we define efficiently computable monotone operators on subpresheaves of $\mathcal{H}_k$ and $\mathcal{I}_k$ and show that they have non-empty greatest fixpoints if and only if $\mathcal{A} \to_k \mathcal{B}$ and $\mathcal{A} \equiv_k \mathcal{B}$ respectively. Proposition 8.2 is reproduced with permission from an unpublished technical report of Samson Abramsky [1] and the formulation of the fixpoint operators is inspired by the same report.

Flasque presheaves and k-consistency

Here, we show that a pair of structures is accepted by the k-consistency algorithm if and only if $\mathcal{H}_k(\mathcal{A}, \mathcal{B})$ contains a non-empty *flasque* subpresheaf, where a presheaf $\mathcal{F}$ is flasque if all of the restriction maps $\mathcal{F}(U \subset U')$ are surjective. To do this we recall the relationship between k-consistency and forth systems from Section 8.1 and prove the following result. This result was originally proved by Samson Abramsky during a collaboration on an earlier version of the work for this chapter and appears in [1].

Proposition 8.2. *For $\mathcal{A}, \mathcal{B}$ relational structures and any k there is a bijection between:*

- *forth systems in $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$, and*
- *non-empty flasque subpresheaves $\mathcal{S} \subset \mathcal{H}_k(\mathcal{A}, \mathcal{B})$.*

Proof. Suppose that $\mathcal{A} \to_k \mathcal{B}$ and that this is witnessed by a non-empty forth system $S \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$. We claim that $\mathcal{S} \subset \mathcal{H}_k(\mathcal{A}, \mathcal{B})$, defined by $\mathcal{S}(U) := \{f \in S \mid \mathbf{dom}(f) = U\}$, is a flasque subpresheaf. As S is downward-closed we have that the restriction maps are well defined on $\mathcal{S}$ and so it remains to prove that it is flasque. To do this, we show that for any $f \in \mathcal{S}(U)$ and $U \subset U'$ f is in the image of $\mathcal{S}(U \subset U') = (\cdot)_{|U}$. We prove this by induction on the size of $d := |U' \setminus U|$. If $d = 0$, then $U' = U$ and the restriction map $\mathcal{S}(U \subset U')$ is the identity map, so the conclusion holds. If $d > 0$, then remove any element $x \in U' \setminus U$ from U' to get U'' which also contains U. By induction f is in the image of $\mathcal{S}(U \subset U'')$, meaning that there is some $f'' \in \mathcal{S}(U'')$ with $f''_{|U} = f$. Now, as $|U''| < k$ by definition we know that $\mathbf{Forth}(S, f'')$ and so there is some $y \in B$ such that $f' := f'' \cup \{(x, y)\} \in S$. Thus we have $f' \in \mathcal{S}(U')$ such that $f'_{|U} = f$, as required.

For the other direction we take a flasque subpresheaf $\mathcal{S}$ of $\mathcal{H}_k(\mathcal{A}, \mathcal{B})$ and define $S := \bigcup_U \mathcal{S}(U)$, a subset of $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$. It is not hard to see that S is a forth system. Indeed, the existence of the restriction maps $\mathcal{S}(U \subset U')$, guarantees downward closure and surjectivity of these maps proves the forth property in the following way. Let $f \in S$ be some partial homomorphism with domain U of size less than k and let $a \in A$ be any element of A. Then we have that $U' := U \cup \{a\} \in A^{\leq k}$ and so $\mathcal{S}(U')$ is defined and the restriction map $\mathcal{S}(U \subset U')$ is surjective. This means that there is some $f' \in \mathcal{S}(U \cup \{a\})$ such that $f'_{|_U} = f$, meaning that f' witnesses the condition $\mathbf{Forth}(S, f)$, as required. □

This gives an alternative description of the k-consistency algorithm as constructing the largest flasque subpresheaf $\overline{\mathcal{H}_k}$ of $\mathcal{H}_k$ and checking if it is empty. As pointed out by Abramsky [1], $\overline{\mathcal{H}_k}$ is the coflasquification of the presheaf $\mathcal{H}_k$ and can be seen as dual to the Godement construction [49], an important early construction in homological algebra. $\overline{\mathcal{H}_k}$ can be computed efficiently as the greatest fixpoint of the presheaf operator $(\cdot)^{\uparrow\downarrow}$ which computes the largest subpresheaf of a presheaf $\mathcal{S} \subset \mathcal{H}_k$ such that every $s \in \mathcal{S}^{\uparrow\downarrow}(U)$ satisfies the forth property $\mathbf{Forth}(\mathcal{S}, s)$.

Greatest fixpoints and k-Weisfeiler-Leman

In a similar way, we saw in the last section that the k-Weisfeiler-Leman algorithm can be formulated as computing the largest bijective forth system between $\mathcal{A}$ and $\mathcal{B}$. This inspires the definition of another efficiently computable presheaf operator $(\cdot)^{\#\downarrow}$ which computes the largest subpresheaf of a presheaf $\mathcal{S} \subset \mathcal{I}_k$ such that every $s \in \mathcal{S}^{\#\downarrow}(U)$ satisfies the bijective forth property $\mathbf{BijForth}(\mathcal{S}, s)$. We call the greatest fixpoint of this operator $\overline{\overline{\mathcal{S}}}$ and we have that $\mathcal{A} \equiv_k \mathcal{B}$ if and only if $\overline{\overline{\mathcal{I}_k}}$ is non-empty.

In the next section, we look at sheaf-theoretic obstructions to forming a global section which come from cohomology and see how these can be used to extend the local methods in this section and define stronger approximations of homomorphism and isomorphism.

8.3 Cohomology for approximating global structure

In this section, we extend the local algorithms, described in the last section in terms of operators on presheaves, by considering efficiently computable obstructions which arise naturally from presheaf cohomology. From this, we derive new cohomological algorithms for approximating homomorphism and isomorphism. While the motivation for these new algorithms comes from the theory of Čech cohomology of presheaves, the actual algorithms require only a single invariant derived from this more general theory. For this reason, we omit a detailed technical description of cohomology from this thesis. Instead, we give an overview of the general technique of presheaf cohomology and its use in quantum contextuality by Abramsky, Barbosa, Kishida, Lal and Mansfield [4] which inspires the invariant of $\mathbb{Z}$-extendability. We then show how to use this invariant to define new algorithms for approximating homomorphism and isomorphism.

8.3.1 Presheaf cohomology and quantum contextuality

The notion of computing cohomology valued in an abelian presheaf $\mathcal{F}$ on a topological space X has a long history in algebraic geometry and algebraic topology which dates back to Grothendieck's seminal paper on the topic [53]. The cohomology of X valued in $\mathcal{F}$ consists of a sequence of abelian groups $H^i(X, \mathcal{F})$ where $H^0(X, \mathcal{F})$ is the free $\mathbb{Z}$-module over global sections of $\mathcal{F}$. As seen in the previous section we may be interested in such global sections but their existence may be difficult to determine. This is where the functorial nature of cohomology is extremely useful. Indeed, any short exact sequence of abelian presheaves

$$0 \to \mathcal{F}_L \to \mathcal{F} \to \mathcal{F}_R \to 0$$

lifts to a long exact sequence of cohomology groups

$$0 \to H^0(X, \mathcal{F}_L) \to H^0(X, \mathcal{F}) \to H^0(X, \mathcal{F}_R) \xrightarrow{\delta} H^1(X, \mathcal{F}_L) \to \dots$$

where we call the group homomorphisms δ between "levels", the *connection maps.* This tells us that the global sections of $\mathcal{F}_R$ which are not images of global sections of $\mathcal{F}$ are mapped to non-trivial elements of the group $H^1(X, \mathcal{F}_L)$ by the maps in this sequence. This means that these higher cohomology groups can be seen as a source of obstacles to lifting "local" solutions in $\mathcal{F}_R$ to "global" solutions in $\mathcal{F}$.

An important recent example of such an application of cohomology to finite structures can be found in the work of Abramsky, Barbosa, Kishida, Lal and Mansfield [4] in quantum foundations. They show that cohomological obstructions of the type described above can be used to detect contextuality (locally consistent measurements which are globally inconsistent) in quantum systems which were earlier given a presheaf semantics by Abramsky and Brandenburger [5].

In this work they represent *quantum measurement scenarios* usuing presheaves. These consist of a set of possible measurements X, a set O of outcomes to each measurement and a cover $M \subset P(X)$ which indicates which measurements can be performed together. The *presheaf of outcomes*, $\mathcal{E}\colon M^{op} \to \mathbf{Set}$, is defined by mapping a set of compatible measurements U to the function space of all joint outcomes on those measurements O^U, with the normal restriction maps. They are particularly interested in flasque subpresheaves $\mathcal{S} \subset \mathcal{E}$ which they call *possibilistic empirical models* as the flasque condition ensures that the "possible measurements" described by $\mathcal{S}$ satisfy the no-signalling law from quantum mechanics. See [5], for more details.

For our purposes we are interested in their treatment of global sections. They say an empirical model $\mathcal{S}$ is *contextual* if it has no global section. And that a local section

$s \in \mathcal{S}(U)$ is *contextual* if there is no global section $\rho\colon \mathbb{I} \implies \mathcal{S}$ such that $\rho_U = s$. They employ cohomology to find sufficient conditions for contextuality. To do this they first construct an abelian presheaf $\mathbb{Z}\mathcal{S}$ by composing $\mathcal{S}$ with the free abelian group functor, so that local sections $r \in \mathbb{Z}\mathcal{S}(U)$ are simply formal $\mathbb{Z}$-linear combinations of elements of $\mathcal{S}(U)$. It is not hard to see that any global section of $\mathcal{S}$ is also a global section of $\mathbb{Z}\mathcal{S}$. So obstructions to global sections of $\mathbb{Z}\mathcal{S}$ are also obstructions to global sections for $\mathcal{S}$. They then use the general cohomological method described above to find such obstructions. We give a sketch of this here and refer to [4] for more details.

From $\mathbb{Z}\mathcal{S}$ and any set of compatible measurements $U \in M$, we can define two other presheaves $\mathbb{Z}\mathcal{S}_{\tilde{U}}$ and $\mathbb{Z}\mathcal{S}_U$, which respectively collect the elements of $\mathbb{Z}\mathcal{S}$ which vanish inside U and outside U. The details of the constructions are unimportant but importantly we have a short exact sequence

$$0 \to \mathbb{Z}\mathcal{S}_{\tilde{U}} \to \mathbb{Z}\mathcal{S} \to \mathbb{Z}\mathcal{S}_U \to 0$$

and the Čech cohomology groups $H^0(\mathbb{Z}\mathcal{S})$ and $H^0(\mathbb{Z}\mathcal{S}_U)$ are generated freely from the global sections of $\mathbb{Z}\mathcal{S}$ and the local sections in $\mathbb{Z}\mathcal{S}(U)$, respectively. Furthermore in the long exact sequence we have the section

$$\ldots \to H^0(\mathbb{Z}\mathcal{S}) \xrightarrow{(\cdot)|_U} H^0(\mathbb{Z}\mathcal{S}_U) \xrightarrow{\delta} H^1(\mathbb{Z}\mathcal{S}_{\tilde{U}}) \to \ldots$$

where δ is the connection map. From the exactness of the sequence, we know that the image of the restriction map is the kernel of δ. This means that, for any $s \in \mathcal{S}(U)$, $\delta(s) \neq 0$ if, and only if, there is no global section $r\colon \mathbb{I} \implies \mathbb{Z}\mathcal{S}$ such that $r_U = s$. This is sufficient for s to be logically contextual in $\mathcal{S}$. For this reason, Abramsky et al. say that $s \in \mathcal{S}(U)$ is *cohomologically contextual* if $\delta(s) \neq 0$. For our purposes, this connection with the connection maps is purely motivational and we don't define it or the higher cohomology groups explicitly. However, these have been used to define other cohomological obstructions to contextuality, for example in the thesis of Giovanni Caru [28]. In this chapter, we stick to the equivalent condition for cohomological contextuality in terms of non-existence of global sections of $\mathbb{Z}\mathcal{S}$ which restrict to s. As our aim is to approximate the existence of global sections we focus on the negation of the cohomological contextuality condition which we will call $\mathbb{Z}$*-extendability.*

8.3.2 $\mathbb{Z}$-local sections and $\mathbb{Z}$-extendability

Returning to presheaves of local homomorphisms and isomorphisms, let $\mathcal{S}$ be a subpresheaf of $\mathcal{H}_k(\mathcal{A}, \mathcal{B})$ or $\mathcal{I}_k(\mathcal{A}, \mathcal{B})$. We fix some terminology inspired by the previous section. Call the abelian presheaf $\mathbb{Z}\mathcal{S}$ the *presheaf of* $\mathbb{Z}$*-linear sections of* $\mathcal{S}$. For any $U \in A^{\leq k}$, elements of $\mathbb{Z}\mathcal{S}(U)$ are called *local* $\mathbb{Z}$*-linear sections of* $\mathcal{S}$ and are written as sums $\sum_s \alpha_s s$ where $\alpha \in \mathbb{Z}$. Global sections of $\mathbb{Z}\mathcal{S}$ are called *global* $\mathbb{Z}$*-linear sections.* Now we can define $\mathbb{Z}$-extendability as follows.

Definition 8.3. *For $\mathcal{S}$ a subpresheaf of $\mathcal{H}_k$ or $\mathcal{I}_k$, say that some $s \in \mathcal{S}(U)$ is $\mathbb{Z}$-extendable in $\mathcal{S}$, and write $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$, if there is a global $\mathbb{Z}$-linear section r of $\mathcal{S}$, such that $r_U = s$. Equivalently, there exists a collection $\{r_{U'} \in \mathbb{Z}\mathcal{S}(U')\}_{U' \in A^{\leq k}}$ such that $r_U = s$ and for all $U', U'' \in M$ we have*

$$(r_{U'})_{|_{U' \cap U''}} = (r_{U''})_{|_{U' \cap U''}}.$$

Importantly for our purposes, deciding the condition $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$ for any $\mathcal{S} \subset \mathcal{H}_k(\mathcal{A}, \mathcal{B})$ is computable in polynomial time in the sizes of A and B. This is because the compatibility conditions for a collection $\{r_U \in \mathbb{Z}\mathcal{S}(U)\}_{U \in A^{\leq k}}$ being a global section of $\mathbb{Z}\mathcal{S}$ can be expressed as a system of polynomially many linear equations in polynomially many variables. Indeed, we write each r_U as $\sum_{s \in \mathcal{S}(U)} \alpha_s s$ where α_s is a variable for each $s \in \mathcal{S}(U)$. This gives a total number of variables bounded by $\mathcal{O}(|A|^k \cdot |B|^k)$, the size of $\mathbf{hom}_k(\mathcal{A}, \mathcal{B})$. For each of the $\mathcal{O}(|A|^{2k})$ pairs of contexts $U, U' \in A^{\leq k}$, the compatibility condition $(r_U)_{|_{U \cap U'}} = (r_{U'})_{|_{U \cap U'}}$ yields a linear equation in the α_s variables for each $s'' \in \mathcal{S}(U \cap U')$ namely,

$$\sum_{\substack{s \in \mathcal{S}(U) \\ s_{|_{U \cap U'}} = s''}} \alpha_s = \sum_{\substack{s' \in \mathcal{S}(U') \\ s'_{|_{U \cap U'}} = s''}} \alpha_{s'}.$$

This leads to a total number of equations bounded by $\mathcal{O}(|A|^{2k} \cdot |B|^k)$. By an algorithm of Kannan and Bachem [65], such a system can be solved in polynomial time in the sizes of A and B. This allows us to define the following efficient algorithms for approximating homomorphism and isomorphism based on removing cohomological obstructions.

8.3.3 Cohomological algorithms for homomorphism and isomorphism

We saw in Section 8.2 that the k-consistency and k-Weisfeiler-Leman algorithms can be recovered as greatest fixpoints of presheaf operators removing local sections which fail the forth and bijective-forth properties respectively. Now that we have from cohomological considerations a new necessary condition $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$ for a local section to feature in a global section of $\mathcal{S}$, we can define natural extensions to the k-consistency and k-Weisfeiler-Leman algorithms as follows.

Cohomological k-consistency

To define the cohomological k-consistency algorithm, we first define an operator which removes those local sections which admit a cohomological obstruction. Let $(\cdot)^{\mathbb{Z}\downarrow}$ be the operator which computes for a given presheaf $\mathcal{S} \subset \mathcal{H}_k$ the subpresheaf $\mathcal{S}^{\mathbb{Z}\downarrow}$ where $\mathcal{S}^{\mathbb{Z}\downarrow}(U)$ contains exactly those local sections $s \in \mathcal{S}(U)$ which satisfy $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$. As this process may remove the local sections in $\mathcal{S}$ which witness the extendability of other local sections we need to take a fixpoint of this operator to get a presheaf with the right extendability properties at every local section. So, we write $\overline{\mathcal{S}}^{\mathbb{Z}}$ for the greatest fixpoint of this operator starting from $\mathcal{S}$. As $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$ is computable in polynomial time in the size of $\mathcal{S}$ and $\overline{\mathcal{S}}^{\mathbb{Z}}$ has a global section if and only if $\mathcal{S}$ has a global section, this allows us to define the following efficient algorithm for approximating homomorphism.

Definition 8.4. *The* cohomological k-consistency algorithm *accepts an instance* $(\mathcal{A}, \mathcal{B})$ *if the greatest fixpoint* $\overline{\mathcal{H}_k(\mathcal{A}, \mathcal{B})}^{\mathbb{Z}}$ *is non-empty and otherwise rejects.*

If $(\mathcal{A}, \mathcal{B})$ *is accepted by this algorithm we write* $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}$ *and say that the instance* $(\mathcal{A}, \mathcal{B})$ *is cohomologically* k*-consistent.*

We begin with two easy observations about this algorithm. Firstly, this algorithm is at least as strong as the k-consistency algorithm.

Observation 8.5. *It is not hard to see that* $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$ *implies both* $\mathbf{Forth}(\mathcal{S}, s)$ *and that for any* $U \subset \mathbf{dom}(s)$, $s_{|_U} \in \mathcal{S}$. *Indeed both of these conditions are subsumed in the global* $\mathbb{Z}$*-linear section witnessing* $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$. *Thus, for any* $\mathcal{A}$ *and* $\mathcal{B}$,

$$\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B} \implies \mathcal{A} \to_k \mathcal{B}$$

The second is that we can rephrase $\to_k^{\mathbb{Z}}$ in terms of a non-empty set of k-local homomorphisms, as we did for $\to_k$ and forth systems in Section 8.1.

Observation 8.6. *For any structures* $\mathcal{A}$ *and* $\mathcal{B}$, $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}$ *if and only if there exists a set* $\emptyset \neq S \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ *in which each element* $s \in S$ *is* $\mathbb{Z}$*-extendable in* S.

We conclude this section by showing that the relation $\to_k^{\mathbb{Z}}$ is transitive.

Proposition 8.7. *For all* k, *given* $\mathcal{A}, \mathcal{B}$ *and* $\mathcal{C}$ *structures over a common finite signature*

$$\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B} \to_k^{\mathbb{Z}} \mathcal{C} \implies \mathcal{A} \to_k^{\mathbb{Z}} \mathcal{C}.$$

Proof. Success of the $\to_k^{\mathbb{Z}}$ algorithm for the pairs $(\mathcal{A}, \mathcal{B})$ and $(\mathcal{B}, \mathcal{C})$ results in two non-empty sets $S^{AB} \subset \mathbf{hom}_k(\mathcal{A}, \mathcal{B})$ and $S^{BC} \subset \mathbf{hom}_k(\mathcal{B}, \mathcal{C})$ in both of which each local section is $\mathbb{Z}$-extendable. By Observation 8.6, to show that $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{C}$, it suffices to show that the set $S^{AC} = \{s \circ t \mid s \in S^{BC}, t \in S^{AB}\}$ has the same property.

To show that every $p_0 = s_0 \circ t_0 \in S^{AC}_{\mathbf{a}_0}$ is $\mathbb{Z}$-extendable in S^{AC} we construct a global $\mathbb{Z}$-linear section extending p_0 from the $\mathbb{Z}$-linear sections $\{r^{t_0}_{\mathbf{a}} := \sum_t z_t t\}_{\mathbf{a} \in A^{\leq k}}$ and $\{r^{s_0}_{\mathbf{b}} := \sum_s w_s s\}_{\mathbf{b} \in B^{\leq k}}$ extending t_0 and s_0 respectively. Define $\{r^{p_0}_{\mathbf{a}}\}_{\mathbf{a} \in A^{\leq k}}$ as

$$r^{p_0}_{\mathbf{a}} = \sum_{t \in S^{AB}_{\mathbf{a}}} \sum_{s \in S^{BC}_{t(\mathbf{a})}} z_t w_s (s \circ t)$$

To show that this is a global $\mathbb{Z}$-linear section extending p_0 we need to show firstly that $r^{p_0}_{\mathbf{a}_0} = p_0$ and secondly that the local sections of r^{p_0} agree on the pairwise intersections of their domains.

To show that $r^{p_0}_{\mathbf{a}_0} = p_0$ we observe that, as r^{t_0} $\mathbb{Z}$-linearly extends t_0, for all $t \in S^{AB}_{\mathbf{a}_0}$ we have $z_{t_0} = 1$ and for other values of t $z_t = 0$ and similarly, for all $s \in S^{BC}_{t_0(\mathbf{a}_0)}$, $w_{s_0} = 1$ and for other values of s $w_s = 0$. From this we have that

$$r^{p_0}_{\mathbf{a}_0} = z_{t_0} w_{s_0}(s_0 \circ t_0) = p_0$$

as required.

Finally, we need to show for any $\mathbf{a}, \mathbf{a}'$ in $A^{\leq k}$ with intersection $\mathbf{a}''$ that

$$r^{p_0}_{\mathbf{a}|_{\mathbf{a}''}} = r^{p_0}_{\mathbf{a}'|_{\mathbf{a}''}}.$$

To do this we show that the left hand side depends only on $\mathbf{a}''$ and not on $\mathbf{a}$. As this argument applies equally to the right hand side, the result follows.

To begin with the left hand side is a dependent sum which loops over $t \in S^{AB}_{\mathbf{a}}$ and $s \in S^{BC}_{t(\mathbf{a})}$ as follows:

$$r^{p_0}_{\mathbf{a}|_{\mathbf{a}''}} = \sum_{t,s} w_s z_t (s \circ t)_{|_{\mathbf{a}''}}$$

To emphasise the dependence on $\mathbf{a}''$ we can group this sum together by pairs t'', s'' with $t'' \in S^{AB}_{\mathbf{a}''}$ and $s'' \in S^{BC}_{t''(\mathbf{a}'')}$. Within each group the the sum loops over $t \in S^{AB}_{\mathbf{a}}$ such that $t_{|_{\mathbf{a}''}} = t''$ and $s \in S^{AB}_{t(\mathbf{a})}$ such that $s_{|_{\mathbf{a}''}} = s''$. We write this as

$$\sum_{t'',s''} \sum_{t_{|_{\mathbf{a}''}}=t''} z_t \sum_{s_{|_{t''(\mathbf{a}'')}}=s''} w_s (s \circ t)_{|_{\mathbf{a}''}}$$

We now show that for each t'', s'' the corresponding part of the sum depends only on t'' and s''. This follows from three observations.

The first observation is that in the sum

$$\sum_{t_{|_{\mathbf{a}''}}=t''} z_t \sum_{s_{|_{t''(\mathbf{a}'')}}=s''} w_s (s \circ t)_{|_{\mathbf{a}''}}$$

the formal variables $(s \circ t)_{|_{\mathbf{a}''}}$ are, by definition, all equal to the variable $(s'' \circ t'')$. Thus we need only consider the coefficients, given by the sum

$$\sum_{t_{|_{\mathbf{a}''}}=t''} z_t \sum_{s_{|_{t''(\mathbf{a}'')}}=s''} w_s$$

The second observation is that for each t such that $t_{|_{\mathbf{a}''}} = t''$ the sum

$$\sum_{s_{|_{t''(\mathbf{a}'')}}=s''} w_s$$

is simply the s'' component of $(r^{s_0}_{t(\mathbf{a})})_{|_{t''(\mathbf{a}'')}}$. As r^{s_0} is a global $\mathbb{Z}$-linear section this is equal to the fixed parameter $w_{s''}$. So the sum in question reduces to

$$w_{s''} \cdot \left(\sum_{t_{|_{\mathbf{a}''}} = t''} z_t \right)$$

The final observation, is that the remaining sum is the t'' component of $(r^{t_0}_{\mathbf{a}})_{|_{\mathbf{a}''}}$ which, as r^{t_0} is a global $\mathbb{Z}$-linear section, is equal to $r^{t_0}_{t''}$. This gives the final form of the expression for $(r^{p_0}_{\mathbf{a}})_{|_{\mathbf{a}''}}$ as

$$\sum_{t'',s''} z_{t''} w_{s''} (t'' \circ s'')$$

It is easy to see that the same arguments apply to $r^{p_0}_{\mathbf{a}'}$ and so

$$(r^{p_0}_{\mathbf{a}})_{|_{\mathbf{a}''}} = (r^{p_0}_{\mathbf{a}'})_{|_{\mathbf{a}''}}$$

as required.

□

Cohomological k-Weisfeiler-Leman

We now define cohomological k-equivalence to generalise k-WL-equivalence in the same way as we did for cohomological k-consistency, by removing local sections which are not $\mathbb{Z}$-extendable. As $\mathbb{Z}$-extendability in $S \subset \mathbf{isom}_k(\mathcal{A}, \mathcal{B})$ is not *a priori* symmetric in $\mathcal{A}$ and $\mathcal{B}$ we need to check both that s is $\mathbb{Z}$-extendable in S and s^{-1} is $\mathbb{Z}$-extendable in $S^{-1} = \{t^{-1} \mid t \in S\}$. We call this s being $\mathbb{Z}$-*bi-extendable* in S and write it as $\mathbb{Z}\mathbf{bext}(\mathcal{S}, s)$. We incorporate this into a new presheaf operator $(\cdot)^{\mathbb{Z}\#}$ as follows. Given a presheaf $\mathcal{S} \subset \mathcal{I}_k$ let $\mathcal{S}^{\mathbb{Z}\#}$ be the largest subpresheaf of $\mathcal{S}$ such that every $s \in \mathcal{S}^{\mathbb{Z}\#}(U)$ satisfies both the bijective forth property $\mathbf{BijForth}(\mathcal{S}, s)$ and the $\mathbb{Z}$-bi-extendability property $\mathbb{Z}\mathbf{bext}(\mathcal{S}, s)$. We write $\overline{\overline{\mathcal{S}}}^{\mathbb{Z}}$ for the greatest fixpoint of this operator starting from $\mathcal{S}$. As both $\mathbf{BijForth}(\mathcal{S}, s)$ and $\mathbb{Z}\mathbf{bext}(\mathcal{S}, s)$ are computable in polynomial time in the size of $\mathcal{S}$ and $\overline{\overline{\mathcal{S}}}^{\mathbb{Z}}$ has a global section if and only if $\mathcal{S}$ has a global section, this allows us to define the following efficient algorithm for approximating isomorphism.

Definition 8.8. *The* cohomological k-Weisfeiler-Leman algorithm *accepts an instance* $(\mathcal{A}, \mathcal{B})$ *if the greatest fixpoint* $\overline{\overline{\mathcal{I}_k(\mathcal{A}, \mathcal{B})}}^{\mathbb{Z}}$ *is non-empty and otherwise rejects.*

If $(\mathcal{A}, \mathcal{B})$ *is accepted by this algorithm we write* $\mathcal{A} \equiv^{\mathbb{Z}}_k \mathcal{B}$ *and say that the instance* $(\mathcal{A}, \mathcal{B})$ *is cohomologically* k*-equivalent.*

Analogously to Observation 8.6, we note that the existence of any non-empty S satisfying these properties is a witness of $\equiv^{\mathbb{Z}}_k$.

Observation 8.9. *For any two structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \equiv_k^{\mathbb{Z}} \mathcal{B}$ if and only if there exists a subset $S \subset \mathbf{Isom}_k(\mathcal{A}, \mathcal{B})$ such that both S and S^{-1} are downward-closed, have the bijective forth property and have $\mathbb{Z}$-extendability for each of their elements.*

Finally, we observe that the existence of a non-empty subpresheaf of $\mathcal{I}_k$ satisfying the **BijForth** and $\mathbb{Z}$**bext** properties also satisfies the conditions for witnessing cohomological k-consistency of the pairs $(\mathcal{A}, \mathcal{B})$ and $(\mathcal{B}, \mathcal{A})$. Formally, we have

Observation 8.10. *For any two structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \equiv_k^{\mathbb{Z}} \mathcal{B}$ implies that $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}$ and $\mathcal{B} \to_k^{\mathbb{Z}} \mathcal{A}$.*

To better understand how the cohomological k-Weisfeiler-Leman algorithm operates, we consider the case of $k = 2$ on undirected graphs where 2-WL coincides with the well-known colour refinement algorithm, as demonstrated for example by Immerman and Lander [60]. The colour refinement algorithm works by starting with a uniform colourings c_0^G and c_0^H of the vertices of the input graphs G and H and refining this colouring in rounds where in each round we define colour $c_i(x)$ by the pair consisting of $c_i(x)$ and the multiset of colours $c_i(y)$ such that y is adjacent to x. The algorithm stops when the partition defined by the c_i is the same as that defined by c_{i+1} for some round i. We call the final colourings c^G and c^H and the pair (G, H) is accepted if the multiset of colours from c^G and c^H are equal. The equivalence of this algorithm with 2-WL can be shown by observing that, if colour refinement accepts (G, H), then the set $C \subset \mathbf{isom}_2(G, H)$ of partial isomorphisms which preserve the final colourings on G and H is a bijective forth system, and the existence of a bijective forth system implies that (G, H) is accepted by colour refinement. Cohomological 2-Weisfeiler-Leman refines this algorithm as follows. Suppose two graphs (G, H) are accepted by colour refinement. Take the bijective forth system $C \subset \mathbf{isom}_2(G, H)$ of isomorphisms which preserve the final colourings and for each such isomorphism $s \in C$ decide whether the linear equations defining $\mathbb{Z}\mathbf{bext}(C, s)$ are satisfiable. The list of maps s which pass this test defines a relation between vertices of G and vertices of H where x and y are related if and only if for each set S of size at most 2 containing either there is an acceptable map s on this set which maps x to y. We use this relation to refine the colourings c^G and c^H, defining two new colourings $(c^G)'$ and $(c^H)'$. If their multisets of colours are not equal, we stop and reject (G, H). If they are equal, we start another round of colour refinement followed by another round of cohomological refinement. We repeat this until the colourings stabilise and perform the same equality test as in the colour refinement. As we will demontrate in the next section this algorithm is strictly stronger than 2-WL.

In Section 8.4, we demonstrate the power of these new algorithms by showing that both cohomological k-consistency and cohomological k-Weisfeiler-Leman are strictly more powerful than their classical counterparts.

8.4 The expressive power of cohomology

In this section, we prove that the new algorithms arising from this cohomological approach to homomorphism and isomorphism are substantially more powerful than the k-consistency and k-Weisfeiler-Leman algorithms. For constraint satisfaction, we show that cohomological k-consistency can decide all CSPs decided by the $BLP_{\mathbb{Z}}$ algorithm detailed in Chapter 7. For isomorphism, we show that for a fixed small k cohomological k-Weisfeiler-Leman can distinguish structures which differ on a very general form of the CFI property, in particular, showing that cohomological k-Weisfeiler-Leman can distinguish a property which Lichter [72] shows not to be expressible in rank logic.

8.4.1 Cohomological k-consistency and ring CSPs

We observed in Section 8.3 that the cohomological k-consistency algorithm is at least as powerful as the k-consistency algorithm and so, as noted in Section 8.1 it decides all CSPs over domains of bounded width. We now show that the new algorithm also decides $CSP(\mathcal{B})$ whenever this is decided by $BLP_{\mathbb{Z}}$, which was studied at length in Chapter 7. In Corollary 7.25, we saw that these structures $\mathcal{B}$ are characterised by the existence of a homomorphism $f \colon \mathbb{D}_{\mathbb{Z}}\mathcal{B} \to \mathcal{B}$. We prove the following theorem by showing that any such homomorphism can reduce a global $\mathbb{Z}$-linear section of $\mathcal{S} \subset \mathcal{H}^k(\mathcal{A}, \mathcal{B})$ to an actual global section of $\mathcal{S}$.

Theorem 8.11. *For any structure $\mathcal{B}$, if $BLP_{\mathbb{Z}}$ decides $CSP(\mathcal{B})$ then there is a k such that cohomological k-consistency algorithm decides $CSP(\mathcal{B})$.*

Proof. As noted in Corollary 7.25, $BLP_{\mathbb{Z}}$ decides $CSP(\mathcal{B})$ if, and only if, there is a homomorphism $f \colon \mathbb{D}_{\mathbb{Z}}\mathcal{B} \to \mathcal{B}$. Fix k as the maximum arity of σ, the signature of $\mathcal{B}$. Cohomological k-consistency decides $CSP(\mathcal{B})$ if for any $\mathcal{A}$ over the same signature $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}$ implies $\mathcal{A} \to \mathcal{B}$. To prove this recall from Observation 8.6 that $\mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}$ is equivalent to the existence of a non-empty subpresheaf $\mathcal{S} \subset \mathcal{H}^k(\mathcal{A}, \mathcal{B})$ such that for every local section $s \in \mathcal{S}(U)$ we have $\mathbb{Z}\mathbf{ext}(\mathcal{S}, s)$. In particular, that means for $\epsilon \in \mathcal{S}(\emptyset)$ there is a global $\mathbb{Z}$-linear section r such that $r_\emptyset = \epsilon$. We use this r to construct a homomorphism $g \colon \mathcal{A} \to \mathcal{B}$ as follows. For each $a \in A$, $r_{\{a\}} = \sum_s \alpha_s s$ where s loops over the elements of $\mathcal{S}(\{a\})$. As r is a global section, we have that $\epsilon = r_\emptyset = \sum_s \alpha_s s_{|\emptyset} = \sum_s \alpha_s \epsilon$ and so for any a $\sum_s \alpha_s = 1$. This means that we can define an element $b_a := \sum_s \alpha_s s(a) \in \mathbb{D}_{\mathbb{Z}}\mathcal{B}$ and use the homomorphism f to map this to a single element of B, giving $g(a) := f(b_a)$. We now show that g is a homomorphism. To this end, consider, for any $\mathbf{a} = (a_1, \ldots, a_m) \in R^{\mathcal{A}}$, the sum $r_{\{a_1,\ldots,a_m\}} = \sum_t \alpha_t t$ for t ranging over $\mathcal{S}(\{a_1, \ldots, a_m\})$. As each t in this sum is a homomorphism, we have that $t(\mathbf{a}) \in R^{\mathcal{B}}$ and, using the naturality of the global section r we see that $\sum_t \alpha_t = 1$ and $\sum_t \alpha_t t(\mathbf{a})$ witnesses that $(b_{a_1}, \ldots b_{a_m}) \in R^{\mathbb{D}_{\mathbb{Z}}\mathcal{B}}$. Then, as f is a homomorphism we see that $g(\mathbf{a}) \in R^{\mathcal{B}}$, completing the proof. $\square$

Remark 8.12. *This theorem is notable because it shows in particular that cohomological k-consistency can solve systems of linear equations with at most k variables per equation over any finite ring. This can be seen by defining for any k and any ring $\mathbf{R}$, a structure $\mathcal{R}$ with the same underlying set as $\mathbf{R}$ and relations $R_{\mathbf{a},b} = \{(x_1, \dots, x_m) \mid \sum a_i x_i = b\}$ for all $\mathbf{a} \in \mathbf{R}^m, b \in \mathbf{R}$ with $m \leq k$. This structure admits a homomorphism $\mathbb{D}_{\mathbb{Z}}\mathcal{R} \to \mathcal{R}$ via the natural $\mathbb{Z}$-action on the underlying abelian addition group of $\mathbf{R}$. This is enough to show that $\to_k^{\mathbb{Z}}$ is strictly stronger than $\to_k$ as even for the ring $\mathbb{Z}_2$ it has been known since Feder and Vardi [44], that $CSP(\mathbb{Z}_2)$ has unbounded width.*

We use this ability to solve equations over all rings to great effect in Section 8.4.3. First however, we show that cohomological k-Weisfeiler-Leman behaves well with respect to logical interpretations.

8.4.2 Working with logical interpretations

We first introduce a slight generalisation of the notion of a logical interpretation between signatures given in Definition 5.1.

Definition 8.13. *For a logic $\mathcal{L}$ and any two (finite) relational signatures σ and $\tau = \{R_1, \dots, R_m\}$ an $\mathcal{L}$-interpretation* of order n *of τ in σ is a collection of $\mathcal{L}[\sigma]$ formulas*

$$I(\mathbf{x}_{R_1}, \dots, \mathbf{x}_{R_m}, \mathbf{z}) = (\phi_R(\mathbf{x}_R, \mathbf{z}_R))_{R\in\tau}$$

where the tuple $\mathbf{x}_R$ has $n \cdot ar(R)$ distinct elements and can be written as a $ar(R)$-tuple of n-tuples as $(\mathbf{x}_R^1, \dots, \mathbf{x}_R^{ar(R)})$ and $\mathbf{z}$ is the tuple of parameters consisting of all unique variables appearing in $\mathbf{z}_R$ for some $R \in \tau$. For each R, $\mathbf{x}_R$ and $\mathbf{z}_R$ are disjoint from each other but the variables in $\mathbf{x}_R$ may be reused in $\mathbf{z}_{R'}$ for some other R'.

For any such interpretation, we have a mapping which sends any σ-structure $\mathcal{A}$ with an assignment $\mathbf{b}$ the variables in $\mathbf{z}$ to a τ-structure $I(\mathcal{A}, \mathbf{b})$ with the underlying set A^n as $\mathcal{A}$ and with related tuples, for each $R \in \tau$, given by

$$R^{I(\mathcal{A},\mathbf{b})} = \{(\mathbf{a}_1, \dots, \mathbf{a}_{ar(R)}) \in (A^n)^{ar(R)} \mid \mathcal{A}, \mathbf{a}_1, \dots, \mathbf{a}_{ar(R)}, \mathbf{b} \models \phi_R(\mathbf{x}_R^1, \dots, \mathbf{x}_R^{ar(R)}, \mathbf{z}_R)\}.$$

In the next result, we show that the equivalence $\equiv_k^{\mathbb{Z}}$ is preserved by C^l-interpretations in the following way.

Proposition 8.14. *For any finite, relational signatures σ and τ, σ-structures A and B, natural numbers n and k, and any order n $\mathcal{C}^{nk}$-interpretation Φ of τ in σ we have that*

$$A \equiv_{nk}^{\mathbb{Z}} B \implies \Phi(A) \equiv_k^{\mathbb{Z}} \Phi(B)$$

Proof. By Observation 8.9, it suffices to show that there is a set $S' \subset \mathbf{isom}_k(\Phi(A), \Phi(B))$ which is downward-closed, satisfies the bijective forth property and in which every map is $\mathbb{Z}$-bi-extendable. As $A \equiv^{\mathbb{Z}}_{nk} B$, there is already a set $S \subset \mathbf{isom}_{nk}(\mathcal{A}, \mathcal{B})$ satisfying these properties. For any $Q \subset A$ we use S_Q to mean the elements of S with domain Q. We now show how to construct a suitable S' from S.

For any $C \subset \Phi(A)$, let $\pi(C)$ be the set of elements in A which appear in some tuple of C. As elements of $\Phi(A)$ are n-tuples over A, it is clear that $|\pi(C)| \leq n|C|$. We can now define S'_C as the set of partial isomorphisms in $S_{\pi(C)}$ applied coordinatewise to C, namely,

$$\{(f, \dots, f)_{|C} \ \mid \ f \in S_{\pi(C)}\}$$

This is well defined for all $C \in (\Phi(A))^{\leq k}$ as $|\pi(C)| \leq nk$. That these maps define partial isomorphisms between $\Phi(A)$ and $\Phi(B)$ follows from Hella's Lemma 5.1 in [58] which states that the elements of $\mathbf{isom}_{nk}(\mathcal{A}, \mathcal{B})$ are exactly those which preserve and reflect C^{nk} formulas. As the relations on $\Phi(A)$ and $\Phi(B)$ are defined by C^{nk} formulas they are preserved and reflected by the members of S. We now show that $S' = \bigcup_{C \in \Phi(A)^{\leq k}} S'_C$ satisfies the required properties.

Downward-closure This follows easily from downward-closure of S. Suppose $\mathbf{f} = (f, \dots, f)_{|C} \in S'$ and $\mathbf{g} \leq \mathbf{f}$. Then there is some $C' \subset C$ such that $\mathbf{g} = \mathbf{f}_{|C'}$ and $\mathbf{g} = (f_{|\pi(C')}, \dots, f_{|\pi(C')})_{|C'}$ but $f_{|\pi(C')} \leq f$ and so is an element of S.

Bijective forth property Let $\mathbf{f} \in S'_C$ with $|C| < k$, with $\mathbf{f}$ given as the coordinatewise application of some $f \in S_{\pi(C)}$. To show that S' has the bijective forth property we must show that there is a bijection $b : \Phi(A) \to \Phi(B)$ such that for any $\mathbf{a} \in \Phi(A)$ the function $\mathbf{f} \cup \{(\mathbf{a}, b(\mathbf{a}))\}$ is in $S'_{C \cup \{\mathbf{a}\}}$. For any such $\mathbf{f}$, we can construct a bijection b whose image on any $\mathbf{a} \in \Phi(A)$ is given as

$$b(\mathbf{a}) = (b^{\epsilon}(a_1), b^{\mathbf{a}_1}(a_2), \dots, b^{(\mathbf{a}_{n-1})}(a_n))$$

where $\mathbf{a}_i$ is the i-tuple of the first i elements in $\mathbf{a}$ and each $b^{\mathbf{a}_i}$ is a bijection $A \to B$. For any $\mathbf{a} \in \Phi(A)$ we choose the bijections $b^{\mathbf{a}_i}$ using the bijective forth property on S. As $\mathbf{f}$ is a coordinatewise application of some $f \in S_{\pi(C)}$ and as $|C| < k$ implies $|\pi(C)| \leq nk - n < nk$, the bijective forth property for S implies the existence of a b_1 such that $f_1 = f \cup \{a_1, b_1(a_1)\} \in S_{\pi(C) \cup \{a_1\}}$. Let $b^{\epsilon} := b_1$. Now suppose for any $i < n$ we have defined the bijections $b^{\epsilon}, b^{\mathbf{a}_1}, \dots, b^{\mathbf{a}_i}$ and $f_i = f \cup \{(a_j, b^{\mathbf{a}_{j-1}}(a_j))\}_{1 \leq j \leq i} \in S_{\pi(C) \cup \{a_1, \dots, a_i\}}$. We still have $|\pi(C) \cup \{a_1, \dots, a_i\}| < nk$ so can use the bijective forth property on S again to find a bijection $b^{\mathbf{a}_i}$ such that $f_{i+1} = f_i \cup \{(a_i, b_{\mathbf{a}_i}(a_i))\} \in S_{\pi(C) \cup \{a_1, \dots, a_{i+1}\}}$. This inductive procedure defines all the required bijections and furthermore shows that $\mathbf{f} \cup \{(\mathbf{a}, b(\mathbf{a})\}$ is the coordinatewise application of some $f_n \in S_{\pi(C \cup \{\mathbf{a}\})}$. This means in particular that $\mathbf{f} \cup \{(\mathbf{a}, b(\mathbf{a})\}$ is in $S'_{C \cup \{\mathbf{a}\}}$, as required.

$\mathbb{Z}$-extendability Our choice of S' makes $\mathbb{Z}$-extendability rather easy. Indeed, we see that any $\mathbf{f} = (f, \dots, f) \in S'_C$ is $\mathbb{Z}$-extendable because the global $\mathbb{Z}$-linear section extending $f \in S_{\pi(C)}$ given as $s_C = \sum_{g \in S_C} \alpha_g g$ can be lifted to a $\mathbb{Z}$-linear extension of $\mathbf{f}$ by defining

$s'_C = \sum_{g \in S_{\pi(C)}} \alpha_g(g, \dots, g)$. The properties of being a $\mathbb{Z}$-linear extension follow from those properties on S.

$\square$

8.4.3 Cohomological k-Weisfeiler-Leman and CFI constructions

The Cai-Fürer-Immerman construction [27] on ordered finite graphs is a very powerful tool for proving expressiveness lower bounds in descriptive complexity theory. While it was originally used to separate the infinitary k variable logic with counting from `PTIME`, it has since been used in adapted forms to prove bounds on invertible maps equivalence [34], computation on Turing machines with atoms [20], symmetric circuits [39] and rank logic [72]. In this section, we show that $\equiv_k^{\mathbb{Z}}$ separates a very general form of this construction.

Recall from Chapter 2 the definition of the CFI structure $\mathbf{CFI}_q(\mathcal{G}, g)$ for any prime power q, totally ordered graph $\mathcal{G} = (G, <)$ and function $g \colon E(G) \to \mathbb{Z}_q$. The construction consists of disjoint *gadgets* A_x for each vertex $x \in G$. Elements of these gadgets are functions $\mathbf{a} \colon \mathcal{N}(x) \to \mathbb{Z}_q$ such that $\sum_{y \in \mathcal{N}(x)} \mathbf{a}(y) = 0$. The elements of these gadgets are connected with those of neighbouring gadgets using relations which are aware of the order $<$, the ring structure $\mathbb{Z}_q$ and the twisting function g. Full details appear in Section 2.4. This construction has the property given in Fact 2.5 that two structures $\mathbf{CFI}_q(\mathcal{G}, g)$ and $\mathbf{CFI}_q(\mathcal{G}, h)$ are isomorphic if, and only if, $\sum g = \sum h$. This leads to the definition of the *CFI property* satisfied by all $\mathbf{CFI}_q(\mathcal{G}, g)$ such that $\sum g = 0$. Two important applications of this construction are given in Chapter 2 as Theorem 2.6 and Theorem 2.7. These results respectively exhibit classes of CFI structures where the CFI property is not expressible in FPC and rank logic.

Despite this CFI property proving to be inexpressible in both FPC and rank logic, we show that (perhaps surprisingly) there is a fixed k such that cohomological k-Weisfeiler-Leman algorithm can separate structures which differ on this property in the following general way.

Theorem 8.15. *There is a fixed k such that for any q, the class of structures $\mathcal{K}_q = \{\mathbf{CFI}_q(\mathcal{G}, g) \mid \sum g = 0\}$ is invariant under $\equiv_k^{\mathbb{Z}}$-equivalence, meaning that for any $\mathcal{A}, \mathcal{B}$ with $\mathcal{A} \equiv_k^{\mathbb{Z}} \mathcal{B}$, $\mathcal{A} \in \mathcal{K}_q \iff \mathcal{B} \in \mathcal{K}_q$.*

As a direct consequence of this result, there is some k such that the set of structures with the CFI property in Lichter's class $\mathcal{K}$ from Theorem 2.7 is closed under $\equiv_k^{\mathbb{Z}}$. This means that, by the conclusion of Theorem 2.7, the equivalence relation $\equiv_k^{\mathbb{Z}}$ can distinguish structures which disagree on a property that is not expressible in rank logic. Indeed, Dawar, Grädel and Lichter [36] show further that this property is also inexpressible in linear algebraic logic. By the definition of our algorithm for $\equiv_k^{\mathbb{Z}}$ this implies that solvability of systems of $\mathbb{Z}$-linear equations is not definable in linear algebraic logic. Furthermore, a similar argument shows that $\equiv_k^{\mathbb{Z}}$ cannot be expressed in any logic for which there is a class of CFI structures on which the logic cannot express the CFI property.

In the next section, we conclude this chapter by proving Theorem 8.15.

8.4.4 Proof of Theorem 8.15

The proof of this theorem proceeds in two parts. The first establishes that the property $\sum g = 0$ for a structure $\mathbf{CFI}_q(\mathcal{G}, g)$ is equivalent to the solvability of a system of equations, $\mathbf{Eq}_q(\mathcal{G}, g)$, over $\mathbb{Z}_q$. The second shows that $\mathbf{Eq}_q(\mathcal{G}, g)$ can be obtained from $\mathbf{CFI}_q(\mathcal{G}, g)$ by an interpretation with a uniform bound on the number of variables per equation. Together with Proposition 8.14, we show that this is enough to prove the theorem.

The first lemma is an adaptation of Lemma 4.36 from Wied Pakusa's PhD thesis [82]. We begin by defining for any $\mathbf{CFI}q(\mathcal{G}, g)$ a system of linear equations over $\mathbb{Z}_q$. This system, $\mathbf{Eq}_q(\mathcal{G}, g)$, is the following collection of equations:

- $X_{\mathbf{a},u}$ for all $u \in G$ and all $\mathbf{a} \in A_u \subset \mathbf{CFI}_q(\mathcal{G}, g)$,
- $I_{\mathbf{a},\mathbf{b},v}$ for all $u \in G$ and $\mathbf{a}, \mathbf{b} \in A_u$ such that $v \in \mathcal{N}(u)$ and there exists $\mathbf{c} \in A_v$ such that $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in R_I$,
- $C_{\mathbf{a},\mathbf{b},v}$ for all $u \in G$ and $\mathbf{a}, \mathbf{b} \in A_u$ such that $v \in \mathcal{N}(u)$ and there exists $\mathbf{c} \in A_v$ $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in R_C$, and
- $E_{\mathbf{a},\mathbf{b},c}$ for all $\mathbf{a} \in A_u, \mathbf{b} \in A_v$ and $(\mathbf{a}, \mathbf{b}) \in R_{E,c}$

where the variables are $w_{\mathbf{a},v}$ for every $u \in G$, $\mathbf{a} \in A_u$ and $v \in \mathcal{N}(u)$ and the equations are given as:

$$\begin{aligned} X_{\mathbf{a},u} : &\quad \sum_{v \in \mathcal{N}(u)} w_{\mathbf{a},v} = 0 \\ I_{\mathbf{a},\mathbf{b},v} : &\quad w_{\mathbf{a},v} - w_{\mathbf{b},v} = 0 \\ C_{\mathbf{a},\mathbf{b},v} : &\quad w_{\mathbf{a},v} - w_{\mathbf{b},v} = 1 \\ E_{\mathbf{a},\mathbf{b},c} : &\quad w_{\mathbf{a},v} + w_{\mathbf{b},u} = c \end{aligned}$$

Then we have the following lemma the proof of which is very similar to that in [82] and so we omit it here.

Lemma 8.16. $\mathbf{CFI}_q(\mathcal{G}, g)$ *a CFI structure, has* $\sum g = 0$ *if and only if* $\mathbf{Eq}_q(\mathcal{G}, g)$ *is solvable in* $\mathbb{Z}_q$

It is not hard to see that the system $\mathbf{Eq}_q(\mathcal{G}, g)$ is first order interpretable in $\mathbf{CFI}_q(\mathcal{G}, g)$. However, in Remark 8.12 we saw that cohomological k-consistency decides satisfiability of systems of equations over any ring with up to k variables per equation. Thus to use this theorem to show that cohomological k-equivalence distinguishes positive and negative instances of the CFI property for some fixed k we need to show that an equivalent system of equations can be interpreted which fixes the number of variables per equation. This is the content of the following lemma.

Lemma 8.17. *For any prime power q, there is an interpretation Φ_q from the signature of the CFI structures $\mathbf{CFI}_q(\mathcal{G}, g)$ to the signature of the ring $\mathbb{Z}_q$ with relations of arity at most 3 such that*

$$\Phi_q(\mathbf{CFI}_q(\mathcal{G}, g)) \to \mathbb{Z}_q \iff \sum g = 0$$

Proof. From Lemma 8.16, we know that interpreting the system of equations $\mathbf{Eq}_q(\mathcal{G}, g)$ would suffice for this purpose. However, the X equations in $\mathbf{Eq}_q(\mathcal{G}, g)$ contain a number of variables which grows with the size of the maximum degree of a vertex in G. As this is, in general, unbounded - and in particular is unbounded in Lichter's class - we need to introduce some equivalent equations in a bounded number of variables. To do this we will introduce some slack variables and utilise the ordering on G to turn any such equation in n variables into a series of equations in 3 variables. We now describe the interpretation Φ_q as follows.

Let 3-$\mathbb{Z}_q$ denote the relational structure on the set $\mathbb{Z}_q$ which contains a relation $T_{\alpha,\beta}$ for each α a tuple of elements of $\mathbb{Z}_q$ size up to 3 and $\beta \in \mathbb{Z}_q$. Each related tuple $(x, y, z) \in T_{\alpha,\beta}$ in a 3-$\mathbb{Z}_q$ structure is an equation

$$\alpha_1 x + \alpha_2 y + \alpha_3 z = \beta$$

To help define the interpretation we introduce some shorthand for some easily interpretable relations on CFI structures A. For $\mathbf{a}, \mathbf{b} \in A$ write $\mathbf{a} \sim \mathbf{b}$ if the two elements belong to the same gadget in A and $\mathbf{a} \frown \mathbf{b}$ if they belong to adjacent gadgets. Both of these relations are easily first-order definable as $\mathbf{a} \sim \mathbf{b}$ if and only if they are incomparable in the $\prec$ relation and $\mathbf{a} \frown \mathbf{b}$ if and only if $(\mathbf{a}, \mathbf{b}) \in R_{E,c}$ for some c. For $\mathbf{a} \frown \mathbf{b}$ in A we will refer to the elements $(\mathbf{a}, \mathbf{a}, \mathbf{b})$ and $(\mathbf{a}, \mathbf{b}, \mathbf{b})$ as $w_{\mathbf{a},\mathbf{b}}$ and $z_{\mathbf{a},\mathbf{b}}$. These will be the variables in the interpreted system of equations. As A comes with a linear pre-order $\prec$ inherited from the order on G, we can also define a local predecessor relation in the neighbourhood of any $\mathbf{a} \in A$. We say that $\mathbf{b}$ is a local predecessor of $\mathbf{b}'$ at $\mathbf{a}$ and write $\mathbf{b} \vdash_{\mathbf{a}} \mathbf{b}'$ if $\mathbf{a} \frown \mathbf{b}$ and $\mathbf{a} \frown \mathbf{b}'$ and there is no $\mathbf{b}''$ with $\mathbf{a} \frown \mathbf{b}''$ such that $\mathbf{b} \prec \mathbf{b}'' \prec \mathbf{b}'$.

Now we define the interpretation on A^3 in three steps, resulting in a system of equations which is solvable if and only if $\mathbf{Eq}_q(\mathcal{G}, g)$ is solvable.

Step 1: Reducing variables We note that in $\mathbf{Eq}_q(\mathcal{G}, g)$ there are only variables $w_{\mathbf{a},y}$ for $\mathbf{a} \in A_x$ and $y \in \mathcal{N}(x)$, whereas the shorthand above describes variables $w_{\mathbf{a},\mathbf{b}}$ and $z_{\mathbf{a},\mathbf{b}}$

for all $\mathbf{a} \in A_x$ and $\mathbf{b} \in A_y$. To reduce the number of variables we want to interpret, for all $\mathbf{a} \frown \mathbf{b}$ and $\mathbf{b} \sim \mathbf{b}'$, the equations $w_{\mathbf{a},\mathbf{b}} = w_{\mathbf{a},\mathbf{b}'}$ and $z_{\mathbf{a},\mathbf{b}} = z_{\mathbf{a},\mathbf{b}'}$. This is done by adding the pairs $(w_{\mathbf{a},\mathbf{b}}, w_{\mathbf{a},\mathbf{b}'})$ and $(z_{\mathbf{a},\mathbf{b}}, z_{\mathbf{a},\mathbf{b}'})$ to the relation $T_{(1,-1),0}$ which can be done as $\frown$ and $\sim$ are definable.

Step 2: Interpreting I, C and E equations Defining these equations in $\Phi(A)$ is straightforward as they all have fewer than 3 variables. In particular we want to add equations

$$w_{\mathbf{a},\mathbf{b}} - w_{\mathbf{a}',\mathbf{b}} = 0$$

for any $(\mathbf{a}, \mathbf{a}', \mathbf{b}) \in R_I$,

$$w_{\mathbf{a},\mathbf{b}} - w_{\mathbf{a}',\mathbf{b}} = 1$$

for any $(\mathbf{a}, \mathbf{a}', \mathbf{b}) \in R_C$, and

$$w_{\mathbf{a},\mathbf{b}} + w_{\mathbf{b},\mathbf{a}} = c$$

for any $(\mathbf{a}, \mathbf{b}) \in R_{E,c}$. These are all easily first-order definable in the $\mathbf{CFI}_q$ signature.

Step 3: Interpreting X equations To interpret the equations for each $u \in G$ and $\mathbf{a} \in A_u$

$$\sum_{v \in \mathcal{N}(u)} w_{\mathbf{a},v} = 0$$

in $\Phi(A)$, we first note that the linear order on G restricts to a linear order on $\mathcal{N}(u)$ which we can write as $\{v_1, \ldots, v_n\}$ where $i < j$ if and only if $v_i < v_j$. To do this it suffices to impose the equations

$$w_{\mathbf{a},\mathbf{b}_1} + \cdots + w_{\mathbf{a},\mathbf{b}_n} = 0$$

for each sequence of elements $\mathbf{b}_1 \vdash_{\mathbf{a}} \ldots \vdash_{\mathbf{a}} \mathbf{b}_n$ with $\mathbf{b}_i \in A_{v_i}$. To do this in equations with at most three variables we employ the auxiliary z variables in the following way. For any $\mathbf{ab} \in A$ such that $\mathbf{a} \frown \mathbf{b}$, if there is no $\mathbf{b}'$ such that $\mathbf{b}' \vdash_{\mathbf{a}} \mathbf{b}$, then we interpret the equation

$$w_{\mathbf{a},\mathbf{b}} - z_{\mathbf{a},\mathbf{b}} = 0$$

if there is $\mathbf{b}'$ such that $\mathbf{b}' \vdash_{\mathbf{a}} \mathbf{b}$ then interpret for all such $\mathbf{b}'$ the equation

$$z_{\mathbf{a},\mathbf{b}'} + w_{\mathbf{a},\mathbf{b}} - z_{\mathbf{a},\mathbf{b}} = 0$$

and if there is no $\mathbf{b}'$ such that $\mathbf{b} \vdash_{\mathbf{a}} \mathbf{b}'$ then interpret the equation

$$z_{\mathbf{a},\mathbf{b}} = 0$$

In this system of equations the $z_{\mathbf{a},\mathbf{b}}$ variables act as running totals for the sum $\sum w_{\mathbf{a},\mathbf{b}_i}$ and so it is not hard to see that solutions to these equations are precisely solutions to the equations $\sum w_{\mathbf{a},\mathbf{b}_i} = 0$. Furthermore, as the relation $\vdash_{\mathbf{a}}$ is definable in the signature of the $\mathbf{CFI}_q$ structures so too are these equations.

To conclude, we have interpreted in $\Phi(\mathbf{CFI}_q(\mathcal{G}, g))$ a system of linear equations with three variables per equation which is solvable over $\mathbb{Z}_q$ if and only if $\mathbf{Eq}_q(\mathcal{G}, g)$ is solvable.

Thus there is a homomorphism $\Phi(\mathbf{CFI}_q(\mathcal{G}, g)) \to \mathbb{Z}_q$ (as 3-$\mathbb{Z}_q$ structures) if and only if $\sum g = 0$. $\square$

We can now conclude with the proof of Theorem 8.15.

Proof of Theorem 8.15. By Fact 2.5, the reverse implication is easy as $\sum h = 0$ implies that $\mathbf{CFI}_q(\mathcal{G}, g) \cong \mathbf{CFI}_q(\mathcal{G}, h)$ and so the structures are cohomologically k-equivalent for any k.

The converse follows from the series of lemmas we have just presented. If $\sum h \neq 0$ then by Lemma 8.17 there is an interpretation Φ_q of order 3 such that $\Phi_q(\mathbf{CFI}_q(\mathcal{G}, g)) \to \mathbb{Z}_q$ but $\Phi_q(\mathbf{CFI}_q(\mathcal{G}, h)) \not\to \mathbb{Z}_q$. By Theorem 8.11, this is means that $\Phi_q(\mathbf{CFI}_q(\mathcal{G}, g)) \to_3^{\mathbb{Z}} \mathbb{Z}_q$ but $\Phi_q(\mathbf{CFI}_q(\mathcal{G}, h)) \not\to_3^{\mathbb{Z}} \mathbb{Z}_q$. So by Observation 8.10, we must have that $\Phi_q(\mathbf{CFI}_q(\mathcal{G}, g)) \not\equiv_3^{\mathbb{Z}} \Phi_q(\mathbf{CFI}_q(\mathcal{G}, h))$. Then noting that the number of variables used in the interpretation Φ_q is some constant c not depending on q and assuming without loss of generality that k is greater than $3c$ then Proposition 8.14 implies that $\mathbf{CFI}_q(\mathcal{G}, g) \not\equiv_k^{\mathbb{Z}} \mathbf{CFI}_q(\mathcal{G}, h)$, as required. $\square$

In this chapter we have introduced a completely new compositional approach to approximating homomorphism and isomorphism using presheaves and presheaf cohomology. This approach and its wider connections to algebraic topology and quantum contextuality suggest new lines of research into algorithms for constraint satisfaction and structure isomorphism. The cohomological algorithms in this chapter were shown to subsume the expressive power of elements of the comonadic and monadic approaches presented in earlier chapters of this thesis but, as yet, no non-trivial upper bounds are known on the expressive power of $\to_k^{\mathbb{Z}}$ and $\equiv_k^{\mathbb{Z}}$. We review some of the open questions in this direction and from other parts of the thesis in the next and final chapter.

Chapter 9

Conclusion and future directions

Since their invention by Abramsky, Dawar and Wang [6], game comonads have developed from a fascinating and rare example of the application of categorical semantics in finite model theory to a substantial framework for uncovering new connections in logic and algorithms. However, this framework has limitations. Before this thesis, it was not possible to apply this new compositional framework to current questions in the descriptive complexity of `PTIME` where the expressive power of the logics and algorithms of interest generally go far beyond the logics $\exists^+\mathcal{L}^\omega_\infty$ and $\mathcal{C}^\omega$. This was due to the lack of game comonads capturing logics or algorithms stronger than those captured by the comonad $\mathbb{P}_k$ in Abramsky, Dawar and Wang's original paper.

This thesis set out to explore the extent to which game comonads can be used for stronger logics than those captured by $\mathbb{P}_k$ and whether other similar constructions from category theory can help to expand the reach of this compositional framework. This goal has been achieved in a variety of ways throughout this thesis. Firstly, we have deepened our understanding of the hitherto "strongest" game comonad, $\mathbb{P}_k$, proving new relationships between its Kleisli maps, counting quantification and unary generalised quantifiers. Secondly, we have created new families of game comonads, $\mathbb{H}_{n,k}$, for capturing logics extended by n-ary quantifiers — an achievement which pushes game comonads beyond the expressive power of $\mathcal{C}^k$ for the first time. Finally, we have provided glimpses of what may lie beyond comonads in the application of category theory to finite model theory. We have shown, in particular, how monads, presheaves and cohomology all provide interesting insights into approximations of homomorphism and isomorphism which are common throughout finite model theory and descriptive complexity.

In this conclusion, we summarise the main contributions of the thesis and we provide some open questions which we hope will form the basis of future work on compositional methods in finite model theory.

9.1 Summary of main results and insights

The contributions of this thesis fall into three main areas: new perspectives on existing game comonads, the construction of new comonads for generalised quantifiers, and new compositional approaches to logic and algorithms.

Unpacking the Kleisli category of $\mathbb{P}_k$ In Chapters 4 and 5, we proved new results about the Kleisli maps for the $\mathbb{P}_k$ comonad. Before this thesis, it was known from Abramsky, Dawar and Wang [6] and Abramsky and Shah [11] that morphisms, isomorphisms and "spans of open pathwise embeddings" in $\mathcal{K}(\mathbb{P}_k)$ correspond to the logics $\exists^+\mathcal{L}^k_\infty$, $\mathcal{L}^k_\infty(\#)$ and $\mathcal{L}^k_\infty$ respectively. In Theorem 4.3 we established further connections between branch-injective, branch-surjective and branch-bijective maps in $\mathcal{K}(\mathbb{P}_k)$ which are intermediate between morphisms and isomorphisms, and the logic $\exists^+\mathcal{L}^k_\infty$ extended by limited forms of counting quantification. Theorem 5.7 gave a new perspective on these connections by showing that different maps in $\mathcal{K}(\mathbb{P}_k)$ correspond to extending infinitary first order logic with different classes of unary generalised quantifiers. Furthering this connection, Theorem 4.16 showed that for a modified version, $\mathbb{P}^*_k$, of the pebbling comonad branch-injective maps are exactly the monomorphisms in $\mathcal{K}(\mathbb{P}^*_k)$.

Game comonads for generalised quantifiers One of the main achievements of this thesis is the construction of the Hella comonad, $\mathbb{H}_{n,k}$ in Theorem 6.1. Made possible by new games for logics with generalised quantifiers introduced in Theorem 5.17, this construction is the first to to capture extensions of first order logic which go beyond $\mathcal{C}^k$. The method of constructing $\mathbb{H}_{n,k}$ as a quotient of the comonad $\mathbb{P}_k$ is also an innovation which opens the door for future constructions of this kind. Theorems 6.13, 6.14 and 6.15 showed how morphisms, isomorphisms and branch-injective/surjective/bijective maps in the Kleisli category of $\mathbb{H}_{n,k}$ relate to k-variable logics extended by different classes of n-ary generalised quantifiers, generalising the connections made in Chapter 5 for $\mathbb{P}_k$. Inspired by this construction, we introduced a new structural decomposition generalising the well-known notion of a tree decomposition of a relational structure, and proved in Theorem 6.19 that the coalgebras of $\mathbb{H}_{n,k}$ correspond to witnesses of this decomposition.

New horizons in compositional constructions Beyond game comonads, this thesis has introduced new constructions from category theory to the study of logic and algorithms.

In Chapter 7, we showed that linear programming approximations to homomorphism can be captured in a very satisfying way using monads. Propositions 7.6 and 7.9 showed how to lift the free left-semimodule and distribution monads $\mathbb{V}_\mathbf{S}$ and $\mathbb{D}_\mathbf{S}$ for any semiring $\mathbf{S}$ to the category of relational structures. This enabled us to prove Theorem 7.16, relating the

Kleisli maps of $\mathbb{D}_{\mathbf{S}}$ to fractional homomorphisms and other linear programming approximations to homomorphism. We also proved Theorem 7.22 which classifies the strength of these approximations in terms of maps of the form $\mathbb{D}_{\mathbf{S}}\mathcal{B} \to \mathcal{B}$.

In Chapter 8, we introduced a new semantics for pebble games and "local" algorithms using presheaves. Building on analogous work from quantum contextuality, we showed how the cohomology of these presheaves could be exploited to define new approximations, $\to_k^{\mathbb{Z}}$ and $\equiv_k^{\mathbb{Z}}$ for homomorphism and isomorphism, and `PTIME` "cohomological" algorithms for computing these. We also demonstrated interesting lower bounds on the strength of these algorithms. Theorem 8.11 showed that $\to_k^{\mathbb{Z}}$ was at least as strong as a combination of k-consistency and the linear programming approximations of Chapter 7, while Theorem 8.15 showed that $\equiv_k^{\mathbb{Z}}$ is capable of distinguishing structures which differ only on properties which are inexpressible in rank logic.

9.2 Open questions and future work

We conclude by recording some of the important questions that emerged from the work presented in this thesis and suggesting directions of further work. Some of these are technical and may make easy questions for future students of this field, while others represent new lines of research whose answers are far from obvious.

Perfecting $\mathbb{P}_k$ In Section 4.2.2, we showed that branch-injective and branch-surjective strategies for the k-pebble game do not correspond to monomorphisms and epimorphisms in the category $\mathcal{K}(\mathbb{P}_k)$. This could be perceived as a deficiency in the definition of $\mathbb{P}_k$ especially in light of Theorem 5.7 which showed that branch-injective and branch-surjective strategies capture generalised quantifiers closed under the monomorphisms and epimorphisms of the category of relational structures. The construction of $\mathbb{P}_k^*$ showed that changing the definition of $\mathbb{P}_k$ could repair this deficiency for monomorphisms. We thus ask if there is a yet-more-perfect version of $\mathbb{P}_k$ which also fixes this discrepancy for epimorphisms.

Question 9.1. *Can Theorem 4.16 be extended to included a relationship between epimorphisms and branch-surjective maps?*
If not, is there an alternative modification, $\tilde{\mathbb{P}_k}$ of $\mathbb{P}_k$ such that monomorphisms and epimorphisms in $\mathcal{K}(\tilde{\mathbb{P}_k})$ correspond to branch-injective and branch-surjective maps?

Generalised quantifiers logics In Section 5.4, we reflected on an often overlooked discrepancy between the generalised quantifier logics used by Hella, and Kolaitis and Väänänen to prove their seminal results (Theorems 5.14 and 5.5) in the theory of finite variable logics extended with generalised quantification. In particular, Proposition 5.28

demonstrated an equivalence between their logics for unary generalised quantifiers that was off-by-one in the variable count. It is is natural to ask if this works for all other arities.

Question 9.2. *Can the result of Proposition 5.28 be generalised to all arities of generalised quantifiers?*

The Hella comonad The construction of the family of game comonads $\mathbb{H}_{n,k}$ has expanded the realm of applicability of game comonads but there is still work to do on better understanding the construction itself. On the technical side, the following is an important question about the related structural decompositions.

Question 9.3. *Can the proof of Theorem 6.19 be improved to drop the condition that the extended tree decompositions are structured?*
Stated equivalently, is it the case that for any $\mathcal{A}$ there exists an extended tree decomposition of $\mathcal{A}$ with width k and arity n if and only if there exists a structured *extended tree decomposition of $\mathcal{A}$ with width k and arity n?*

Another important question is whether the comonad can be used to prove, via Theorem 3.22, a version of Lovasz's theorem [74] for generalised quantifier logic.

Question 9.4. *Does $\mathbb{H}_{n,k}$ satisfy the conditions given by Dawar, Jakl and Reggio [38] that would guarantee a Lovasz-type type theorem linking $\equiv_{\mathcal{L}^k_\infty(\mathbf{Q}_n)}$ and homomorphism counts from structures admitting (structured) extended tree decompositions of width k and arity n?*

Finally, we reflected at the end of Chapter 6 on the use of generalised quantifiers in candidate logics for `PTIME` including in rank logic [35] and linear algebraic logic [36]. As Spoiler-Duplicator games exist for bounding these logics, such as the matrix equivalence and invertible maps games of Dawar and Holm [37], we ask whether we can use $\mathbb{H}_{n,k}$ to construct a game comonad for these games.

Question 9.5. *Can we use a similar approach to the construction of $\mathbb{H}_{n,k}$ to capture the matrix equivalence and invertible maps games of Dawar and Holm [37], or, more generally, to provide a comonadic semantics for rank or linear algebraic logic?*

Uniting game comonads with monads and presheaves As the constructions in Chapters 7 and 8 introduce new objects from category theory, distinct from comonads, an interesting and important question is whether these constructions can be related to game comonads in any formal way. This leads us to ask the following two questions.
Firstly, we ask if the monads in Chapter 7 can be related to game comonads. Relating monads and comonads formally has been studied before in category theory, for example

by Power and Watanabe [83]. This usually involves the definition of distributive laws between a monad $\mathbb{M}$ and a comonad $\mathbb{T}$ which are natural transformations of the form $\lambda\colon \mathbb{TM} \implies \mathbb{MT}$ satisfying certain laws. Forthcoming work of Amin Karamlou and Nihil Shah has investigated the existence of such distributive laws for game comonads and could be instrumental in answering the following question.

Question 9.6. *Can we formally relate, via a distributive law or other natural transformation, the pebbling comonad, $\mathbb{P}_k$, to the monads $\mathbb{V}_{\mathbf{S}}$ or $\mathbb{D}_{\mathbf{S}}$?*
If so, does the resulting joint structure provide a way to give a semantics to a form of linear algebraic logic?

Another question relates to the sheaf-theoretic approach in Chapter 8. We saw in that chapter that this formalism was able to characterise the relations $\to_k$ and $\equiv_k$ which are captured by the morphisms and isomorphisms in $\mathcal{K}(\mathbb{P}_k)$. It is therefore interesting to ask whether the cohomological algorithms defined in that chapter can be captured by comonads or similar constructions.

Question 9.7. *Does there exist a comonad $\mathbb{C}_k$ for which the notion of morphism and isomorphism in the Kleisli category are $\to_k^{\mathbb{Z}}$ and $\equiv_k^{\mathbb{Z}}$?*
If not, is there some other way to capture these relations using a combination of monads and comonads?

Cohomology and `PTIME` The cohomological algorithms introduced in Chapter 8 for approximating homomorphism and isomorphism appear to be truly novel approaches to the problems of constraint satisfaction and structure isomorphism. In light of recent advances in these fields we ask how these algorithms compare to state-of-the-art approaches to these problems.
Bulatov and Zhuk's recent independent resolutions of the Feder-Vardi conjecture [25, 91], show that for all domains $\mathcal{B}$ either $\mathbf{CSP}(\mathcal{B})$ is `NP-Complete` or $\mathcal{B}$ admits a weak near-unanimity polymorphism and $\mathbf{CSP}(\mathcal{B})$ is tractable. As the cohomological k-consistency algorithm expands the power of the k-consistency algorithm which features as one case of Bulatov and Zhuk's general efficient algorithms, we ask if it is sufficient to decide all tractable **CSP**s.

Question 9.8. *For all domains $\mathcal{B}$ which admit a weak near-unanimity polymorphism, does there exists a k such that for all $\mathcal{A}$*

$$\mathcal{A} \to \mathcal{B} \iff \mathcal{A} \to_k^{\mathbb{Z}} \mathcal{B}?$$

As cohomological k-Weisfeiler-Leman is an efficient algorithm for distinguishing some non-isomorphic relational structures we ask if it distinguishes all non-isomorphic structures. As the best known structure isomorphism algorithm is quasi-polynomial [15], we do not expect a positive answer to this question but expect that negative answers would aid our understanding of the hard cases of structure isomorphism in general.

Question 9.9. *For every signature σ does there exists a k such that for all σ-structures $\mathcal{A}, \mathcal{B}$*

$$\mathcal{A} \cong \mathcal{B} \iff \mathcal{A} \equiv_k^{\mathbb{Z}} \mathcal{B}?$$

Bibliography

[1] ABRAMSKY, S. Notes on presheaf representations of strategies and cohomological refinements of k-consistency and k-equivalence, 2022.

[2] ABRAMSKY, S. Structure and power: an emerging landscape. *Fundam. Informaticae 186*, 1-4 (2022), 1–26.

[3] ABRAMSKY, S., BARBOSA, R. S., DE SILVA, N., AND ZAPATA, O. The Quantum Monad on Relational Structures. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)* (Dagstuhl, Germany, 2017), K. G. Larsen, H. L. Bodlaender, and J.-F. Raskin, Eds., vol. 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 35:1–35:19.

[4] ABRAMSKY, S., BARBOSA, R. S., KISHIDA, K., LAL, R., AND MANSFIELD, S. Contextuality, Cohomology and Paradox. In *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)* (Dagstuhl, Germany, 2015), S. Kreutzer, Ed., vol. 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 211–228.

[5] ABRAMSKY, S., AND BRANDENBURGER, A. The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics 13*, 11 (Nov 2011), 113036.

[6] ABRAMSKY, S., DAWAR, A., AND WANG, P. The pebbling comonad in finite model theory. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (2017), pp. 1–12.

[7] ABRAMSKY, S., JAKL, T., AND PAINE, T. Discrete density comonads and graph parameters, 2022.

[8] ABRAMSKY, S., AND MARSDEN, D. Comonadic semantics for guarded fragments. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (Los Alamitos, CA, USA, Jul 2021), IEEE Computer Society, pp. 1–13.

[9] ABRAMSKY, S., AND MARSDEN, D. Comonadic semantics for hybrid logic. In *47th International Symposium on Mathematical Foundations of Computer Science*

(MFCS 2022) (Dagstuhl, Germany, 2022), S. Szeider, R. Ganian, and A. Silva, Eds., vol. 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 7:1–7:14.

[10] Abramsky, S., and Reggio, L. Arboreal Categories and Resources. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)* (Dagstuhl, Germany, 2021), N. Bansal, E. Merelli, and J. Worrell, Eds., vol. 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 115:1–115:20.

[11] Abramsky, S., and Shah, N. Relating structure and power: Comonadic semantics for computational resources. *Journal of Logic and Computation 31*, 6 (08 2021), 1390–1428.

[12] Aigner, M., and Fromme, M. A game of cops and robbers. *Discrete Applied Mathematics 8*, 1 (1984), 1–12.

[13] Atserias, A., Bulatov, A., and Dalmau, V. On the power of k-consistency. In *Automata, Languages and Programming* (Berlin, Heidelberg, 2007), L. Arge, C. Cachin, T. Jurdziński, and A. Tarlecki, Eds., Springer Berlin Heidelberg, pp. 279–290.

[14] Atserias, A., Bulatov, A., and Dawar, A. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science 410*, 18 (2009), 1666–1683. Automata, Languages and Programming (ICALP 2007).

[15] Babai, L. Graph isomorphism in quasipolynomial time. *CoRR abs/1512.03547* (2015).

[16] Babai, L., and Luks, E. M. Canonical labeling of graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1983), STOC '83, Association for Computing Machinery, p. 171–183.

[17] Baldwin, J. Finite and infinite model theory-a historical perspective. *Logic Journal of the IGPL 8*, 5 (2000), 605–628.

[18] Barto, L., Bulín, J., Krokhin, A., and Opršal, J. Algebraic approach to promise constraint satisfaction. *J. ACM 68*, 4 (Jul 2021).

[19] Barto, L., and Kozik, M. Constraint satisfaction problems of bounded width. pp. 595–603.

[20] Bojanczyk, M., Klin, B., Lasota, S., and Torunczyk, S. Turing machines with atoms. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science* (2013), pp. 183–192.

[21] Bonato, A., and Nowakowski, R. *The Game of Cops and Robbers on Graphs.* Student mathematical library. American Mathematical Society, 2011.

[22] Brady, Z. Notes on CSPs and Polymorphisms, 2022.

[23] Brakensiek, J., Guruswami, V., Wrochna, M., and Živný, S. The power of the combined basic linear programming and affine relaxation for promise constraint satisfaction problems. *SIAM Journal on Computing 49*, 6 (2020), 1232–1248.

[24] Bulatov, A., Jeavons, P., and Krokhin, A. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing 34*, 3 (2005), 720–742.

[25] Bulatov, A. A. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017), pp. 319–330.

[26] Butti, S., and Dalmau, V. Fractional homomorphism, weisfeiler-leman invariance, and the sherali-adams hierarchy for the constraint satisfaction problem. In *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia* (2021), F. Bonchi and S. J. Puglisi, Eds., vol. 202 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 27:1–27:19.

[27] Cai, J.-Y., Fürer, M., and Immerman, N. An optimal lower bound on the number of variables for graph identification. *Combinatorica 12*, 4 (Dec. 1992), 389–410.

[28] Caru, G. *Logical and topological contextuality in quantum mechanics and beyond.* PhD thesis, University of Oxford, 2019.

[29] Chandra, A., and Harel, D. Structure and complexity of relational queries. *Journal of Computer and System Sciences 25*, 1 (1982), 99–128.

[30] Chen, H., Dalmau, V., and Grussien, B. Arc consistency and friends. *Journal of Logic and Computation 23*, 1 (Nov. 2011), 87–108.

[31] Dalmau, V. Linear Datalog and Bounded Path Duality of Relational Structures. *Logical Methods in Computer Science Volume 1, Issue 1* (Apr. 2005).

[32] DAWAR, A. Generalized Quantifiers and Logical Reducibilities. *Journal of Logic and Computation 5*, 2 (Apr. 1995), 213–226.

[33] Dawar, A. Constraint Satisfaction, Graph Isomorphism, and the Pebbling Comonad, 2022.

[34] Dawar, A., Grädel, E., and Pakusa, W. Approximations of Isomorphism and Logics with Linear-Algebraic Operators. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)* (Dagstuhl, Germany, 2019), C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, Eds., vol. 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 112:1–112:14.

[35] Dawar, A., Grohe, M., Holm, B., and Laubner, B. Logics with rank operators. In *Proceedings of the Twenty-Fourth Annual IEEE Symposium on Logic in Computer Science (LICS 2009)* (August 2009), IEEE Computer Society Press, pp. 113–122.

[36] Dawar, A., Grädel, E., and Lichter, M. Limitations of the invertible-map equivalences. *Journal of Logic and Computation* (Sept. 2022).

[37] Dawar, A., and Holm, B. Pebble games with algebraic rules. *Fundamenta Informaticae Vol. 150, nr 3/4* (2017), 281–316.

[38] Dawar, A., Jakl, T., and Reggio, L. *Lovász-Type Theorems and Game Comonads.* Association for Computing Machinery, New York, NY, USA, 2021.

[39] Dawar, A., and Wilsenach, G. Lower Bounds for Symmetric Circuits for the Determinant. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)* (Dagstuhl, Germany, 2022), M. Braverman, Ed., vol. 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 52:1–52:22.

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

[41] Ebbinghaus, H.-D. Extended logics: The general framework. In *Model-Theoretic Logics*, J. Barwise and S. Feferman, Eds. Springer-Verlag, New York, 1985, pp. 25–76.

[42] Ehrenfeucht, A. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae 49*, 2 (1961), 129–141.

[43] Fagin, R. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation 7* (1974), 43–73.

[44] Feder, T., and Vardi, M. Y. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing 28*, 1 (1998), 57–104.

[45] Flum, J., and Grohe, M. On fixed-point logic with counting. *The Journal of Symbolic Logic 65*, 2 (2000), 777–787.

[46] Foniok, J., and Tardif, C. Digraph functors which admit both left and right adjoints. *Discrete Mathematics 338*, 4 (2015), 527–535.

[47] Fritz, T. Convex spaces i: Definition and examples, 2009.

[48] Gehrke, M., Petrişan, D., and Reggio, L. Quantifiers on languages and codensity monads. *Mathematical Structures in Computer Science 30*, 10 (2020), 1054–1088.

[49] Godement, R. Topologie algébrique et théorie des faisceaux. *The Mathematical Gazette 44* (1960), 69.

[50] Grohe, M. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, vol. 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017.

[51] Grohe, M. Counting bounded tree depth homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science* (New York, NY, USA, 2020), LICS '20, Association for Computing Machinery, p. 507–520.

[52] GROHE, M., and OTTO, M. Pebble games and linear equations. *The Journal of Symbolic Logic 80*, 3 (2015), 797–844.

[53] Grothendieck, A. Sur quelques points d'algèbre homologique, I. *Tohoku Mathematical Journal 9*, 2 (1957), 119 – 221.

[54] Gurevich, Y. *Logic and the Challenge of Computer Science*. Computer Science Press, July 1988, pp. 1–57.

[55] Hall, P. On representatives of subsets. *Journal of the London Mathematical Society s1-10*, 1 (1935), 26–30.

[56] Halpern, J. Y., Harper, R., Immerman, N., Kolaitis, P. G., Vardi, M. Y., and Vianu, V. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic 7*, 2 (2001), 213–236.

[57] Hell, P., and Nešetřil, J. The core of a graph. *Discrete Mathematics 109*, 1 (1992), 117 – 126.

[58] Hella, L. Logical hierarchies in PTIME. *Information and Computation 129*, 1 (1996), 1 – 19.

[59] Immerman, N. *Descriptive complexity.* Graduate texts in computer science. Springer, 1999.

[60] Immerman, N., and Lander, E. *Describing Graphs: A First-Order Approach to Graph Canonization.* Springer New York, New York, NY, 1990, pp. 59–81.

[61] Jacobs, B. Probabilities, distribution monads, and convex categories. *Theoretical Computer Science 412*, 28 (2011), 3323–3336. Festschrift in Honour of Jan Bergstra.

[62] Jakl, T., Marsden, D., and Shah, N. A game comonadic account of Courcelle and Feferman-Vaught-Mostowski theorems, 2022.

[63] Jeavons, P., Cohen, D., and Gyssens, M. A unifying framework for tractable constraints. In *Principles and Practice of Constraint Programming — CP '95* (Berlin, Heidelberg, 1995), U. Montanari and F. Rossi, Eds., Springer Berlin Heidelberg, pp. 276–291.

[64] Joyal, A., Nielsen, M., and Winskel, G. Bisimulation and open maps. In *In Proc. LICS'93, Eighth Annual Symposium on Logic in Computer Science* (1993), pp. 418–427.

[65] Kannan, R., and Bachem, A. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM Journal on Computing 8*, 4 (11 1979), 499–9.

[66] Kiefer, S. *Power and limits of the Weisfeiler-Leman algorithm.* Dissertation, RWTH Aachen University, Aachen, 2020. Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2020.

[67] Kolaitis, P., and Vardi, M. On the expressive power of Datalog: Tools and a case study. *Journal of Computer and System Sciences 51*, 1 (1995), 110–134.

[68] Kolaitis, P. G., and Väänänen, J. A. Generalized quantifiers and pebble games on finite structures. *Annals of Pure and Applied Logic 74*, 1 (1995), 23 – 75.

[69] Krokhin, A. A., Oprsal, J., Wrochna, M., and Zivný, S. Topology and adjunction in promise constraint satisfaction. *Electron. Colloquium Comput. Complex. TR20-040* (2020).

[70] Leinster, T. *Categories, functors and natural transformations.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2014, p. 9–40.

[71] Libkin, L. *Elements Of Finite Model Theory (Texts in Theoretical Computer Science. An Eatcs Series).* SpringerVerlag, 2004.

[72] Lichter, M. Separating rank logic from polynomial time. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021* (2021), IEEE, pp. 1–13.

[73] Lindström, P. First order predicate logic with generalized quantifiers. *Theoria 32*, 3 (1966), 186–195.

[74] Lovász, L. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica 18*, 3 (Sept. 1967), 321–328.

[75] Luks, E. M. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences 25*, 1 (1982), 42–65.

[76] Mac Lane, S., and Moerdijk, I. *Sheaves of Sets*. Springer New York, New York, NY, 1994, pp. 64–105.

[77] Montacute, Y., and Shah, N. The pebble-relation comonad in finite model theory, 2021.

[78] Ó Conghaile, A. Cohomology in constraint satisfaction and structure isomorphism. In *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria* (2022), S. Szeider, R. Ganian, and A. Silva, Eds., vol. 241 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 75:1–75:16.

[79] Ó Conghaile, A., and Dawar, A. Game comonads & generalised quantifiers. *CoRR abs/2006.16039* (2020).

[80] Ó Conghaile, A., and Dawar, A. Game Comonads & Generalised Quantifiers. In *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)* (Dagstuhl, Germany, 2021), C. Baier and J. Goubault-Larrecq, Eds., vol. 183 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 16:1–16:17.

[81] Paine, T. A pebbling comonad for finite rank and variable logic, and an application to the equirank-variable homomorphism preservation theorem. *Electronic Notes in Theoretical Computer Science 352* (2020), 191–209. The 36th Mathematical Foundations of Programming Semantics Conference, 2020.

[82] Pakusa, W. *Linear Equation Systems and the Search for a Logical Characterisation of Polynomial Time*. PhD thesis, RWTH Aachen University, 2016.

[83] Power, J., and Watanabe, H. Combining a monad and a comonad. *Theoretical Computer Science 280*, 1 (2002), 137–162. Coalgebraic Methods in Computer Science.

[84] Ramana, M. V., Scheinerman, E. R., and Ullman, D. Fractional isomorphism of graphs. *Discrete Mathematics 132*, 1 (1994), 247–265.

[85] Read, R. C., and Corneil, D. G. The graph isomorphism disease. *Journal of Graph Theory 1*, 4 (1977), 339–363.

[86] Robertson, N., and Seymour, P. Graph minors. iii. Planar tree-width. *Journal of Combinatorial Theory, Series B 36*, 1 (1984), 49–64.

[87] Tinhofer, G. Graph isomorphism and theorems of Birkhoff type. *Computing 36*, 4 (Dec. 1986), 285–300.

[88] Tinhofer, G. A note on compact graphs. *Discrete Applied Mathematics 30*, 2 (1991), 253–264.

[89] Watanabe, K., Eberhart, C., Asada, K., and Hasuo, I. A compositional approach to parity games. In *Proceedings 37th Conference on Mathematical Foundations of Programming Semantics, MFPS 2021, Hybrid: Salzburg, Austria and Online, 30th August - 2nd September, 2021* (2021), A. Sokolova, Ed., vol. 351 of *EPTCS*, pp. 278–295.

[90] Weisfeiler, B., and Lehman, A. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia 9*, 9 (1968), 12 – 16. Cited by: 348.

[91] Zhuk, D. A proof of the CSP dichotomy conjecture. *J. ACM 67*, 5 (aug 2020).