Homomorphism Counts, Database Queries, and Modal Logics

MSc Thesis *(Afstudeerscriptie)*

written by

Jesse A. Comer
(born March 24th, 1995 in Oceanside, California)

under the supervision of **Dr. Balder ten Cate**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
July 12, 2023	Dr. Malvin Gattinger (chair)
	Dr. Balder ten Cate (supervisor)
	Dr. Nick Bezhanishvili
	Dr. Ronald de Haan

Institute for Logic, Language and Computation

Abstract

We study applications of restrictions of *homomorphism vectors* for finite relational models in database theory and modal logic. Assume some fixed enumeration $(M_i)_{i\in\omega}$ of all finite relational models (up to isomorphism), and let M be some finite relational model. The left homomorphism vector of M is the countably infinite vector whose i^{th} entry is the number of homomorphisms from M_i to M. Similarly, the right homomorphism vector of M is the countably infinite vector whose i^{th} entry is the number of homomorphisms from M to M_i. A *restriction* of a homomorphism vector is a vector obtained by removing some of the entries.

In particular, we study *(right) finite characterizations*, which are restrictions of the *right homomorphism vector* of a model M to a finite number of entries which characterize M up to isomorphism. Interpreting the characterized model as a *canonical model* of a *conjunctive query*, a finite characterization can equivalently be seen as a collection of database instances, where the answers to the query in each instance of the collection under the *bag semantics* determine the query up to isomorphism. Given an arbitrary finite model M, we construct a finite characterization of M containing at most exponentially-many examples, each of which has domain size bounded by the domain size of M. We also construct polynomially-large finite characterizations for certain special classes of models.

Additionally, we study which relations preserving different modal languages can be captured by restricting the left homomorphism vector of labeled transition systems. In this vein, we show that simulation and graded bisimulation are captured by the restriction of the left homomorphism vector to the class of *directed tree-shaped* labeled transition systems under the Boolean and counting semantics, respectively. We then lift these results to show that modal languages with backward and global modalities are captured by the restriction of the left homomorphism vector to appropriate classes of *tree-shaped* and *directed forest* models, respectively. We also show that no relation finer than directed simulation and coarser than bisimulation can be captured by a restriction of the left homomorphism vector under the Boolean or the counting semantics.

Acknowledgements

I would like to express my deepest gratitude to Balder, whose mentoring and guidance, both prior to and during the writing of this thesis, have been invaluable to my learning and professional development. Your dedication, expertise, and support have shaped my academic journey, from the first coordinated research project, through the individual project, the PhD admissions cycle, and finally to this thesis. I'm thankful to have found such an invested advisor. I would also like to thank Tanja Kassenar, Paul Dekker, and Yde Venema, who, through their understanding and flexibility, have made it possible for me to complete the Master of Logic.

Contents

Chapter 1

Introduction

Homomorphisms and Homomorphism Vectors

This thesis is primarily concerned with *finite model theory*, defined in our case as the study of logical languages interpreted over the class of *finite relational structures* (or *finite models*). The simplest example of such a structure is a directed graph, although in general we allow relations of arbitrary arity, each corresponding to a particular relation symbol in our formal language. Between structures of the same signature, there may or may not exist one or more structure-preserving maps, called *homomorphisms* (cf. Figure 1.1). A map is *structure-preserving* if, whenever a tuple of elements in the first structure occurs in some relation, then the image of that tuple must also occur in the corresponding relation in the second structure. Homomorphisms need not be injective nor surjective, and the relations only need to be preserved from the first structure to the second. Thus homomorphisms generalize *isomorphisms*, which must be bijective and preserve relations in both directions.

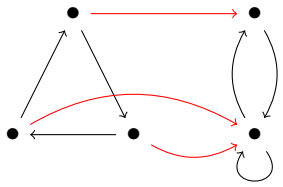


Figure 1.1: A homomorphism (in red) between two directed graphs.

Given two finite models, a natural question is: does a homomorphism exist between them? And if so, how many? These are the *homomorphism existence* and *homomorphism counting* problems, respectively. Our primary object of study will be *homomorphism vectors*, which encode the answers to the homomorphism counting problem between a fixed model M and all other finite models. More explicitly, assume some fixed enumeration $(M_i)_{i\in\omega}$ of all finite models. The left homomorphism vector (or *left profile*) of M is the countably infinite vector whose i^{th} entry is the number of homomorphisms from M_i to M. Similarly, the right homomorphism vector (or *right profile*) of M is the countably infinite vector whose i^{th} entry is the number of homomorphisms from M to M_i. Note that we use the terms "homomorphism vector" and "profile" interchangeably. The terms "left" and "right" indicate whether we are considering homomorphism in to or out of the model M (cf. figure 1.2).

In our initial presentation, we have defined each entry of a homomorphism vector as an answer to the homomorphism counting problem. However, homomorphism vectors can be defined with regard to various "semantics." In some sense, the most natural semantics

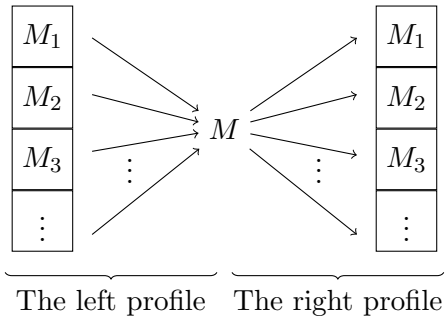


Figure 1.2: A depiction of the left and right profiles of a model M

is the one in which the counting is done over the domain of the natural numbers. This is the *counting semantics*, in which each entry of the vector corresponds to an answer to an instance of the homomorphism counting problem, as seen above. We could also define homomorphism vectors in which counting is done over only Boolean values, where all entries greater than 0 are rounded to 1. This is the *Boolean semantics*, in which each entry of the vector corresponds to an answer to an instance of the homomorphism existence problem. In general, we can count over the elements of an arbitrary *semiring*.

An immediate question of interest is: what does it mean for two models to have the same homomorphism vector under some fixed semantics? In the case of the Boolean semantics, two models have the same homomorphism count vector (either left or right) if and only if there exist homomorphisms in both directions between them (i.e., they are *homomorphically equivalent)* [AKW21]. In the case of the counting semantics, a well-known theorem of Lovász states that two finite relational models are isomorphic if and only if their left profiles are identical [Lov67]. Similarly, a theorem of Chaudhuri and Vardi states that isomorphism between models is also characterized by right profile equivalence [CV93]. However, one might ask: in order to verify isomorphism between two models, do we need to check equivalence of the entire right profile, or could we check only a finite number of entries? If so, how many entries do we need to check? Another natural question is: which equivalence relations strictly weaker than isomorphism can be captured by restricting the left profile? Our goal is to shed light on these kinds of questions.

Up to this point, we have discussed our subject matter entirely from the perspective of finite relational models and morphisms between them. From a technical standpoint, this is entirely sufficient, and will be our primary perspective throughout the majority of this thesis. However, the motivation for the questions investigated come from database theory and process theory. In the case of database theory, this is because database instances and queries can be represented as finite relational models and logical formulas, respectively. In the case of process theory, we can view finite-state computational processes as finite relational models, and we can describe properties of these processes using logical formulas.

Finite Characterizations of Conjunctive Queries

Modern database systems are increasingly large and complex, and the queries posed by users to these databases have correspondingly grown in size and complexity. As a result, it can become increasingly difficult for users to write queries with the intended semantic meaning. Furthermore, users may outsource query-writing to artificial intelligence language models to generate a query with a particular meaning described in natural language. However, it may be difficult to verify correctness of such a query. One solution, given an AI-generated query, is to automatically generate a finite number of pairs of labeled

database examples where the label is the answer to Q when it is posed to the database example. If we can guarantee that any query Q' which produces the same answer as Q on each database example must be answer-equivalent on all databases, then such a finite collection of examples could be used to verify the correctness of Q. We refer to such a collection as a *finite characterization* of Q.

Another application of a finite characterization is in exact learning of queries. This is a supervised learning context in which the learner maintains a "guess" regarding the query Q to be learned, updating it iteratively after each example. At each training step, the guess must be consistent with all examples seen up to that point (i.e., it must return the labeled answer). In exact learning, the goal is for the learning algorithm to halt, after some finite number of training examples, with the final guess for the query being exactly the desired query (up to answer-equivalence). A necessary condition for exact learning to be possible is that there exists a *finite* collection of training examples such that any query Q' which is consistent with the collection must be answer-equivalent to Q. Otherwise, there will be some non-equivalent query whose answers match Q on all of the training examples, in which case the algorithm is not guaranteed to halt with the correct query.

In database theory, an important class of database queries are the *conjunctive queries*, equivalent to the formulas of first-order logic using only conjunction and existential quantification. Because of this equivalence, the task of a database query language, when determining an answer to a query in some database instance, is equivalent to determining the truth value of a logical formula in a finite model M representing the database instance. In the case of conjunctive queries, this problem turns out to be closely related to the homomorphism problem. For each conjunctive query Q, up to query equivalence, there exists a unique model M, called the *canonical model* of Q, such that a satisfying assignment for Q in a model N corresponds precisely to an appropriate homomorphism from M to N. Under the usual semantics of first-order logic, this means that the problem of determining whether or not a query is satisfied by some tuple in a model reduces to the homomorphism existence problem.

Most work in database theory assumes *set semantics*, in which the answer to a query in a database instance is the set of tuples satisfying the query. In [CV93], Chaudhuri and Vardi pointed out that this assumption was not applicable in most database languages. Instead, languages like SQL return multi-sets of tuples satisfying the query, where the multiplicity of a tuple is the number of assignments witnessing that the tuple satisfies the query. We refer to this as *bag semantics*. This extends the usual semantics of first-order logic for conjunctive queries: the truth value of a formula is not just a Boolean value (i.e., an indicator of the existence of a satisfying assignment), but is actually the number of satisfying assignments. Since satisfying assignments for conjunctive queries correspond to homomorphisms, this implies that the problem of determining the multiplicity of a tuple in the answer to a query (under the bag semantics) reduces to the homomorphism counting problem.

Because of the correspondence between conjunctive queries and finite relational models, we can define the left and right homomorphism vectors of a conjunctive query as those of its canonical model. Furthermore, due to the relationship between satisfying assignments and homomorphisms, we can now take a different perspective on these vectors. Suppose Q is a conjunctive query and I is a database instance which is represented by the canonical model of Q. Under the Boolean semantics, the left profile indicates which conjunctive queries are satisfied in I, while the right profile indicates which database instances satisfy Q. Similarly, under the counting semantics, the left profile indicates the number of satisfying assignments for each query in I, while the right profile indicates the

number of satisfying assignments for Q in all other models. This implies that two conjunctive queries have the same left (or right) profiles if and only if the results of evaluating those queries on all database instances under the bag semantics are identical.

The right profile of a conjunctive query is particularly interesting, representing the answers to that query in all database instances. However, this is not particularly useful from a computational perspective: the right profile is infinite. However, recall our earlier question: can we characterize a model up to isomorphism by considering only a finite number of entries of its right profile? Restated in terms of queries and databases, this question becomes: can we characterize a query up to answer-equivalence (under the bag semantics) with only a finite collection of labeled examples, where an example is a database instance, and the associated label is the number of satisfying assignments for the query in that instance? If so, can we derive bounds on the number and size of the examples needed? Answering these questions regarding *finite characterizations* under the bag semantics is the focus of Chapter 3.

Characterizing Modal Equivalence Relations by Restricting the Left Profile

Lovász's theorem states that isomorphism between finite models is captured by left profile equivalence. It is natural to ask if there are other relations weaker than isomorphism which can be captured by some restriction of the left profile to a smaller class of structures; as it turns out, there are interesting results in this area. The most striking results show that the left profile of an undirected graph restricted to the class of trees characterizes four seemingly disparate notions from logic, linear algebra, graph theory, and artificial intelligence (see Section 4.1 for an overview of these results).

Most of the literature in this area has confined attention to undirected graphs. However, many other natural structures appearing in computer science applications are not best modeled by graphs. In particular, *labeled transition systems* are models with a finite number of *directed* binary relations, called *actions*, as well as a finite number of unary predicates, called *proposition letters*. In this setting, elements of the domain of a model might represent possible states of a computational process. The unary predicates would then represent properties of a given state, while the binary predicates represent transitions from one state to another.

In the context of labeled transition systems, isomorphism between structures is not particularly interesting. Instead, there are more natural notions of process equivalence, such as *simulation*, *directed simulation*, *bisimulation*, or *graded bisimulation*. These notions are known to correspond, over the class of finite labeled transition systems, to invariance of structures over different *modal logics*. These modal logics are expressive enough to describe many interesting properties of processes, and are useful due to their well-known computational qualities. The focus of Chapter 4 is to determine whether or not these *modal equivalence relations* can be characterized by restricting the left profiles of labeled transition systems.

Chapter 2

Preliminaries

2.1 Notation and Basic Definitions for First-Order Logic

In this section, we will establish our notational conventions and basic definitions for first-order logic (FO). We use σ and τ to denote first-order signatures. Throughout this thesis, we will confine attention to relational signatures: those with relation symbols, but no constants or function symbols. We write $\varphi, \psi, \chi, \gamma$ to denote arbitrary FO-formulas, and α, β to denote atomic formulas. We will also refer to atomic formulas as facts.

We typically use x_i, y_i, z_i to denote variables and $a_i, b_i, c_i, d_i, m_i, n_i$ to denote objects in models. We write Vars for the set of all variables. We will frequently drop the subscripts on variables and objects. When working with canonical models (to be defined later), variables will also appear as elements of models; this will be clear from the context. We denote tuples of variables with $\overline{x}, \overline{y}, \overline{z}, \overline{u}, \overline{v}$, and tuples of elements of models with $\overline{a}, \overline{b}, \overline{c}, \overline{d}, \overline{m}, \overline{n}$. We will frequently make the free variables of a formula φ explicit by writing $\varphi(x_1, \ldots, x_n)$ or $\varphi(\overline{x})$.

We use A, B, C, M, N, T to denote models. Given a σ-model M, we write $\mathrm{dom}(M)$ for the domain of M, and for each relation symbol $R \in \sigma$, we write R^M for the interpretation of R in M. If $(a_1, \ldots, a_k) \in R^M$, then we say that $R^M(a_1, \ldots, a_k)$ holds, and say that $R(a_1, \ldots, a_k)$ is a *fact* of M. We write $\mathrm{Facts}(M)$ for the collection of all facts of M, and $\mathrm{Rel}(M)$ for the collection of all possible facts on M. For an arbitrary fact α of a model M, we write $el(\alpha)$ for the tuple of elements occurring in α, and $rel(\alpha)$ for the relation symbol occurring in α.

We frequently work with models with *distinguished elements*. Given a tuple $\overline{a} = a_1, \ldots, a_n$ of (not necessarily distinct) elements of M, we write $(M, a_1, \ldots, a_n)$ or $(M, \overline{a})$ to denote the model M along with distinguished elements $a_1, \ldots, a_n$. All elements not among $a_1, \ldots, a_n$ are *non-distinguished elements* of $(M, a_1, \ldots, a_n)$. In particular, a model M without designated distinguished elements has only non-distinguished elements. Given a fixed relational signature σ, we write $\mathcal{M}_n^\sigma$ for the class of all finite σ-models with n distinguished elements.

If $A \subseteq dom(M)$ contains all distinguished elements of $(M, \overline{a})$, then we write $(A, \overline{a})$ to denote the submodel of M induced by S (i.e., the model with domain A and relations restricted to A). When no confusion results, we will also sometimes equate a model with its domain. We are interested only in finite models over relational signatures; as a result, we will assume that all models are finite and non-empty unless explicitly stated.

We write $M \models \varphi$ to indicate that the model M satisfies an FO-formula φ. Given a formula $\varphi(x_1, \ldots, x_k)$, we write $M \models \varphi(a_1, \ldots, a_k)$ to indicate that the model M satisfies the formula $\varphi(x_1, \ldots, x_k)$ with the variable assignment $x_i \mapsto a_i$ for $1 \leq i \leq k$. We will

write $M, g \models \varphi$ to indicate that M satisfies φ under the variable assignment g. If $M \models \varphi$ whenever $M \models \psi$, then we write $\psi \models \varphi$. If $\psi \models \varphi$ and $\varphi \models \psi$, we write $\varphi \equiv \psi$.

Definition 2.1.1. Let $(M, \overline{a})$ and $(N, \overline{b})$ be models, where $\overline{a} = a_1 \dots a_k$ and $\overline{b} = b_1 \dots b_k$.

1. A *homomorphism* from $(M, \overline{a})$ to $(N, \overline{b})$ is a map $h : dom(M) \to dom(N)$ such that $a_i \mapsto b_i$ for $i \leq k$, and $R^M(m_1, \dots, m_n)$ implies $R^N(h(m_1), \dots, (m_n))$ for all n-ary relation symbols R.

2. An *isomorphism* from $(M, \overline{a})$ to $(N, \overline{b})$ is a bijective homomorphism $f : \mathrm{dom}(M) \to \mathrm{dom}(N)$ whose inverse is also a homomorphism.

We refer to a homomorphism $h : (M, \overline{a}) \to (M, \overline{a})$ as an *endomorphism*. Let $h : (M, \overline{a}) \to (N, \overline{b})$ be a homomorphism. We say that h is *injective* when, for all elements $a, b \in M$, we have that $h(a) = h(b)$ only if $a = b$. We say that h is *surjective* when, for each $b \in N$, there's some $a \in M$ such that $h(a) = b$. We say that h is *fully surjective* if it is surjective and if, for all relation symbols R in the signature, if $(b_1, \dots, b_k) \in R^N$, then there exists a tuple $(a_1, \dots, a_k) \in R^M$ such that $(h(a_1), \dots, h(a_k)) = (b_1, \dots, b_k)$.

If there exists a homomorphism $h : (M, \overline{a}) \to (N, \overline{a})$, then we say that $(M, \overline{a})$ *homomorphically maps to* $(N, \overline{a})$ (notation: $(M, \overline{a}) \to (N, \overline{a})$). We say that $(M, \overline{a})$ and $(N, \overline{a})$ are *homomorphically equivalent* (notation: $(M, \overline{a}) \rightleftarrows (N, \overline{a})$) if $(M, \overline{a}) \to (N, \overline{a})$ and $(N, \overline{a}) \to (M, \overline{a})$. Clearly homomorphic equivalence is symmetric, and the identity map is an endomorphism on any model. Then because homomorphisms are preserved under functional composition, we have that homomorphic equivalence constitutes an equivalence relation on models. If there exists an isomorphism $f : (M, \overline{a}) \to (N, \overline{a})$, then we say that $(M, \overline{a})$ and $(N, \overline{a})$ are *isomorphic* (notation: $(M, \overline{a}) \cong (N, \overline{a})$. As with homomorphic equivalence, isomorphism also induces an equivalence relation on models.

2.2 Conjunctive Queries

The primary type of FO-formula of interest to us is the *conjunctive query* (CQ), which is a formula of the form

$$\varphi(x_1, \dots, x_n) := \exists y_1 \dots \exists y_m (\bigwedge_{i \in I} \alpha_i),$$

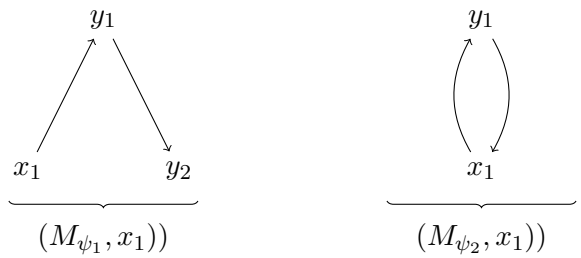
where each α_i is an atomic relation, possibly an equality, whose free variables are among $\{x_1, \dots, x_n, y_1, \dots, y_m\}$. We further assume that every CQ satisfies the *safety condition*, which requires that every variable which occurs in the formula also occurs in some (non-equality) atomic relation in the formula. The arity of a conjunctive query is the number of free variables in the query. We write CQ for the class of all conjunctive queries, and CQ^k for the class of all k-ary conjunctive queries.

We define the *canonical model* of a conjunctive query $\varphi(x_1, \dots, x_n)$ to be the model $(M_\varphi, x_1, \dots, x_n)$ whose elements are the variables of the query and whose relations are defined by the atomic formulas of the query. We also take the free variables to be distinguished elements of the model. Furthermore, we define the *canonical query* of a model $(M, a_1, \dots, a_n)$ to be the conjunctive query $\varphi_M(x_1, \dots, x_n)$ whose free variables correspond to $a_1, \dots, a_n$, whose bound variables correspond to the number of non-distinguished elements of the model, and whose facts are those occurring in M, replacing elements of each fact by the appropriate variables.

Example 2.2.1. Consider the queries

$$\psi_1(x_1) := \exists y_1 \exists y_2 \left(R(x_1, y_1) \wedge R(y_1, y_2)\right), \text{ and}$$
$$\psi_2(x_1) := \exists y_1 \left(R(x_1, y_1) \wedge R(y_1, x_1)\right).$$

The canonical models of these queries are (M_{ψ_1}, x_1) and (M_{ψ_2}, x_1), depicted below:



The canonical model and canonical query constructions are inverses of each other, and so we have a bijective correspondence between the collection of conjunctive queries and the collection of finite relational models (up to isomorphism). The following well-known result creates a correspondence between satisfying assignments to conjunctive queries and homomorphisms from the canonical model of the query. We include the proof because it is illustrative of a common theme throughout this thesis.

Lemma 2.2.2 (Magic Lemma, [CV93])**.** Given a CQ $\varphi(\overline{x})$ and a model $(M, \overline{a})$, the following are equivalent:

1. $M \models \varphi(\overline{a})$.
2. There exists a homomorphism $h : (M_\varphi, \overline{x}) \to (M, \overline{a})$.

Proof.
Let $\varphi(\overline{x}) := \exists y_1 \ldots \exists y_m(\bigwedge_{i \in I} \alpha_i)$. For the forward direction, suppose that $(M, \overline{a}) \models \varphi(\overline{x})$. Let h be some satisfying variable assignment such that $h(x_i) = a_i$ for each $a_i \in \overline{a}$. We claim that $h : (M_\varphi, \overline{x}) \to (M, \overline{a})$ is a homomorphism. To see this, consider some fact α_i of $(M_\varphi, \overline{x})$. By construction, α_i is a fact of $\varphi(\overline{x})$, and so $M, h \models \alpha_i$. It then follows that h is a homomorphism. We argue similarly for the reverse direction, showing that any homomorphism $g : (M_\varphi, \overline{x}) \to (M, \overline{a})$ is an assignment such that $M, g \models \varphi(\overline{x})$ (and hence $M \models \varphi(\overline{a})$). □

Due to the fact that homomorphisms are preserved under composition, the preceding lemma can be extended further, showing that the existence of a homomorphism between two canonical models equates to logical implication between the appropriate queries.

Theorem 2.2.3. (Chandra-Merlin Theorem, [CV93]) Given CQs $\varphi(\overline{x})$ and $\psi(\overline{x})$, the following are equivalent:

1. $\psi(\overline{x}) \models \varphi(\overline{x})$.
2. There exists a homomorphism $h : (M_\varphi, \overline{x}) \to (M_\psi, \overline{x})$.

Proof.
For the forward direction, suppose that $\psi(\overline{x}) \models \varphi(\overline{x})$. Clearly $M_\psi \models \psi(\overline{x})$, and so $M_\psi \models \varphi(\overline{x})$. Then by the magic lemma, there exists a homomorphism $h : (M_\varphi, \overline{x}) \to (M_\psi, \overline{x})$. For the reverse direction, suppose that there exists a homomorphism h :

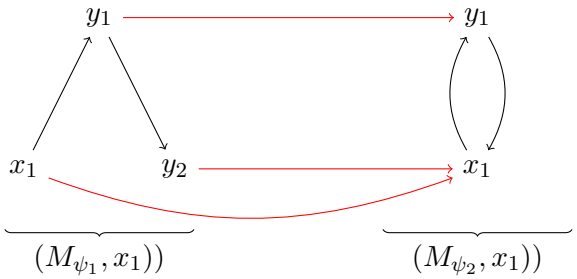
$(M_\varphi, \overline{x}) \to (M_\psi, \overline{x})$, and that $N \models \psi(\overline{a})$ for some model N. By the magic lemma, there exists a homomorphism $g : (M_\psi, \overline{x}) \to (N, \overline{a})$. Then $g \circ h : (M_\varphi, \overline{x}) \to (N, \overline{a})$ is a homomorphism, and so we have again by the magic lemma that $N \models \varphi(\overline{a})$. □

The following proposition follows immediately from the Chandra-Merlin theorem.

Proposition 2.2.4. Given CQs $\varphi(\overline{x}), \psi(\overline{x})$, we have that

$$\varphi(\overline{x}) \equiv \psi(\overline{x}) \iff (M_\varphi, \overline{x}) \rightleftarrows (M_\psi, \overline{x}).$$

Example 2.2.5. In light of the magic lemma, we can revisit the structures in example 2.2.1. We can define a homomorphism $(M_{\psi_1}, x_1) \to (M_{\psi_2}, x_1)$ as follows:



By the magic lemma, we then have that $(M_{\psi_2}, x_1) \models \psi_1(x_1)$. However, it's easy to see that there is no homomorphism $(M_{\psi_2}, x_1) \to (M_{\psi_1}, x_1)$, and so, again by the magic lemma, we have that $(M_{\psi_1}, x_1) \not\models \psi_2(x_1)$.

Thus far, we have that the transformation mapping queries to their canonical models constitutes a one-for-one correspondence between CQs and the class of all finite models (up to isomorphism). The magic lemma extends this correspondence by relating the satisfaction of a CQ in a model to the existence of an appropriate homomorphism between the canonical models. This is further extended by the Chandra-Merlin theorem, which equates validity (of implication) between two CQs to the existence of homomorphisms. The next definition, defined originally for undirected graphs but easily extended to arbitrary finite models, provides canonical representatives (up to isomorphism) of each $\rightleftarrows$-equivalence class.

Definition 2.2.6. A core of a model $(M, \overline{a})$ is an induced submodel $(C_M, \overline{a})$ of M such that $(C_M, \overline{a}) \rightleftarrows (M, \overline{a})$ and such that every endomorphism $h : (C_M, \overline{a}) \to (C_M, \overline{a})$ is surjective. If $C_M = M$, then we say that $(M, \overline{a})$ is a core.

Proposition 2.2.7 ([HN92])**.** The following are some basic properties of cores:

1. Every finite model has a core.
2. The core of a finite model is unique up to isomorphism.
3. The cores of any two homomorphically equivalent models are isomorphic.

2.3 Semiring Semantics for Conjunctive Queries

We now introduce *semiring semantics* for conjunctive queries, which generalize the typical semantics of first-order logic [GKT07].

Definition 2.3.1. A *semiring* is an algebraic structure $\langle S, +, \cdot, 0, 1\rangle$, where S is a domain set, 0 and 1 are constants in S, and $+$ and $\cdot$ are binary operations on S satisfying the following properties:

1. $a + (b + c) = (a + b) + c$
2. $a + 0 = 0 + a = a$
3. $a + b = b + a$
4. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
5. $a \cdot 1 = 1 \cdot a = a$
6. $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
7. $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$
8. $a \cdot 0 = 0 \cdot a = 0$

Some important semirings are the Boolean semiring $\langle \{0, 1\}, \vee, \wedge, 0, 1\rangle$ and the semiring of the natural numbers $\langle \mathbb{N}, +, \cdot, 0, 1\rangle$. When no confusion results, we will identify a semiring $\langle S, +, \cdot, 0, 1\rangle$ with its domain set S. In particular, we will write $\mathbb{B}$ for the Boolean semiring, and $\mathbb{N}$ for the semiring of the natural numbers. To define semiring semantics in their most general form, we first need an expanded notion of a finite model. Recall that $\mathrm{Rel}(M)$ denotes the collection of all possible facts on the model M over its signature.

Definition 2.3.2. Let S be a semiring. An S-labeled model (or simply *S-model*) is a pair (M, λ), where M is a finite relational model, and $\lambda : \mathrm{Rel}(M) \to S$ is called an *S-interpretation* (on M), and is required to satisfy that $\lambda(R^M(a_1, \ldots, a_k)) \neq 0$ for any k-ary relation symbol R such that $R^M(a_1, \ldots, a_k)$ holds, and $\lambda(R^M(a_1, \ldots, a_k)) = 0$ otherwise.

In other words, an S-labeled model is one in which each possible fact is assigned some element (a *truth value*) in the semiring S, with facts of the model assigned nonzero truth values, while atomic relations which do not hold in the model are assigned 0. We are now ready to define semiring semantics.

Definition 2.3.3. Let S be a semiring, and let (M, λ) be an S-labeled model. Given a variable assignment $g : \mathrm{Vars} \to M$, the *S-semantics* are the map $\nu^S_{(M,\lambda),g} : \mathrm{CQ} \to S$ defined recursively as follows:

$$\begin{aligned}
\nu^S_{(M,\lambda),g}(R(\overline{x})) &:= \lambda(R(g(\overline{x}))),\\
\nu^S_{(M,\lambda),g}(x = y) &:= \begin{cases} 1 & \text{if } g(x) = g(y)\\ 0 & \text{otherwise,}\end{cases}\\
\nu^S_{(M,\lambda),g}(\varphi \wedge \psi) &:= \nu^S_{(M,\lambda),g}(\varphi) \cdot \nu^S_{(M,\lambda),g}(\psi), \text{ and}\\
\nu^S_{(M,\lambda),g}(\exists x \varphi) &:= \sum_{a \in M} \nu^S_{(M,\lambda),g[x \mapsto a]}(\varphi),
\end{aligned}$$

where the notation $g[x \mapsto a]$ denotes the assignment which maps x to a and which agrees with g on all other variables. Note that product operation in the clause for conjunction is defined by the $\cdot$ operation for S, while the summation in the clause for existential quantification is defined by the $+$ operation for S.

It is easy to verify, given a semiring S, an S-model (M, λ), and a conjunctive query $\varphi(x_1, \ldots, x_k)$, that for any variable assignments g and g' such that $g(x_i) = g'(x_i)$ for each $i \leq k$, we have that

$$\nu^S_{(M,\lambda),g}(\varphi) = \nu^S_{(M,\lambda),g'}(\varphi).$$

In other words, given a tuple $\overline{a} = a_1, \ldots, a_k$, we can use the more concise notation

$$\nu^S_{(M,\lambda)}(\varphi(\overline{a})) := \nu^S_{(M,\lambda),g}(\varphi),$$

where g is any variable assignment such that $x_i \mapsto a_i$ for each $i \leq k$. Furthermore, observe that for any finite relational model M and any semiring S, the following map is a valid S-labeling:

$$\nu^{\mathbb{N}}_{(M,\lambda)}(R(\overline{a})) = \begin{cases} 1 & \text{if } M \models R(\overline{a}) \\ 0 & \text{otherwise.} \end{cases}$$

We refer to this as the *trivial labeling*. When λ is the trivial labeling, we will simplify our notation, writing

$$\nu^S_M(\varphi(\overline{a})) := \nu^S_{(M,\lambda)}(\varphi(\overline{a})).$$

We refer to $\mathbb{B}$-semantics as the *Boolean semantics*. It is worth noting that the trivial labeling is the only valid $\mathbb{B}$-labeling on any finite relational model M. The following proposition equates Boolean semantics with the usual semantics of first-order logic.

Proposition 2.3.4. Let M be a model and let $\lambda : \mathrm{Rel}(M) \to \mathbb{B}$ be the trivial labeling. Then for any tuple $a_1, \ldots, a_k$ in M and CQ $\varphi(x_1, \ldots, x_k)$, we have that

$$\nu^{\mathbb{B}}_{(M,\lambda)}(\varphi(\overline{a})) = 1 \iff M \models \varphi(\overline{a}).$$

We omit the proof of this proposition because it is developed nearly in parallel to the following analysis of the $\mathbb{N}$-semantics.

For the $\mathbb{N}$-semantics, which we will prefer to call the counting semantics, we first consider the special case of an $\mathbb{N}$-labeled model (M, λ), where λ is the trivial labeling. We can view such a model as a typical first-order structure, in the sense that

$$\nu^{\mathbb{N}}_M(R(\overline{a})) = \begin{cases} 1 & \text{if } M \models R(\overline{a}) \\ 0 & \text{otherwise.} \end{cases}$$

Let $\varphi(\overline{x}) := \exists y_1 \ldots \exists y_m(\bigwedge_{i \in I} \alpha_i)$ be a CQ with free variables $\overline{x} = x_1, \ldots, x_k$ and bound variables $\overline{y} = y_1, \ldots, y_m$. Let

$$\psi(\overline{x}, \overline{y}) := \bigwedge_{i \in I} \alpha_i.$$

Since conjunction under $\mathbb{N}$-semantics is interpreted using the $\cdot$ operation, we have that

$$\nu^{\mathbb{N}}_M(\psi(\overline{a}, \overline{b})) = \begin{cases} 1 & \text{if } M \models \psi(\overline{a}, \overline{b}) \\ 0 & \text{otherwise,} \end{cases}$$

where $\overline{a} = a_1, \ldots, a_k$ and $\overline{b} = b_1, \ldots, b_m$ are tuples of elements in M. So far, we have not departed from the typical semantics of first-order logic. However, it is the $\mathbb{N}$-semantics for the existential quantifier that justify the name "counting semantics." Since the $+$ operation in the semiring of the natural numbers is standard addition, we have that

$$\nu^{\mathbb{N}}_M(\varphi(\overline{a})) = \sum_{b_1, \ldots, b_m \in M} \nu^{\mathbb{N}}_M(\psi(\overline{a}, b_1, \ldots, b_m))$$

is the number of satisfying assignments g in M for $\varphi(x_1, \ldots, x_k)$ such that $g(x_i) = a_i$ for each $i \leq k$. Note that a summation of several 1s under the Boolean semiring simply yields another 1. As a result, if we were to view λ as the $\mathbb{B}$-interpretation on M, then we would have that $\nu_M^{\mathbb{B}}(\varphi(\overline{a})) = 1$ if and only if there's at least one appropriate satisfying assignment, which justifies Proposition 2.3.4.

More generally, since $1 \cdot 1 = 1$ for all semirings S, we have for all semirings S that

$$\nu_M^S(\psi(\overline{a}, \overline{b})) = \begin{cases} 1 & \text{if } M \models \psi(\overline{a}, \overline{b}) \\ 0 & \text{otherwise,} \end{cases}$$

and thus

$$\nu_M^S(\varphi(\overline{a})) = \sum_{b_1, \ldots, b_m \in M} \nu_M^S(\psi(\overline{a}, b_1, \ldots, b_m)).$$

We can go further, observing that, by the proof of Lemma 2.2.2, each satisfying assignment g for $\varphi(\overline{x})$ in a model M such that $g(x_i) = a_i$ for each $i \leq k$ corresponds to a homomorphism from the canonical model $(M_\varphi, \overline{x})$ to $(M, \overline{a})$. It follows that $\nu_M^{\mathbb{N}}(\varphi(\overline{a}))$ is the number of homomorphisms from $(M_\varphi, \overline{x})$ to $(M, \overline{a})$, while $\nu_M^{\mathbb{B}}(\varphi(\overline{a}))$ indicates whether or not such a homomorphism exists. In most of the later parts of this thesis, we will assume the trivial labeling, and so the above analysis will hold. Given the relationship between homomorphism counting and S-semantics, we now define a more convenient notation for tracking homomorphisms.

Definition 2.3.5. For finite models $(M, \overline{a})$ and $(N, \overline{b})$, we write $\mathrm{Hom}((N, \overline{b}), (M, \overline{a}))$ for the collection of homomorphisms from $(N, \overline{b})$ to $(M, \overline{a})$. Then we write

$$\hom_S((N, \overline{b}), (M, \overline{a})) := \sum_{h \in \mathrm{Hom}((N, \overline{b}), (M, \overline{a}))} 1_S,$$

where 1_S is the 1 element of S and the summation is defined by the $+$ operation for S.

Then by our preceding discussion, we obtain the following proposition.

Proposition 2.3.6. Let $(M, \overline{a})$ and $(N, \overline{b})$ be finite models. Then

$$\hom_S((N, \overline{b}), (M, \overline{a})) := \nu_M^S(\varphi_N(\overline{a})).$$

In particular, we have the following proposition for the special cases of the $\mathbb{B}$-semantics and $\mathbb{N}$-semantics, which essentially restates the magic lemma and generalizes it to the counting semantics.

Proposition 2.3.7. Let $(M, \overline{a})$ and $(N, \overline{b})$ be finite models. Then we have that

$$\hom_{\mathbb{B}}((N, \overline{b}), (M, \overline{a})) = \begin{cases} 1 & \text{if } \mathrm{Hom}((N, \overline{b}), (M, \overline{a})) \neq \emptyset \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\hom_{\mathbb{N}}((N, \overline{b}), (M, \overline{a})) = |\mathrm{Hom}((N, \overline{b}), (M, \overline{a}))|.$$

For finite models $(M, \overline{a})$ and $(N, \overline{b})$, we will also write $\mathrm{Sur}((M, \overline{a}), (N, \overline{b}))$ for the set of surjective homomorphisms from $(M, \overline{a})$ to $(N, \overline{b})$, and we will write

$$\mathrm{sur}_{\mathbb{N}}((M, \overline{a}), (N, \overline{b})) = |\mathrm{Sur}((M, \overline{a}), (N, \overline{b}))|.$$

For arbitrary $\mathbb{N}$-models where λ may not be the trivial labeling, the only change to our analysis is that $\nu^{\mathbb{N}}_{(M,\lambda)}(\psi(a_1,\ldots,a_k,b_1,\ldots,b_m)$ becomes the product over all the $\mathbb{N}$-labels for the facts occurring in the conjunction under the assignment g such that $y_i \mapsto b_i$ for each $i \leq m$. We then sum over this product for all possible tuples $b_1,\ldots,b_m$ in M to determine $\nu^{\mathbb{N}}_{(M,\lambda)}(\varphi(a_1,\ldots,a_k))$. In fact, this more general context applies to any semiring S: each term of the sum corresponds to a distinct homomorphism, and each term is the product of the S-labels for the facts of the query for the corresponding assignment.

2.4 Homomorphism Vectors

We now introduce homomorphism vectors, the central concept of this thesis. Fix a finite relational signature σ and some enumeration $\langle (M_i, \overline{a}^{M_i})\rangle_{i\in\omega}$ of all finite σ-models with n distinguished elements. Recall that $\mathcal{M}^\sigma_n$ denotes the class of all σ-models with n distinguished elements.

Definition 2.4.1. Let S be a semiring and let $(M,\overline{a})$ be an arbitrary σ-model with n distinguished elements. Then the left homomorphism vector of $(M,\overline{a})$ under S (or S *left profile* of $(M,\overline{a})$) is the countably infinite vector $\hom_S(\mathcal{M}^\sigma_n,(M,\overline{a}))$ whose k^{th} entry is $\hom_S((M_k,\overline{a}^{M_i}),(M,\overline{a}))$.

Definition 2.4.2. Let S be a semiring and let $(M,\overline{a})$ be an arbitrary σ-model with n distinguished elements. Then the right homomorphism vector of $(M,\overline{a})$ under S (or S *right profile* of $(M,\overline{a})$) is the countably infinite vector $\hom_S(M,\mathcal{M}^\sigma_n)$ whose k^{th} entry is $\hom_S((M,\overline{a}),(M_k,\overline{a}^{M_i}))$.

The use of $\mathcal{M}^\sigma_n$ in the notation will be justified when we consider *restrictions* of homomorphism vectors.

Remark 2.4.3. We refer to $\hom_{\mathbb{B}}(\mathcal{M}^\sigma_n,(M,\overline{a}))$ and $\hom_{\mathbb{B}}((M,\overline{a}),\mathcal{M}^\sigma_n)$ as the Boolean left profile of $(M,\overline{a})$ and the Boolean right profile of $(M,\overline{a})$, respectively. We refer to $\hom_{\mathbb{N}}(\mathcal{M}^\sigma_n,(M,\overline{a}))$ and $\hom_{\mathbb{N}}((M,\overline{a}),\mathcal{M}^\sigma_n)$ as the counting left profile of $(M,\overline{a})$ and the counting right profile of $(M,\overline{a})$, respectively.

It is worth noting, at this point, that we have abstracted away any reference to conjunctive queries in our notation. However, the correspondence between CQs and finite relational models, as well as the correspondence between satisfying assignments to CQs and homomorphisms between canonical models, provide that these vectors are rich in semantic content. If we knew every entry of, say, the Boolean left profile of a σ-model $(M,\overline{a})$ with n distinguished elements, then we would know which n-ary CQs are satisfied by the tuple $\overline{a}$ in M and which are not. Similarly, if we knew the Boolean right profile of $(M,\overline{a})$, then we would know which finite models with n distinguished elements satisfy its canonical query $\varphi_M(\overline{x})$.

We will soon see that the Boolean homomorphism vectors capture homomorphic equivalence, while the counting homomorphism vectors capture isomorphism. These were some of the earliest results on the *expressive power* of homomorphism vectors. In particular, these results capture the expressive power of homomorphism vectors with respect to the class of all finite relational models, $\mathcal{M}^\sigma_n$. However, we can consider smaller classes, and ask which relations between models are captured by homomorphism vectors restricted to that class. To make this notion precise, we need the following definition:

Definition 2.4.4. Let S be a semiring, $\mathcal{C}$ be any class of finite relational models with n distinguished elements, and $\langle (M_{k_i},\overline{a})\rangle_{i\in I}$ be the subsequence of the fixed enumeration

$\langle (M_k, \overline{a}) \rangle_{k \in \omega}$ containing only those models which occur in $\mathcal{C}$. Then, given a finite model $(M, \overline{a})$,

1. the *left profile of* $(M, \overline{a})$ *restricted to* $\mathcal{C}$ (notation: $\hom_S(\mathcal{C}, (M, \overline{a}))$) is the vector (of length $|I|$) whose i^{th} entry, for $i \in I$, is $\hom_S((M_{k_i}, \overline{a}^{M_{k_i}}), (M, \overline{a}))$, and

2. the *right profile of* $(M, \overline{a})$ *restricted to* $\mathcal{C}$ (notation: $\hom_S((M, \overline{a}), \mathcal{C})$) is the vector (of length $|I|$) whose i^{th} entry, for $i \in I$, is $\hom_S((M, \overline{a}), (M_{k_i}, \overline{a}^{M_{k_i}}))$.

Definition 2.4.4 justifies the use of $\mathcal{M}_n^\sigma$ in the notation for Definitions 2.4.1 and 2.4.2. At a high level, restrictions of homomorphism vectors are the primary focus of our later chapters. We will now review some of the key known results regarding the expressive power of homomorphism vectors over $\mathcal{M}_n^\sigma$.

Theorem 2.4.5. ([AKW21]) For models $(M, \overline{a})$ and $(N, \overline{b})$, the following are equivalent:

1. $(M, \overline{a})$ and $(N, \overline{b})$ are homomorphically equivalent.

2. $(M, \overline{a})$ and $(N, \overline{b})$ have the same Boolean left profile.

3. $(M, \overline{a})$ and $(N, \overline{b})$ have the same Boolean right profile.

Proof.
To obtain (2) from (1), suppose that $(M, \overline{a})$ and $(N, \overline{b})$ are homomorphically equivalent. Then there exist homomorphisms $h : (M, \overline{a}) \to (N, \overline{b})$ and $g : (N, \overline{b}) \to (M, \overline{a})$. Now consider $(M_k, \overline{a}^{M_k})$, the k^{th} model of the enumeration. If there's a 1 in the k^{th} entry of the Boolean left profile of $(M, \overline{a})$, then there's a homomorphism $f : (M_k, \overline{a}^{M_k}) \to (M, \overline{a})$. Hence $h \circ f : (M_k, \overline{a}^{M_k}) \to (N, \overline{b})$ is a homomorphism, and so there's a 1 in the k^{th} entry of the Boolean left profile of $(N, \overline{b})$ as well. A symmetric argument establishes that $(M, \overline{a})$ and $(N, \overline{b})$ must have the same value in each entry of their Boolean left profiles. A nearly identical argument gives (3) from (1).

To obtain (1) from (2), suppose $(M, \overline{a})$ and $(N, \overline{b})$ have the same left profile. The identity maps $id_M : (M, \overline{a}) \to (M, \overline{a})$ and $id_N : (N, \overline{b}) \to (N, \overline{b})$ are clearly homomorphisms. Furthermore, $(M, \overline{a})$ and $(N, \overline{b})$ must occur somewhere in the enumeration. As a result, since $(M, \overline{a})$ and $(N, \overline{b})$ have homomorphisms to themselves, and they have the same Boolean left profiles, they must also have homomorphisms to each other. A similar argument gives (1) from (3). □

The next two theorems state that isomorphism is captured by the counting profiles.

Theorem 2.4.6 (Lovász Theorem, [Lov67])**.** Two models $(M, \overline{a})$ and $(N, \overline{b})$ have the same (counting) left profile if and only if $(M, \overline{a}) \cong (N, \overline{b})$.

Theorem 2.4.7 (Chaudhuri-Vardi Theorem, [CV93])**.** Two models $(M, \overline{a})$ and $(N, \overline{b})$ have the same (counting) right profile if and only if $(M, \overline{a}) \cong (N, \overline{b})$.

Definition 2.4.8. Let $\mathcal{C}$ be some class of models. We write $\text{Inj}(\mathcal{C})$ for the class of models $(N, \overline{b})$ such that there exists some injective homomorphism $h : (N, \overline{b}) \to (M, \overline{a})$ for some $(N, \overline{b}) \in \mathcal{C}$. We write $\text{Sur}(\mathcal{C})$ for the class of models $(N, \overline{b})$ such that there exists some fully-surjective homomorphism $h : (M, \overline{a}) \to (N, \overline{b})$ for some $(N, \overline{b}) \in \mathcal{C}$. We define the *extension class* of $\mathcal{C}$ to be $\text{Ext}(\mathcal{C}) := \text{Inj}(\mathcal{C}) \cap \text{Sur}(\mathcal{C})$.

Clearly $\text{Ext}(\mathcal{M}_n^\sigma) = \mathcal{M}_n^\sigma$, and so the next theorem generalizes Theorems 2.4.6 and 2.4.7.

Theorem 2.4.9 ([AKW21])**.** Let $\mathcal{C}$ be a non-empty class of finite structures with n distinguished elements. For all $(M, \overline{a}), (N, \overline{b}) \in \mathcal{C}$, the following are equivalent:

1. $(M, \overline{a}) \cong (N, \overline{b})$;
2. $\hom_{\mathbb{N}}(\mathrm{Ext}(\mathcal{C}), (M, \overline{a})) = \hom_{\mathbb{N}}(\mathrm{Ext}(\mathcal{C}), (N, \overline{b}))$;
3. $\hom_{\mathbb{N}}((M, \overline{a}), \mathrm{Ext}(\mathcal{C})) = \hom_{\mathbb{N}}((N, \overline{b}), \mathrm{Ext}(\mathcal{C}))$.

2.5 Important Classes of Finite Models

In this section, we introduce some important properties of finite models.

Definition 2.5.1. A *fact path* in a model M is a finite sequence of facts $P = f_1 \dots f_k$ with $el(f_i) \cap el(f_{i+1}) \neq \emptyset$ for each $i < k$. Given $a, b \in M$, we say that P is a fact path from a to b if $a \in f_1$ and $b \in f_k$.

Given a model M whose signature contains a relation symbol R, we refer to a fact path containing only facts of R as an R^M-path.

Definition 2.5.2. A model M is *connected* if there's a fact path between any $a, b \in M$.

An important class of CQs are those which are "acyclic." To make this notion precise, we need the following preliminary definitions.

Definition 2.5.3. The *incidence graph* of a model $(M, \overline{a})$ is the (undirected) bipartite multigraph $(V = \mathrm{dom}(M) \cup \mathrm{Facts}(M), E)$, where $(b, f) \in E$ whenever $b \in M$ occurs in $f \in \mathrm{Facts}(M)$. Note that we allow duplicate edges if an element b occurs more than once in a fact f. A path in the incidence graph of a model M is a sequence of edges $P = (a_0, f_0) \dots (a_k, f_k)$ such that $a_i = a_{i+1}$ or $f_i = f_{i+1}$ for each $i < k$. P is a simple path if all of its elements are distinct, and P is a simple cycle if all but its first and last elements are distinct, and the first and last elements are equivalent.

See Figure 2.1 for an example of an incidence graph.

Definition 2.5.4. A model is *acyclic* if there are no simple cycles in its incidence graph, and *c-acyclic* if every simple cycle in its incidence graph contains a distinguished element.

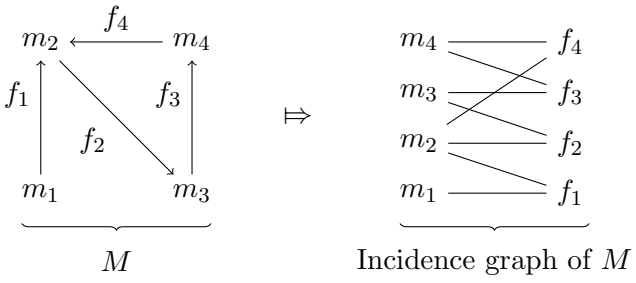


Figure 2.1: A model M along with its incidence graph.

Observe that the incidence graph of the model M depicted in Figure 2.1 contains a simple cycle, and so the model is *not* acyclic. The incidence graph of any CQ is the incidence graph of its canonical model, and we call a CQ acyclic (resp. c-acyclic) if its canonical model is acyclic (resp. c-acyclic).

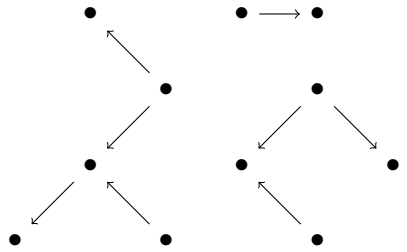


Figure 2.2: Example of a forest model.

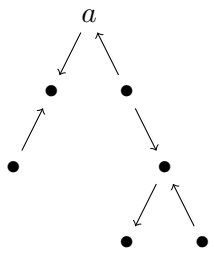


Figure 2.3: Example of a tree-shaped model (M, a).

Definition 2.5.5. A *forest* is an acyclic model over a signature with no relation symbols of arity greater than 2.

Definition 2.5.6. A *tree-shaped* model is a connected forest with one distinguished element.

Figure 2.2 depicts a forest, while Figure 2.3 depicts a tree-shaped model. Given a tree-shaped model (M, a), we refer to a as the *root* of the model. By acyclicity, there's a unique fact path from a to each element $m \in M$, and we define $\mathrm{depth}(m)$ to be the length of this fact path. The depth of (M, a) is $\max_{m \in M} \mathrm{depth}(m)$. If n is the unique element occurring with m in the last fact of this path, then we write $n = \mathrm{parent}(m)$.

Definition 2.5.7. Given a model M, a binary relation $R \subseteq M \times M$, and $a, b \in M$, an *R-directed path* from a to b is a sequence $(a, c_1)(c_1, c_2) \dots (c_{k-1}, c_k)(c_k, b)$ of tuples in R.

Definition 2.5.8. Let R_i be a binary relation symbol for each $i \in I$, where I is a finite index set. A *directed tree* with respect to $\{R_i \mid i \in I\}$ is a tree-shaped model (M, a) such that every element $m \in M$ can be reached by an R-directed path from a, where $R = \bigcup_{i \in I} R_i$.

Figure 2.4 depicts a directed tree.

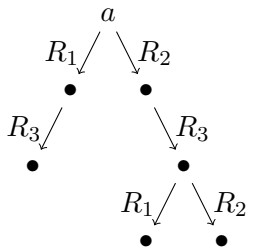


Figure 2.4: A directed tree (M, a) with respect to $\{R_i \mid 1 \leq i \leq 3\}$.

Definition 2.5.9. Let (M_i, a_i) be a directed tree for each $i \in I$, where I is some finite index set. Let (M, a) be the model obtained by taking the disjoint union $M = \biguplus_{i \in I} M_i$ and setting $a = a_i$ for some $i \in I$. We say that (M, a) is a *directed forest*.

Proposition 2.5.10. Let (M, a) be a directed tree with respect to $\{R_i \mid i \in I\}$, and let (N, b) be an arbitrary model. Then

$$\hom((M, a), (N, b)) = \prod_{i \in I} \prod_{m : R_i^M(a,m)} \sum_{n : R_i^N(b,n)} \hom((M', m), (N', n)),$$

where (M', m) and (N', n) are the subtrees of M and N rooted at m and n, respectively.

Proof.
We will show this by induction of the depth of the directed tree (M, a). For the base case, a has no successors, and so the result is trivial. For the inductive step, suppose that (M, a) has depth $k + 1$, and that

$$\hom((T, c), (N, b)) = \prod_{i \in I} \prod_{m : R_i^M(c,m)} \sum_{n : R_i^N(b,n)} \hom((M', m), (N', n))$$

holds for all directed trees (T, c) of depth k. Any homomorphism $h : (M, a) \to (N, b)$ must have that $h(a) = b$. It follows that each subtree rooted at an R_i-child m of a must have its root mapped to some R_i-child n of b. Then h restricted M' is a homomorphism from (M', m) to (N', n). Thus we can choose to map each subtree rooted at a child of a to any homomorphism from (M', m) to (N', n). The total number of such maps is given by

$$\sum_{n : R_i^N(b,n)} \hom((M', m), (N', n)).$$

Thus, choosing without replacement, we have that

$$\hom((M, a), (N, b)) = \prod_{i \in I} \prod_{m : R_i^M(a,m)} \sum_{n : R_i^N(b,n)} \hom((M', m), (N', n)).$$

This completes the proof. □

The above recursive formula will be useful in Chapter 4.

Chapter 3

Finite Characterizations of Database Queries under Bag Semantics

In database query languages such as SQL, users can pose *queries* to structured data stored in *database instances*. These queries are formal expressions evaluated under some *semantics* to determine an *answer*. Most of the literature on the theory of such queries has assumed *set semantics*. Under this assumption, the answer to a query is a collection of objects (without duplicates) in the database instance satisfying the requirements of the query. However, in SQL and many other languages, the default setting is to instead return *multi-sets*, where objects satisfying a query may appear more than once, representing *how many different ways* the objects satisfy the requirements of the query. This is the *bag semantics*.

A finite characterization of a query Q is a collection of *examples* of database instances, such that any other query Q' which returns the same answer as Q on each of the examples must also agree with Q on *all* possible database instances. The problem of finding and bounding the number of examples needed to finitely characterize a conjunctive query has been explored under the set semantics [CD22]. In this chapter, we explore these same problems under the bag semantics.

3.1 Database Theory and Logic

In this section, we will develop the basics of database theory, and justify the sufficiency of a finite model-theoretic perspective in addressing database-theoretic questions.

The central objects in database theory are database instances and database queries. A *database instance* is a finite collection $I = (T_1, \ldots, T_n)$ of finite two-dimensional tables, each with an associated *name* as well as an *arity* equal to the number of columns in the table. See Figure 3.1 for an example of a database instance. The collection of all table names for a given database instance, along with their associated arities, constitutes a *database schema*. The terminology of a "database instance" is justified by the observation that there might be many instances of the same database schema. Typically, the entries of a database instance will be some kind of data object, like a string or a number. For our purposes, we will just refer to these entries as "elements" of the table, and ignore what type of objects they actually are.

Family Information		
Student	**Parent 1**	**Parent 2**
Kurt	Rudolf	Marianne
Lewis	Matthew	Marta
Alvin	Cornelius	Lettie

Alma Mater	
Student	**University**
Kurt	Vienna University
Lewis	West Point
Alvin	Yale University

Figure 3.1: A database instance with two tables.

The *active domain* of a database instance I, denoted $\text{adom}(I)$, is the collection of elements occurring in some table in I. For example, the active domain of the database instance in Figure 3.1 is the set

$$\{\text{Kurt, Lewis, Alvin, Rudolf, Matthew, Cornelius, Marianne,}$$
$$\text{Marta, Lettie, Vienna University, West Point, Yale University}\}.$$

Given a database instance $I = (T_1, \ldots, T_n)$, we can define a corresponding finite relational model M_I as follows. The signature of the model will contain a relation symbol for each table name whose arity is the arity of the table, and the domain of the model is $\text{adom}(I)$. For a given relation symbol R of arity k in the signature, we will say that $R^{M_I}(a_1, \ldots, a_k)$ holds if there is a row in the R-table containing the entries $a_1, \ldots, a_k$. In other words, we view each table as a relation on the active domain, where each row represents a fact of that relation. Figure 3.2 depicts this transformation.

R	
a	c
b	d
c	b
d	c

$\Rightarrow$

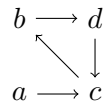


Figure 3.2: A simple translation from a database instance to a finite model.

A *query*, in the database-theoretic setting, is some function mapping database instances to *answers*. We additionally require that a query always produces the same answer on isomorphic database instances (i.e., database instances which are identical up to a re-ordering of the rows of the tables and a renaming of their entries). Typically, queries take the form of expressions written in some formal database query language. The answer to a query written in a formal language (notation: $E(I)$) is either a collection of tuples of elements (all of the same length) of the database instance "satisfying" the query under some *database semantics*, or otherwise is some semantic value (like "true" or "false"). If a query returns tuples, then we refer to the length of the tuples as the *arity* of the query. The problem of computing the answer to a query in a database instance is referred to as *query evaluation*.

There are various *database languages* in which a query can be written, including the relational calculus (comparable to first-order logic) and relational algebra (comparable to SQL syntax). We say that a query Q is *expressible* in a formal language L if there exists some expression E in L such that $Q(I) = E(I)$ on all database instances I. We ignore the details of the syntax and semantics for relational algebra. For our purposes, it suffices to note that these languages are roughly equally expressive, by the following well-known result.

Theorem 3.1.1. (Codd's Theorem, [Cod71]) Let Q be a k-ary query. Then the following are equivalent.

1. There is an expression E of the relational calculus such that $Q(I) = E(I)$ for all database instances I.
2. There is a formula $\varphi(\overline{x})$ of the relational algebra (FO) such that $Q(I) = \{\langle \overline{a} \in M_I \rangle \mid M_I \models \varphi((\overline{a}))\}$ for all database instances I.

Note that (2) of the preceding theorem implicitly relies on the assumption that the formula φ is interpreted over the active domain of the database instance. In other words, to evaluate a query of the relational algebra (and hence of commonly-used languages like SQL) in a database instance I, it suffices to determine the collection of tuples in M_I satisfying an appropriately-constructed FO-formula.

In our previous description of queries, we have been intentionally vague about what constitutes an answer to a query in a formal database query language. We will now clarify this point by defining two important database semantics for FO-formulas.

Definition 3.1.2. The *set semantics* for conjunctive queries of arity k in a database instance I is the map $[\![\cdot]\!]_I^{set_k} : \mathrm{CQ}^k \to \mathcal{P}(I^k)$ given by

$$[\![\varphi(x_1, \ldots, x_k)]\!]_I^{set_k} := \{\langle a_1, \ldots, a_k \rangle \mid M_I \models \varphi(a_1, \ldots, a_k)\}.$$

We write $[\![\cdot]\!]_I^{set}$ for the union of these maps for each k.

Observe that the set semantics are assumed for query evaluation in Codd's theorem. Furthermore, recall that first-order semantics for conjunctive queries are precisely the Boolean semantics:

Proposition 2.3.4. Let M be a model and let $\lambda : \mathrm{Rel}(M) \to \mathbb{B}$ be the trivial labeling. Then for any tuple $a_1, \ldots, a_k$ in M and CQ $\varphi(x_1, \ldots, x_k)$, we have that

$$\nu^{\mathbb{B}}_{(M,\lambda)}(\varphi(\overline{a})) = 1 \iff M \models \varphi(\overline{a}).$$

Thus we can equivalently define the map $[\![\cdot]\!]_I^{set_k}$ as

$$[\![\varphi(x_1, \ldots, x_k)]\!]_I^{set_k} := \{(a_1, \ldots, a_k) \mid \nu^{\mathbb{B}}_{M_I}(\varphi(a_1, \ldots, a_k)) = 1\}.$$

Thus far, our development has been predicated upon two (unwarranted) assumptions, which we will now address. The first assumption is that query evaluation is performed under the set semantics. However, as was pointed out by Chaudhuri and Vardi in [CV93], most database query languages (like SQL) do *not* use set semantics by default. Instead, the answer to a SQL query is a multi-set, where the multiplicity of a tuple in the answer is the number of assignments under which the query is satisfied for that tuple. In other words, the default semantics for most database query languages is as in the following definition.

Definition 3.1.3. The *bag semantics* for conjunctive queries of arity k in a database instance I is the map $[\![\cdot]\!]_I^{bag} : CQ^k \to \mathcal{P}(I^k)$ given by

$$[\![\varphi(x_1, \ldots, x_k)]\!]_I^{bag} := \{(\langle a_1, \ldots, a_k \rangle, l) \mid l = \nu^{\mathbb{N}}_{M_I}(\varphi(a_1, \ldots, a_k))) > 0\}.$$

We write $[\![\cdot]\!]_I^{bag}$ for the union of these maps for each k.

Thus the answer to a CQ $\varphi(x_1, \ldots, x_k)$ under the bag semantics is a multi-set, where the multiplicity of a tuple $a_1, \ldots, a_k$ in the answer is the number of satisfying assignments for φ such that $g(x_i) = a_i$ for each $i \leq k$. It follows that the bag semantics more faithfully represents the true semantics of database query languages than does the set semantics.

Our second unwarranted assumption is that database instances can be represented using an appropriately-defined standard finite first-order model. To see that this is not always the case, recall that the rows of a database instance I determine the facts of the corresponding finite model M_I. However, it is allowed in database query languages, and occurs frequently in actual database instances, that duplicate rows may appear in some table of the instance. Furthermore, the output of a query might change dependent on the existence and number of duplicate rows. To preserve all of the information in the table, we need to extend our translation of database instances.

To do this, given a database instance I, we will augment the standard first-order model M_I with a $\mathbb{N}$-interpretation λ_{M_I} (cf. Definition 2.3.2) to obtain an $\mathbb{N}$-model (M_I, λ_{M_I}). For each $\alpha \in \text{Facts}(M_I)$, we define $\lambda_{M_I}(\alpha)$ to be the number of times that the row corresponding to α occurs in the database table in I. Note that, for the special case in which I does not have any duplicate rows, the resulting $\mathbb{N}$-interpretation is the trivial interpretation, which is in concord with our earlier analysis of the counting semantics for standard first-order models. We will refer to a database instance without duplicate rows as a *set* database instance, while a database instance which may have duplicates is a *bag* database instance. In particular, we will view set databases as a special case of bag databases.

3.2 Finite Characterizations

We now turn our attention to the notion of finite characterizations of finite models, which can be defined with respect to any semiring S. We will focus only on finite characterizations of the right profile, but the notion can be easily extended to the left profile.

Definition 3.2.1. Let S be a semiring and $(M, \overline{a})$ be a finite σ-model. A (right) finite characterization of $(M, \overline{a})$ under S is a finite collection $\Sigma \subseteq \mathcal{M}^\sigma_n$ such that, for all finite models $(N, \overline{b})$, we have that

$$\hom_S((M, \overline{a}), \Sigma) = \hom_S((N, \overline{b}), \Sigma) \iff \hom_S((M, \overline{a}), \mathcal{M}^\sigma_n) = \hom_S((N, \overline{b}), \mathcal{M}^\sigma_n).$$

We refer to elements of such a set Σ as *examples*, and we say that two models $(M, \overline{a})$ and $(N, \overline{b})$ *agree* on a model (i.e., an example) $(A, \overline{c})$ if $\hom_S((M, \overline{a}), (A, \overline{c})) = \hom_S((N, \overline{b}), (A, \overline{c}))$. Otherwise, we say that $(M, \overline{a})$ and $(N, \overline{b})$ are *separated* by $(A, \overline{c})$. Given a collection Σ of models, we say that $(M, \overline{a})$ and $(N, \overline{b})$ agree on Σ if and only if $(M, \overline{a})$ and $(N, \overline{b})$ agree on each $(A, \overline{c}) \in \Sigma$. So far, we have described finite characterizations of models; however, we can also speak in terms of finite characterizations of conjunctive queries, where we define a finite characterization of a CQ $\varphi(\overline{x})$ to be a finite characterization of its canonical model $(M_\varphi, \overline{x})$.

Figure 3.3 depicts a finite characterization for the two element linear order (L_2, a), where the first element of the order is a distinguished element of the model. Any model which agrees with (L_2, a) on the first example must have the same domain size. The next four examples, as we will see in the proof of Theorem 3.4.2, separate any model (N, b) with a different counting right profile than (L_2, a) such that $|\text{dom}(N)| = |\text{dom}(L_2)|$. It then follows by the Chaudhuri-Vardi theorem that any model which agrees with (L_2, a) on the entire collection of examples is isomorphic to (L_2, a).

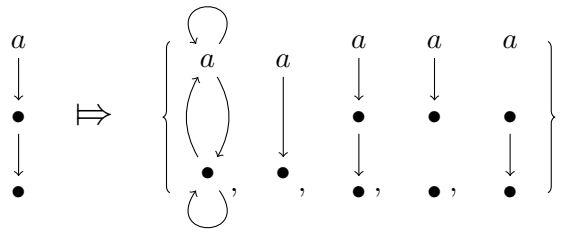


Figure 3.3: A finite characterization under the counting semantics for the two-element linear order (L_2, a) with the first element of the order as a distinguished element.

Manilla and Räihä were the first to explore characterizing a database query Q by giving examples of database instances [MR89]. In their setting, they were interested in finding a single database instance I such that any query Q' which returned the same answer as Q in I must be equivalent (under the set semantics) to Q. The use of database instances is noteworthy, since the answer under the set semantics for a database instance I is any tuple of elements from I which satisfy the query, and so their examples have the form $(I, [\![Q]\!]_I^{set})$, where I is a database instance. This would be equivalent, in our setting, to using a collection of the form $\{(M, \overline{a}) \mid \overline{a} \in M_I\}$ for some database instance I as a finite characterization under the Boolean semantics of Q. In other words, their single database instance I with active domain of size n translates in our context to a collection of $\binom{n}{k}$-many models (i.e., to $O(n^k)$-many examples).

This work was extended by Böttcher et. al. to the bag semantics [BLZ14], with attention confined to a very restricted class of queries. Say that a query is *self-join free* if it contains at most one occurrence of each relation symbol in the signature. In [BLZ14], the authors were interested in giving a single database instance which distinguished a self-join free query from any other non-equivalent self-join free query over the same signature. It is worth noting, in this case, that this restriction means that the query to be characterized needs only to be distinguished from finitely many other queries. Furthermore, following Manilla and Räihä, they were interested in the full answer to the query under the bag semantics, and so their single example has the form $(I, [\![Q]\!]_I^{bag})$ and corresponds to a finite characterization of polynomially-many examples. The last notable difference is that the authors permitted the use of constants.

We seek a strictly more general setting than that of Böttcher et. al. Our examples are models $(M, a_1, \ldots, a_k)$ with distinguished elements matching the arity of the query to be characterized. Our labels are implicit, corresponding to the multiplicity of the tuple $a_1, \ldots, a_k$ in the answer to the query in M, rather than the full answer to the query in M. However, this distinction is not significant, since the existence of a characterizing set Σ of examples of size n of our form for a query Q implies the existence of a characterizing set S of labeled database examples of size n of the form $(I, Q(I))$. To see this, consider $S = \{(I, Q(I)) \mid (M_I, \overline{a}) \in \Sigma\}$. Furthermore, we want to characterize arbitrary conjunctive queries relative to all other conjunctive queries. As a consequence, it is not immediately obvious that a finite characterization should exist at all. Finally, we will study finite characterizations of queries without constants and briefly state how to extend our results to queries containing constants.

By Theorem 2.4.5, two finite models have the same Boolean right profile if and only if they are homomorphically equivalent. By Proposition 2.2.4, two conjunctives queries are equivalent if and only if their canonical models are homomorphically equivalent. Thus a

finite characterization of $(M, \overline{a})$ under $\mathbb{B}$ is a finite set $\Sigma \subseteq \mathcal{M}_n^\sigma$ such that

$$\begin{aligned}(M,\overline{a}) \text{ and } (N,\overline{b}) \text{ agree on each } A \in \Sigma &\iff (M,\overline{a}) \text{ and } (N,\overline{b}) \text{ have the same Boolean right profile} \\ &\iff (M,\overline{a}) \rightleftarrows (N,\overline{b}) \\ &\iff \varphi_M \equiv \varphi_N.\end{aligned}$$

Suppose that Σ is a finite characterization under $\mathbb{B}$ of a conjunctive query $\varphi(\overline{x})$. Then any query $\psi(\overline{x})$ whose canonical model agrees with $(M_\varphi, \overline{x})$ on all examples in Σ is satisfied by all and only tuples $\overline{c}$ in a model A at which $\varphi(\overline{x})$ is satisfied. We claim that, when posed to a database instance I under the set semantics, $\varphi(\overline{x})$ and $\psi(\overline{x})$ always return the same answers. To see this, simply observe that for any model A, we have

$$\{\overline{c} \in A \mid A \models \psi(\overline{c})\} = \{\overline{c} \in A \mid A \models \varphi(\overline{c})\}.$$

Then directly from the definition of the set semantics, we obtain the following proposition.

Proposition 3.2.2. Let Σ be a finite characterization under $\mathbb{B}$ of a conjunctive query $\varphi(\overline{x})$. Then for any conjunctive query $\psi(\overline{x})$, we have that

$$[\![\varphi(\overline{x})]\!]_I^{set} = [\![\psi(\overline{x})]\!]_I^{set}$$

for all set database instances I if and only if $(M_\varphi, \overline{x})$ and $(N_\varphi, \overline{x})$ agree on Σ.

Finite characterizations under $\mathbb{B}$ are characterized by the following theorem.

Theorem 3.2.3 ([CD22])**.** A model has a finite characterization under $\mathbb{B}$ if and only if its core is c-acyclic. Furthermore, c-acyclic queries have polynomially-large finite characterizations.

What about under the bag semantics? We will refer to a finite characterization under $\mathbb{N}$ as a *counting* finite characterization. Let's now suppose that Σ is a counting finite characterization of a model $(M, \overline{a})$. Then we have by Theorem 2.4.6 that

$$\begin{aligned}(M,\overline{a}) \text{ and } (N,\overline{b}) \text{ agree on each } A \in \Sigma &\iff (M,\overline{a}) \text{ and } (N,\overline{b}) \text{ have the same counting right profile} \\ &\iff (M,\overline{a}) \cong (N,\overline{b}).\end{aligned}$$

Since $(M, \overline{a})$ and $(N, \overline{b})$ have the same counting right profiles, it follows that $\varphi_M(\overline{x})$ and $\varphi_N(\overline{x})$ have the same number of satisfying assignments in any model B. This implies that the multiplicity of a tuple $\overline{a} \in B$ in the answer to $\varphi_M(\overline{x})$ equals its multiplicity in the answer to $\varphi_N(\overline{x})$. In other words, when posed to a set database I under the bag semantics, φ_M and φ_N always return the same answers.

What about for arbitrary bag database instances? We have defined finite characterizations to be collections of finite models of first order logic. However, we saw in the preceding section that bag databases are modeled using $\mathbb{N}$-models. It is then natural to ask: do $\varphi_M(\overline{x})$ and $\varphi_N(\overline{x})$ always return the same answer on any bag database?

The answer to this question is yes. To see why, we need only look at the canonical query construction and the $\mathbb{N}$-semantics (cf. Definition 2.3.3). By the canonical query construction, it's easy to see that φ_M and φ_N are the same conjunctive query, up to a reordering of the atomic facts and a renaming of the variables; in database theoretic terms,

they are *isomorphic*. It then follows from our analysis of the $\mathbb{N}$-semantics for arbitrary $\mathbb{N}$-labeled models that

$$\nu_A^{\mathbb{N}}(\varphi_M(\overline{a})) = \nu_A^{\mathbb{N}}(\varphi_N(\overline{a}))$$

for any tuple $\overline{a}$ in an $\mathbb{N}$-labeled model (A, λ). Thus we obtain the following proposition.

Proposition 3.2.4. Let $\varphi(\overline{x})$ be a conjunctive query with a finite characterization Σ (under the counting semantics). Then for any conjunctive query $\psi(\overline{x})$, we have that

$$[\![\varphi(\overline{x})]\!]_I^{bag} = [\![\psi(\overline{x})]\!]_I^{bag}$$

for all database instances I if and only if $(M_\varphi, \overline{x})$ and $(M_\psi, \overline{x})$ agree on Σ.

This proposition shows that our finite characterizations, defined only in terms of standard finite models, are sufficient to capture equivalence of queries under the bag semantics.

3.3 Finite Characterizations for Arbitrary Models

Fix some arbitrary relational signature σ. In this section, we prove that every σ-model with k distinguished elements has a finite characterization under the counting semantics. In particular, we provide an upper bound of the size of such finite characterizations – exponential in the domain size of the model. The first step of our construction is to argue that we can confine attention to models of the same size. We do this with two lemmas.

Lemma 3.3.1. There exists a collection Γ_σ of models such that, for any σ-models $(M, \overline{a})$ and $(N, \overline{b})$ with $\overline{a} = a_1, \ldots, a_k$ and $\overline{b} = b_1, \ldots, b_k$, if $(M, \overline{a})$ and $(N, \overline{b})$ agree on Γ_σ, then

$$a_i = a_j \text{ if and only if } b_i = b_j \qquad (\dagger)$$

holds for all $i, j \leq k$.

Proof.
For any partition P of the set $[k]$, we can define a σ-model $(X_P, \overline{c}^P)$ with $\overline{c}^P = c_1^P, \ldots, c_k^P$ as follows. Every element of $\mathrm{dom}(X_P)$ will be a distinguished element, and the model will satisfy that

$$i \text{ and } j \text{ occur in the same set in } P \text{ if and only if } c_i^P = c_j^P.$$

The facts of the model will be all possible facts on its domain. Define

$$\Gamma_\sigma = \{(X_P, \overline{c}^P) \mid P \text{ is a partition of } [k]\}.$$

Since homomorphisms must preserve distinguished elements, it follows easily that for any σ-model $(M, \overline{a})$ with $\overline{a} = a_1, \ldots, a_k$, we have that $\hom_{\mathbb{N}}((M, \overline{a}), (X_P, \overline{c}^P)) > 0$ implies

$$c_i^P = c_j^P \implies a_i = a_j.$$

Consider the partition P^M such that i and j occur in the same set in P^M if and only if $a_i = a_j$. This is clearly the finest partition P to which $\hom_{\mathbb{N}}((M, \overline{a}), (X_P, \overline{c}^P)) > 0$. Furthermore, $\hom_{\mathbb{N}}((M, \overline{a}), (X_{P'}, \overline{c}^{P'})) > 0$ for any partition P' coarser than P. Note that this analysis also holds for all other σ-models with k distinguished elements, and so any such model which agrees with $(M, \overline{a})$ on Γ_σ satisfies $(\dagger)$ for all $i, j \leq k$. □

It is worth noting that, given a fixed signature, the size of the collection of models in Lemma 3.3.1 depends only on the fixed number of distinguished elements k, and hence is constant in the size of the domain of the model. Additionally, it follows that any models which agree on Γ_σ must have exactly the same number of distinct distinguished elements. The next lemma will allow us to also count the non-distinguished elements of σ-models.

Lemma 3.3.2. There exists a σ-model $(X_\sigma, \overline{c})$ with $\overline{c} = c_1, \ldots, c_k$ such that, for any σ-models $(M, \overline{a})$ and $(N, \overline{b})$ with $\overline{a} = a_1, \ldots, a_k$ and $\overline{b} = b_1, \ldots, b_k$, if $(M, \overline{a})$ and $(N, \overline{b})$ do not have the same number of distinct non-distinguished elements, then $(M, \overline{a})$ and $(N, \overline{b})$ are separated by $(X_\sigma, \overline{c})$.

Proof.
Define $(X_\sigma, \overline{c})$ as follows. The domain of the model will have two elements, d_1 and d_2. We set $c_i = d_1$ for each $i \leq k$. The facts of the model will be all possible facts over its domain. Let $(M, \overline{a})$ be any model, and define

$$n = |\{m \in \mathrm{dom}(M) \mid m \neq a_i \text{ for any } i \leq k\}|.$$

In other words, n is the number of non-distinguished elements of $(M, \overline{a})$. Clearly, any map $h : \mathrm{dom}(M) \to \mathrm{dom}(X_\sigma)$ such that $a_i \mapsto d_1$ for each $i \leq k$ is a homomorphism. To count the number of such maps, we can observe that any non-distinguished elements of $(M, \overline{a})$ can be mapped freely to either element of $(X_\sigma, \overline{c})$. It follows that

$$\hom_{\mathbb{N}}((M, \overline{a}), (X_\sigma, \overline{c})) = 2^n.$$

In particular, $(M, \overline{a})$ has exactly n-many non-distinguished elements. Since this analysis also holds for any other σ-model, we have that any $(N, \overline{b})$ with $\overline{b} = b_1, \ldots, b_k$ which agrees with $(M, \overline{a})$ on $(X_\sigma, \overline{c})$ must have the same number of non-distinguished elements. □

It follows from Lemma 3.3.1 and Lemma 3.3.2 that whenever σ-models $(M, \overline{a})$ and $(N, \overline{b})$ agree on all examples of $\Gamma_\sigma \cup \{(X_\sigma, \overline{c})\}$, we must have that $|\mathrm{dom}(M)| = |\mathrm{dom}(N)|$. Recall now that our goal is, given a model $(M, \overline{a})$, to provide a collection of examples Σ such that any model $(N, \overline{b})$ which agrees with $(M, \overline{a})$ on Σ must be isomorphic to $(M, \overline{a})$.

We begin with a naive, non-constructive argument which makes use of the Chaudhuri-Vardi theorem (Theorem 2.4.7). We can start our finite characterization for a σ-model $(M, \overline{a})$ with the collection $\Gamma_\sigma \cup \{X_\sigma, \overline{c}\}$. By the preceding two lemmas, all σ-models which do not have domain size $|\mathrm{dom}(M)|$ are separated by this collection. Hence we need only to ensure that all non-isomorphic σ-models with domain size $|\mathrm{dom}(M)|$ (of which there are finitely many) are separated by some example in our finite characterization. By the Chaudhuri-Vardi theorem, we know that any $(N, \overline{b})$ which is not isomorphic to $(M, \overline{a})$ must disagree on the k^{th} entry of their right profiles for some $k \in \mathbb{N}$. For each such model, we can add the model $(M_k, \overline{a}^{M_k})$ to our collection. It follows that any model not isomorphic to $(M, \overline{a})$ is separated from $(M, \overline{a})$ by some example in the collection, and so it is a finite characterization of $(M, \overline{a})$.

The argument in the preceding paragraph has several undesirable qualities. The first is that its use of the Chaudhuri-Vardi theorem to find separating examples is non-constructive, and so it is not clear how we would compute such a finite characterization in practice. The second issue is that we have no way of bounding the size of these non-constructively produced examples. We know that they exist, but it could be that they have a domain significantly larger than $|\mathrm{dom}(M)|$. Finally, given a signature σ whose maximum-arity relation symbol has arity k, the number of σ-models of size n is $O(2^{n^k})$.

We will now refine our argument to address these issues, providing a finite characterization Σ for an arbitrary finite model $(M, \overline{a})$ which is constructive, contains only models of domain size at most $|\text{dom}(M)|$, and has a total size of $O(2^{\text{dom}(M)})$. To this end, we will construct Σ first by including $(X_\sigma, \overline{c})$ and Γ_σ to enforce that any model $(N, \overline{b})$ which agrees with $(M, \overline{a})$ on Σ has domain size $|\text{dom}(M)|$. Then we will include examples to ensure that any non-isomorphic σ-model of size $|\text{dom}(M)|$ is separated by some example in Σ. It follows that Σ is a finite characterization of $(M, \overline{a})$. To do this, we first need the following technical lemma.

Lemma 3.3.3. Let $(M, \overline{a})$, $(N, \overline{b})$, and $(A, \overline{c})$ be σ-models with $\overline{a} = a_1, \ldots, a_k$, $\overline{b} = b_1, \ldots, b_k$, and $\overline{c} = c_1, \ldots, c_k$. If $g \in \text{Sur}_{\mathbb{N}}((M, \overline{a}), (N, \overline{b}))$ and $h_i : (N, \overline{b}) \to (A, \overline{c})$ is distinct for each $i \leq n$, then $h_i \circ g : (M, \overline{a}) \to (A, \overline{c})$ is a distinct homomorphism for each $i \leq n$.

Proof.
Suppose that $g \in \text{Sur}_{\mathbb{N}}((M, \overline{a}), (N, \overline{b}))$ and $h_i : (N, \overline{a}) \to (A, \overline{a})$ is distinct for each $i \leq n$, and suppose for a contradiction that there exists h_j, h_k for distinct $j, k \leq n$ such that $h_j \circ g = h_k \circ g$. Then for each $m \in \text{dom}(M)$, we have that $h_j \circ g(m) = h_k \circ g(m)$. Let $n \in \text{dom}(N)$ be arbitrary. Then since g is surjective, there's some $m \in \text{dom}(M)$ such that $g(m) = n$. Thus $h_j(n) = h_j \circ g(m) = h_k \circ g(m) = h_k(n)$. Therefore, $h_j = h_k$, which is a contradiction. □

The next definition provides useful collections of σ-models that we will use to build our finite characterizations.

Definition 3.3.4. For any σ-model $(M, \overline{a})$ with $\overline{a} = a_1, \ldots, a_k$, we define

1. $F^{-1}((M, \overline{a}))$ to be the collection of σ-models obtained by removing one fact from $(M, \overline{a})$;
2. $Sub((M, \overline{a}))$ to be the collection of induced submodels of $(M, \overline{a})$;
3. $S^{max}((M, \overline{a}))$ to be the collection of induced submodels of $(M, \overline{a})$ of domain size $|\text{dom}(M)| - 1$.

The next two lemmas will show how to use various examples, or collections of examples, to separate any non-isomorphic models $(M, \overline{a})$ and $(N, \overline{b})$ of the same domain size.

Lemma 3.3.5. Let $(M, \overline{a})$ and $(N, \overline{b})$ be non-isomorphic σ-models with $\overline{a} = a_1, \ldots, a_k$ and $\overline{b} = b_1, \ldots, b_k$, and suppose that $|\text{dom}(M)| = |dom(N)|$. If $\text{sur}_{\mathbb{N}}((N, \overline{b}), (M, \overline{a})) \neq 0$, then $(M, \overline{a})$ and $(N, \overline{b})$ are separated by some $(A, \overline{a}) \in F^{-1}((M, \overline{a}))$.

Proof.
Suppose that $\text{sur}_{\mathbb{N}}((N, \overline{b}), (M, \overline{a})) \neq 0$, and let $g : (N, \overline{b}) \to (M, \overline{a})$ be a surjective homomorphism. Since $|\text{dom}(M)| = |\text{dom}(N)|$, we have that g is also injective, and so $|\text{Facts}(N)| \leq |\text{Facts}(M)|$. If $|\text{Facts}(N)| = |\text{Facts}(M)|$, then g would be fully surjective (and hence an isomorphism). Then because we know that $(M, \overline{a}) \ncong (N, \overline{b})$, the inequality must be strict: $|\text{Facts}(N)| < |\text{Facts}(M)|$. This implies that an isomorphic copy of $(N, \overline{b})$ can be obtained by removing facts from $(M, \overline{a})$. Consequently, there exists some $(A, \overline{a}) \in F^{-1}((M, \overline{a}))$ such that g can also be seen as a surjective homomorphism from $(N, \overline{b})$ to $(A, \overline{a})$. However, since $(A, \overline{a})$ has fewer facts than $(M, \overline{a})$, we have that $\text{sur}_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = 0$. Let $\text{Hom}_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = \{h_1, \ldots, h_k\}$. Then

by applying Lemma 3.3.3, we have that $S = \{h_i \circ g \mid i \leq k\}$ is a set of k distinct homomorphisms, none of which are surjective. Hence

$$\hom_{\mathbb{N}}((N, \overline{b}), (A, \overline{a})) \geq k + 1 > \hom_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})),$$

which completes the proof. □

Lemma 3.3.6. Let $(M, \overline{a})$ and $(N, \overline{b})$ be non-isomorphic σ-models with $\overline{a} = a_1, \ldots, a_k$ and $\overline{b} = b_1, \ldots, b_k$, and suppose that $|\mathrm{dom}(M)| = |dom(N)|$. If $\mathrm{sur}_{\mathbb{N}}((N, \overline{b}), (M, \overline{a})) = 0$, then $(M, \overline{a})$ and $(N, \overline{b})$ are separated by some $(A, \overline{a}) \in Sub((M, \overline{a}))$.

Proof.
Suppose that $\mathrm{sur}_{\mathbb{N}}((N, \overline{b}), (M, \overline{a})) = 0$. We want to show that $(M, \overline{a})$ and $(N, \overline{b})$ are separated by at least one element of $Sub((M, \overline{a}))$, possibly $(M, \overline{a})$ itself. To this end, suppose that $(M, \overline{a})$ and $(N, \overline{b})$ are not separated by any proper substructure $(A, \overline{a})$ of $(M, \overline{a})$. In other words,

$$\hom_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = \hom_{\mathbb{N}}((N, \overline{b}), (A, \overline{a}))$$

for each $(A, \overline{a}) \in Sub((M, \overline{a})) \setminus \{(M, \overline{a})\}$. We will now argue that $(M, \overline{a})$ and $(N, \overline{b})$ are separated by $(M, \overline{a})$. First, we observe that, for any σ-models $(C, \overline{c})$ and $(A, \overline{a})$, we have that

$$\begin{aligned}
\hom_{\mathbb{N}}((C, \overline{c}), (A, \overline{a})) &= \sum_{(A', \overline{a}) \in Sub((A, \overline{a}))} \mathrm{sur}_{\mathbb{N}}((C, \overline{c}), (A', \overline{a})) \\
&= \left(\sum_{(A', \overline{a}) \in Sub((A, \overline{a})) \setminus \{A, \overline{a}\}} \mathrm{sur}_{\mathbb{N}}((C, \overline{c}), (A', \overline{a})) \right) \\
&\qquad + \mathrm{sur}_{\mathbb{N}}((C, \overline{c}), (A, \overline{a})). \qquad (\dagger)
\end{aligned}$$

Consider the following claim.

Claim: For each $(A, \overline{a}) \in Sub((M, \overline{a})) \setminus \{(M, \overline{a})\}$, we have that

$$\mathrm{sur}_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = \mathrm{sur}_{\mathbb{N}}((N, \overline{b}), (A, \overline{a})).$$

Proof.
We proceed by induction on $|\mathrm{dom}(A)|$. For the base case, $|\mathrm{dom}(A)| = 1$, and so every homomorphism into $(A, \overline{a})$ must be surjective. Thus, by the assumption that

$$\hom_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = \hom_{\mathbb{N}}((N, \overline{b}), (A, \overline{a})),$$

we have that

$$\mathrm{sur}_{\mathbb{N}}((M, \overline{a}), (A, \overline{a})) = \mathrm{sur}_{\mathbb{N}}((N, \overline{b}), (A, \overline{a})).$$

For the inductive step, suppose that

$$\mathrm{sur}_{\mathbb{N}}((M, \overline{a}), (A', \overline{a})) = \mathrm{sur}_{\mathbb{N}}((N, \overline{b}), (A', \overline{a}))$$

for all $(A', \overline{a}) \in Sub((M, \overline{a}))$ such that $|\mathrm{dom}(A')| < |\mathrm{dom}(A)|$. Then, in particular, since

$$Sub((A, \overline{a})) \subseteq Sub((M, \overline{a})),$$

we have that

$$\mathrm{sur}_{\mathbb{N}}((M,\overline{a}),(A',\overline{a})) = \mathrm{sur}_{\mathbb{N}}((N,\overline{b}),(A',\overline{a}))$$

for each $(A',\overline{a}) \in Sub((A,\overline{a}))$. Then since we have that

$$\hom_{\mathbb{N}}((M,\overline{a}),(A,\overline{a})) = \hom_{\mathbb{N}}((N,\overline{b}),(A,\overline{a})),$$

we can conclude by (†) and the inductive hypothesis that

$$\mathrm{sur}_{\mathbb{N}}((M,\overline{a}),(A,\overline{a})) = \mathrm{sur}_{\mathbb{N}}((N,\overline{b}),(A,\overline{a})).$$

This completes the inductive step, proving the claim. □

From the claim, we can conclude that

$$\sum_{(A,\overline{a})\in Sub((M,\overline{a}))\setminus\{(M,\overline{a})\}} \mathrm{sur}_{\mathbb{N}}((M,\overline{a}),(A,\overline{a})) = \sum_{(A,\overline{a})\in Sub((M,\overline{a}))\setminus\{(M,\overline{a})\}} \mathrm{sur}_{\mathbb{N}}((N,\overline{b}),(A,\overline{a})).$$

Then since $(M,\overline{a})$ also has the surjective homomorphism $id : (M,\overline{a}) \to (M,\overline{a})$, while by assumption $\mathrm{sur}((N,\overline{b}),(M,\overline{a})) = 0$, we conclude by (†) that $\hom_{\mathbb{N}}((M,\overline{a}),(M,\overline{a})) \geq k+1 > \hom_{\mathbb{N}}(((N,\overline{b}),\overline{b}),(M,\overline{a}))$, and so $(M,\overline{a})$ and $(N,\overline{b})$ are separated by $(M,\overline{a})$. □

We are now ready to prove our main result.

Theorem 3.3.7. Every σ-model $(M,\overline{a})$ can be finitely characterized with $O(2^{|\mathrm{dom}(M)|})$ examples, where the domain size of each example is at most $|\mathrm{dom}(M)|$.

Proof.
Fix some σ-model $(M,\overline{a})$, and define

$$\Sigma((M,\overline{a})) := \{(X_\sigma,\overline{c})\} \cup \Gamma_\sigma \cup F^{-1}((M,\overline{a})) \cup Sub((M,\overline{a})).$$

We observed earlier that $\{(X_\sigma,\overline{c})\} \cup \Gamma_\sigma$ has a size that is constant in $|\mathrm{dom}(M)|$. Since $|\mathrm{Facts}(M)|$ is polynomial in $|\mathrm{dom}(M)|$, we have that $F^{-1}((M,\overline{a}))$ is polynomial in $|\mathrm{dom}(M)|$, and since $Sub((M,\overline{a})) < 2^{|\mathrm{dom}(M)|}$, we obtain that $|\Sigma((M,\overline{a}))| = O(2^{|\mathrm{dom}(M)|})$. To see that $\Sigma((M,\overline{a}))$ is a finite characterization of $(M,\overline{a})$, fix an arbitrary σ-model $(N,\overline{b})$ which is not isomorphic to $(M,\overline{a})$. If $|\mathrm{dom}(M)| \neq |\mathrm{dom}(N)|$, then by Lemmas 3.3.1 and 3.3.2, $(M,\overline{a})$ and $(N,\overline{b})$ are separated by either $(X_\sigma,\overline{c})$ or some $(X_P,\overline{c}^P) \in \Gamma_\sigma$. Now suppose $|\mathrm{dom}(M)| = |\mathrm{dom}(N)|$, and distinguish cases:

1. If $\mathrm{sur}_{\mathbb{N}}((N,\overline{b}),(M,\overline{a})) \neq 0$, then $(M,\overline{a})$ and $(N,\overline{b})$ are separated by some $(A,\overline{a}) \in F^{-1}((M,\overline{a}))$ (Lemma 3.3.5).
2. If $\mathrm{sur}_{\mathbb{N}}((N,\overline{b}),(M,\overline{a})) = 0$, then $(M,\overline{a})$ and $(N,\overline{b})$ are separated by some $(A,\overline{a}) \in Sub((M,\overline{a}))$ (Lemma 3.3.6).

Thus every non-isomorphic σ-model $(N,\overline{b})$ is separated from $(M,\overline{a})$ by some example in $\Sigma((M,\overline{a}))$. Therefore, $\Sigma((M,\overline{a}))$ is a finite characterization of $(M,\overline{a})$. □

Theorem 3.3.7 provides an explicit construction, for any model $(M,\overline{a})$, of a set $\Sigma((M,\overline{a}))$ which contains a separating example for any non-isomorphic σ-model. This implies that any non-isomorphic σ-models must disagree on some entry of their counting right profiles. In particular, this construction constitutes an alternative proof of the Chaudhuri-Vardi theorem.

We will now briefly explain how these results can be extended to hold for queries whose signature contains some finite number of constants. Let $(M, \overline{a})$ be the canonical model of the query to be characterized. First, observe that our treatment of distinguished elements is very similar to constants: we require that they be preserved by homomorphisms. Then we can construct a set S analogous to the construction of Γ_σ which enforces that equalities between interpretations of constants hold in a model which agrees with $(M, \overline{a})$ on S if and only if those same equalities hold in $(M, \overline{a})$. Aside from this, the only other significant change is that we must define induced submodels to always contain all interpretations of constants. The remainder of the upper bound proof remains the same. It follows that our setting is strictly more general than that of Böttcher et. al.

An additional point worthy of mention is the complexity of enumerating the finite characterization in Theorem 3.3.7. Each model in the characterization for $(M, \overline{a})$ has domain size at most the size of $\mathrm{dom}(M)$. Furthermore, we claim that we can compute the examples using linear space and exponential time. That we can do this for $(X_\sigma, \overline{c})$ and Γ_σ is clear. We can also do this in the case of $F^{-1}((M, \overline{a}))$ by removing one fact at a time, while in the case of $Sub((M, \overline{a}))$ we need only to traverse the power set of $\mathrm{dom}(M)$ in a fixed order.

3.4 Finite Characterizations for Restricted Classes

In this section, we will establish the existence of even smaller finite characterizations for more restricted classes of σ-models, where σ is an arbitrary finite relational signature. To begin, first note that, in the construction of $\Sigma((M, \overline{a}))$ in Theorem 3.3.7, the source of the exponential size in general of the finite characterization is due to the inclusion of $Sub((M, \overline{a}))$. However, it's worth noting that many classes of models do not have exponentially-many induced submodels. Hence we obtain the following result.

Corollary 3.4.1. Let $\mathcal{C}$ be any class of σ-models such that the number of non-isomorphic induced submodels of each model with domain size n in $\mathcal{C}$ is $O(n^k)$ for some fixed k. Then there exist finite characterizations of the models in $\mathcal{C}$ which are bounded by some polynomial in the domain size of the models.

For an example of a class satisfying the requirements of Corollary 3.4.1, consider the class of cliques with self-loops. In other words, the collection of all τ-models $(M, \overline{a})$, with $R^M(m, n)$ for each $m, n \in \mathrm{dom}(M)$, where τ is a signature containing only one binary relation R. Each model of this class has at most one submodel (up to isomorphism) of each cardinality smaller than $\mathrm{dom}(M)$, and so $|Sub(M)| = O(|\mathrm{dom}(M)|)$.

We can also prove better bounds on the size of finite characterizations for cores. Recall that a core is a model with only surjective endomorphisms. Many important classes of models are cores, including linear orders, cliques (without self-loops), and directed cycles. Below, we show that each model in the class of cores has a finite characterization whose size is polynomial in the domain size of the model.

Theorem 3.4.2. Any σ-model $(M, \overline{a})$ which is a core can be finitely characterized with $O(|\mathrm{dom}(M)|^{k_\sigma})$ examples, where k_σ denotes the maximum arity of any relation symbol in σ.

Proof.
Let $(M, \overline{a})$ be a σ-model, and define

$$\Delta((M, \overline{a})) = \{(X_\sigma, \overline{c}), (M, \overline{a})\} \cup \Gamma_\sigma \cup F^{-1}((M, \overline{a})) \cup S^{max}((M, \overline{a})).$$

We claim that $\Delta((M,\overline{a}))$ is a finite characterization of $(M,\overline{a})$; since the size of $\Delta((M,\overline{a}))$ is dominated by $|F^{-1}((M,\overline{a}))| = O(|\text{dom}(M)|^{k_\sigma})$, the result follows. Let $(N,\overline{b})$ be any model which agrees with $(M,\overline{a})$ on $\Delta((M,\overline{a}))$. By Lemmas 3.3.1 and 3.3.2, since $(M,\overline{a})$ and $(N,\overline{b})$ agree on $(X_\sigma,\overline{c})$ and Γ_σ, we have that M and N have the same domain size. Since $(M,\overline{a})$, as a core, has no non-surjective endomorphisms, we conclude that $\hom_{\mathbb{N}}((M,\overline{a}),(B,\overline{a})) = 0$ for all $(B,\overline{a}) \in S^{max}((M,\overline{a}))$, and hence $\hom_{\mathbb{N}}((N,\overline{b}),(B,\overline{a})) = 0$ for all $(B,\overline{a}) \in S^{max}((M,\overline{a}))$. Thus every homomorphism from $(N,\overline{b})$ to $(M,\overline{a})$ must be surjective.

Let $h : (N,\overline{b}) \to (C_N,\overline{b})$ be any homomorphism, which by the definition of cores must be surjective. Note that no non-surjective homomorphism $h' : (C_N,\overline{b}) \to (M,\overline{a})$ can exist (otherwise, $h' \circ h : (N,\overline{b}) \to (M,\overline{a})$ would be a non-surjective homomorphism). Thus for any homomorphism $h : (N,\overline{b}) \to (M,\overline{a})$, we have that the map $h \restriction \text{dom}(C_N) : (C_N,\overline{b}) \to (M,\overline{a})$, which is also a homomorphism, must be surjective. Hence $|\text{dom}(N)| = |\text{dom}(M)| \leq |\text{dom}(C_N)|$, and so $\text{dom}(N) = \text{dom}(C_N)$, which implies that $(N,\overline{b})$ is a core.

Since $|\text{dom}(M)| = |\text{dom}(N)|$ and $(N,\overline{b})$ has a surjective homomorphism to $(M,\overline{a})$, it follows that $|Facts(N)| \leq |Facts(M)|$. If $|Facts(N)| < |Facts(M)|$, then there's a homomorphism $g : (N,\overline{b}) \to (A,\overline{a})$ for some $(A,\overline{a}) \in F^{-1}((M,\overline{a}))$. Then because $(M,\overline{a})$ and $(N,\overline{b})$ agree on every example in $F^{-1}((M,\overline{a}))$, there's some homomorphism $f : (M,\overline{a}) \to (A,\overline{a})$. Since f can also be seen as a homomorphism from $(M,\overline{a})$ to itself, we conclude that f must be surjective. Hence we have that $(M,\overline{a})$ has at most as many facts as $(A,\overline{a})$, contradicting the definition of $(A,\overline{a})$. Thus $(N,\overline{b})$ must have the same number of facts as $(M,\overline{a})$, and so $(N,\overline{b}) \cong (M,\overline{a})$. □

3.5 Conclusion

Recall that our initial motivation for studying finite characterizations of models was to obtain characterizations of conjunctive queries up to answer-equivalence under the bag semantics. By the correspondence between finite relational models and conjunctive queries, we have succeeded in this regard. Phrased in database theoretic terms, the results of this chapter imply that, for any conjunctive query Q, we can find a finite collection of labeled database examples such that any query Q' which fits all of the labels on each example is answer-equivalent to Q under the bag semantics. These finite collections can be computed in exponential time in general, and in polynomial time in certain special cases. In particular, we were interested in computing characterizing examples for use in debugging, allowing users to verify correctness of complex queries; given the query Q, our constructions are all enumerable and permit this application. Furthermore, our proof of the existence of finite characterizations shows that a necessary condition for exact learning of conjunctive queries under the counting semantics holds.

In finite model-theoretic terms, we have provided exponentially-large constructions of finite characterizations for arbitrary models and polynomially-large constructions for some special cases of models, where all examples are bounded by the domain size of the characterized model. These constructions only provide upper bounds on the size of finite characterizations, and we leave open the question of whether or not we could make do with less. In particular, the problem of determining a tight lower bound on the size of finite characterizations in the general case appears to be combinatorially difficult. We are also interested in whether or not c-acyclic conjunctive queries can be finitely characterized

under the counting semantics with only polynomially-many examples, as they can be under the Boolean semantics.

Finite characterizations can be seen within the larger context of *right query algorithms*. A right query algorithm for a class of conjunctive queries takes as input a query and poses that query to a sequence of examples, determining membership of the query in the class based on the answers. In an *adaptive* right-query algorithm, later examples in the sequence are allowed to be conditioned on the answers to earlier examples. These were studied in [CFLX22] under the counting semantics for the class of undirected graphs. In a *non-adaptive* right-query algorithm, the sequence of examples is fixed in advance. In [CDKW23], ten Cate et. al. expanded the notion of query algorithms under the counting semantics to arbitrary semiring semantics. Under the counting semantics, a finite characterization can then be seen as providing a non-adaptive right-query algorithm which determines membership in the class of models isomorphic to the characterized model.

Chapter 4

Left-Homomorphism Vectors and Modal Relations

Lovász's Theorem (Theorem 2.4.7) provides that equivalence of the counting left profile captures isomorphism between finite models. A natural follow-up question is to ask: which equivalence relations between finite models (coarser than isomorphism) can be captured by an appropriate restriction of the left profile? In this chapter, we explore this question for the natural model-theoretic relations arising in process theory and modal logic, namely, simulation, directed simulation, bisimulation, and graded bisimulation.

4.1 Characterizing Equivalence Relations with Restricted Left Profiles

Lovász's theorem grew out of the study of a fundamental computational problem in graph and complexity theory: the (undirected) graph isomorphism problem. This problem is significant because it is not known to be solvable in polynomial time, but is also not known to be **NP**-complete. In fact, recent work by Babai has shown that the problem can be resolved in quasipolynomial time [Bab16], and it is considered to be a potential member of the conjectured class of **NP**-intermediate problems, which exist if and only if $\mathbf{P} \neq \mathbf{NP}$. Due to the relatively high running time of known exact algorithms for the graph isomorphism problem, and the difficulty in determining a lower bound on its complexity, researchers have developed many heuristic algorithms that have found much use in practice. One such algorithm is the color-refinement algorithm, which can very quickly distinguish many (but not all) non-isomorphic graphs.

The study of restrictions of the left homomorphism vector was initiated by Dvořák, who showed that two undirected graphs have the same left profile restricted to trees if and only if they are indistinguishable by the color-refinement algorithm [Dvo10]. This result, later proven independently by Dell et. al. [DGR18], is analogous to that of Lovász: rather than characterizing isomorphism with left profile equivalence over the class of all structures, it characterizes indistinguishability by the color-refinement algorithm with left profile equivalence over the class of trees. In fact, Dvořák and Dell et. al. provided a more general result: the left profile restricted to graphs of tree width k captures indistinguishability by the k-dimensional Weisfeiler-Leman (WL) method, where the color-refinement algorithm is the special case of the k-dimensional WL method for $k = 1$.

This is not the end of the story: there are other ways to characterize classes of graphs indistinguishable by the color refinement algorithm. Given two graphs with adjacency

matrices A and B, an isomorphism between them can be interpreted as a permutation matrix X such that $AX = B$. When we drop the requirement that X contain only binary values, allowing instead positive rational number entries such that each column and row sums to 1, then we say that X is a *fractional isomorphism*. The existence of a fractional isomorphism between two graphs is strictly weaker than the existence of an isomorphism, and so induces a less-refined equivalence relation on the class of all graphs. Fractional isomorphisms are an inherently *linear algebraic* notion, and yet it has also been shown that two graphs are indistinguishable by the color-refinement algorithm if and only if a fractional isomorphism exists between their adjacency matrices [Tin86, Tin91].

In logic, the *two-variable fragment* (FO^2) is the fragment of first-order logic in which only two variable are allowed. An important extension of this language is the *two-variable fragment with counting quantifiers* (C^2), which contains quantifiers of the form $\exists^{\geq k}$, such that $\exists^{\geq k} x\varphi(x)$ asserts the existence of at least k elements satisfying $\varphi(x)$. The logic C^2 is important because its satisfiability problem is decidable, unlike that of full first-order logic. In addition, a theorem of Cai et. al. shows that two graphs are invariant under C^2 (satisfying the same formulas) if and only if they are indistinguishable by the the color-refinement algorithm [CFI92]. In fact, they show that two graphs are invariant under the k-variable fragment with counting quantifiers (C^k), which naturally generalizes C^2, if and only if they are distinguishable by the $(k-1)$-dimensional WL method (for $k \geq 2$).

Finally, in artificial intelligence, *graph neural networks* (GNNs) are a type of machine learning architecture, represented using graphs, which have found applications in areas such as bioinformatics, web analysis, and natural language processing. In [MRF+19], Morris et. al. showed that the power of GNNs to distinguish non-isomorphic graphs is precisely the same as that of the color-refinement algorithm. Inspired by the observation that C^2 and the color-refinement algorithm can be lifted to C^k and the k-dimensional WL method, respectively, the authors proposed a natural generalization of GNNs, which they call k-dimensional GNNs. They showed that these k-dimensional GNNs can distinguish non-isomorphic graphs with the same expressive power as the k-dimensional WL method.

We have now seen that several seemingly distinct notions – the color-refinement algorithm from graph theory, fractional isomorphism from linear algebra, the two-variable fragment with counting quantifiers from logic, and graph neural networks from machine learning – all induce the same equivalence class on the class of undirected graphs. Furthermore, they are all undergirded by the same phenomenon: the expressive power of homomorphism vectors restricted to a particular class of graphs. These results, as well as some others found in the literature, are depicted in Figure 4.1.

Invariance Relation	**Restriction**	**Citation**
Color-refinement indistinguishability Fractional isomorphism C^2 equivalence GNN indistinguishability	$\hom_{\mathbb{N}}(\mathbb{T}, G)$	[Dvo10, DGR18] [Tin86, Tin91] [CFI92] [MRF+19]
k-dimensional WF indistinguishability C^{k+1} equivalence k-dimensional GNN indistinguishability	$\hom_{\mathbb{N}}(\mathbb{T}_k, G)$	[Dvo10, DGR18] [CFI92] [MRF+19]
Quantifier depth k FO-invariance	$\hom_{\mathbb{N}}(\mathbb{T}^k, G)$	[Gro20]
Co-spectrality	$\hom_{\mathbb{N}}(\mathbb{C}, G)$	[DGR18]

Figure 4.1: Summary of known characterization results for equivalence relations on graphs, where $\mathbb{T}$ is the class of undirected trees, $\mathbb{T}_k$ is the class of graphs of tree width k, $\mathbb{T}^k$ is the class of graphs of tree depth k, and $\mathbb{C}$ is the class of undirected cycles.

Because of these deep results, Atserias et. al. set out to determine what equivalence relations between graphs can be expressed by restricting homomorphism vectors to some fixed class of graphs [AKW21], pursuing both positive and negative results. For negative results, they showed that chromatic equivalence and FO^k-equivalence cannot be captured by any restriction of the left homomorphism vector to a class of graphs. In this chapter, we pursue a similar line of research, with an emphasis on *modal logics*.

In particular, Barcelo et. al. recently showed that nodes of undirected graphs are indistinguishable by a special case of GNNs (aggregate-combine GNNs) if and only if they satisfy the same formulas of graded modal logic (i.e., are *graded modal equivalent*), a syntactic fragment of C^2 [BKM$^+$20]. Given that indistinguishability by standard GNNs and C^2 can be captured by the restriction of the left homomorphism vector to the class of undirected trees, this result suggests that a similar restriction should exist which captures graded modal logic. However, the structures over which graded modal logic is typically interpreted are not undirected graphs; rather, they are a more general type of structure arising in process theory: *labeled transition systems*.

In this chapter, we are interested in finding classes which capture the natural model-theoretic equivalence relations arising in process theory and modal logic, or otherwise proving that no such class exists. Similar to the results we have seen in the undirected case, we will see that *tree-like* structures will be of critical importance in this regard. However, a consequence of the more general setting of labeled transition systems will be that we must consider structures with directed edges. As we will see, different notions of tree-like structures will characterize different modal logics, and so our more general setting is of great consequence for our results.

4.2 Basic Modal Logic

Definition 4.2.1. A *labeled transition system* (LTS) is any model whose signature contains binary relations R_i for $i \in I$ (where I is some finite index set), called *actions*, and some finite number of unary predicates, called *proposition letters*. We refer to domain elements of such a model as *states*.

Note that we will work with both finite and infinite LTSs throughout this chapter, and so we will always explicitly state when models are finite or infinite. We will primarily work with *pointed* LTSs, which are LTSs with one distinguished element. For a pointed LTS (M, a), we will prefer the notation M_a. Given an LTS M and an element $a \in M$, we write $R_i[a]$ for the set $\{b \in M \mid R_i^M(a, b)\}$, and we refer to the elements of this set as *R_i-successors* of a in M; we define $R[a] := \bigcup_{i \in I} R_i[a]$. We use p, q, r to denote proposition letters, and we write Prop to denote the set of proposition letters. It will sometimes be convenient to refer to the set of proposition letters true at a state $m \in M$, which we call the *marking* of m (notation: $\pi(m)$).

Definition 4.2.2. The *basic modal language* is the collection BML of formulas generated by the following recursive grammar:

$$\varphi := p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond_i \varphi \mid \Box_i \varphi,$$

where $p \in \mathrm{Prop}$ and $i \in I$. Symbols of the form $\Diamond_i$ and $\Box_i$ are called *modalities*; each modality corresponds to a different action in the signature.

We have been primarily dealing with conjunctive queries, which are formulas of first-order logic. As it turns out, there is a simple syntactic translation from formulas of the

basic modal language to equivalent formulas of first-order logic. This so-called "standard translation" witnesses that basic modal logic is a semantic fragment of first-order logic.

Definition 4.2.3. The *standard translation* of BML formulas is given by the following recursive definition.

$$\begin{aligned} ST_x(p) &:= P(x), \\ ST_x(\varphi \wedge \psi) &:= ST_x(\varphi) \wedge ST_x(\psi), \\ ST_x(\neg\varphi) &:= \neg ST_x(\varphi), \\ ST_x(\Diamond_i\varphi) &:= \exists y(R_i(x,y) \wedge ST_y(\varphi)), \text{ and} \\ ST_x(\Box_i\varphi) &:= \forall y(R_i(x,y) \rightarrow ST_y(\varphi)). \end{aligned}$$

Two restrictions of modal logic that we will consider later on are *positive modal logic* and *positive-existential modal logic*. Positive modal logic (notation: ML^+) is the fragment of ML which excludes negation, while positive-existential modal logic (notation: $\mathrm{ML}^+_\Diamond$) excludes both negation and the $\Box_i$ modalities. It is not difficult to see that the standard translation of any formula of $\mathrm{ML}^+_\Diamond$ is, in fact, expressible as a conjunctive query. Later on, we will say more about the structure of the queries expressible in $\mathrm{ML}^+_\Diamond$.

We have previously claimed that the basic modal language represents a semantic fragment of first-order logic. To see that this is actually the case, let us now take a look at the semantics of the basic modal language, from which it is easily seen that the standard translation does, in fact, preserve the semantics of the BML formulas being translated.

Definition 4.2.4. Given a pointed LTS M_a, we recursively define the modal satisfaction relation recursively as follows:

$M, a \models p$	if $a \in p^M$,
$M, a \models \varphi \wedge \psi$	if $M, a \models \varphi$ and $M, a \models \psi$,
$M, a \models \neg\varphi$	if $M, a \not\models \varphi$,
$M, a \models \Diamond_i\varphi$	if there's some $b \in R_i[a]$ such that $M, b \models \varphi$, and
$M, a \models \Box_i\varphi$	if for all $b \in R_i[a]$, we have that $M, b \models \varphi$.

We say that a formula φ of the basic modal language is *satisfied at* a pointed LTS M_a whenever $M_a \models \varphi$. We define the relation of *basic modal equivalence* (notation: $\equiv_{ML}$) between LTSs M_a and N_b to hold if and only if they satisfy the same formulas of the basic modal language. We similarly define *positive modal equivalence* (notation $\equiv_{\mathrm{ML}^+_\Diamond}$).

Definition 4.2.5. A *bisimulation* between pointed LTSs M_a and N_b is a binary relation $Z \subseteq M \times N$ with $(a, b) \in Z$ and meeting the following three conditions:

(prop) If $(m, n) \in Z$, then $\pi(m) = \pi(n)$.

(forth) For each $i \in I$, if $(m, n) \in Z$ and there's $s \in M$ such that $R_i^M(m, s)$, then there's some $t \in N$ such that $R_i^N(n, t)$ and $(s, t) \in Z$.

(back) For each $i \in I$, if $(m, n) \in Z$ and there's $t \in N$ such that $R_i^N(n, t)$, then there's some $s \in M$ such that $R_i^M(m, s)$ and $(s, t) \in Z$.

If such a relation $Z \subseteq M \times N$ exists, then we say M_a and N_b are *bisimilar* (notation: $M_a \leftrightarrows N_b$). Bisimilarity between two LTSs implies basic modal equivalence.

Theorem 4.2.6. (Bisimulation Invariance, [Ben14]) For LTSs M_a and N_b, if $M_a \leftrightarrows N_b$, then $M_a \equiv_{ML} N_b$.

It is natural to ask whether the converse of Theorem 4.2.6 holds. As it turns out, this it not true in general, but holds over the class of *image-finite* LTSs, which are those models M_a such that the successor set $R_i[m]$ is finite for each element $m \in M$ and $i \in I$.

Theorem 4.2.7. (Hennessy-Milner Theorem, [HM85]) For image-finite LTSs M_a and N_b, if $M_a \equiv_{ML} N_b$, then $M_a \leftrightarrows N_b$.

It is clear that all finite pointed LTSs must be image-finite. Thus Theorem 4.2.7 applies, in particular, to the class of finite LTSs. The name "bisimulation" indicates that the relation is bidirectional, which is captured by the equality in the (prop) clause, and the presence of the (back) clause. The next notion simultaneously weakens both of these.

Definition 4.2.8. A *simulation* between pointed LTSs M_a and N_b is a binary relation $Z \subseteq M \times N$ with $(a, b) \in Z$ and meeting the following three conditions:

(prop) If $(m, n) \in Z$, then $\pi(m) \subseteq \pi(n)$.

(forth) For each $i \in I$, if $(m, n) \in Z$ and there's $s \in M$ such that $R_i^M(m, s)$, then there's some $t \in N$ such that $R_i^N(n, t)$ and $(s, t) \in Z$.

We say M_a and N_b are *simulation equivalent* (notation: $M_a \leftrightarrows_s N_b$) if there exists a simulation $Z \subseteq M \times N$ as well as a simulation $Z' \subseteq N \times M$ with $(a, b) \in Z$ and $(b, a) \in Z'$. Simulation equivalence preserves formulas of $\mathrm{ML}^+_\Diamond$.

Theorem 4.2.9. (Simulation Equivalence Invariance, [JLW20]) For image-finite LTSs M_a and N_b, we have that $M_a \leftrightarrows_s N_b$ if and only if $M_a \equiv_{\mathrm{ML}^+_\Diamond} N_b$.

We can also weaken only the (prop) clause to obtain the following definition.

Definition 4.2.10. A *directed simulation* between pointed LTSs M_a and N_b is a binary relation $Z \subseteq M \times N$ with $(a, b) \in Z$ and meeting the following three conditions:

(prop) If $(m, n) \in Z$, then $\pi(m) \subseteq \pi(n)$.

(forth) For each $i \in I$, if $(m, n) \in Z$ and there's $s \in M$ such that $R_i^M(m, s)$, then there's some $t \in N$ such that $R_i^N(n, t)$ and $(s, t) \in Z$.

(back) For each $i \in I$, if $(m, n) \in Z$ and there's $t \in N$ such that $R_i^N(n, t)$, then there's some $s \in M$ such that $R_i^M(m, s)$ and $(s, t) \in Z$.

We say M_a and N_b are *directed simulation equivalent* (notation: $M_a \leftrightarrows_d N_b$) if there exists a directed simulation $Z \subseteq M \times N$ as well as a directed simulation $Z' \subseteq N \times M$ with $(a, b) \in Z$ and $(b, a) \in Z'$. Directed simulation equivalence preserves formulas of ML^+.

Theorem 4.2.11. (Directed Simulation Equivalence Invariance, [KR97]) For image-finite LTSs M_a and N_b, we have that $M_a \leftrightarrows_d N_b$ if and only if $M_a \equiv_{\mathrm{ML}^+} N_b$.

For most of this chapter, we will be interested in characterizing the relations induced between LTSs by restricting their Boolean and counting homomorphism vectors to the class of directed tree-shaped LTSs, defined as follows.

Definition 4.2.12. A *directed tree-shaped LTS* is a pointed LTSs M_a that is a directed tree with respect to $R_1^M, \ldots, R_n^M$ (cf. Definition 2.5.8). We write $\mathcal{T}$ for the class of all finite directed tree-shaped LTSs, and we write $\mathcal{T}^k$ for the class of all finite directed tree-shaped LTSs of depth k.

Recall the following definition from Chapter 2:

Definition 2.4.8. Let $\mathcal{C}$ be some class of models. We write $\mathrm{Inj}(\mathcal{C})$ for the class of models $(N, \bar{b})$ such that there exists some injective homomorphism $h : (N, \bar{b}) \to (M, \bar{a})$ for some $(N, \bar{b}) \in \mathcal{C}$. We write $\mathrm{Sur}(\mathcal{C})$ for the class of models $(N, \bar{b})$ such that there exists some fully-surjective homomorphism $h : (M, \bar{a}) \to (N, \bar{b})$ for some $(N, \bar{b}) \in \mathcal{C}$. We define the *extension class* of $\mathcal{C}$ to be $\mathrm{Ext}(\mathcal{C}) := \mathrm{Inj}(\mathcal{C}) \cap \mathrm{Sur}(\mathcal{C})$.

The proof of the next proposition is adapted from a similar proof for undirected trees from [AKW21].

Proposition 4.2.13. $\mathcal{T} = \mathrm{Ext}(\mathcal{T})$.

Proof.
It's easy to see that $\mathcal{T} \subseteq Ext(\mathcal{T})$, and so we need only show that every $M \in \mathrm{Ext}(\mathcal{T}) = \mathrm{Inj}(\mathcal{T}) \cap \mathrm{Sur}(\mathcal{T})$ is a directed tree-shaped LTS. To see this, first observe that, since $M \in \mathrm{Inj}(\mathcal{T})$, we have that M is a submodel of a directed tree-shaped LTS. Furthermore, since $M \in \mathrm{Sur}(\mathcal{T})$, and every directed tree-shaped LTS is connected, we must have also that M is connected. Since M is a connected submodel of a directed tree-shaped LTS, it must also be a directed tree-shaped LTS. □

4.3 Capturing Simulation Equivalence

In this section, we will characterize $\mathrm{ML}^{+}_{\Diamond}$ equivalence for image-finite LTSs via the restriction of their left-profiles to the class of directed tree-shaped LTSs. By Theorem 4.2.9, this is equivalent for the class of image-finite models to simulation invariance. Before stating the theorem, we begin with a key lemma.

Lemma 4.3.1. If φ is a $\mathrm{ML}^{+}_{\Diamond}$ formula, then $ST_x(\varphi)$ is a conjunctive query whose canonical model is a finite directed tree-shaped LTS.

Proof.
We proceed by induction on the complexity of formulas. For the base case, φ is just a proposition letter, in which case the canonical model of $ST_x(\varphi)$ is just a single element at which the proposition letter p is true. For the inductive step, either $ST_x(\varphi) = ST_x(\psi_1) \wedge ST_x(\psi_2)$ for some formulas ψ_1, ψ_2, or $ST_x(\varphi) = \exists y(R_i(x, y) \wedge \psi)$ for some formula ψ. In the first case, the canonical model of $ST_x(\varphi)$ is just the directed tree-shaped LTS obtained by equating the roots of the canonical models of $ST_x(\psi_1)$ and $ST_x(\psi_2)$. In the second case, the standard translation of $ST_x(\varphi)$ is the tree obtained by adding a new root to the canonical model of $ST_x(\psi)$ with the old root as an R_i-successor. □

Up to this point, we have only defined counting left homomorphism vectors with respect to finite models. However, given an arbitrary image-finite LTS M_a, we can define the left homomorphism vector of M_a with respect to $\mathcal{T}$, since there will always be finitely many homomorphisms from a finite directed tree-shaped LTS T_c to M_a. To see this, observe that any homomorphism $h : T_c \to M_a$ must map c to a and must map fact paths of length n in T_c to fact paths of length n in M_a. Furthermore, all actions in T_c are oriented in the same direction (away from c). Hence only elements of $\mathrm{dom}(M)$ occurring in the image of some homomorphism $h : T_c \to M_a$ are those reachable from a

by a directed path of actions (cf. Definition 2.5.7) of length at most depth(T_c). Then by image-finiteness of M_a, it follows that

$$\bigcup_{h \in Hom(T_c, M_a)} Im(h)$$

is a finite set, and so $\mathrm{Hom}(T_c, M_a)$ must also be finite. Thus $\hom_{\mathbb{N}}(\mathcal{T}, M_a)$ is well-defined for all image-finite LTSs, and so we will prove our next characterization result for all image-finite LTSs, rather than only finite LTSs.

Theorem 4.3.2. If M_a, N_b are image-finite LTSs, then the following are equivalent:

1. $\hom_{\mathbb{B}}(\mathcal{T}, M_a) = \hom_{\mathbb{B}}(\mathcal{T}, N_b)$,
2. $M_a \leftrightarrows_s N_b$, and
3. $M_a \equiv_{\mathrm{ML}^+_\Diamond} N_b$.

Proof.
By the Theorem 4.2.11, it suffices to show (1) if and only if (3). For (1) to (3), suppose that $\hom_{\mathbb{B}}(\mathcal{T}, M_a) = \hom_{\mathbb{B}}(\mathcal{T}, N_b)$. Let φ be any formula of $\mathrm{ML}^+_\Diamond$. By Lemma 4.3.1, we have that $(M_{ST_x(\varphi)}, x) \in \mathcal{T}$. By the magic lemma, we have that there's a homomorphism from $(M_{ST_x(\varphi)}, x)$ to M_a if and only if $M_a \models ST_x(\varphi)$ (and similarly for N_b). It then follows from our initial assumption that $M_a \models \varphi$ if and only if $N_b \models \varphi$. Hence $M_a \equiv_{\mathrm{ML}^+_\Diamond} N_b$.

For (3) to (1), we will show that for any finite directed tree-shaped LTS T_c, there is a formula $H^\exists(T_c)$ of $\mathrm{ML}^+_\Diamond$ such that, for all image-finite LTSs M, and any element $a \in M$, we have that $M_a \models \varphi$ if and only if there exists a homomorphism from T_c to M_a. From this, it clearly follows that all image-finite LTSs satisfying the same $\mathrm{ML}^+_\Diamond$ formulas must have the same left profile with respect to finite directed tree-shaped LTSs. We will proceed by induction on the height of the tree T_c.

Base Case
For any element b of an LTS N, let

$$\mathrm{mark}^N_b := \bigwedge_{p \in \pi(b)} p.$$

If $\mathrm{height}(T_c) = 0$, then T_c is a single (distinguished) element. Then $\hom_{\mathbb{B}}(T_c, M_a) = 1$ if and only if $\pi(c) \subseteq \pi(a)$; otherwise, $\hom_{\mathbb{B}}(T_c, M_a) = 0$. It follows easily that mark^T_c expresses $H^\exists(T_c)$.

Inductive Step
Suppose $\mathrm{ML}^+_\Diamond$ can express $H^\exists(T'_{c'})$ for all finite directed tree-shaped LTSs $T'_{c'}$ of height less than k. Let T_c be an arbitrary tree of height k. Then $\mathrm{ML}^+_\Diamond$ can express $H^\exists(T_c)$ with the formula

$$\mathrm{mark}_c \wedge \bigwedge_{i \in I} \bigwedge_{d \in R_i[c]} \Diamond H^\exists(T_d).$$

To see that this is correct, observe that the first conjunct implies that $\pi(c) \subseteq \pi(a)$. Furthermore, we have that each R_i-successor of c has a homomorphism to some R_i-successor of a. Hence T_c has a homomorphism to M_a. □

4.4 Graded Modal Logic

In this section, we introduce graded modal logic. Graded modal logic is an extension of the basic modal language whose modalities allow for counting the number of successors satisfying some formula. As we will see, the primary notion used to indicate that two LTSs satisfy the same modal formulas is that of the *graded bisimulation*. However, a more convenient characterization of graded modal equivalence will be *unraveling invariance*.

Definition 4.4.1. Graded modal logic is the collection $\mathrm{ML}_{\#}$ of formulas generated by the following recursive grammar:

$$\varphi := p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond_i^{\geq k}\varphi,$$

where $p \in \mathrm{Prop}$, $i \in I$, and $k \in \mathbb{N}$.

For the common operators, the semantics of $\mathrm{ML}_{\#}$ and BML are identical, so we need only define the semantics for the modalities of $\mathrm{ML}_{\#}$.

Definition 4.4.2. Given a pointed LTS M_a and an $\mathrm{ML}_{\#}$-formula φ, we define

$$M, a \models \Diamond_i^{\geq k}\varphi \quad \text{if there exist at least } k\text{-many elements } b \in R_i[a] \text{ such that } M, b \models \varphi.$$

It's easy to see that the semantics of $\Diamond_i^{\geq 1}\varphi$ and $\Diamond_i\varphi$ are identical. Thus, for any formula φ of BML, it's clear that $\mathrm{ML}_{\#}$ contains an equivalent formula, and so we can say that $\mathrm{ML}_{\#}$ is a semantic extension of BML. We will also make use of the abbreviation $\Diamond_i^{=k}\varphi := \Diamond_i^{\geq k}\varphi \wedge \neg\Diamond_i^{\geq k+1}\varphi$, where clearly

$$M, a \models \Diamond_i^{=k}\varphi \quad \text{if there are exactly } k \text{ many elements } b \in R_i[a] \text{ such that } M, b \models \varphi.$$

As in the case of the basic modal language, we say that a formula φ of $\mathrm{ML}_{\#}$ is *satisfied* in a pointed LTS M_a if $M, a \models \varphi$. We similarly define the relation of graded modal equivalence (notation: $\equiv_{\mathrm{ML}_{\#}}$) between pointed LTSs in the natural way. Just as bisimulation between pointed LTSs preserves the satisfaction of basic modal formulas, the following notion preserves formulas of graded modal logic.

Definition 4.4.3. A *graded bisimulation* between pointed LTSs M_a and N_b is a binary relation $Z \subseteq M \times N$ with $(a, b) \in Z$ and meeting the following three conditions:

(prop) If $(m, n) \in Z$, then $\pi(m) = \pi(n)$.

(forth) If $(m, n) \in Z$, then for all $n \in \mathbb{N}$ and all distinct $s_1, \ldots s_n \in R_i[m]$, there are distinct R_i-successors $t_1, \ldots, t_n \in R_i[n]$ such that

- For every s_i, there's some t_j such that $(s_i, t_j) \in Z$, and
- For every t_j, there's some s_i such that $(s_i, t_j) \in Z$.

(back) If $(m, n) \in Z$, then for all $n \in \mathbb{N}$ and all distinct $t_1, \ldots t_n \in R_i[n]$, there are distinct R_i-successors $s_1, \ldots, s_n \in R_i[m]$ such that

- For every t_i, there's some s_j such that $(s_j, t_i) \in Z$, and
- For every s_i, there's some t_j such that $(s_i, t_j) \in Z$.

If a graded bisimulation between M_a and N_b exists, then we write $M_a \leftrightarrows_{\#} N_b$.

Theorem 4.4.4. (Graded Bisimulation Invariance, [Rij00]) For LTSs M_a and N_b, if $M_a \leftrightarrows_{\#} N_b$, then $M_a \equiv_{\mathrm{ML}_{\#}} N_b$.

While graded bisimulations preserves satisfaction of $\mathrm{ML}_{\#}$-formulas in pointed LTSs, they are generally cumbersome and difficult to use. However, there is an alternative way to show graded modal equivalence, based on the notion of unravelings.

Definition 4.4.5. Let M_a be a pointed LTS. The *unraveling* of M at a is the model $unr(M_a)$ such that

1. $\mathrm{dom}(unr(M_a))$ is all non-empty finite strings over M,
2. $R_i^{unr(M_a)} = \{(w, w^\frown u) \mid (\mathrm{last}(w), u) \in R_i^M\}$, and
3. $w \in p^{unr(M_a)} \iff last(w) \in p^M$,

where $\langle a \rangle$ is the unique distinguished element of the model, *last* is the function mapping strings to their last element, and $w^\frown u$ denotes string concatenation. It is a well known fact that $unr(M_a)$ is a directed tree-shaped LTS. We write $unr^k(M_a)$ for the submodel of M_a induced the by set of states at most depth k from the root $\langle a \rangle$.

Theorem 4.4.6 (Unraveling Invariance, [BCV09])**.** Let M_a be a pointed LTS. Then $M, a \models \varphi$ if and only if $unr(M_a), \langle a \rangle \models \varphi$.

The graph of the map $\mathrm{last} : \mathrm{dom}(unr(M_a)) \to \mathrm{dom}(M_a)$ is easily seen to be a graded bisimulation between M_a and its unraveling. Furthermore, the composition of two graded bisimulations is also a graded bisimulation. It follows that, to show that a graded bisimulation exists between two image-finite LTSs M_a and N_b, it suffices to show that their unravelings are isomorphic.

4.5 Capturing Graded Bisimulation

In this section, using a similar methodology to the proof of Theorem 4.3.2, we will establish that, under the counting semantics, if two image-finite LTSs have the same left-profile with respect to the class of finite directed tree-shaped LTSs, then their unravelings are isomorphic. We will prove this result through a series of lemmas.

Lemma 4.5.1. For all finite directed tree-shaped LTSs T_c and all $n \in \mathbb{N}$, there's a formula φ of $\mathrm{ML}_{\#}$ such that, for all image-finite LTSs M, and all $a \in M$, we have that $M_a \models \varphi$ if and only if $\hom(T_c, M_a) = n$.

Proof.
Since $\mathrm{ML}_{\#}$ extends $\mathrm{ML}_{\Diamond}^{+}$, we have by the proof of Lemma 4.3.2 that $\mathrm{ML}_{\#}$ can express $H^{\exists}(T_c)$ for all finite directed tree-shaped LTSs T_c. We want to show that, for all finite directed tree-shaped LTSs T_c and all $n \in \mathbb{N}$, there's a formula $H^{=}(n, T_a)$ of $\mathrm{ML}_{\#}$ such that, for all image-finite pointed LTSs M_a, we have that $M_a \models H^{=}(n, T_a)$ if and only if $\hom(T_c, M_a) = n$. We will show this by induction on the height of T_c.

Base Case
Recall that we defined

$$\mathrm{mark}_c^T := \bigwedge_{p \in \pi(c)} p.$$

Since there can only be at most one map from a directed tree-shaped LTS T_c of height 0 to M_a (namely, $c \mapsto a$), we can express $H^{=}(0, T_a)$ with $\neg\text{mark}_d$ and $H^{=}(1, T_a)$ with mark_d, while $H^{=}(k, T_a)$ for any $k > 1$ is expressed by $\bot$.

Inductive Step
Now suppose inductively that $\text{ML}_{\#}$ can express $H^{=}(n, T'_{c'})$ for any $n \in \mathbb{N}$ and all finite directed tree-shaped LTSs $T'_{c'}$ of height less than k. Let T_c be an arbitrary finite directed tree-shaped LTS of height k. We will construct $H^{=}(n, T_c)$, for any $n \in \mathbb{N}$ in several parts. To start, we will give a formula $H_i^{=}(l, n, T'_{c'})$ (for any tree $T'_{c'}$ of height less than k) such that

$$M_a \models H_i^{=}(l, n, T'_{c'}) \iff \text{there are exactly } l\text{-many } R_i\text{-successors of } a \text{ such that } H_i^{=}(n, T_c) \text{ holds.}$$

By the inductive hypothesis, we can express this for any $n \in \mathbb{N}$ with the $\text{ML}_{\#}$-formula

$$\Diamond_i^{=k}(H_i^{=}(n, T'_{c'})).$$

We will now define a formula $H_i^{l}(T_c)$ such that

$$M_a \models H_i^{l}(T_c) \iff \text{there are exactly } l\text{-many } R_i\text{-successors of } a \text{ such that } H^{\exists}(T_c).$$

We can express this with the $\text{ML}_{\#}$-formula

$$\Diamond_i^{=l} H^{\exists}(T_c).$$

Let T'_d be the subtree of T_c rooted at some $d \in R_i[c]$. Note that T'_d has height less than k. Then we claim that, by the inductive hypothesis, we can express, for each $n \in \mathbb{N}$, a formula $H_i^{\Sigma}(n, T'_d)$, such that

$$M_a \models H_i^{\Sigma}(n, T'_d) \iff \sum_{x \in R_i[a]} \hom_{\mathbb{N}}(T'_d, M_x) = n.$$

To see this, we define $H_i^{\Sigma}(n, T'_d)$ to be the $\text{ML}_{\#}$-formula

$$\bigvee_{m \leq n} H_i^{m}(T'_d) \wedge \left(\bigvee_{f:[m]\to[n]:\sum_{j\leq m} f(j)=n} \left(\bigwedge_{k \leq n} H_i^{=}(|f^{-1}(k)|, k, T'_d) \right) \right).$$

To see that this formula works, suppose that $M_a \models H_i^{\Sigma}(n, T'_d)$. Fix any $m \leq n$ and consider the corresponding disjunct. We first have that $M_a \models H_i^{m}(T'_d)$; thus there are exactly m R_i-successors x of a such that there's a homomorphism from d to M_x. The next disjunction ranges over all functions $f : [m] \to [n]$ such that $\sum_{j\leq m} f(j) = n$. We interpret $f(j) = l$ to mean that there are exactly l-many homomorphisms to the j^{th} R_i-successor of a to which T'_d has some homomorphism. Since there can be at most n successors y of a such that $hom(T'_d, M_y) \geq 1$ while $\sum_{x\in R_i[a]} = n$, it follows that M_a satisfies a disjunct of this formula if and only if $\sum_{x\in R_i[a]} = n$, as required. Finally, we can express that there are exactly N homomorphisms from T_a to M_x with

$$\text{mark}_c^T \wedge \left(\bigvee_{f:R[a]\to[N]:\prod_{c\in R[a]} f(c)=N} \left(\bigwedge_{i\in I} \bigwedge_{c\in R_i[a]} H_i^{\Sigma}(f(c), T_c) \right) \right).$$

The correctness of this formula is justified by Proposition 2.5.10. □

Note that if M_a is an image-finite LTS, then $unr(M_a)$ is also image-finite, and thus has a well-defined left profile with respect to the class of finite directed tree-shaped LTSs. We use this fact in the following lemma.

Lemma 4.5.2. For any image-finite pointed LTS M_a, we have that

$$\hom_{\mathbb{N}}(\mathcal{T}, M_a) = \hom_{\mathbb{N}}(\mathcal{T}, unr(M_a)).$$

Proof.
Let T_c be an arbitrary finite directed tree-shaped LTS. We will show that $|\mathrm{Hom}(\mathcal{T}, M_a)| \leq |\mathrm{Hom}(\mathcal{T}, unr(M_a))|$ and $|\mathrm{Hom}(\mathcal{T}, M_a)| \geq |\mathrm{Hom}(\mathcal{T}, unr(M_a))|$, thereby establishing the claim.

($\leq$) For each $h \in Hom(T_c, M_a)$, define a new map $g : T_c \to unr(M_a)$ by the following recursion on the depth of the elements of T_c:

$$g(c) = \langle a \rangle$$
$$g(m) = g_i(\mathrm{parent}(m))^\frown h(m)$$

We claim that g is a homomorphism. To see that g preserves proposition letters, observe that, for any proposition letter $p \in P$, we have that

$$\begin{aligned} p^T(m) &\implies p^M(h(m)) && \text{(since } h \text{ is a homomorphism)} \\ &\implies p^{unr(M_a)}(g(m)) && \text{(since } h(m) = \mathrm{last}(g(m))\text{)} \end{aligned}$$

To show that g preserves actions, it suffices to show that, for any $i \in I$ and any node $m \in T_c$ with R_i-children $s_1, \ldots, s_r$, we have that $R_i^{unr(M_a)}(g(m), s_j)$ for each $j \leq r$. For this to hold, we need only to make the following two observations:

1. $g(s_j)$ extends $g(m)$ (by the definition of g);
2. $R_i^M(\mathrm{last}(g(x)), \mathrm{last}(g(s_j)))$ if and only if $R_i^M(h(x), h(s_j))$ (since h_i is a homomorphism).

Thus g is a homomorphism. Furthermore, it's clear that this construction constitutes an injective map from $\mathrm{Hom}(T_c, M_a)$ to $\mathrm{Hom}(T_c, unr(M_a))$, and so we're done.

($\geq$) For each $g \in \mathrm{Hom}(T_c, unr(M_a))$, define a map $h : T_c \to M_a$ by $m \mapsto \mathrm{last}(g(m))$. That h is a homomorphism is immediate from the definition of unravelings. We need to also show that this construction is injective. To see this, let $h, h' : T_c \to unr(M_a)$ be homomorphisms, and let g, g' be the corresponding maps in $\mathrm{Hom}(T_c, M_a)$. Suppose that $h = h'$.

We now show by induction on depth of the elements of T_c that $g = g'$. The base case is immediate, since $g(c) = g'(c) = \langle a \rangle$. Now suppose inductively that g and g' agree on all elements of depth less than k, and let $m \in T_c$ be some element of depth k. By assumption, we have that $h(m) = h'(m)$, and so $last(g(m)) = last(g'(m))$. Furthermore, by the inductive hypothesis, since $\mathrm{parent}(m)$ has depth less than k, we have that $g(\mathrm{parent}(m)) = g'(\mathrm{parent}(m))$. Since g and g' are homomorphisms and $R_i^T(\mathrm{parent}(m), m)$, we have that $R_i^{unr(M_a)}(g(\mathrm{parent}(m)), g(m))$ and $R_i^{unr(M_a)}(g'(\mathrm{parent}(m)), g'(m))$. Then by definition of the actions for unravelings, we have $g(m) = g'(m)$. This completes the induction, and so $g = g'$. Hence our construction is an injective map from $Hom(T_c, M_a)$ to $Hom(T_c, unr(M_a))$. □

Lemma 4.5.3. Let M_a and N_b be finite directed tree-shaped LTSs of depth k. If $\hom(\mathcal{T}^k, M_a) = \hom(\mathcal{T}^k, N_b)$, then $M_a \cong N_b$.

Proof.
Let T_c be any finite directed tree-shaped LTS of depth strictly greater than k. Since M_a and N_b have no directed paths of length greater than k, clearly $\hom(T_c, M_a) = \hom(T_c, N_b) = 0$. Hence $\hom(\mathcal{T}, M_a) = \hom(\mathcal{T}, N_b)$. Then by Proposition 4.2.13 and Theorem 2.4.9, we have that $M_a \cong N_b$. □

Lemma 4.5.4. If M_a, N_b are image-finite directed tree-shaped LTSs and $M_a^k \cong N_b^k$ for all $k \in \mathbb{N}$, then $M_a \cong N_b$.

Proof.
If M_a or N_b is finite, then the result is trivial, so suppose they are both infinite. Consider the sequence of sets $(S_i)_{i\in\omega}$ such that S_i is the collection of isomorphisms $M_a^i \to N_b^i$, and let $S = \bigcup_{i\in\omega} S_i$. By assumption, each S_i is nonempty. Since M_a and N_b are image-finite, we have also that each M_a^k and N_b^k is finite, and hence each S_i is finite. Furthermore, it's easy to see that, for all $i \in \mathbb{N}$, we have that each $f \in S_i$ must extend some $g \in S_j$ for any $j < i$. We will now recursively construct sequences $(g_i)_{i\leq n}$ for each $n \in \mathbb{N}$ such that, for each $i \in \mathbb{N}$, the following properties hold:

(1) $g_i : M_a^i \to N_b^i$ is an isomorphism,

(2) $g_i \subseteq g_{i+1}$, and

(3) there exist infinitely many isomorphisms in S extending g_n.

Let g_0 be the map $a \mapsto b$. This is clearly an isomorphism from M_a^0 to N_b^0, and every map in S extends g_0; this concludes the base case of the construction.

Now suppose inductively that $(g_i)_{i\leq n}$ for some $n \in \mathbb{N}$ is a sequence meeting conditions (1)-(3). Let $T \subseteq S$ denote the set of infinitely many isomorphisms in S extending g_n. Since S_{n+1} is finite, we have that $T \setminus S_{n+1}$ is also infinite. Furthermore, for each function f in $T \setminus S_{n+1}$, there's some function g in S_{n+1} such that f extends g. Since S_{n+1} is finite, there must be at least one g_{n+1} in S_{n+1} such that infinitely many of the functions in $T \setminus S_{n+1}$ extend g_{n+1}. This completes the construction.

By our construction, we have a sequence of isomorphisms $(g_i)_{i\in\omega}$ meeting properties (1)-(3). Let $g = \bigcup_{i\in\omega} g_i$. It follows immediately that $g : M_a \to N_b$ is an isomorphism, and so $M_a \cong N_b$. □

We are now ready to prove the main result.

Theorem 4.5.5. Let M_a and N_b be image-finite LTSs. Then the following are equivalent:

1. $\hom_{\mathbb{N}}(\mathcal{T}, M_a) = \hom_{\mathbb{N}}(\mathcal{T}, N_b)$,
2. $unr(M_a) \cong unr(N_b)$,
3. $M_a \leftrightarrows_{\#} N_b$,
4. $M_a \equiv_{\mathrm{ML}_{\#}} N_b$.

Proof.
For the direction (1) to (2), we first apply Lemma 4.5.2 to obtain that $\hom_{\mathbb{N}}(\mathcal{T}, M_a) = \hom_{\mathbb{N}}(\mathcal{T}, unr(M_a)) = \hom_{\mathbb{N}}(\mathcal{T}, unr(N_b)) = \hom_{\mathbb{N}}(\mathcal{T}, N_b)$. Then clearly, for each $k \in \mathbb{N}$, we have that $\hom_{\mathbb{N}}(\mathcal{T}, unr^k(M_a)) = \hom_{\mathbb{N}}(\mathcal{T}, unr^k(N_b))$, and so by Lemma 4.5.3, we have that $unr^k(M_a) \cong unr^k(N_b)$. Since M_a and N_b are image-finite LTSs, their unravelings are image-finite directed tree-shaped LTSs. Thus we have by Lemma 4.5.4 that $unr(M_a) \cong unr(N_b)$.

The direction (2) to (3) follows from the fact that every LTS has a graded bisimulation to its unraveling, as well as the fact that the composition of two graded bisimulations is also a graded bisimulation. The direction (3) to (4) is immediate from Theorem 4.4.4. For (4) to (1), suppose that M_a and N_b satisfy the same graded modal formulas. It then follows immediately from Lemma 4.5.1 that $\hom_{\mathbb{N}}(T_c, M_a) = \hom_{\mathbb{N}}(T_c, N_b)$ for any finite directed tree-shaped LTS T_c. Therefore, we have that $\hom_{\mathbb{N}}(\mathcal{T}, M_a) = \hom_{\mathbb{N}}(\mathcal{T}, N_b)$. □

Observe that we have proven the following result, analogous to Theorem 4.2.7.

Corollary 4.5.6. (Graded Hennessy-Milner Theorem) For image-finite LTSs M_a and N_b, if $M_a \equiv_{\mathrm{ML}_\#} N_b$, then $M_a \leftrightarroweq_\# N_b$.

4.6 Backward and Global Modalities

In this section, we will lift our results from earlier sections to languages containing *backward* and *global* modalities. The corresponding extended languages will be captured by restricting the left profile to more general classes of structures than $\mathcal{T}$. In this section, it will be important to track signatures, and so we will write that a model M_a is a σ-LTS if all facts of M are interpretations of actions and proposition letters in the signature σ. Furthermore, we will parametrize classes of LTSs by the relevant signature. For example, we will write $\mathcal{T}^\sigma$ for the class of directed tree-shaped σ-LTSs.

We will now lift our previous results to languages containing backward modalities. Given a σ-LTS M and an element $a \in M$, we define $R_i^{-1}[a] := \{b \in M \mid R_i^M(b, a)\}$, and we refer to the elements of this set as R_i*-predecessors* of a in M.

Definition 4.6.1. Given a pointed σ-LTS M_a and an $\mathrm{ML}_\#$-formula φ, we define

$$M, a \models \blacklozenge_i^{\geq k}\varphi \quad \text{if there exist at least } k\text{-many elements } b \in R_i^{-1}[a] \text{ such that } M, b \models \varphi.$$

We refer to $\blacklozenge_i^{\geq k}$ as a *backward modality* for the action R_i, where $i \in I$. Let $\mathrm{ML}_\Diamond^{+,B}$ denote the extension of $\mathrm{ML}_\Diamond^+$ with the backward modalities for $k = 1$, and let $\mathrm{ML}_\#^B$ denote the extension of $\mathrm{ML}_\#$ with the backward modalities for each $k \in \mathbb{N}$. We define positive-existential modal equivalence with backward modalities (notation: $\equiv^\sigma_{\mathrm{ML}_\Diamond^{+,B}}$) and graded modal equivalence with backward modalities (notation: $\equiv^\sigma_{\mathrm{ML}_\#^B}$) between pointed σ-LTSs in the natural way. The relations preserving these languages are *back-and-forth simulation equivalence* and *back-and-forth graded bisimulation*, respectively [DNV95].

Definition 4.6.2. Let σ be a fixed signature whose actions are R_i for each $i \in I$, where I is a finite index set. Define the expanded signature $\sigma_B \supseteq \sigma$ to have actions R_i and B_i for each $i \in I$, as well as the same collection of proposition letters occurring in σ. For each $i \in I$, we write $\Diamond_{B,i}^{\geq k}$ for the graded modality associated with the action B_i, and we retain

the notation $\Diamond_i^{\geq k}$ for the graded modality associated with R_i. Given a pointed σ-LTS M_a, we define the *backward expansion* of M_a to be the σ_B-LTS M_a^B which is identical to M_a, except that we provide an interpretation for B_i for each $i \in I$ as follows: $B_i^{M^B}(m,n)$ holds for $m, n \in \mathrm{dom}(M)$ if and only if $R_i^M(n,m)$ holds.

Figure 4.2 depicts the backward expansion transformation.

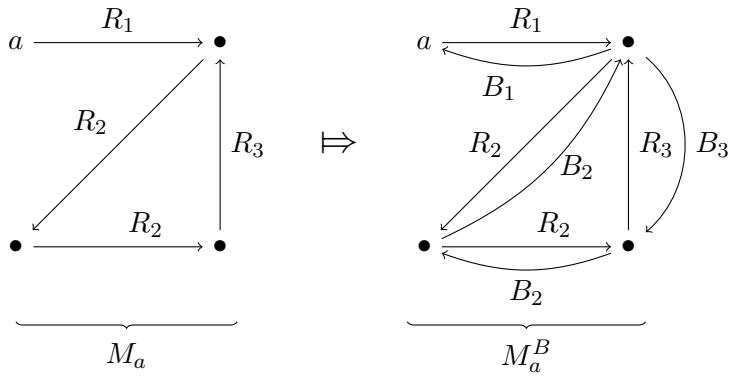


Figure 4.2: Transformation from σ-LTS M_a to its backward expansion.

Definition 4.6.3. Let $\mathcal{A}^\sigma$ denote the class of finite tree-shaped (cf. Definition 2.5.6) σ-LTSs with one distinguished element.

Definition 4.6.4. Let $T_c \in \mathcal{A}^\sigma$. Then the *directed tree around* T_c is the σ_B-LTS $T_c^\downarrow$ with the same domain as T and the same proposition letters true at each state, and whose interpretation of the actions in σ_B are defined by the following rules:

1. If $R_i^T(m,n)$ holds where $\mathit{depth}(m) < \mathit{depth}(n)$, then $R_i^{T_c^\downarrow}(m,n)$ holds.
2. If $R_i^T(m,n)$ holds where $\mathit{depth}(n) < \mathit{depth}(m)$, then $B_i^{T_c^\downarrow}(m,n)$ holds.

Note that, given some $T_c \in \mathcal{T}^{\sigma_B}$, we can define the *tree-shaped model around* T_c to be the reverse construction, denoted by $T_c^\uparrow$. These transformations are exact inverses of one another: $T_c = (T_c^\downarrow)^\uparrow$ for any $T_c \in \mathcal{A}^\sigma$. Figure 4.3 depicts this transformation.

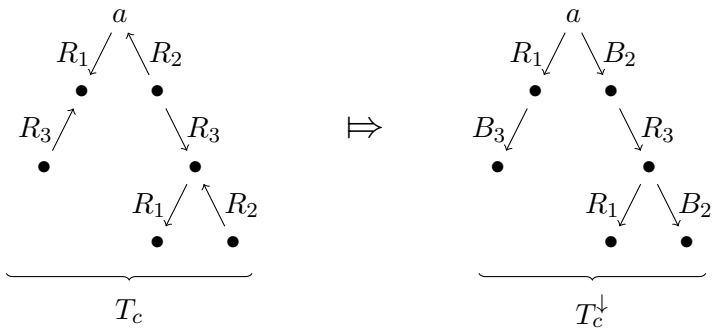


Figure 4.3: A tree-shaped σ-LTS T_c and the directed tree around T_c.

Observe that $T_c^\downarrow$ is a directed tree with respect to $\{R_i \mid i \in I\} \cup \{B_i \mid i \in I\}$ for any $T_c \in \mathcal{A}^\sigma$. This implies that if $T_c \in \mathcal{A}^\sigma$, then $T_c^\downarrow \in \mathcal{T}^{\sigma_B}$.

In Sections 4.3 and 4.5, we stated our results for arbitrary image-finite LTSs M_a. This assumption was sufficient in those cases because we were only considering homomorphisms

from finite directed tree-shaped LTSs to M_a. However, we will now consider (possibly non-directed) finite tree-shaped LTSs T_c, and so image-finiteness is not enough to guarantee that $\mathrm{Hom}(T_c, M_a)$ is finite. To remedy this, we will assume a slightly stronger condition for M_a. We say that M_a is *degree-finite* if $R_i[m]$ and $R_i^{-1}[m]$ are both finite for all $m \in \mathrm{dom}(M)$. By assuming that M_a is degree-finite, it follows easily that

$$\bigcup_{h \in Hom(T_c, M_a)} Im(h)$$

is finite. Thus only finitely many homomorphisms from a tree-shaped LTS T_c to M_a can exist, and so $\hom_{\mathbb{N}}(\mathcal{A}^\sigma, M_a)$ is well-defined. Furthermore, note that if M_a is a degree-finite σ-LTS, then M_a^B is also a degree-finite σ^B-LTS.

Proposition 4.6.5. Let M_a be a degree-finite σ-LTS, and T_c be an arbitrary finite tree-shaped σ-LTS. Then $\hom_{\mathbb{N}}(T_c, M_a) = \hom_{\mathbb{N}}(T_c^{\downarrow}, M_a^B)$.

Proof.
Let $h : T_c \to M_a$ be any homomorphism. Clearly h is also a map from $T_c^{\downarrow}$ to M_a^B, and we claim that it is a homomorphism. Since T_c and $T_c^{\downarrow}$ satisfy the same proposition letters at the same states, clearly h preserves proposition letters from $T_c^{\downarrow}$ to M_a^B. For actions, we distinguish cases.

1. Suppose that $R_i^{T_c^{\downarrow}}(m, n)$ holds for some $m, n \in T_c^{\downarrow}$. Then, by construction of $T_c^{\downarrow}$, we have that $R_i^{T_c}(m, n)$ must also hold. Then because h preserves actions from T_c to M_a, we have that $R_i^{M_a}(h(m), h(n))$ also holds. Then by construction of M_a^B, we have that $R_i^{M_a^B}(h(m), h(n))$ holds.
2. Suppose that $B_i^{T_c^{\downarrow}}(m, n)$ holds for some $m, n \in T_c^{\downarrow}$. Then, by construction of $T_c^{\downarrow}$, we have that $R_i^{T_c}(n, m)$ must also hold. Then because h preserves actions from T_c to M_a, we have that $R_i^{M_a}(h(n), h(m))$ also holds. Then by construction of M_a^B, we have that $B_i^{M_a^B}(h(m), h(n))$ holds.

Thus h preserves all facts from $T_c^{\downarrow}$ to M_a^B, and so h is a homomorphism between these structures. Using a similar argument, we also obtain that any homomorphism $g : T_c^{\downarrow} \to M_a^B$ is also a homomorphism from T_c to M_a. It then follows immediately that $\hom_{\mathbb{N}}(T_c, M_a) = \hom_{\mathbb{N}}(T_c^{\downarrow}, M_a^B)$. □

The following proposition is immediate from Proposition 4.6.5 and the observation that, for each $T_c' \in \mathcal{T}^{\sigma_B}$, there's some $T_c \in \mathcal{A}^\sigma$ such that $T_c' = T_c^{\downarrow}$ (and hence $T_c'^{\uparrow} = T_c$).

Proposition 4.6.6. Let M_a be a degree-finite σ_B-LTS, and T_c be an arbitrary directed tree-shaped σ_B-LTS. Then $\hom_{\mathbb{N}}(T_c, M_a^B) = \hom_{\mathbb{N}}(T_c^{\uparrow}, M_a)$.

Lemma 4.6.7. Let M_a and N_b be degree-finite σ-LTSs. Then $\hom_{\mathbb{N}}(\mathcal{A}^\sigma, M_a) = \hom_{\mathbb{N}}(\mathcal{A}^\sigma, N_b)$ if and only if $\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, M_a^B) = \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, N_b^B)$.

Proof.
For the forward direction, we proceed by contraposition. Suppose that

$$\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, M_a^B) \neq \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, N_b^B),$$

so that there is some $T_c \in \mathcal{T}^{\sigma_B}$ such that

$$\hom_{\mathbb{N}}(T_c, M_a^B) \neq \hom_{\mathbb{N}}(T_c, N_b^B).$$

Then by Proposition 4.6.6, we have that

$$\hom_{\mathbb{N}}(T_c^{\uparrow}, M_a) \neq \hom_{\mathbb{N}}(T_c^{\uparrow}, N_b).$$

Since $T_c^{\uparrow} \in \mathcal{A}^{\sigma}$, this implies that

$$\hom_{\mathbb{N}}(\mathcal{A}^{\sigma}, M_a) \neq \hom_{\mathbb{N}}(\mathcal{A}^{\sigma}, N_b).$$

For the reverse direction, we again proceed by contraposition. Suppose that

$$\hom_{\mathbb{N}}(\mathcal{A}^{\sigma}, M_a) \neq \hom_{\mathbb{N}}(\mathcal{A}^{\sigma}, N_b),$$

so that there's some $T_c \in \mathcal{A}^{\sigma}$ such that

$$\hom_{\mathbb{N}}(T_c, M_a) \neq \hom_{\mathbb{N}}(T_c, N_b).$$

Then by Proposition 4.6.5, we have that

$$\hom_{\mathbb{N}}(T_c^{\downarrow}, M_a^B) \neq \hom_{\mathbb{N}}(T_c^{\downarrow}, N_b^B).$$

Since $T_c^{\downarrow} \in \mathcal{T}^{\sigma_B}$, this implies that

$$\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, M_a^B) \neq \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_B}, N_b^B),$$

which is what we wanted to show. □

Lemma 4.6.8. Let M_a and N_b be degree-finite σ-LTSs. Then $M_a \equiv^{\sigma}_{\mathrm{ML}^B_{\#}} N_b$ if and only if $M_a^B \equiv^{\sigma_B}_{\mathrm{ML}_{\#}} N_b^B$.

Proof.
Given an $\mathrm{ML}^B_{\#}$-formula φ over σ, let $tr_B(\varphi)$ denote the $\mathrm{ML}_{\#}$-formula over σ_B obtained by replacing all occurrences of the backward modality $\blacklozenge_i^{\geq k}$ with the graded modality $\Diamond_{B,i}^{\geq k}$. Using the construction of M_a^B, it is then a straightforward induction to show that

$$M_a \models \varphi \iff M_a^B \models tr_B(\varphi),$$

and similarly, that

$$N_b \models \varphi \iff N_b^B \models tr_B(\varphi).$$

Furthermore, tr_B is clearly a bijective translation, and so it follows immediately that

$$M_a \equiv^{\sigma}_{\mathrm{ML}^B_{\#}} N_b \iff M_a^B \equiv^{\sigma_B}_{\mathrm{ML}_{\#}} N_b^B.$$

This completes the proof. □

The following theorem shows that $\mathrm{ML}^B_{\#}$-equivalence is captured by restricting the counting left profile to the class of tree-shaped LSTs.

Theorem 4.6.9. Let M_a and N_b be degree-finite σ-LTSs. The following are equivalent.

1. $\hom(\mathcal{A}^{\sigma}, M_a) = \hom(\mathcal{A}^{\sigma}, N_b)$, and
2. $M_a \equiv^{\sigma}_{\mathrm{ML}^B_{\#}} N_b$.

Proof.
Let M_a and N_b be degree-finite σ-LTSs. Then we have that

$$\begin{aligned}
\hom(\mathcal{A}^\sigma, M_a) = \hom(\mathcal{A}^\sigma, N_b) &\iff \hom(\mathcal{T}^{\sigma_B}, M_a^B) = \hom(\mathcal{A}^{\sigma_B}, N_b^B) && \text{(Lemma 4.6.7)}\\
&\iff M_a^B \equiv^{\sigma_B}_{\mathrm{ML}_\#} N_b^B && \text{(Theorem 4.5.5)}\\
&\iff M_a \equiv^{\sigma}_{\mathrm{ML}_\#^B} N_b. && \text{(Lemma 4.6.8)}
\end{aligned}$$

This completes the proof. □

We now turn our attention to languages with global modalities.

Definition 4.6.10. Given a pointed σ-LTS M_a and an $\mathrm{ML}_\#$-formula φ, we define

$$M, a \models \mathrm{E}^{\geq k}\varphi \quad \text{if there exist at least } k\text{-many elements } b \in M \text{ such that } M, b \models \varphi.$$

We refer to $\mathrm{E}^{\geq k}$ as a *global modality*. Let $\mathrm{ML}_\Diamond^{+,G}$ denote the extension of $\mathrm{ML}_\Diamond^+$ with the global modality for $k = 1$, and let $\mathrm{ML}_\#^G$ denote the extension of $\mathrm{ML}_\#$ with the global modalities for each $k \in \mathbb{N}$. We write $\Diamond_{G,i}^{\geq k}$ for the graded modalities associated with the action R_G. We define positive-existential modal equivalence with global modalities (notation: $\equiv^{\sigma}_{\mathrm{ML}_\Diamond^{+,G}}$) and graded modal equivalence with global modalities (notation: $\equiv^{\sigma}_{\mathrm{ML}_\#^G}$) between pointed LTSs in the natural way. The relations preserving these languages are *global simulation equivalence* and *global graded bisimulation*, respectively [OP08].

Definition 4.6.11. Let σ be a fixed signature whose actions are R_i for $i \in I$, where I is a finite index set. Define the expanded signature $\sigma_G \supseteq \sigma$ to have actions R_i for $i \in I$, a fresh action R_G, and the same collection of proposition letters occurring in σ. Given a pointed σ-LTS M_a, we define the *global expansion* of M_a to be the σ_G-LTS M_a^G which is identical to M_a, except that $R_G^{M^G}(m, n)$ holds for all $m, n \in \mathrm{dom}(M)$.

Definition 4.6.12. Let $\mathcal{F}^\sigma$ denote the class of σ-LTSs which are finite directed forests (cf. Definition 2.5.9) with one distinguished element. We refer to such models as *forest σ-LTSs*.

Definition 4.6.13. Let $T_c \in \mathcal{F}^\sigma$. We say that a σ_G-expansion T_c' of T_c is an *R_G-connection of T_c* if T_c' is a directed tree-shaped σ_G-LTS and $T_c = T_c' \restriction \sigma$.

It's easy to see that every $T_c' \in \mathcal{T}^{\sigma_G}$ is an R_G-connection of some $T_c \in \mathcal{F}^\sigma$. In other words, we have that

$$\mathcal{T}^{\sigma_G} = \{T_c' \mid T_c' \text{ is an } R_G\text{-connection of some } T_c \in \mathcal{F}^\sigma\}. \qquad (\dagger)$$

Furthermore, any $T_c \in \mathcal{F}^\sigma$ can be formed by simply removing the interpretation of R_G from some $T_c' \in \mathcal{T}^{\sigma_G}$. Thus we have that

$$\mathcal{F}^\sigma = \{T_c \mid T_c = T_c' \restriction \sigma \text{ for some } T_c' \in \mathcal{T}^{\sigma_G}\}. \qquad (\ddagger)$$

In the case of the global modality, we will state our results only for finite σ-LTSs, to ensure that the counting left homomomorphism vector with respect to $\mathcal{F}^\sigma$ is always well-defined. To see that this is necessary, observe that, since there exist models T_c in $\mathcal{F}^\sigma$ with connected components which do not contain c, these connected components could be mapped anywhere in an infinite degree-finite LTS M_a. Hence there might, in general, be infinitely many homomorphisms from T_c to M_a, even if M_a is degree-finite. Note that, for any finite σ-LTS M_a, we have that M_a^G is finite (and hence image-finite) as well. However, if M_a were infinite, then M_a^G would not be image-finite.

Proposition 4.6.14. Let M_a be a finite σ-LTS. Then for any $T_c \in \mathcal{F}^\sigma$ and any T'_c which is an R_G-connection of T_c, we have that $\hom_{\mathbb{N}}(T_c, M_a) = \hom_{\mathbb{N}}(T'_c, M_a^G)$.

Proof.
To see that this is the case, let $h : T_c \to M_a$ be a homomorphism. Clearly h is also a map from T'_c to M_a^G. By the construction of M_a^G, it's easy to see that h preserves all facts from T'_c to M_a^G from the signature σ. Furthermore, since R^G is the total relation on M_a^G, clearly h also preserves the facts for the R_G action from T'_c to M_a^G as well. Furthermore, since $T_c = T'_c \restriction \sigma$ and $M_a = M_a^G \restriction \sigma$, clearly any homomorphism $g : T'_c \to M_a^G$ is also a homomorphism from T_c to M_a. It follows that $\hom_{\mathbb{N}}(T_c, M_a) = \hom_{\mathbb{N}}(T'_c, M_a^G)$. □

Lemma 4.6.15. Let M_a and N_b be finite σ-LTSs. The following are equivalent.

1. $\hom_{\mathbb{N}}(\mathcal{F}^\sigma, M_a) = \hom_{\mathbb{N}}(\mathcal{F}^\sigma, N_b)$, and
2. $\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, M_a^G) = \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, N_b^G)$.

Proof.
For the forward direction, we proceed by contraposition. Suppose that

$$\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, M_a^G) \neq \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, N_b^G),$$

so that there is some $T'_c \in \mathcal{T}^{\sigma_G}$ such that

$$\hom_{\mathbb{N}}(T'_c, M_a^G) \neq \hom_{\mathbb{N}}(T'_c, N_b^G).$$

Then by (†), there's some $T_c \in \mathcal{F}^\sigma$ such that $T_c = T'_c \restriction \sigma$, and by Proposition 4.6.14, we have that

$$\hom_{\mathbb{N}}(T_c, M_a) \neq \hom_{\mathbb{N}}(T_c, N_b).$$

Since $T_c \in \mathcal{F}^\sigma$, this implies that

$$\hom_{\mathbb{N}}(\mathcal{F}^\sigma, M_a) \neq \hom_{\mathbb{N}}(\mathcal{F}^\sigma, N_b).$$

For the reverse direction, we again proceed by contraposition. Suppose that

$$\hom_{\mathbb{N}}(\mathcal{F}^\sigma, M_a) \neq \hom_{\mathbb{N}}(\mathcal{F}^\sigma, N_b),$$

so that there's some $T_c \in \mathcal{F}^\sigma$ such that

$$\hom_{\mathbb{N}}(T_c, M_a) \neq \hom_{\mathbb{N}}(T_c, N_b).$$

Then by (‡), there's some $T'_c \in T^{\sigma_G}$ such that $T_c = T'_c \restriction \sigma$, and by Proposition 4.6.14, we have that

$$\hom_{\mathbb{N}}(T_c, M_a^G) \neq \hom_{\mathbb{N}}(T_c, N_b^G).$$

Since $T_c \in \mathcal{T}^{\sigma_G}$, this implies that

$$\hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, M_a^G) \neq \hom_{\mathbb{N}}(\mathcal{T}^{\sigma_G}, N_b^G),$$

which is what we wanted to show. □

Lemma 4.6.16. Let M_a and N_b be finite σ-LTSs. Then $M_a \equiv^\sigma_{\mathrm{ML}^G_\#} N_b$ if and only if $M_a^G \equiv^{\sigma_G}_{\mathrm{ML}_\#} N_b^G$.

Proof.
Given an $\mathrm{ML}^G_\#$-formula φ over σ, let $tr_G(\varphi)$ denote the $\mathrm{ML}_\#$-formula over σ_G obtained by replacing all occurrences of the global modality $\mathrm{E}^{\geq k}{}_i$ with the modality $\Diamond^{\geq k}_{G,i}$. By the construction of M^G_a, it is a straightforward induction to show that

$$M_a \models \varphi \iff M^G_a \models tr_G(\varphi),$$

and similarly, that

$$N_b \models \varphi \iff N^G_b \models tr_G(\varphi).$$

Furthermore, tr_G is clearly a bijective translation, and so it follows immediately that

$$M_a \equiv^\sigma_{\mathrm{ML}^G_\#} N_b \iff M^G_a \equiv^{\sigma_G}_{\mathrm{ML}_\#} N^G_b.$$

This completes the proof. □

Theorem 4.6.17. Let M_a and N_b be finite σ-LTSs. The following are equivalent.

1. $\hom_{\mathbb{N}}(\mathcal{F}^\sigma, M_a) = \hom_{\mathbb{N}}(\mathcal{F}^\sigma, N_b)$, and
2. $M_a \equiv^\sigma_{\mathrm{ML}^G_\#} N_b$.

Proof.
Let M_a and N_b be finite σ-LTSs. Then

$$\begin{aligned}
\hom(\mathcal{F}^\sigma, M_a) = \hom(\mathcal{F}^\sigma, N_b) &\iff \hom(\mathcal{T}^{\sigma_G}, M^G_a) = \hom(\mathcal{T}^{\sigma_G}, N^G_b) && \text{(Lemma 4.6.15)}\\
&\iff M^G_a \equiv^{\sigma_G}_{\mathrm{ML}_\#} N^G_b && \text{(Theorem 4.5.5)}\\
&\iff M_a \equiv^\sigma_{\mathrm{ML}^B_\#} N_b. && \text{(Lemma 4.6.16)}
\end{aligned}$$

This completes the proof. □

We have chosen to work out the proofs for the more complicated case of extending $\mathrm{ML}_\#$ with backwards and global modalities. However, using highly parallel arguments, we also obtain the following results.

Theorem 4.6.18. Let M_a and N_b be degree-finite σ-LTSs. The following are equivalent.

1. $\hom_{\mathbb{B}}(\mathcal{A}^\sigma, M_a) = \hom_{\mathbb{B}}(\mathcal{A}^\sigma, N_b)$, and
2. $M_a \equiv^\sigma_{\mathrm{ML}^{+,B}_\Diamond} N_b$.

Theorem 4.6.19. Let M_a and N_b be finite σ-LTSs. The following are equivalent.

1. $\hom_{\mathbb{B}}(\mathcal{F}^\sigma, M_a) = \hom_{\mathbb{B}}(\mathcal{F}^\sigma, N_b)$, and
2. $M_a \equiv^\sigma_{\mathrm{ML}^{+,G}_\Diamond} N_b$.

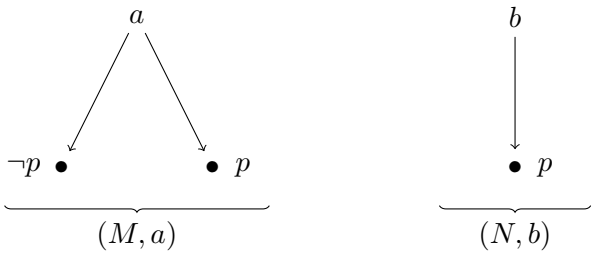
Note that, even in the Boolean case, we still require the assumption that M_a is degree-finite (for Theorem 4.6.18) and finite (for Theorem 4.6.19). These assumptions are required to guarantee that M^B_a and M^G_a are image-finite, allowing us to apply Theorem 4.3.2 at the appropriate point in the proofs.

4.7 Negative Results

We have seen that simulation and graded bisimulation are captured by restricting the Boolean and counting left profiles, respectively, to the class of directed tree-shaped LTSs. Could a similar restriction of the Boolean or counting left profiles capture the notions of directed simulation or bisimulation? As it turns out, this is not possible. Let $\sim$ and $\approx$ be equivalence relations on pointed LTSs. We say that $\sim$ is a *refinement* of $\approx$ if $M_a \sim N_b$ implies $M_a \approx N_b$, in which case we say that $\sim$ is *finer* than $\approx$, and $\approx$ is *coarser* than $\sim$. We show that no relation finer than directed simulation can be captured by restricting the Boolean left profile, and that no relation finer than directed simulation and coarser than bisimulation can be captured by restricting the counting left profile.

Proposition 4.7.1. Let $\sim$ be an equivalence relation on LTSs finer than directed simulation. Then there's no class $\mathcal{C}$ such that $\hom_{\mathbb{B}}(\mathcal{C}, M_a) = \hom_{\mathbb{B}}(\mathcal{C}, N_b)$ if and only if $M_a \sim N_b$.

Proof.
Suppose for a contradiction that some such $\mathcal{C}$ exists. Consider the following LTSs:



Since N_b is the core of M_a, we have that $M_a \rightleftarrows N_b$, and so $\hom_{\mathbb{B}}(\mathcal{C}, M_a) = \hom_{\mathbb{B}}(\mathcal{C}, N_b)$. However, there's no successor of b whose proposition letters are a subset of those satisfied at the left successor of a. It follows that $M_a \not\leftrightarroweq_d N_b$. Then since $\sim$ is finer than $\leftrightarroweq_d$, we have that $M_a \not\sim N_b$, which is a contradiction. Therefore, no such class $\mathcal{C}$ exists. □

Note that the example in Proposition 4.7.1 uses directed tree-shaped LTSs. This implies that equivalence relations finer than directed simulation cannot be characterized even if we confine attention only to the class of directed tree-shaped LTSs. Furthermore, *two-way directed simulation equivalence*, *two-way bisimulation*, *global directed simulation equivalence*, and *global bisimulation* are all equivalence relations finer than directed simulation equivalence, and so cannot be captured by a restriction of the Boolean left profile.

Proposition 4.7.2. Let $\sim$ be any equivalence relation on LTSs finer than directed simulation and coarser than bisimulation. Then there's no class $\mathcal{C}$ such that $\hom_{\mathbb{N}}(\mathcal{C}, M_a) = \hom_{\mathbb{N}}(\mathcal{C}, N_b)$ if and only if $M_a \sim N_b$.

Proof.
Suppose for a contradiction that some such class $\mathcal{C}$ exists. Consider two cases:

1. For each $A_c \in \mathcal{C}$, we have $|A_c| = 1$. Consider again the example from proposition 4.7.1. For each $A_c \in \mathcal{C}$, we have that $\hom_{\mathbb{N}}(A_c, M_a) = \hom_{\mathbb{N}}(A_c, N_b) = 1$ if and only if $\pi(c) = \emptyset$, and $\hom_{\mathbb{N}}(A_c, M_a) = \hom_{\mathbb{N}}(A_c, N_b) = 0$ otherwise. Hence $\hom_{\mathbb{N}}(\mathcal{C}, M_a) = \hom_{\mathbb{N}}(\mathcal{C}, N_b)$. Since no directed simulation exists between M_a and N_b, we have that $M_a \not\leftrightarroweq_d N_b$, and since $\sim$ is finer than directed simulation, we have $M_a \not\sim N_b$, contradicting our initial assumption about $\mathcal{C}$.

2. There's some $A_c \in \mathcal{C}$ such that $|A_c| > 1$. Let K_a^n and K_b^{n+1} be the LTSs with n and $n+1$ elements whose actions hold between any two elements, and whose states satisfy all proposition letters. It follows easily that $Z = K^n \times K^{n+1}$ is a bisimulation, and so $K_a^n \leftrightarrows K_b^{n+1}$. Then since $\sim$ is coarser than bisimulation, we have that $K_a^n \sim K_b^{n+1}$. However, clearly any maps from A_c to K_a^n such that $c \mapsto a$ or from $A_c \to K_b^{n+1}$ such that $c \mapsto b$ are homomorphisms. Clearly, then, we have that $\hom_{\mathbb{N}}(A, K_b^{n+1}) > \hom_{\mathbb{N}}(A, K_a^n)$. Thus $\hom_{\mathbb{N}}(\mathcal{C}, K_a^n) \neq \hom_{\mathbb{N}}(\mathcal{C}, K_b^{n+1})$, contradicting our initial assumption about $\mathcal{C}$.

It follows that no such class $\mathcal{C}$ exists. □

Call a class $\mathcal{C}$ *trivial* if it contains only one-element structures. It is worth nothing that, if we consider only non-trivial classes, then we can complete the proof of Proposition 4.7.2 without considering the first case of the case distinction. Auditing the proof, we then see that we would only need to appeal the assumption that $\sim$ is coarser than bisimulation, but not the assumption that it is finer than directed simulation. Thus we obtain the following proposition, which we interpret to say that the counting left profile restricted to any non-trivial class is too powerful to capture relations weaker than bisimulation.

Proposition 4.7.3. Let $\sim$ be any equivalence relation on LTSs coarser than bisimulation. Then there's no non-trivial class $\mathcal{C}$ such that $\hom_{\mathbb{N}}(\mathcal{C}, M_a) = \hom_{\mathbb{N}}(\mathcal{C}, N_b)$ if and only if $M_a \sim N_b$.

4.8 Conclusion

Figure 4.4 summarizes the main results of this chapter.

Language	Invariance Relation	Characterizing Vector
$\mathrm{ML}^{+}_{\Diamond}$	Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{T}, M_a)$
ML^{+}	Directed Simulation Equivalence	None for $\mathbb{B}$ or $\mathbb{N}$
BML	Bisimulation	None for $\mathbb{B}$ or $\mathbb{N}$
$\mathrm{ML}_{\#}$	Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{T}, M_a)$
$\mathrm{ML}^{+,B}_{\Diamond}$	Back-and-Forth Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{A}, M_a)$
$\mathrm{ML}^{+,G}_{\Diamond}$	Global Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{F}, M_a)$
$\mathrm{ML}^{B}_{\#}$	Back-and-Forth Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{A}, M_a)$
$\mathrm{ML}^{G}_{\#}$	Global Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{F}, M_a)$

Figure 4.4: Summary of Characterization Results.

Note that we stated our results for various different classes: finite, image-finite, and degree-finite LTSs. These results imply that the LTSs within these classes can be represented, up to various different notions of process equivalence, by infinite-dimensional vectors. These representations create geometric abstractions, which, if made finite, could be used as feature vectors for applications in machine learning. We leave for future work the task of determining which classes of LTSs can be represented, up to these notions of equivalence, using only finite-length vectors. Furthermore, it is worth noting that modal logics and LTSs commonly occur outside of process theory in research areas such as knowledge representation and description logics. We leave for future work whether these results can be applied in these areas.

Some other common variations of LTSs are *weighted* and *probabilistic* LTSs, which assign weights and probabilities to the actions of the model, respectively. These can be

naturally interpreted as S-labeled models for appropriately chosen semirings S (depending on the application). The process-theoretic equivalence notions (like bisimulation) can be extended to these types of structures. We leave for future work the task of characterizing these structures via a restriction of the left profile to some class (or showing that no such characterizing class exists).

Additionally, a well-studied extension of modal logic is the *guarded fragment*, which can also be extended with counting quantifiers to form the *counting guarded fragment*. These are appropriately preserved by *guarded bisimulations* and *graded guarded bisimulations*. We leave for future work the task of characterizing these structures via a restriction of the left profile to some class (or showing that no such characterizing class exists). We conjecture that guarded bisimulation cannot be characterized by a restriction of the counting left profile, and that graded guarded bisimulation can be characterized by the restriction of the counting left profile to the class of *hypergraph-acyclic* structures.

Chapter 5

Conclusion

In this thesis, we studied restrictions of homomorphism vectors of finite models. These were defined very broadly, over arbitrary semirings. We primarily confined attention to the homomorphism vectors defined under the Boolean and counting semantics. Furthermore, we defined both left and right homomorphism vectors. In Chapter 3, we studied restrictions of the counting right homomorphism vector, while in Chapter 4, we studied restrictions of both the Boolean and counting left homomorphism vector.

In Chapter 3 we studied finite characterizations of models, which can be seen as finite restrictions of the (counting) right profile capturing a model up to isomorphism. We motivated this line of research by its applications in database theory: a finite characterization of a finite model can also be interpreted as a characterization of the model's canonical query. In this way, a finite characterization could be manually checked to verify correctness of a complex query. However, one might also study finite characterizations of models with respect to the left profile. This can equivalently be interpreted as a (counting) left characterization of a database instance I, which is a finite collection of queries whose answers in I under the bag semantics determine I up to isomorphism. We leave this for future work.

In Chapter 4, we studied which modal relations can be characterized by restrictions of the left profile. At a high level, these investigations were motivated by similar elegant results relating C^2-equivalence to different notions in linear algebra, machine learning, and heuristics for the graph isomorphism problem. In a more fine-grained view, these were motivated by the observation in [BKM$^+$20] that invariance under formulas of graded modal logic correspond to indistinguishability by certain restricted forms of graph neural networks. However, one could ask which relations between models can be characterized by restrictions of the right profile. This line of research was initiated in [AKW21], in which the authors showed that several relations of interest, including co-spectrality, fractional isomorphism, and C^2-equivalence, cannot be captured by restrictions of the counting right profile.

In general, homomorphism vectors over various semirings are a powerful tool for capturing various finite model-theoretic relations. While we confined attention to the Boolean semiring and the semiring of the natural numbers, numerous other semirings exist whose corresponding semantics are meaningful in various applications in computer science. We leave exploration of restrictions of homomorphism vectors over these semirings for future work.

Bibliography

[AKW21] Albert Atserias, Phokion G Kolaitis, and Wei-Lin Wu. On the expressive power of homomorphism counts. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2021.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[BCV09] Johan van Benthem, Balder ten Cate, and Jouko Vaananen. Lindstrom theorems for fragments of first-order logic. *Logical Methods in Computer Science*, 5, 2009.

[Ben14] Johan van Benthem. *Modal correspondence theory*. PhD thesis, University of Amsterdam, 2014.

[BKM+20] Pablo Barceló, Egor V Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan-Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[BLZ14] Stefan Böttcher, Sebastian Link, and Lin Zhang. Pulling conjunctive query equivalence out of the bag. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 41–50, 2014.

[CD22] Balder ten Cate and Victor Dalmau. Conjunctive queries: Unique characterizations and exact learnability. *ACM Transactions on Database Systems*, 47(4):1–41, 2022.

[CDKW23] Balder ten Cate, Víctor Dalmau, Phokion G Kolaitis, and Wei-Lin Wu. When do homomorphism counts help in query algorithms? *arXiv e-prints*, pages arXiv–2304, 2023.

[CFI92] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992.

[CFLX22] Yijia Chen, Jörg Flum, Mingjun Liu, and Zhiyang Xun. On algorithms based on finitely many homomorphism counts. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[Cod71] E. F. Codd. A data base sublanguage founded on the relational calculus. In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, SIGFIDET '71, page 35–68, New York,

NY, USA, 1971. Association for Computing Machinery.

[CV93] Surajit Chaudhuri and Moshe Y Vardi. Optimization of real conjunctive queries. In *Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 59–70, 1993.

[DGR18] Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász meets Weisfeiler and Leman. *arXiv preprint arXiv:1802.08876*, 2018.

[DNV95] Rocco De Nicola and Frits Vaandrager. Three logics for branching bisimulation. *Journal of the ACM (JACM)*, 42(2):458–487, 1995.

[Dvo10] Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010.

[GKT07] Todd J Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, 2007.

[Gro20] Martin Grohe. Counting bounded tree depth homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 507–520, 2020.

[HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM (JACM)*, 32(1):137–161, 1985.

[HN92] Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1-3):117–126, 1992.

[JLW20] Jean Christoph Jung, Carsten Lutz, and Frank Wolter. Least general generalizations in description logic: Verification and existence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2854–2861, Apr. 2020.

[KR97] Natasha Kurtonina and Maarten de Rijke. Simulating without negation. *Journal of Logic and Computation*, 7(4):501–522, 1997.

[Lov67] László Lovász. Operations with structures. *Acta Math. Acad. Sci. Hungar*, 18(3-4):321–328, 1967.

[MR89] Heikki Mannila and Kari-Jouko Räihä. Automatic generation of test data for relational queries. *Journal of Computer and System Sciences*, 38(2):240–258, 1989.

[MRF+19] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[OP08] Martin Otto and Robert Piro. A lindström characterisation of the guarded fragment and of modal logic with a global modality. *Advances in modal logic*, 7:273–287, 2008.

[Rij00] Maarten de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.

[Tin86] Gottfried Tinhofer. Graph isomorphism and theorems of birkhoff type. *Computing (Wien. Print)*, 36(4):285–300, 1986.

[Tin91] Gottfried Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30(2-3):253–264, 1991.